МИНИСТЕРСТВО СЕЛЬСКОГО ХОЗЯЙСТВА
И ПРОДОВОЛЬСТВИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ
АГРАРНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра прикладной информатики

В. Л. Быков, Н. Г. Серебрякова

ИНФОРМАТИКА

Рекомендовано
Учебно-методическим объединением
по образованию в области сельского хозяйства
в качестве учебно-методического пособия
для студентов высших учебных заведений
группы специальностей 74 06 «Агроинженерия»

Минск
БГАТУ
2013

УДК 004(07)
ББК 32.81я7
Б96

Рецензенты:
заведующий кафедрой информатики и прикладной математики,
кандидат физико-математических наук,
доцент Брестского государственного университета им. А. С. Пушкина
В. Ф. Савчук;
декан факультета непрерывного и дистанционного обучения
Белорусского государственного университета информатики
и радиоэлектроники, кандидат технических наук, доцент
В. М. Бондарик

Быков, В. Л.

Б96 Информатика : пособие / В. Л. Быков, Н. Г. Серебрякова. – Минск : БГАТУ. – 2013. – 656 с. : ил.

ISBN 978-985-519-564-2

Излагаются общие сведения об информатике как основе познания и ее категориях, системах счисления, применяемых в вычислительной технике, об истории развития вычислительной техники, устройстве компьютера и его составных частей, программном обеспечении: операционная система *Windows 7*, сервисная оболочка *Total Commander*, программы архивации; приведены сведения об антивирусных программах. Изложены общие сведения о работе в *MathCad*. Подробно рассмотрены вопросы алгоритмизации и программирования в среде *Visual Basic for Applications*. Приведены основные сведения о приложениях *Microsoft Office 2010*: *Word*, *Excel*, *Power Point*. Даны краткие сведения о сетевых технологиях.

Ориентировано на студентов технических специальностей, но может быть полезно всем студентам, магистрантам, аспирантам, изучающим информатику самостоятельно.

УДК 004(07)
ББК 32.81я7

ISBN 978-985-519-564-2

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Предмет «Информатика» является естественно-научной дисциплиной для всех технических направлений и специальностей. Современный уровень развития науки, который немыслим без знания вычислительной техники, по-новому поставил вопрос о подготовке специалистов в вузах. Если в начале XX века молодое правительство России поставило задачу ликвидировать общую неграмотность, то в настоящее время, в начале XXI века, возникла необходимость ликвидации компьютерной неграмотности. В соответствии с образовательными стандартами студенты вузов должны иметь представление об информатике как особом способе познания мира, роли информатики в обеспечении экономической мощи государства, о современных достижениях в области вычислительной техники и информационных технологий; знать и уметь использовать современные средства вычислительной техники и численные методы решения инженерных и экономических задач; иметь навыки численного решения математических задач, задач математической физики, использования стандартных и технологических программ, смысловой постановки прикладных задач, алгоритмизации и программирования задач отрасли, математического моделирования объектов и оценки пределов применения полученных результатов».

С учетом роли информатики в современном обществе был принят закон «Об информации, информатизации и защите информации». Данный закон учитывает опыт применения ранее принятого закона, изменения, произошедшие в обществе, и накопленный опыт использования информационных систем, вновь возникшие проблемы и требования времени. В этом законе определена государственная политика в сфере информатизации, приведены основные термины и определения, определены принципы информатизации и правовые отношения в сфере информатизации и защиты информации.

Целью дисциплины «Информатика» является приобретение студентами знаний, умений и навыков по эффективному использова-

нию современной вычислительной техники и ее программного обеспечения.

Предметом изучения в дисциплине «Информатика» в рамках учебной программы являются базовое программное обеспечение, операционная система, сервисные оболочки, язык программирования *Visual Basic*, сетевое программное обеспечение, прикладное программное обеспечение: пакеты прикладных программ *Word*, *Excel*, *PowerPoint*, математическая система *MathCad*.

Чем вызвана необходимость написания данного учебного пособия, при том, что учебников и учебных пособий по данной дисциплине достаточно много? Это объясняется следующим: большинство учебных пособий написано применительно к экономическим и гуманитарным специальностям. Основное внимание в них уделяется прикладному программному обеспечению – *Word*, *Excel*, сетевому программному обеспечению, подготовке презентаций, и в то же время вопросы алгоритмизации, программирования не находят в этих изданиях должного отражения. Поэтому авторы надеются, что изданием данного пособия удастся восполнить указанный пробел. Предмет «Информатика» настолько обширен, что в рамках одного учебного пособия невозможно подробно изложить все его аспекты. Поэтому авторы придерживаются типовой программы, утвержденной Министерством образования Республики Беларусь. Однако ограниченность времени, отводимого на изучение предмета, поставила перед авторами альтернативу: либо изложить понемногу сведения о всех приложениях *Windows*, либо сосредоточить внимание на наиболее важных, с точки зрения авторов, для студентов технических специальностей приложениях. Авторы придерживаются второго направления.

Пособие состоит из семи частей.

В первой части приведены основные понятия об информации и информатике; даны основные термины и определения, форматы представления информации; дано понятие о системах счисления, истории и перспективах развития вычислительной техники, устройстве, принципах работы персонального компьютера и его составных частей; приведена классификация программного обеспечения и краткие сведения об операционной системе *Windows*.

Во второй части приведено описание математической системы *MathCad* и рассмотрены основные приемы работы в ее среде, решения типовых математических задач.

Третья часть посвящена вопросам алгоритмизации. Базовые структуры языка программирования и алгоритмы решения типовых задач рассматриваются на основе языка программирования *Visual Basic 6.0*. Язык программирования *Visual Basic 6.0* выбран по ряду причин. Этот язык прост в изучении. Синтаксис большинства его операторов совпадает со структурами алгоритмического языка программирования, изучаемого в школах. Кроме того, этот язык достаточно развит и приближается по своим возможностям к таким языкам программирования высокого уровня, как *Pascal*, *Delphi*, *C*. Язык *Visual Basic* является объектно-ориентированным языком программирования и позволяет создавать 32-х разрядные приложения для работы в среде *Windows*. Следующим и весьма важным аргументом в пользу выбора языка программирования *VB 6.0* в качестве базового языка программирования для студентов технических специальностей, не программистов, является то, что различные версии этого языка используются во всех стандартных приложениях *Windows*, таких как, например, *Word*, *Excel*, *Access* для разработки пользовательских процедур и функций, автоматизации управления разрабатываемыми приложениями.

Четвертый раздел посвящен текстовому процессору *Microsoft Word*. Основное внимание здесь уделено оформлению документов средствами текстового процессора.

В пятом разделе приведены сведения о средствах для разработки презентаций и дано описание программы *Power Point*.

В шестом разделе приведены основные сведения об электронных таблицах *Microsoft Excel* и использовании этого приложения для решения инженерных задач.

Седьмой раздел посвящен сетевому программному обеспечению. Излагаются основные понятия о глобальной сети *Internet*, протоколах и адресации. Рассмотрена структура браузера *Internet Explorer* и назначение его основных элементов. Рассмотрены принципы работы электронной почты.

Учебное пособие предназначено для студентов технических специальностей, может быть полезно и для студентов других специальностей, а также для всех желающих самостоятельно изучить основы дисциплины «Информатика».

Авторы выражают благодарность и признательность сотрудникам Брестского государственного технического университета, оказавшим помощь в подготовке отдельных разделов настоящего пособия. Особую благодарность авторы выражают рецензентам: заве-

дующему кафедрой «Информатика и прикладная математика» Брестского государственного университета им. А. С. Пушкина, кандидату физико-математических наук, доценту В. Ф. Савчуку, декану факультета непрерывного и дистанционного образования Белорусского государственного университета информатики и радиоэлектроники, кандидату технических наук, доценту В. М. Бондарику за ценные замечания, способствовавшие улучшению структуры и содержания учебного пособия.

1. ОСНОВНЫЕ ПОНЯТИЯ И КАТЕГОРИИ ИНФОРМАТИКИ

1.1. ОСНОВНЫЕ ПОНЯТИЯ И ОПРЕДЕЛЕНИЯ

Ключевые слова: информация, данные, базы данных, банки данных, знания, базы знаний, информатизация, информационные технологии, информационный ресурс, информационное общество.

1.1.1. Информация – основа познания и преобразования мира

Термин «информация» происходит от латинского слова *information* – изложение, разъяснение. Под информацией понимают сведения, передаваемые людьми устным, письменным или иным способом.

Об информации можно говорить в широком и узком смысле слова.

В *широком смысле* слова информация – это отражение реального мира во всем его многообразии, в *узком смысле* слова – это любые сведения, являющиеся объектом регистрации, хранения, передачи и преобразования. В законодательстве дается следующее определение информации[1]: *информация – сведения о лицах, предметах, фактах, событиях, явлениях и процессах независимо от формы их представления*.

Информация как объект научного исследования стала рассматриваться лишь во второй половине XX столетия, хотя информацией человечество пользовалось во все времена.

Информация о любом материальном объекте может быть получена различными способами: наблюдением, измерением его параметров, путем вычислительного эксперимента над ним или путем логического вывода на основе накопленных знаний. В связи с этим

[1] Определения даются в соответствии с Законом Республики Беларусь от 10 ноября 2008 г. № 455-З «Об информации, информатизации и защите информации».

информацию делят на доопытную, или *априорную*, и послеопытную, или апостериорную полученную в результате проведенного эксперимента.

Информация обладает рядом важных свойств, к которым относятся: структура, форма, измеримость, ценность, достоверность, качество.

Структура информации – это то, что определяет взаимосвязи между ее составными элементами. Фундаментальным свойством информации является ее *системность*. Это означает, что информация обладает в совокупности такими свойствами, какими не обладает ни один из составляющих ее элементов. Например, фраза несет больше информации, чем отдельные слова, из которых она построена.

Форма информации – форма информации определяется способом ее представления. В зависимости от этого традиционно различают символьно-текстовую, графическую, звуковую информацию. Можно также выделить и другие, нетрадиционные формы представления информации, такие как: тепло, нервное раздражение кожи, запах и др., которые современная техника в состоянии измерять и представлять в ЭВМ.

Измеримость информации – информация может быть измерена. Потребность в измерении информации возникла в связи с необходимостью ее передачи, накопления и хранения.

Информация возникает в процессе взаимодействия трех объектов: источника информации, среды, через которую передается информация, и приемника информации. Например, человек стоит на берегу моря и слышит шум прибоя. Морской прибой – источник информации, воздух – среда, в которой распространяется звук, человек – приемник информации. Или другой пример: микрофон – источник информации, среда – проводной или воздушный канал передачи информации со средствами усиления и преобразования исходного сигнала, магнитофон – приемник информации.

Ценность информации. Сообщение, которое тем или иным способом уменьшает наше незнание о предмете или о явлении, может считаться ценным. В противном случае это сообщение не представляет для нас никакой ценности. Основываясь на этом подходе, один из основоположников теории информации Клод Шеннон определил информацию как *снятую неопределенность*.

Американский инженер Р. Хартли в 1928 году предложил формулу для определения количества информации I, содержащейся

в сообщении, выбранном из множества N равновероятных сообщений:

$$I = \log_2 N.$$

Клод Шеннон развил идею Хартли для оценки количества информации при неодинаковой вероятности сообщений в наборе. Формула Шеннона имеет вид:

$$I = -\sum(p_i \log_2 p_i),$$

где p_i – вероятность появления того или иного сообщения.

При оценке информации различают три аспекта: синтаксический, семантический и прагматический.

Синтаксический аспект связан со способом представления информации. Этот аспект важен при измерении информации, ее передаче и хранении. Информацию, рассматриваемую только в синтаксическом аспекте, называют *данными*. С синтаксическим аспектом информации связано понятие о количестве информации и объеме данных, необходимых для хранения информации.

Семантический аспект связан со смысловым содержанием информации и соотносимостью ее с известной ранее информацией.

Для количественной оценки семантической информации Ю. А. Шрейдером предложена *тезаурусная мера информации*. Когда получатель не воспринимает или не понимает информацию, тезаурус получателя информации равен нулю: $S_п = 0$. Для получателя информации, которому все известно об исследуемом объекте, тезаурус также равен нулю. В обоих случаях количество семантической информации $I_с = 0$. Максимальное значение семантическая информация приобретает при согласовании смыслового содержания информации S^* с тезаурусом пользователя $S_п$, когда поступающая информация понятна пользователю и несет ему ранее не известные (отсутствующие в его тезаурусе) сведения.

Прагматический аспект определяет полезность информации. Его определяют иногда как вероятность достижения цели с учетом получения информации:

$$I_п = \log_a \left(\frac{P_0}{P_1} \right),$$

где $I_п$ – прагматическое количество информации;

P_0 – вероятность достижения цели до получения информации;

P_1 – вероятность достижения цели после получения информации;

$a > 1$ – основание логарифма.

Качество информации определяют как совокупность свойств, обусловливающих возможность ее использования для удовлетворения определенных, в соответствии с ее назначением, потребностей. К таким свойствам относят: *репрезентативность*, *содержательность*, *достаточность*, *доступность*, *своевременность*, *устойчивость*, *точность*, *достоверность*, *актуальность* и *ценность*.

Из этих свойств раскроем содержание только одного свойства – репрезентативность. *Репрезентативность* – это правильность, качественная адекватность отражения заданных свойств объекта. Содержание других понятий интуитивно понятно.

В зависимости от области знаний различают научную, техническую, экономическую, производственную, правовую, патентную и другие виды информации. Каждый из этих видов информации имеет особую смысловую нагрузку и ценность, свои требования к точности и достоверности, преимущественные технологии обработки, формы представления и хранения информации.

Процесс усвоения знаний и применений их на практике представляет замкнутый цикл. Человек, изучая объекты окружающего мира, получает информацию, которую фиксирует на различных носителях: книгах, магнитных лентах, магнитных дисках. Собранная информация обрабатывается, анализируется, обобщается и накапливается на тех же носителях информации. Теперь эта научная информация (нас интересует прежде всего эта информация) может быть достоянием каждого желающего. Появление и внедрение в научные учреждения компьютеров ускорило процесс поиска и использования этой информации. Сведения, полученные при использовании научной информации, позволяют расширять наши знания об объекте, создавать новые методы исследования, получать новую информацию.

Место информации в системе познания человека отражает схема, представленная на рисунке 1.1.

В одном терминологическом ряду с понятием «информация» стоят понятия «данные» и «знания».

*Под **данными** понимают документированную информацию, циркулирующую в процессе ее обработки на электронно-вычислительных машинах.*

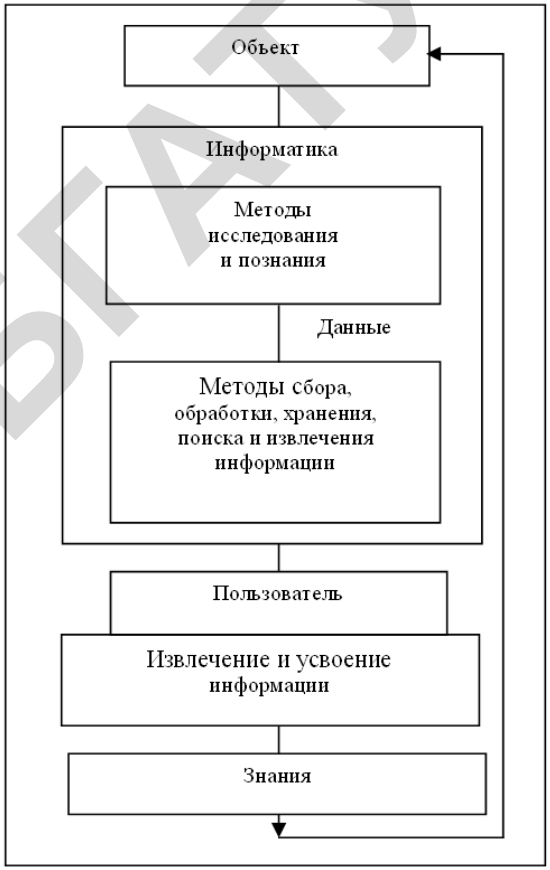


Рис. 1.1. Цикл получения информации

Данные хранятся в базах данных. ***База данных*** *– совокупность структурированной и взаимосвязанной информации, организованной по определенным правилам на материальных носителях.* Базы данных могут объединяться в банки данных. ***Банк данных*** *– организационно-техническая система, включающая одну или несколько баз данных и систему управления ими.*

Знания *– это информация, на основании которой путем логических рассуждений могут быть получены определенные выводы.* Для обобщения накопленных знаний и обеспечения возможности использования их в практической деятельности, они так же, как

и данные, организуются в базы знаний. ***База знаний*** – *совокупность формализованных знаний об определенной предметной области, представленных в виде фактов и правил.*

В наш бурно развивающийся век информация с каждым годом приобретает все большее значение. Она оказывает влияние на все стороны человеческой деятельности: науку, экономику, политику, социальную сферу.

Характерными чертами информации являются следующие:

1. Информация не может существовать без носителя информации, при передаче информация не теряется;

2. Это наиболее важный ресурс современного производства: он снижает потребность в земле, труде, капитале, уменьшает расход сырья и энергии.

Информация оказывает непосредственное влияние на производство. Современные технологии выпуска продукции все в большей степени связаны с переработкой информации. Любая промышленная технология может быть охарактеризована объемом требуемой информации на единицу продукции. Причем технология большей информационной емкости характеризуется большей эффективностью, меньшим потреблением материальных ресурсов. Происходит удивительное явление: материальные ресурсы как бы замещаются «невещественной» информацией. Происходит экономия энергии, металлов, нефти и других ресурсов за счет процессов переработки информации.

3. Благодаря этому информация становится *товаром*, *ресурсом*, и доступ к этому ресурсу определяет эффективность всех видов деятельности.

4. Информация вызывает к жизни новые производства. Например, изобретение лазерного луча явилось причиной возникновения, развития и внедрения лазерных технологий, получивших широкое применение в промышленном производстве, медицине, военном деле, например, резание металла, точечная сварка, электронный скальпель, винтовки с лазерным прицелом, запись информации на компакт-диск и ее чтение.

5. Информация придает дополнительную ценность другим ресурсам, в частности, трудовым. Действительно, работник с высшим образованием ценится больше, чем работник со средним образованием.

Интенсивное развитие всех отраслей знания вызвало колоссальный рост количества информации, информационный взрыв. Особенно ярко это стало проявляться с середины XX века. В 70-е годы

объем информации удваивался каждые 5–7 лет. В 80-е годы удвоение информации происходило уже за 20 месяцев, а в 90-е годы – ежегодно. Например, в конце 80-х годов в СССР в области естественных и общественных наук ежегодно регистрировалось 1,5 млн. отечественных и зарубежных первоисточников. Найти, однако, интересующие сведения в этом потоке информации чрезвычайно сложно, возник информационный голод. Это *противоречие* века информации: рост объема информации и трудность ее получения.

Разрешение этого противоречия возможно только на пути *информатизации* и *компьютеризации* общества: создания интегрированных банков данных, сетей обработки и передачи информации, широкого внедрения компьютеров, обеспечивающих доступ к всемирной энциклопедии знаний.

За несколько последних десятилетий в связи с большими достижениями в производстве микросхем, использовании новых технологий произошли серьезные изменения в состоянии всей индустрии информатики: появились мобильные телефоны, цифровые фотоаппараты и видеокамеры. Растет число нетрадиционных информационных устройств, включая беспроводные счетчики, автомобильные навигационные системы, промышленное оборудование, радиочастотные датчики и интеллектуальные контроллеры. Увеличивается число пользователей сети, взаимодействий между людьми по электронной почте, через системы обмена мгновенными сообщениями, социальные сети и т. п. Это привело к значительному росту объема цифровой информации. Исследования, проведенные корпорацией EMC – мировым лидером в области решения информационных инфраструктур, – показали, что в 2011 году совокупный объем цифровой информации составил **161 миллион гигабайт** (161 эксабайт). Указанная цифра приблизительно в 3 миллиона раз больше объема информации, содержащейся во всех когда-либо написанных книгах. По данным этой организации, в 2011 году создано 3 892 179 868 480 350 000 000 (**3 секстиллиона 892 квинтиллиона 179 квадриллионов 868 триллионов 480 миллиардов 350 миллионов**) бит новой цифровой информации. По прогнозам за период с 2006 по 2012 год объем информации увеличился в шесть раз. Этот прогноз выдвигает перед компаниями, занимающимися вопросами разработки и внедрения систем телекоммуникации, новые, трудные задачи по обеспечению надежности, безопасности, конфиденциальности, скорости передачи и доступности информации, соблюдению нормативных требований.

Информатизация

Информатизация – организационный, социально-экономический и научно-технический процесс, обеспечивающий условия для формирования и использования информационных ресурсов и реализации информационных отношений.

Информатизация – это процесс, включающий комплекс взаимосвязанных и взаимообусловленных мер по обеспечению полного использования достоверных, исчерпывающих и современных знаний во всех общественно значимых сферах человеческой деятельности. Это процесс существенного изменения роли информации как совокупности знаний и зависимостей между ними в общественной жизни. Создание индустрии информатики и превращение информационного продукта в товар приводит к глубоким социальным изменениям в обществе, меняет само общество. Оно трансформируется из индустриального общества в *информационное общество*.

Информатизация – это сложный социальный процесс, связанный со значительными изменениями в образе жизни населения. Он требует серьезных усилий на многих направлениях, включая ликвидацию компьютерной неграмотности, формирование культуры использования новых информационных технологий и др. Движущей силой развития общества должно стать производство информационного, а не материального продукта. В информационном обществе изменяется не только производство, но и весь уклад жизни, система ценностей, возрастает значимость культурного досуга по отношению к материальным ценностям. В информационном обществе производятся и потребляются интеллект, знания, что приводит к увеличению доли умственного труда. От человека потребуется способность к творчеству, возрастает спрос на знания. Материальной и технологической базой информатизации общества станут различного рода системы на базе компьютерной техники и компьютерных сетей, информационной технологии, телекоммуникационной связи.

Темпы экономического развития многих капиталистических стран (США, Япония, Англия и др.) в настоящее время определяются, главным образом, ростом производительности труда именно в сфере информатизации. По официальным оценкам, доля информационных видов деятельности в валовом национальном продукте США достигла 50 % уже к концу 70-х годов прошлого столетия. Во второй половине 70-х годов в США в создании вычислительного потенциала и организации его эффективного использования начался качественно новый этап интеграции его элементов. Этот

этап характеризуется формированием многоотраслевого информационно-вычислительного комплекса, ядром которого служит индустрия переработки информации, представляющая собой особую отрасль экономики США. Основная функция этой отрасли – создание материально-технической базы для удовлетворения информационных потребностей промышленной и деловой сфер, органов государственного управления, других отраслей деятельности общества. Основу такой базы составляют ЭВМ, системы связи и передачи информации, базы данных, базы знаний, математические модели, программные средства, информационные технологии, контингент специалистов в области информатики и вычислительной техники, другие составляющие инфраструктуры информационного обслуживания.

В настоящее время ближе всех стран к информационному обществу находятся США, Япония, Англия, страны Западной Европы.

Информационные технологии

В индустрии переработки информации все возрастающую роль приобретают так называемые информационные технологии. *Информационная технология – это совокупность процессов и методов осуществления поиска, получения, передачи, сбора, обработки, накопления, хранения, распространения и (или) предоставления информации, а также пользования информацией и защиты информации.*

Информационные технологии включают два основных элемента – машинный и человеческий (социальный), причем социальный элемент выступает главным. Новые информационные технологии имеют большое значение в решении задач информатизации. Поэтому информатизацию можно рассматривать как целенаправленную деятельность по созданию и широкомасштабному использованию во всех сферах жизни общества новых информационных технологий с целью интенсификации экономики, ускорения научно-технического прогресса, совершенствования стиля и повышения уровня жизни, развития производственных и общественных отношений. Складываются новые отношения между людьми в обществе – информационные отношения. *Информационные отношения – это отношения, возникающие при поиске, получении, передаче, сборе, обработке, накоплении, хранении, распространении и (или) предоставлении информации, пользовании информацией, защите информации, а также при применении информационных технологий.*

Информационный ресурс

Информационные технологии выступили новым средством превращения знаний в информационный ресурс общества. *Информационный ресурс – организованная совокупность документированной информации, включающая базы данных, другие совокупности взаимосвязанной информации в информационных системах.*

В последней четверти двадцатого столетия информация, по утверждению специалистов, стала для промышленно развитых стран одним из наиболее важных национальных ресурсов. В двадцать первом веке информационные ресурсы станут основным национальным богатством, а эффективность их промышленного использования будет определять экономическую и оборонную мощь страны в целом.

Информационное общество

Информационное общество – это теоретическая концепция постиндустриального общества. Термин этот, полагают, предложил впервые Ю. Хаяши, профессор токийского технологического института (Япония), в 1969 году. Предполагают, что это будет историческая фаза возможного развития цивилизации, в которой главными продуктами производства становятся информация и знания. Отличительными чертами этого общества будут:

увеличение роли информации, знаний и информационных технологий в жизни общества;

возрастание числа людей, занятых информационными технологиями, коммуникациями и производством информационных продуктов и услуг в валовом внутреннем продукте;

нарастающая информатизация общества с использованием телефонии, радио, телевидения, сети Интернет, а также традиционных и электронных средств массовой информации;

создание глобального информационного пространства, обеспечивающего эффективное информационное взаимодействие людей, их доступ к мировым информационным ресурсам и удовлетворение их потребностей в информационных продуктах и услугах.

Защита информации

В зависимости от категории доступа законодательство устанавливает два вида информации: общедоступную информацию и информацию, распространение и (или) предоставление которой ограничено.

К информации первого вида относится вся информация, определяющая права человека, предусмотренные Конституцией

государства, например, информация о правах, свободах и законных интересах физических лиц, правах и законных интересах юридических лиц и о порядке реализации прав, свобод и законных интересов; о деятельности государственных органов, общественных объединений; о чрезвычайных ситуациях, экологической, санитарно-эпидемиологической обстановке, гидрометеорологической и иной информации, отражающей состояние общественной безопасности, и др.

К информации второго вида относятся, например, информация, составляющая государственную, коммерческую или профессиональную тайну, сведения о частной жизни физических лиц и персональные данные и др., определенное законодательством государства.

Оба вида информации нуждаются в защите.

Требования по защите общедоступной информации могут устанавливаться только в целях недопущения ее уничтожения, модификации (изменения), блокирования правомерного доступа к ней.

Требования по защите информации в государственных информационных системах, а также информационных системах, содержащих информацию, распространение и (или) предоставление которой ограничено, определяются законодательством.

Контрольные вопросы

1. Что такое информация?
2. Что понимают под информацией в широком смысле слова?
3. Что понимают под информацией в узком смысле слова?
4. Как влияют научно-технические достижения на объем и распространение информации?
5. Что составляет научно-техническую базу процесса информатизации общества?
6. Какое место занимает информатика в системе познания человеком окружающего мира и научных знаний?
7. Как влияет информация на уровень жизни общества?
8. Что понимается под информационными технологиями?
9. Что такое информационный ресурс общества?
10. В чем состоит цель защиты общедоступной информации?

1.1.2. Информатика

Информатика – это наука о законах и методах организации и переработки информации в естественных и искусственных системах с применением ЭВМ.

Само понятие «информатика» возникло уже с появлением ЭВМ и подразумевалось вначале как наука о вычислениях. Благодаря ЭВМ появилась возможность представлять, хранить и передавать информацию в одной форме – двоичной, независимо от ее вида. Рост объемов информации и проблемы, связанные с ее получением, привели к возникновению новой отрасли науки – информатики.

На сегодня информатика представляет собой *комплексную научно-техническую дисциплину*. Она объединяет под своим названием довольно обширный комплекс наук, каждая из которых занимается изучением одного из аспектов понятия «информатика». Прилагаются интенсивные усилия ученых по сближению наук, составляющих информатику. Информатика представляет собой основу для информатизации общества и перехода его от индустриального общества к информационному обществу.

Основными задачами информатики как науки являются:

1) разработка методов и алгоритмов автоматизированного сбора, хранения, поиска и передачи информации;

2) разработка методов и алгоритмов автоматизированной обработки и преобразования информации;

3) разработка технологий и электронно-вычислительной техники, позволяющих развивать первые два направления.

В развитии ЭВМ, в связи с рассматриваемым вопросом, просматриваются три этапа:

вычислительный;

общеинформационный;

интеллектуальный.

Вычислительный этап охватывает период 40–60-х годов XX столетия, когда доступ пользователей к ЭВМ был ограничен. Общеинформационный этап охватывает период с 60-х годов до конца 80-х годов XX столетия. В настоящее время наука и техника находятся на третьем этапе, этапе развития машинного интеллекта.

История развития информатики

В истории развития информатизации общества выделяют несколько информационных скачков:

освоение устной речи;

возникновение письменности и книгопечатание;

развитие компьютерной техники и технических средств связи;

развитие микропроцессорной техники и сетей ЭВМ;

развитие нанотехнологий.

Первый информационный скачок произошел еще на заре развития человечества, когда люди путем обмена информацией об окружающем мире благодаря членораздельной речи стали получать больше сведений, чем путем наследственной передачи. В начале зарождения человеческого общества объем информации был невелик. Она, то есть информация, представляла собой сведения о природе, роде, племени. Информация эта накапливалась и передавалась из поколения к поколению в виде устных преданий. В обществе проявилось такое качество информации, как *экономия познания*, при которой процесс познания через пробы и ошибки одного из членов сообщества служит источником знаний для остальных членов сообщества. Способность передавать информацию через пространство и время при помощи знаков присуща лишь человеку. Это качество сыграло важнейшую роль в развитии цивилизации, а в будущем оно, как утверждают специалисты, станет важнейшим.

Второй информационный скачок произошел примерно пять тысяч лет назад с появлением различных форм письменности. Развитие книгопечатания было на этом пути хотя и эволюционным, но значительным шагом вперед. Появилась возможность накапливать, хранить, передавать информацию. Каждое очередное техническое новшество: изобретение паровой машины, радио, телеграфа, телевидения – ускоряло процесс распространения информации, в том числе и научной.

Третий информационный скачок произошел в 70-е годы XX столетия в связи с появлением персональных компьютеров. Впервые стало возможным не только запоминание и хранение информации, но и «накопление интеллекта» в виде профессиональных знаний и их тиражирование в виде программ для компьютеров.

Четвертый информационный скачок произошел в конце XX и начале XXI столетия. Он связан с технологическим прорывом в области микроминиатюризации. Произошла незаметная для глаза технологическая революция. Это привело к появлению переносных и карманных компьютеров, мобильных телефонов, смартфонов, цифровых фотоаппаратов, видеокамер, цифровых систем передачи данных.

Пятый информационный скачок мы переживаем в настоящее время в связи с развитием нанотехнологий – технологии, соответствующие размерам 10^{-9} м. На этом уровне меняются свойства материалов, появляются материалы с новыми свойствами.

Контрольные вопросы

1. Что понимают под термином «информатика»?
2. Назовите основные этапы развития информатики.
3. Какие задачи стоят перед информатикой как наукой?
4. Какие этапы просматриваются в истории развития информатики?

1.1.3. Способы представления информации

Ключевые слова: сообщение, источник информации, приемник информации, канал связи, кодирование информации, разрядная сетка.

Передача информации

С практической точки зрения информация всегда представляется в виде *сообщения*. Информационное сообщение связано с источником информации (ИИ), каналами связи (КС), приемником информации (ПИ). Кроме того, в канале передачи информации могут присутствовать аналого-цифровые (АЦП) и цифроаналоговые (ЦАП) преобразователи (рис. 1.2).

Рис. 1.2. Передача информации

Источник информации служит для измерения параметров контролируемой среды и преобразования ее к виду, удобному для передачи по каналам связи или использования в ЭВМ. Источником информации могут быть различные электронные устройства, хранящие, преобразующие и воспроизводящие информацию, а также сами ЭВМ.

Приемником информации в одних случаях выступает ЭВМ, в других случаях – различные устройства автоматики: электромагниты, машинные усилители, пусковая аппаратура электродвигателей и др.

Сообщение от источника информации к приемнику информации передается в *материально-энергетической форме* (электрической, звуковой, световой). В качестве каналов передачи информации могут использоваться воздушная среда, проводные, кабельные или

волоконно-оптические линии связи. Информационное сообщение можно представить как изменение во времени параметров материально-энергетической среды и описать его функцией от времени, например:

$$y = x(t).$$

В зависимости от физической природы измеряемой величины, а также способа измерения и преобразования, измеряемая величина может быть представлена в *непрерывной* – аналоговой форме или в виде *дискретных* сообщений. В ряде случаев переход от непрерывного сообщения к дискретному сообщению дает преимущества при передаче, хранении и обработке информации.

В настоящее время для управления технологическими процессами и техническими системами, например, изготовления полимерных волокон, управления сложными механизмами, применяются цифровые ЭВМ. Поэтому возникает объективная необходимость преобразования непрерывных параметров, характеризующих состояние технологического процесса (температуры, давления, влажности, процентного содержания отдельных компонентов и т. д.), в дискретную форму. Устройства, осуществляющие такие преобразования, называются *аналого-цифровыми преобразователями* (АЦП). Цифровые ЭВМ выдают информацию также в дискретной форме, однако для управления некоторыми устройствами (машинные усилители, магнитные усилители, преобразователи энергии, двигатели постоянного тока) необходимы непрерывные сигналы. Обратное преобразование сигнала из дискретной формы в цифровую форму осуществляется с помощью *цифро-аналоговых преобразователей* (ЦАП).

Кодирование информации

Для представления информации в ЭВМ используется *алфавитный способ*, основой которого является использование фиксированного конечного набора символов любой природы, называемого *алфавитом*. Символы из набора алфавита называются *буквами*, а любая конечная последовательность букв этого алфавита – *словом*. При этом не требуется, чтобы слово обязательно имело языковое смысловое значение. Примеры слов: AAB, 10110110.

Символы алфавита при вводе в ЭВМ должны быть преобразованы в код. В цифровых ЭВМ преимущественное распространение получило двоичное кодирование, при котором символы вводимой в ЭВМ информации представляются средствами двоичного алфави-

та, состоящего из двух символов 0 и 1. Двоичный алфавит по числу входящих в него символов является минимальным, поэтому при двоичном кодировании алфавита, включающего большее число букв, каждой букве ставится в соответствие последовательность нескольких двоичных знаков или двоичное слово. Такие последовательности называются *кодовыми комбинациями*. Процесс получения кодовых комбинаций для представления букв одного алфавита средствами другого алфавита называется *кодированием*. Процесс обратного преобразования информации относительно ранее выполненного кодирования называется *декодированием*. Например, представление буквы А русского алфавита с помощью двоичных символов 11100001 есть процесс кодирования, обратный процесс получения буквы А из двоичного кода 11100001 есть декодирование. Полный набор кодовых комбинаций, соответствующий представлению всех букв одного алфавита средствами другого алфавита, называется *кодом*.

Число символов, составляющих кодовую комбинацию, называется *длиной кода* или *разрядностью кода*. Максимальное число кодовых комбинаций N при заданной разрядности n определяется выражением:

$$N = 2^n.$$

Различают коды равномерные и неравномерные. В равномерных кодах число символов во всех кодовых комбинациях одинаковое, в неравномерных кодах – разное. Примером неравномерного кода является азбука Морзе, где, например, буква Е имеет один короткий сигнал, а буква Ш – четыре длинных сигнала.

В вычислительной технике используются обычно равномерные коды. В IBM совместимых ЭВМ для внутреннего представления используется ASCII код (американский стандарт кодов для обмена информацией), позволяющий закодировать 256 символов. В настоящее время в приложениях операционной системы Windows применяется UNICOD, который позволяет закодировать 65 536 символов.

Для измерения объема передаваемой или хранимой информации используются следующие единицы измерения:

бит – один двоичный символ. Он позволяет представить в двоичной форме одно из двух различимых состояний объекта – «0» или «1»;

байт – восемь двоичных символов;

слово – два байта для 16-разрядных ЭВМ или 4 байта для 32 разрядных ЭВМ, 8 байт для 64-разрядных ЭВМ;

килобайт – 1024 бита (2^{10});

мегабайт – 1 048 576 байт (2^{20}), и более крупные единицы измерения: *гигабайт* – 2^{30} байт, *терабайт* – 2^{40} байт, *петабайт* – 2^{50} байт, *эксабайт* – 2^{60}, *зеттабайт* – 2^{70}, *йоттабайт* – 2^{80}.

Форматы представления информации

При работе с ЭВМ приходится иметь дело с различными видами информации: числовой, буквенной, графической, звуковой. Для представления этой информации разработаны определенные форматы. Форматы представления информации определяются *разрядной сеткой* ЭВМ.

Под разрядной сеткой понимают совокупность двоичных разрядов, используемых для хранения и обработки машинных слов.

В ЭВМ число этих разрядов фиксировано и кратно восьми: 8, 16, 32, 64, 128 и т. д.

Форматы представления чисел

Числа могут быть представлены в двух формах: *естественной* и *нормальной.* При естественном представлении чисел в ЭВМ, например, таких чисел, как: 12 560; 0,003572; 4,89760, – устанавливается длина разрядной сетки, а также длина целой и дробной частей. При этом распределение числа разрядов между целой частью и дробной частью в числе не изменяется и остается постоянным, независимо от величины числа. Такое представление чисел называется представлением чисел в форме с фиксированной запятой (рис. 1.3).

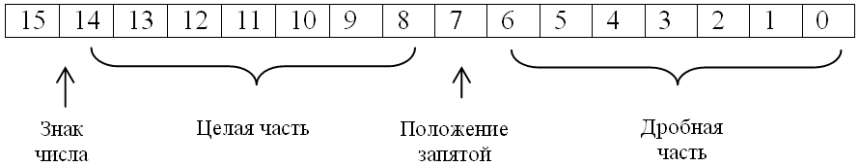


Рис. 1.3. Формат представления числа с фиксированной запятой

В современных ЭВМ эта форма используется преимущественно для представления целых чисел. При таком представлении целых чисел один разряд (старший) отводится под знак числа, остальные пятнадцать разрядов – под поле числа. Диапазон изменения чисел от $-(2^n - 1)$ до $(2^n - 1)$.

Представлением чисел в нормальной форме называют представление числа в форме с плавающей запятой. Общий вид числа:

$$A_n = m_A q^{P_A},$$

где m_A – мантисса числа A;

P_A – характеристика числа A;

q – основание системы счисления.

Например, в числе $0{,}14518 \cdot 10^4$ число 0,14518 – мантисса числа, 4 – характеристика, или порядок числа, 10 – основание системы счисления. Мантисса числа, во избежание неоднозначности представления чисел, должна находиться в пределах $q^{-1} \le |m_A| < 1$.

В шестнадцатиразрядном слове десять разрядов отводится для представления мантиссы, а шесть разрядов отводится для записи порядка чисел. При этом в каждой группе один разряд отводится под знак числа или знак порядка соответственно. Распределение разрядов при 32-х разрядной сетке приведено на рисунке 1.4.

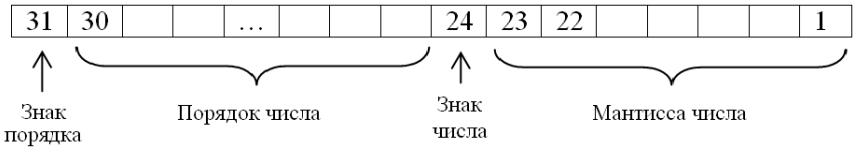


Рис. 1.4. Формат представления числа с плавающей запятой

Форматы представления букв

Для представления букв, цифр, знаков арифметических и логических операций, знаков сравнения и специальных знаков используется один байт. При этом можно закодировать 256 символов.

Особенности представления графической информации

В режиме отображения графической информации экран видеомонитора представляет собой поле точек 320×200, 640×200, 640×480, 800×600, 1280×1024, возможны и другие режимы. Для каждой точки графического изображения необходимо хранить значения уровней яркости, а при цветном изображении – цвет и оттенок. Современные дисплеи позволяют получать 16, 64, 256 и более различных цветов. В каждый момент времени необходимо хранить информацию о всех точках экрана, для чего требуется большой объем памяти. Кроме того, в ЭВМ могут храниться стандартные графические символы, сформированные самим

пользователем. Для этой цели ЭВМ имеет специально отведенную область памяти.

Изображение, выдаваемое на экран дисплея в закодированном виде, хранится в специально отведенной для этого области памяти компьютера, называемой памятью регенерации изображения. Данные из памяти регенерации периодически считываются, преобразуются в видеосигнал и отображаются на экране. Изображение на экране меняется с определенной частотой, обеспечивающей ему четкость и стабильность (на современных мониторах частота обновления может изменяться в пределах от 60 до 100 Гц). Электронный луч обегает экран построчно с верхнего левого угла в нижний правый угол. Время возвращения луча в верхний левый угол (когда он гасится) используется для изменения информации в памяти регенерации.

Память регенерации может быть разделена на страницы, каждая из которых содержит информацию об изображении полного экрана. Многостраничная организация памяти регенерации позволяет удобно реализовать эффекты движения графических изображений.

Особенности представления звуковой информации

Простейшим примером, поясняющим особенности кодирования звуковой информации, может служить нотная запись музыкального произведения. Обозначение нот, их длительности, пауз, знаков усиления и снижения громкости звука, музыкального ударения и другие знаки, составляющие набор нотной азбуки, образуют алфавит, с помощью которого можно представить в закодированном виде музыкальную мелодию. Последовательность символов, кодирующих музыкальную мелодию, хранится и обрабатывается в ЭВМ как обычная алфавитно-цифровая информация. Это позволяет использовать ЭВМ не только для анализа, но и для создания музыкальных произведений.

Перспективы развития ЭВМ связаны с созданием систем ввода и вывода речевой информации. Устное сообщение можно представить как последовательность элементарных звуков, называемых *фонемами*, и пауз между ними. От числа фонем, выделяемых в устной речи, зависит точность ее описания. На практике для кодирования русской устной речи выделяют 40–45 фонем, каждой из которых ставится в соответствие определенный код. Последовательность кодов, описывающих фонемы устного сообщения, вводится в память ЭВМ, хранится в ней и при необходимости выводится из нее через специальные устройства, называемые синтезаторами речи.

В настоящее время разработаны устройства, позволяющие переводить письменный текст в соответствующее фонемное представление, что позволяет воспроизводить этот текст на экране видеомонитора или через синтезаторы речи.

Весьма перспективным является создание средств общения человека с ЭВМ посредством голоса. Такие системы находятся в стадии развития и пока не получили широкого распространения.

Представление команд в ЭВМ

Управление вычислительным процессом в ЭВМ осуществляется с помощью команд, хранящихся в памяти. Команда состоит из двух частей: кода операции (КОП) и адреса. Код операции определяет действие, которое должна выполнить ЭВМ, а адресная часть команды содержит адреса *операндов* (чисел, знаков, функций), участвующих в операции. Иначе говоря, адресная часть команды определяет, откуда необходимо выбрать информацию или куда следует поместить информацию. Различают одноадресные, двухадресные и трехадресные команды (рис. 1.5). Могут быть и безадресные команды. В этом случае команда содержит только код операции, а адреса заранее записываются в определенные регистры процессора. Рассмотрим, как может выполняться операция сложения с помощью трехадресной команды. В этом случае код операции содержит указание процессору выполнить операцию сложения. Адрес первого слагаемого (операнда) содержится в ячейке с адресом А1, адрес второго слагаемого (операнда) хранится в ячейке с адресом А2, а в ячейке с адресом А3 хранится адрес, куда следует поместить результат вычисления. Для выполнения операции сложения с помощью одноадресной команды потребуется три команды. То есть, чем больше адресов содержится в адресной части команды, тем быстрее будут выполняться вычислительные операции.

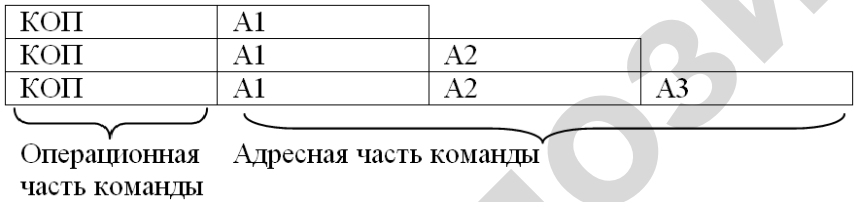


Рис. 1.5. Представление команд в ЭВМ

Все машинные команды можно разделить на несколько групп по виду выполняемых операций:

операции пересылки данных;
арифметические операции;
логические операции;
операции обращения к внешним устройствам;
операции передачи управления;
обслуживающие и вспомогательные операции.

Повышение разрядности ЭВМ позволяет разработчикам программного обеспечения разрабатывать более быстрые алгоритмы выполнения вычислительных и логических операций, что наряду с повышением тактовой частоты способствует увеличению быстродействия ЭВМ.

Контрольные вопросы

1. Что представляет собой информационное сообщение, в каком виде оно передается?
2. Опишите принцип работы системы передачи информации.
3. Что такое код? Какое количество кодовых комбинаций необходимо для кодирования всех символов русского алфавита, белорусского алфавита?
4. Что такое разрядная сетка ЭВМ? Чем она определяется?
5. Какие форматы используются для представления: а) целых; б) вещественных чисел?
6. Сколько кодовых комбинаций можно представить с помощью одного байта?
7. В чем состоит особенность представления графической информации?
8. Что такое фонема? Сколько фонем необходимо для представления звуков русского алфавита?
9. Какие устройства используются для воспроизведения речи?
10. Каким образом представляются команды в ЭВМ?

1.1.4. Системы счисления

Ключевые слова: базис, динамический диапазон, основание системы счисления, системы счисления.

Для хранения и передачи информации человечество на протяжении своего исторического развития разработало различные системы знаков. Но машина не понимает эту систему знаков. Ее этому надо «научить». В силу конструктивных особенностей цифровые ЭВМ способны различать сигналы разного уровня в виде нулей и единиц,

которые можно использовать для кодирования информации. Для представления данных в ЭВМ нашли применение различные системы счисления.

Под системой счисления понимают совокупность приемов и правил для записи чисел цифровыми знаками.

Различают позиционные и непозиционные системы счисления.

Примером *непозиционной* системы счисления является римская система, использующая набор символов I, V, X, C, L, D. В этой системе значение символов не меняется в зависимости от положения их в числе. Так, в числах VI и IV символ I имеет одно и то же значение – 1, в числах LX и XL символ X также принимает одно и то же значение – 10, но в первом случае это число прибавляется к 100, во втором случае вычитается из 100.

В *позиционной* системе счисления значение цифры определяется положением в числе: один и тот же знак принимает различные значения. Примером позиционной системы счисления является десятичная система счисления, известная всем со школьной скамьи. Например, в числе 222 первая цифра слова означает число двести, вторая – двадцать, третья – два. Вес числа возрастает в 10 раз при движении справа налево. Рассмотрим вещественное число 135,28:

$$135{,}28 = 100 + 30 + 5 + 0{,}2 + 0{,}08 = \\ = 1 \cdot 10^2 + 3 \cdot 10^1 + 5 \cdot 10^0 + 2 \cdot 10^{-1} + 8 \cdot 10^{-2}.$$

Приведенный пример записи вещественного числа в десятичной системе счисления позволяет установить общую форму записи чисел в позиционной системе счисления. Произвольное число A можно представить в виде полинома:

$$A(q) = a_{n-1}q^{n-1} + a_{n-2}q^{n-2} + \ldots + a_i q^i + \ldots + a_1 q^1 + a_0 q^0 + \\ + a_{-1}q^{-1} + \ldots + a_{-m}q^{-m}, \qquad (1.1)$$

где a_i – символы алфавита системы счисления;

q – основание системы счисления;

n, m – число целых и дробных разрядов соответственно.

Пример 1.1. Определите значение числа, представленного двоичным кодом 101101.011_2.

Решение. Число содержит целую и дробную части $n = 6$, $m = 3$:

$$A(2) = 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2} + \\ + 1 \cdot 2^{-3} = 1 \cdot 3\,2 + 0 \cdot 16 + 1 \cdot 8 + 1 \cdot 4 + 0 \cdot 2 + 1 \cdot 1 + 0 \cdot 1/2 + \\ + 1 \cdot 1/4 + 1 \cdot 1/8 = 32 + 8 + 4 + 1 + 1/4 + 1/8 = 45{,}375.$$

Любая позиционная система счисления характеризуется *основанием* (*базисом*). Основание позиционной системы счисления – это число знаков или символов для изображения цифр в данной системе.

В настоящее время в вычислительной технике нашли наибольшее применение позиционные системы счисления, приведенные в таблице 1.1.

Таблица 1.1

Позиционные системы счисления

Наименование	Алфавит	Основание системы счисления
Двоичная	0, 1	2
Восьмеричная	0, 1, 2, 3, 4, 5, 6, 7	8
Десятичная	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	10
Шестнадцатеричная	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F	16

Если задано число $A_{\max}(q)$, то можно определить требуемое число разрядов для его представления – n:

$$n = \log_a(A_{\max}(q) + 1). \qquad (1.2)$$

И наоборот, если известна длина разрядной сетки n, то можно определить максимальное число $A_{\max}(q)$, которое можно представить с использованием данной разрядной сетки:

$$A_{\max}(q) = q^n - 1. \qquad (1.3)$$

Например, с помощью одного байта (восьми разрядов) можно представить число 255:

$$2^8 - 1 = 256 - 1 = 255,$$

а с помощью 15 разрядов – число 32 767:

$$2^{15} - 1 = 32\,768 - 1 = 32\,767.$$

То есть числа заключены между некоторым минимальным значением и некоторым максимальным значением. Интервал числовой оси, заключенный между максимальным и минимальным числами, называют *динамическим диапазоном*.

Современные вычислительные машины, как известно, оперируют в основном только двоичными знаками, поэтому числа, представленные в других системах счисления, ЭВМ предварительно перево-

дит в двоичную систему счисления. С клавиатуры числа вводятся, как правило, в десятичной системе счисления, но могут вводиться также в восьмеричной или шестнадцатеричной системах счисления. При выводе числовой информации на экран ЭВМ производит обратные преобразования, т. е. переводит числа из двоичной системы счисления в десятичную, восьмеричную или шестнадцатеричную системы счисления. Указанные преобразования осуществляются в соответствии с определенными алгоритмами. Правила перевода приведены в таблицах 1.2 и 1.3.

Таблица 1.2

Перевод чисел из десятичной системы счисления в двоичную систему счисления и обратно

Перевод целых чисел	Перевод дробных чисел
Правило перевода. Разделить число на основание системы счисления. Остаток от деления записать в виде кода двоичного числа. Целую часть полученного числа снова разделить на основание системы счисления. Вверху – младший разряд, внизу – старший разряд. 29 : 2 = 14 ↑ 1 – мл. разряд 14 : 2 = 7 | 0 7 : 2 = 3 | 1 3 : 2 = 1 | 1 1 : 2 = 0 | 1 – ст. разряд Код числа: 11101	Правило перевода. Умножить число на основание системы счисления. Целая часть полученного числа образует код двоичного числа, а остаток используется для последующего умножения. Вверху – старший разряд, внизу – младший. Умножение продолжается до достижения требуемой точности. 0.35 · 2 = 0.70 | 0 – ст. разряд 0.70 · 2 = 1.4 | 1 0.4 · 2 = 0.8 | 0 0.8 · 2 = 1.6 | 1 0.6 · 2 = 1.2 | 1 0.2 · 2 = 0.4 ↓ 0 – мл. разряд Код числа: 01011

Для перевода двоичных чисел в восьмеричные числа двоичное число делится на триады, начиная с младшего разряда, каждая из которых переводится в соответствующее восьмеричное число:

$$110\ 111\ 001\ 101\ 010_2 = 67152_8.$$

При обратном переводе каждое число восьмеричного кода заменяется тремя двоичными знаками:

Таблица 1.3

Перевод чисел из двоичной системы счисления в десятичную и обратно

Перевод целых чисел	Перевод дробных чисел
Правило перевода. Умножить старший разряд на основание системы счисления и прибавить к полученному результату следующий разряд. Полученное число снова умножить на основание системы счисления и прибавить следующий разряд. 1 = 1 | – ст. разряд 1 · 2 + 1 = 3 3 · 2 + 1 = 7 7 · 2 + 0 = 14 14 · 2 + 1 = 29 ↓ – мл. разряд	Правило перевода. Разделить младший разряд на основание системы счисления. Прибавить к полученному числу следующий разряд и разделить результат на основание системы счисления. Прибавить к полученному результату следующий разряд и снова разделить полученный результат на основание системы счисления. 1 : 2 = 0.5 ↑ – мл. разряд (0.5 + 1) : 2 = 0.75 (0.75 + 0) : 2 = 0.375 (0.375 + 1) : 2 = 0.6875 (0.6875 + 0) : 2 = 0.34375 | – ст. разряд

Примечание. Перевод десятичных чисел в восьмеричную и шестнадцатеричную системы счисления и обратно осуществляется по приведенным выше правилам.

$$731254_8 = 111\ 011\ 001\ 010\ 101\ 100_2.$$

Для перевода двоичных чисел в шестнадцатеричные числа двоичное число делится на тетрады, начиная с младшего разряда, каждая из которых переводится в соответствующее шестнадцатеричное число:

$$0110\ 1110\ 0110\ 1010_2 = 6E6A_{16}.$$

При обратном переводе каждое число шестнадцатеричного кода заменяется четырьмя двоичными знаками:

$$5D6B_{16} = 0101\ 1101\ 0110\ 1011_2.$$

Контрольные вопросы

1. Что такое система счисления?
2. Что называется основанием (базисом) системы счисления?
3. Какая система счисления называется позиционной системой счисления?

4. Представьте число $172{,}356_{10}$ в виде полинома.

5. Какое максимальное число можно представить с помощью двух (трех, восьми, шестнадцати) разрядов?

6. Как перевести число из двоичной системы счисления в восьмеричную (шестнадцатеричную) систему счисления?

7. Сформулируйте правило перевода десятичных чисел в двоичную (восьмеричную, шестнадцатеричную системы счисления).

8. Сформулируйте правило перевода двоичных (восьмеричных, шестнадцатеричных) чисел в десятичные.

1.1.5. История и перспективы развития вычислительной техники

Электронно-вычислительная машина (ЭВМ) представляет собой устройство, предназначенное для выполнения вычислительных и логических операций, а также обработки информации.

В своем развитии ЭВМ прошли достаточно большой путь от замысла до воплощения в реальные машины. В развитии вычислительной техники принято выделять ряд этапов:

ручной (начало не установлено);

механический (с середины XVII века);

электромеханический (с 90–х годов XIX века);

электронный (с 40-х годов XX века).

Подробные сведения об этих этапах можно найти в соответствующей литературе [17]. Ручной, механический и электромеханический этапы развития вычислительной техники создали предпосылки для создания ЭВМ. Но отсутствие соответствующей элементной базы, технологий не позволяло создать полностью автоматическую вычислительную машину. И только в начале XX века была построена первая вычислительная машина. В 1936 году немецкий инженер *Конрад Цузе* начал конструировать вычислительный аппарат, работающий в двоичной системе счисления. В 1941 году он сумел построить действующую модель такого устройства (*Zuse 3*), которая состояла из 600 реле счетного устройства и 2000 реле устройства памяти. В 1943–44 годах в Англии было разработано полностью автоматическое устройство Колосс II, предназначенное для дешифровки военных сообщений. Еще одна полностью автоматическая вычислительная машина была изобретена и построена *Говардом Эйкеном*, профессором Гарвардского университета (США) при участии группы инженеров фирмы IBM. Эта машина называлась *Марк 1*,

она состояла из 750 тысяч компонентов, на операцию умножения затрачивала около 4 секунд.

Новые возможности по созданию вычислительных машин открылись с появлением электронных ламп и последующим бурным развитием электроники. Это новый период развития вычислительной техники. Он делится на этапы, непосредственно связанные с уровнем развития элементной базы электронной техники, конструктивно-технологическим исполнением, логической организацией, математическим обеспечением, удобством общения человека с машиной. Смена поколений ЭВМ происходила революционно, ей сопутствовало изменение технико-экономических показателей этих машин: быстродействие, надежность, потребляемая мощность, стоимость, габариты. Изменение периферийного оборудования и программного обеспечения шло в основном эволюционным путем.

В настоящее время выделяют шесть этапов (поколений) в развитии электронной вычислительной техники, связанных с развитием элементной базы и промышленных технологий[2]:

- ЭВМ на электронных лампах (1945–1956 гг.);
- ЭВМ на дискретных полупроводниковых и магнитных элементах (диоды, биполярные транзисторы, тороидальные ферритовые микротрансформаторы и ферромагнитные ячейки памяти) (1956–1964 гг.);
- ЭВМ на интегральных элементах малой плотности (1964–1971 гг.);
- ЭВМ на микропроцессорных элементах (1971–1990 гг.);
- ЭВМ на сверхбольших ИС и многопроцессорные системы (с 1990 по настоящее время);
- ЭВМ на новой элементной базе и новых принципах работы (настоящее и будущее).

Появлению первых ЭВМ предшествовали такие фундаментальные изобретения, как электронная лампа (1879), триод (1913). Триод, в отличие от двух электродной лампы, имеет еще один электрод – сетку. Благодаря наличию этого электрода появилась возможность управлять потоком электронов в лампе и создавать на их основе элементы памяти. Работы по созданию отдельных элементов и узлов ЭВМ были начаты в 1937 г. в США *Дж. Атанасовым*. Им были за-

[2] В разных источниках приводятся различные даты начала и конца соответствующего периода. Отдельные авторы отходят от данного принципа периодизации, в результате чего границы этапов существенно смещаются.

патентованы первые электронные схемы отдельных узлов ЭВМ. В 1942 г. им совместно с *К. Берри* была построена электронная машина *ABC*. Первая ЭВМ полностью на электронных лампах была названа *ENIAC* (электронный числовой интегратор и вычислитель). Она была изобретена Эккертом и Маучли и создана в США в 1946 году.

Основные этапы развития ЭВМ приведены в таблице 1.4.

На этапах с первого по третий большой вклад в развитие вычислительной техники внесли советские ученые: С. А. Лебедев, И. С. Брук. Под их руководством были созданы ЭВМ первого поколения МЭСМ – 1951 год (малая электронная счетная машина), БЭСМ – 1952–1953 гг. (большая электронная счетная машина). К машинам первого поколения можно отнести МЭСМ, БЭСМ, М-1, М-2, М-3, «Стрела», «Минск-1», «Урал-1», «Урал-2», «Урал-3», М-20, «Сетунь», БЭСМ-2, «Раздан». ЭВМ «Сетунь» была построена в 1953 году Н. П. Бруснецовым. В ней использовалась *троичная система счисления*, которая в ряде случаев позволяла создавать более эффективные программы.

В развитие машин второго поколения большой вклад внес В. М. Глушков. К машинам второго поколения относятся следующие отечественные ЭВМ: М-40, М-50, «Урал-11», «Урал-14», «Урал-16», «Минск-2», «Минск-22», БЭСМ-3, БЭСМ-4, БЭСМ-6, М-20, М-220, М-222, «МИР-1».

В разработку машин третьего поколения большой вклад внесли ученые и конструкторы М. А. Карцев, Б. И. Рамаев, Ю. А. Базилевский, Б. Н. Малиновский и др. К ЭВМ третьего поколения относятся машины Единой Системы (ЕС) и малые ЭВМ (СМ): «Днепр-2», ЕС-1010, ЕС-1020, ЕС-1030, ЕС-1040, ЕС-1050, ЕС-1060, «МИР-2», «Наири-2».

К ЭВМ четвертого поколения относится ряд ЭВМ: ЕС-1015, ЕС-1025, …, ЕС-1065, ЕС-1036, ЕС-1046, ЕС-1066; малые ЭВМ: СМ-1420, СМ-1600, СМ-1700; персональные ЭВМ «Электроника МС 0501», «Электроника-85», «Искра-226», ЕС-1840, ЕС-1841, ЕС-1842, ЕС-1845, ЕС-1861 и др.

Однако уже в этот период стало заметно отставание СССР в области микроэлектроники от стран Запада. А с развалом Советского Союза в 1991 году производство собственных ЭВМ для бытовых целей полностью прекратилось, так как продукция отечественной промышленности не смогла конкурировать с ЭВМ, производимыми на Западе, ни по стоимости, ни по надежности, ни по производительности.

Таблица 1.4

Основные этапы развития электронной вычислительной техники

Поколение ЭВМ	Предшествующие научные открытия	Год начала этапа	Элементная база	Назначение или тип	Новые свойства	Программное обеспечение
1	1879 – изобретена электронная лампа, 1913 – триод	1945	Электронные лампы	Инженерно-технические расчеты	Программное управление, машинный язык	Операционная система практически отсутствовала. 1949 – создан первый язык программирования Short Code. Языки программирования высокого уровня: Фортран – 1957, Лисп – 1956, Кобол – 1959, АЛГОЛ – 1960
2	1948 – изобретен транзистор	1956	Полупроводниковые приборы	Обработка данных, управлениемировательные техническими объектами	Языки программирования, широкая периферия	Операционные системы для работы с накопителями на магнитных барабанах и магнитных лентах
3	1956 – изобретен жесткий диск, 1958–1959 – ИС	1964	Интегральные микросхемы (ИС)	СуперЭВМ, малые ЭВМ, настольные ЭВМ	Программная совместимость, модульный принцип организации технического и программного обеспечения	Новые языки программирования: Бейсик – 1964, ПЛ/1 – 1964. Языки программирования: Паскаль – 1967, Симула –1967, ЛОГО – 1968, 1968 – текстовый процессор

Окончание табл. 1.4

Поколение ЭВМ	Предшествующие научные открытия	Год начала этапа	Элементная база	Назначение или тип	Новые свойства	Программное обеспечение
4	1967 – идея МП на одном кристалле	1971	Микропроцессоры (МП)	Персональные, профессиональные ЭВМ	Децентрализация вычислений	Операционная система UNIX, системы программирования С – 1972, ПРОЛОГ – 1978, 1973 – операционная система для ПК CP/M
5		1979	Сверхбольшие интегральные схемы	Экспертные системы	Искусственный интеллект	1981 – операционная система MS-DOS, языки программирования Ада –1979, C++ – 1980, HTML – 1989, *Delphi* – 1995, *Java* – 1995
			Задача разработки ЭВМ пятого (шестого) поколения сформулирована в 1979 году в Японии. Ведутся разработки. Имеются элементы с биологическими принципами обработки информации, нейронные сети. Проводятся научные исследования. Современная элементная база основана на использовании кристаллов кремния. Проводятся исследования по созданию элементной базы на основе углерода			

В начале нового столетия наметился определенный сдвиг в развитии собственной элементной базы. В России в середине 2001 года был введен в строй 768-процессорный суперкомпьютер МВС-1000М, обеспечивавший производительность в 1 Терафлоп. После этого Россия вышла на третье место в мире по мощности производимых суперкомпьютеров. В последующие годы совместными усилиями российских и белорусских ученых создан ряд СуперЭВМ серии СКИФ [2], входящих в первую сотню мирового рейтинга компьютеров по производительности. Самый мощный в России, СНГ и Восточной Европе суперкомпьютер «СКИФ МГУ» занимает 22-е место в мировом рейтинге суперкомпьютеров TOP-500. Пиковая производительность суперкомпьютера «СКИФ МГУ» составляет 60 триллионов операций в секунду (60 Терафлоп). Это в 10 раз меньше, чем у лидера списка TOP-500 американского суперкомпьютера LLNL с пиковой производительностью 596 378 Терафлоп, и примерно в 3,5 раза меньше пиковой производительности немецкого *FZ Juelich* с пиковой производительностью 222 822 Терафлоп.

ЭВМ пятого поколения не связаны с изменением элементной базы. В основу периодизации здесь для отличия их от ЭВМ четвертого поколения положены особенности архитектуры и организация вычислительного процесса. ЭВМ пятого поколения характеризуются наличием сверхсложных микропроцессоров с параллельно-векторной структурой, а также СуперЭВМ, содержащих в своей структуре сотни параллельно работающих процессоров, позволяющих строить системы обработки данных и знаний, эффективные сетевые компьютерные системы.

Современный этап развития ЭВМ можно охарактеризовать как этап развития машинного интеллекта. Вычислительные системы будущего будут ориентированы на обработку знаний и должны располагать развитыми возможностями логического вывода. Важнейшая черта их должна состоять в том, чтобы используемый интерфейс был непосредственно рассчитан на человека. Главными особенностями машин будущего будут речевой ввод-вывод информации и самообучаемость.

Технический базис ее должна составить развивающаяся технология сверхбольших интегральных схем, создание памяти повышенного объема, возрастающие возможности высокоскоростных элементов.

Основу архитектуры должны составить системы с распределительными функциями, сетевая архитектура, машина базы данных,

быстродействующая машина для численных расчетов, высокоуровневая система человеко-машинного общения.

Основными системами программного обеспечения должны стать системы управления базами знаний, системы решения проблем и логического вывода, системы интеллектуального интерфейса.

Основными прикладными системами могут стать системы машинного перевода, вопросно-ответная система, прикладные системы понимания речи, изображений, рисунков, прикладные системы решения проблем.

О практических результатах в области разработки машины шестого поколения говорить пока рано. Первыми практическими результатами в области искусственного интеллекта стали экспертные системы, с помощью которых достигнуты значительные результаты в области медицины, геологии, технической диагностики.

Невиданные успехи в области миниатюризации уже в последнее десятилетие привели к колоссальному росту объемов памяти и быстродействию вычислительной техники, а также к коренному изменению взглядов на ее использование. Ученые предполагают создание голографической памяти, которая позволит хранить терабайты информации на кубический дюйм. Кассеты для нового поколения цифровых видеомагнитофонов смогут хранить более 100 Гигабайт информации, то есть на одну-единственную ленту удастся записать все разговоры, которые человек ведет на протяжении жизни. Постепенное развитие компьютеров и технологии производства мониторов приведет к созданию почти невесомой, универсальной электронной книги. В коробке размером с обыкновенную книгу будет находиться дисплей, способный показывать текст, картинки и видеоматериалы с высоким разрешением. Перелистывать страницы можно будет пальцем или отдавать команды голосом. Современные карманные персональные компьютеры и смартфоны уже позволяют выполнять многие операции.

Будущее развитие вычислительной техники связывают с вычислительными сетями. Техническую основу их составят электронные и оптоэлектронные компьютеры с массовым параллелелизмом, нейронной структурой и распределенной сетью микропроцессоров, моделирующих архитектуру нейронных биологических систем. Предполагают, что глобальные сети превратятся в универсальный рынок и центральный универмаг всего мира. Проникновение вычислительной техники во все сферы человеческой деятельности

изменит быт людей, социальные отношения. Появятся электронные банки, и клиенты банка смогут оплачивать счета, заказывать и приобретать товары и услуги, пользуясь электронным бумажником. Уже в настоящее время реализуется концепция «Цифровой дом». Эта концепция имеет в виду создание комплекса взаимодействующих домашних и офисных устройств, предоставляющих возможности интерактивного доступа в любой момент времени к любой информации, необходимой для работы и отдыха. Будут созданы системы биометрической защиты, которые смогут опознавать человека по отпечаткам пальцев, цвету радужной оболочки глаза, голосу. Системы виртуальной реальности позволят архитекторам видеть, например, устройство квартиры или офиса как бы изнутри. В костюмах виртуальной реальности, снабженных миллионами сенсорных датчиков, человек сможет путешествовать в космосе, в пустыне Сахаре или джунглях Амазонки и ощущать космическую невесомость, зной пустыни или прикосновение лиан.

Успехи в области электроники последнего десятилетия позволяют с оптимизмом смотреть в будущее, все, что сказано выше, – не пустые фантазии, а вопрос времени, и, возможно, не такого уж далекого.

Контрольные вопросы

1. Назовите основные этапы развития вычислительной техники.
2. Назовите ученых, которые внесли существенный вклад в развитие вычислительной техники.
3. Что положено в основу периодизации развития электронной вычислительной техники?
4. Какие научные открытия предшествовали появлению первых вычислительных машин на электронных лампах?
5. Назовите особенности и направления развития вычислительной техники шестого поколения.

1.2. УСТРОЙСТВО И РАБОТА ЭВМ

Ключевые слова: видеомонитор, видеокарта, дополнительные устройства, интерфейс, классификация, компьютер, микропроцессор, память, принтер, принцип программного управления, процессор, электронная вычислительная машина – ЭВМ.

1.2.1. Классификация ЭВМ

В настоящее время создан большой парк ЭВМ различного назначения. По мере увеличения количества ЭВМ, изменения их технических характеристик расширялась и область их применения: от стационарных ЭВМ большой производительности до микропроцессоров, встраиваемых в бытовую технику, от ЭВМ для управления космическим аппаратом до бытового компьютера. Можно выделить следующие основные классификационные признаки: вид обрабатываемой информации (принцип действия), вычислительная мощность, назначение, уровень организации, конструктивно-технологическое исполнение.

По виду обрабатываемой информации (принципу действия) ЭВМ делятся на три больших класса: аналоговые, цифровые и гибридные. Аналоговые ЭВМ служат для обработки медленно меняющихся сигналов тока или напряжения. Цифровые ЭВМ обрабатывают информацию, поступающую на их входы в дискретной форме или в виде цифровых кодов. Гибридные ЭВМ могут работать с информацией, представленной и в цифровой, и в аналоговой форме; они сочетают в себе достоинства и аналоговых, и цифровых ЭВМ. Область применения гибридных ЭВМ – управление сложными быстродействующими техническими комплексами. В дальнейшем будут рассматриваться только цифровые ЭВМ, которые для краткости будем называть просто ЭВМ.

По вычислительной мощности ЭВМ делятся на суперкомпьютеры, мэйнфреймы, мини-компьютеры и микрокомпьютеры (супербольшие, большие, малые, сверхмалые).

Суперкомпьютеры обладают высоким быстродействием и имеют огромные вычислительные мощности. Они используются для сложных расчетов в аэродинамике, метеорологии, космических и физических исследованиях, экономике и финансовом управлении.

Мэйнфреймы обладают значительными ресурсами для решения сложных задач в финансовой области, в управлении регионами, отраслями промышленности, большими предприятиями, в том числе предприятиями торговли, в военной области.

Миникомпьютеры используются для управления предприятиями и организациями. К ним относятся *серверы* старшего уровня, используемые для управления локальными компьютерными сетями.

Микрокомпьютеры – это самые массовые модели вычислительных машин. К ним относятся персональные компьютеры

(ПК), рабочие станции. Рабочие станции используются в локальных вычислительных сетях, в том числе и в учебных заведениях.

По назначению (или специализации) ЭВМ делятся на вычислительные машины общего применения (универсальные), специализированные (проблемно-ориентированные), информационно-вычислительные, управляющие, персональные, проблемно-ориентированные процессоры. Основными параметрами, по которым различаются подклассы определенных средств вычислительной техники, являются разрядность, производительность и тип системного интерфейса.

Универсальные ЭВМ ориентированы на решение широкого круга задач, причем во всех классах задач они развивают одинаковую производительность. Эти ЭВМ имеют архитектуру, позволяющую подключать разнообразные периферийные устройства, варьировать их число и технические параметры, обеспечивать различные виды обработки данных и режимы взаимодействия с пользователями.

Специализированные ЭВМ предназначены для решения определенного класса конкретных прикладных задач либо для ограниченных сфер применения и, как правило, отличаются от универсальных ЭВМ рядом ограничений на виды обработки информации и/или возможности подключения периферийных устройств. Они оснащаются специальным программным обеспечением. Например, для обработки информации в геологических партиях при разведке полезных ископаемых.

Информационно-вычислительные ЭВМ ориентированы на обработку больших объемов текстовой (справочной) информации.

Управляющие ЭВМ применяются для управления производством, техническими системами и технологическими процессами.

Персональные ЭВМ (ПЭВМ) предназначены для облегчения труда инженеров, ученых, все настойчивее проникают в наш быт, ориентированы на использование отдельным пользователем. В настоящее время за этим классом ЭВМ закрепилось название *персональный компьютер* (ПК). По функциональным возможностям персональный компьютер относится к универсальным ЭВМ.

Проблемно-ориентированные процессоры подключаются, как правило, к персональным ЭВМ и служат для повышения быстродействия при решении задач вычислительной математики, преобразования и обработки аналого-цифровой информации.

По уровню организации различают однопроцессорные (автономные) ЭВМ, вычислительные системы и вычислительные комплексы, сети ЭВМ (см. раздел 7).

Вычислительные системы – совокупность взаимосвязанных или взаимодействующих процессов или ЭВМ, периферийного оборудования, программного обеспечения, предназначенного для автоматизации процессов приема, хранения, обработки и выдачи информации. ЭВМ в вычислительной системе работают под управлением одной операционной системы.

Вычислительные комплексы отличаются от вычислительных систем тем, что каждая из ЭВМ имеет свою оперативную память и работает под управлением своей операционной системы. Вычислительные комплексы применяются в автоматизированных системах управления технологическими процессами для повышения производительности и надежности управления. Они отличаются от однопроцессорных ЭВМ повышенными характеристиками надежности и готовности, а также концентрацией вычислительных мощностей в сочетании с их лучшим использованием по сравнению с набором независимых ЭВМ.

Сетью ЭВМ называется совокупность ЭВМ, взаимосвязанных и распределенных по некоторой территории.

Признак ***конструктивно-технологического исполнения*** можно разделить на два дополнительных признака: место установки и число плат.

По месту установки ЭВМ делятся на: стационарные, бортовые, настольные, переносные, встраиваемые, а *по числу плат* ЭВМ делятся на многоплатные и одноплатные.

Стационарные ЭВМ размещаются в специально оборудованных помещениях, где создаются необходимые условия по температурно-влажностному режиму и уровню шумов.

Бортовые ЭВМ размещаются на морских, речных, воздушных судах, космических аппаратах. К ним предъявляются повышенные требования по ударным нагрузкам, вибростойкости, устойчивости к ионизирующим излучениям, надежности.

Переносные ЭВМ размещаются в специальных чемоданах, сумках. Одной из разновидностей переносных ЭВМ являются карманные компьютеры.

Встраиваемые ЭВМ не оформляются в виде самостоятельных приборов, а встраиваются непосредственно в оборудование (станки, приборы, узлы, агрегаты), образуют с ними конструктивное целое.

Одноплатные ЭВМ имеют сравнительно небольшой объем памяти. Расширение их возможностей идет за счет добавления модулей памяти, а также устройств, обеспечивающих связь с внешними

устройствами (контроллеров), источников питания. Названные устройства выполняются, как правило, в виде отдельных плат. За счет дополнительных устройств одноплатные ЭВМ превращаются в *многоплатные*.

1.2.2. Структура и принцип работы ЭВМ

Структура ЭВМ

Типичная ЭВМ состоит из четырех основных частей: процессора, памяти, устройств ввода и вывода информации. Все составные части ЭВМ связаны между собой шинами адреса, данных и управления. Обобщенная структурная схема фон-неймановской ЭВМ приведена на рисунке 1.6.

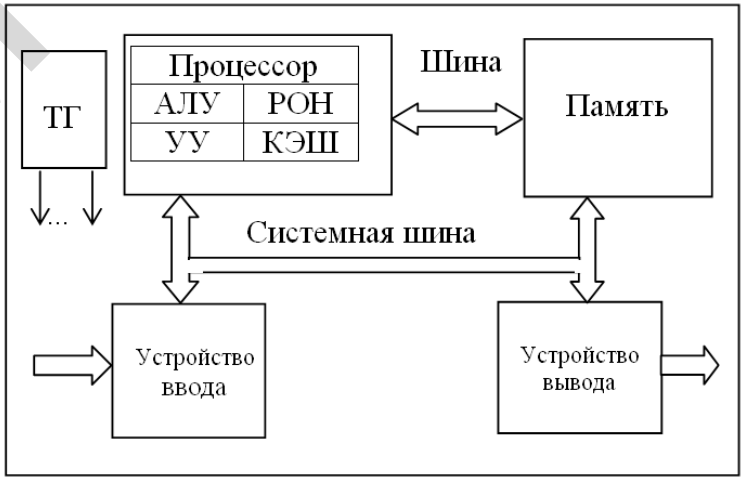


Рис. 1.6. Упрощенная структура компьютера

Тактовый генератор (ТГ) формирует тактовые импульсы, синхронизирующие работу всех устройств компьютера. Ввиду того, что быстродействие различных устройств ЭВМ различно, генератор тактовых импульсов формирует сигналы на нескольких частотах.

Процессор является ядром вычислительной системы и предназначен для обработки информации и управления работой всей системы. Основными его узлами являются арифметико-логическое

устройство (АЛУ), устройство микропрограммного управления (УУ), регистры общего назначения (РОН) и кэш-память.

Арифметико-логическое устройство предназначено для выполнения простейших операций: сложения, сдвига, логических операций И, ИЛИ, исключающего ИЛИ. Составив соответствующую программу, на основе этих операций можно выполнять и более сложные операции, такие как умножение, возведение в степень и т. д. АЛУ связано с регистрами, аккумулятором (специальный однобайтовый регистр) и устройством управления.

Устройство управления является одним из важнейших узлов процессора. Совместно с генератором тактовых импульсов оно обеспечивает правильную последовательность выполнения всех операций в ЭВМ. После извлечения команды из памяти и ее дешифрации устройство управления генерирует последовательность сигналов, необходимую для выполнения команды. Он выполняет и другие функции, например, прерывание программы.

Регистры имеют разное назначение, но, независимо от этого, их общая функция состоит во временном хранении информации: команд, адресов, данных или признаков состояния АЛУ.

Кэш-память – оперативная память компьютера первого уровня. Служит для хранения наиболее часто используемых программ и данных. Кэш-память управляется процессором.

Память компьютера служит для хранения программ и данных. Она состоит из ячеек, связанных с шинами данных и адреса (рис. 1.7).

Ячейки памяти (элементы памяти) служат для хранения информации. Элементом памяти в компьютерах является обычно байт. Число ячеек в блоках памяти зависит от разрядности шины адреса. Например, при 8-разрядной шине адреса блок памяти может содержать 256 ячеек, а при 16-разрядной шине адреса число ячеек в блоке равно $2^{16} = 65\,536$, или 64 кбайт. Ячейкам присваиваются номера от 0 до 65 535. Ячейки памяти объединяются, как правило, в матрицы.

Селектор адреса состоит из селекторов строк и столбцов и предназначен для выбора ячейки с заданным адресом с целью передачи содержимого ячейки в шину данных или наоборот, записи данных в ячейку. По окончании селекции ячейки с заданным адресом он выдает в процессор по шине управления сигнал «Готов». При записи информации в память этот сигнал указывает на то, что данные могут вводиться в память. При чтении данных сигнал «Готов» разрешает процессору начать прием данных.

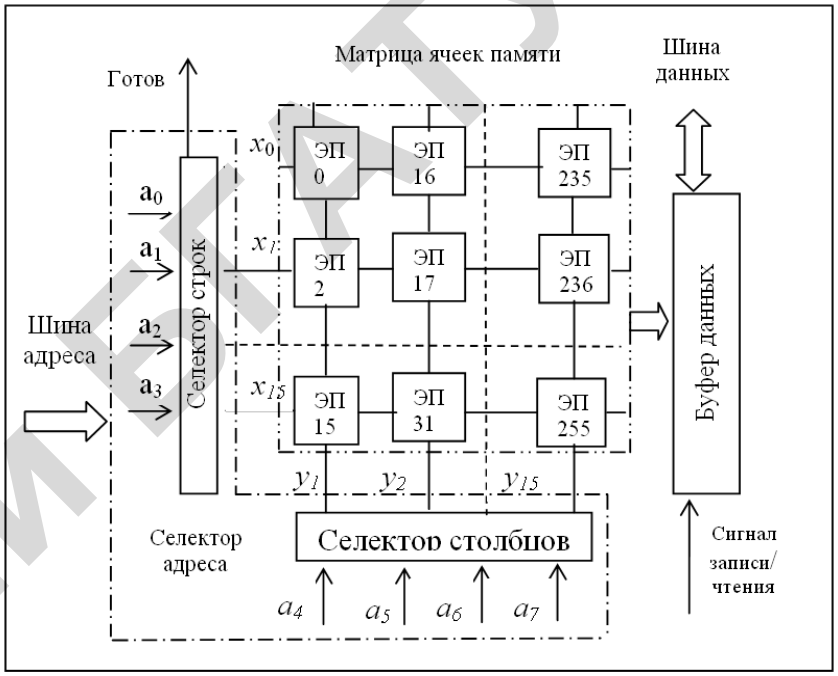


Рис. 1.7. Структура блока памяти

Буфер данных служит для временного хранения принимаемой или выдаваемой из памяти информации. После получения от селектора адреса сигнала «Готов» процессор выдает по шине управления команду на запись или чтение данных.

Шина представляет собой стандартный комплект токопроводящих линий с устройствами согласования и служит для передачи данных и адреса для выбора ячеек, а также сигналов управления процессора. Шина включает в себя шину адреса, шину данных и шину управления.

Шина адреса имеет 16 или 32 разряда, однонаправленная, служит для передачи адреса выбираемого элемента памяти или адреса *порта* ввода-вывода к соответствующим устройствам. *Шина данных* может быть двунаправленной и может иметь 8, 16, 32 или 64 разряда, в зависимости от разрядности процессора. Она служит для передачи данных от микропроцессора к устройствам ввода-вывода и памяти или от устройств ввода-вывода и памяти к процессору.

Шина управления двунаправленная. Она служит для передачи сигналов записи, чтения, тактовых сигналов, сигналов синхронизации от процессора к блокам памяти, устройствам ввода-вывода, а также для передачи сигналов готовности от указанных устройств процессору.

Устройства ввода обеспечивают связь процессора с источниками информации, согласуют входные сигналы по уровню напряжения и формату с уровнем сигналов микросхем, используемых в процессоре, обеспечивают гальваническую развязку электрических цепей источников информации и процессора, а также источников информации и устройств ввода. В качестве источников информации могут быть использованы машинные носители программ (перфокарты, перфоленты, магнитные диски и др.), клавиши устройств ввода информации, датчики состояния систем управления (температуры, давления, магнитных полей и др.).

Устройства вывода обеспечивают связь процессора с исполнительными устройствами и выполняют практически те же функции, что и устройства ввода. Отличие состоит в направлении передачи информации. В устройствах вывода информация передается от процессора к внешним устройствам. Кроме того, выходные устройства могут обеспечивать усиление выходных сигналов по напряжению и мощности до величины, необходимой для управления исполнительными устройствами. В качестве исполнительных устройств могут быть устройства вывода информации на машинные носители (магнитные диски, перфокарты, перфоленты и др.), дисплеи, графопостроители, индикаторы, исполнительные устройства систем управления техническими системами и технологическим оборудованием (реле, электроприводы, пусковая аппаратура систем электроснабжения), устройства формирования звуковых сигналов и др.

Устройства ввода-вывода состоят из механической части – собственно устройства – и электронного компонента, который называют *контроллером* или *адаптером*. Контроллеры имеют обычно набор регистров для обмена информацией с центральным процессором или памятью. Адреса регистров устройств ввода-вывода принято называть *портами*. В некоторых компьютерах адреса регистров устройств ввода-вывода являются частью физического адресного пространства процессора. В других компьютерах они образуют собственное адресное пространство за счет введения специальных операций ввода-вывода.

Принцип программного управления

В ЭВМ используется *принцип программного управления*. Один из способов его реализации был предложен в 1945 году американским математиком Д. Нейманом, и с тех пор неймановский принцип программного управления используется в качестве основного принципа построения персональных ЭВМ. Этот принцип состоит в следующем:

- Информация кодируется в двоичной форме и разделяется на единицы информации – слова.
- Разнотипные слова информации различаются по способу использования, но не по способам кодирования.
- Слова информации размещаются в памяти ЭВМ и идентифицируются номерами ячеек, которые называются номерами слов.
- *Алгоритм* представляется в виде последовательности управляющих слов – команд, которые определяют наименование операции и слова информации, участвующие в операциях. Алгоритм, представленный в терминах машинных команд, называется *программой*.
- Выполнение вычислений, предписанных алгоритмом, сводится к последовательному выполнению команд в порядке, однозначно определяемом программой. Первой выполняется команда, заданная пусковым адресом программы. Обычно это адрес первой команды программы. Адрес следующей команды однозначно определяется в процессе выполнения текущей команды и может быть либо адресом следующей по порядку команды, либо адресом любой другой команды. Процесс вычислений продолжается до тех пор, пока не будет выполнена команда, предписывающая прекращение вычислений.

В процессе работы микропроцессора возможны *прерывания*. Прерывания возникают в результате неисправностей аппаратуры, запросов подключенных внешних устройств на обслуживание, запросов выполняемых программ на обслуживание или при возникновении ошибок вычисления. Прерывания принято делить на логические, программные и аппаратные. *Аппаратные прерывания* инициируются аппаратурой, например, сигналом от принтера, нажатием клавиши на клавиатуре, сигналом таймера. *Логические прерывания* возникают при нестандартных ситуациях в работе микропроцессора, например, деление на ноль, переполнение регистра, несоответствия типов данных, отсутствие затребованного файла и др. *Программные прерывания* инициируются программой пользователя, когда работающая

программа хочет получить доступ к каким-либо ресурсам компьютера или другой программе. Каждое прерывание имеет уникальный номер, и с ним связана определенная программа обработки прерывания. Процессор обрабатывает прерывания в соответствии с их приоритетом. Сигнал прерывания приостанавливает работу процессора, состояние обрабатываемого процесса запоминается, и, в зависимости от кода прерывания, подключается модуль операционной системы, который обслуживает это прерывание. Затем операционная система начинает реализовывать функции, определяемые кодом прерывания. После окончания обработки прерывания процессор переходит к выполнению процесса, который выполнялся в момент поступления сигнала прерывания.

Основные характеристики ЭВМ

Характеристики ЭВМ определяют ее назначение, область применения и потребительские качества. К ним относятся следующие показатели.

Состав и типы подключаемых внешних устройств.

Тип процессора. Наибольшее распространение в персональных ЭВМ имеют процессоры *Pentium*, *Celeron*, *Sempron* фирмы *Intel*, *Athlon*, *Duro*n фирмы *AMD*, а в последнее время преимущественное распространение получают многоядерные процессоры этих же фирм.

Разрядность. Разрядность ЭВМ определяется разрядностью процессора и характеризует точность вычислений и производительность машины. Различают 8-, 16-, 32-, 64-разрядные ЭВМ.

Быстродействие - число элементарных операций, выполняемых в единицу времени (оп./с – *флоп*). Быстродействие определяется тактовой частотой задающего генератора. Первые ПК имели тактовую частоту 4, 8, 16 МГц. В настоящее время частота тактового генератора подошла к порогу 4 ГГц. Двуядерные процессоры имеют меньшую тактовую частоту, чем одноядерные процессоры, но производительность их выше за счет распараллеливания процесса вычисления.

Объем памяти компьютера определяет возможности ЭВМ по использованию современных пакетов прикладных программ. Максимальная оперативная память 32-х разрядных ЭВМ достигает 4096 Мбайт, кэш-память первого и второго уровней составляет 128–2048 кбайт.

Емкость внешних запоминающих устройств (ВЗУ) определяет объем хранимой и используемой информации. Емкость накопителей на жестком диске достигает 1500 и более Гбайт.

Программное обеспечение: операционная система, системы программирования, пакеты прикладных программ.

Массогабаритные характеристики. Они не являются определяющими.

Стоимость.

1.2.3. Устройство персонального компьютера

Состав блоков персонального компьютера

Персональные компьютеры выпускаются разными фирмами, но независимо от фирмы-производителя в состав современного ПК входят системный блок, видеомонитор, клавиатура, манипулятор типа мышь. Перечисленные блоки составляют базовую (минимальную) конфигурацию ПК. Для получения твердой копии документов необходимо также печатающее устройство (принтер) индивидуального или коллективного пользования.

Кроме перечисленных блоков к ПК могут подключаться дополнительные внешние устройства и модули, обеспечивающие профессиональную ориентацию компьютера.

Системный блок предназначен для размещения электронных плат, источников питания, накопителей на магнитных дисках, элементов управления. На лицевой панели блока могут быть размещены:

накопитель для компакт-дисков (CD-ROM) или DVD-дисков;

кнопки *Power* – включения питания;

индикаторы включения питания и работы накопителя на жестком диске. На лицевой панели могут размещаться также разъемы USB.

Внутри корпуса системного блока размещаются системная плата, платы адаптеров периферийных устройств, видеомонитора, накопитель на жестком диске (НМД), накопители на оптических дисках или магнитооптических дисках, блок питания, вентилятор для охлаждения блока питания.

Системная плата (материнская плата) представляет собой одноплатную МикроЭВМ и может, при необходимости, функционировать самостоятельно. На ней размещены:

микропроцессор с вентилятором для охлаждения микропроцессора – кулером;

чипсет – набор микросхем, отвечающих за работу с внутренними и внешними устройствами. Как правило, чипсет состоит из двух

микросхем: северный мост и южный мост. Северный мост осуществляет связь микропроцессора с оперативной памятью и видеокартой, а южный мост обеспечивает связь микропроцессора с жестким диском и другими внешними устройствами;

Кэш-память второго уровня, основная память, блок управления, программируемый системный таймер, 4-х канальный блок прямого доступа к памяти, адаптеры связи с динамиком и клавиатурой, системная шина;

аккумуляторная батарея, которая обеспечивает питание перепрограммируемого запоминающего устройства.

Системная шина является сложным устройством, представляющим собой набор проводников и устройств сопряжения с процессором, памятью и другими устройствами. К системной шине могут подключаться также дополнительные платы расширения, например, аудиоплаты для поддержки работы со звуком, сетевая карта для подключения к сети Интернет. Эти платы размещаются внутри компьютера.

Системная плата через разъемы системной шины, называемые *слотами*, соединяется с выполненными на отдельных печатных платах модулями дополнительного оборудования – *адаптерами*.

Адаптеры также являются сложными электронными устройствами, в состав которых могут входить и микропроцессоры. Адаптеры обеспечивают связь внешних устройств с центральным процессором.

Блок питания служит для обеспечения элементов электронных схем и дисководов стабилизированным и нестабилизированным напряжением требуемых номиналов.

На заднюю панель системного блока выведены разъемы, через которые подключаются внешние устройства компьютера: видеомонитор, принтер, клавиатура, манипулятор мышь и другие устройства. Персональные компьютеры могут работать с периферийными устройствами, имеющими связь по параллельному интерфейсу (*Centronics*) или последовательному интерфейсу (*RS-232C*, *USB*). ПК может иметь несколько разъемов для последовательного интерфейса.

Микропроцессор

Основные характеристики

Микропроцессор – функционально законченное программно управляемое устройство обработки информации, выполненное в виде нескольких больших или сверхбольших интегральных схем.

Микропроцессор (процессор) является ядром вычислительной системы и предназначен для обработки информации и управления работой всей системы. Процессоры выпускаются многими фирмами. Однако наибольшее распространение для бытовых компьютеров имеют процессоры американских фирм Intel и AMD. Основные характеристики процессоров приведены в таблице 1.5. К ним относятся тип микропроцессора, разрядность, тактовая частота, число команд. Плотность размещения элементов на кристалле (степень интеграции) характеризуется числом транзисторов, а также технологией изготовления – *технологический процесс*. Под *технологическим процессом* понимается наименьший размер одного элемента, например, транзистора, диода, конденсатора. Технология изготовления процессора влияет и на его быстродействие: чем меньше размеры элементов и расстояние между ними, тем быстрее передаются сигналы в процессоре и элементах памяти. Разрядность шин данных и адреса также влияет на скорость работы процессора. Увеличение разрядности шин данных и адреса ускоряет процесс выборки данных, а следовательно, повышает и скорость вычислений.

Таблица 1.5.

Характеристики процессоров

Модель (тип) МП	Разрядность, бит		Тактовая частота, МГц	Число команд	Технология, мк	Число транзисторов, тыс.	Год выпуска
	шины данных	шины адреса					
1	2	3	4	5	6	7	8
Процессоры фирмы Intel							
4004	4	4	0,75	45		2,3	1971
8080	8	8	4,77			10	1974
8086	16	16	4,77 и 8	134		29	1978
80286	16	24	6–12		1,5	130	1982
80386	32	32	16–33	240	1,5–1	275	1985
80486	32	32	33–100	240	1–0,8	1 200	1989
Pentium	64	32	75–200	240	0,8	3 100	1993
Pentium II	64	36	233–300	MMX	0,35	7 500	1997
Pentium III	64	36	450–600	MMX	0,18	9 500	1999
Pentium 4	64	36	1000–3600	SSE2	0,13–0,09	42 000	2000

Окончание табл. 1.5.

1	2	3	4	5	6	7	8
Процессоры фирмы AMD							
K5	64	32	75–166		0,6–0,35		–
K6	64	32	166–233		0,35–0,25		1997
Athlon	64	32	3500				2004

Из анализа данных таблицы 1.5 можно сделать вывод о том, какими темпами развивалась микропроцессорная техника с момента появления первого компьютера. В 2006 году промышленно выпускались и реализовывались через торговую сеть процессоры Celeron и Sempron с тактовой частотой до 2800 МГц, Pentium 4 и Athlon с частотой до 3500 МГц.

Процессоры 8086, 80286 принято обозначать i86, а процессоры старшего ряда: 80386, 80486, Pentium – i386.

Быстродействие компьютера принято определять числом элементарных операций, выполняемых в единицу времени (оп./с). Быстродействие зависит от многих факторов и в первую очередь от тактовой частоты задающего генератора. Если первые ПК имели тактовую частоту 4, 8, 16 МГц, то в настоящее время частота тактового генератора подошла к порогу 4 ГГц. Например, в ноябре 2000 г. был выпущен процессор Pentium 4 с тактовой частотой 1,5 ГГц, изготовленный по 0,18-микронной технологии. А в настоящее время уже выпускаются процессоры данного типа с тактовой частотой 3,8 ГГц.

Однако дальнейшее повышение тактовой частоты связано с определенными трудностями: повышение тепловыделения и трудности отвода тепла от процессора. Для разрешения этих проблем перешли к созданию многоядерных процессоров. Это позволяет добиваться высокой производительности процессора при меньшей тактовой частоте.

Кроме повышения тактовой частоты увеличение производительности процессоров достигается путем разработки новых архитектур и алгоритмов обработки информации.

Имеются различные *архитектуры* процессоров. Под «архитектурой» процессора понимают конструкцию процессора и систему команд (инструкций). По способу представления команд различают четыре типа процессоров:

CISC с набором системы полных команд;

RISC с набором системы усеченных команд. Процессоры второго типа нацелены на быстрое выполнение небольшого набора простых команд;

VLIW со сверхдлинным командным словом;

MISC с минимальным набором системы команд и весьма высоким быстродействием.

К основным способам повышения производительности компьютера относятся:

суперскалярная архитектура;

конвейерная обработка информации;

использование разделенных кэш-памяти для команд и данных;

использование блока предсказания адреса;

использование блока вычислений с плавающей точкой;

поддержка многопроцессорного режима работы;

многопоточный режим (технология гипертрединг, сокращенно HT);

использование средств обработки ошибок;

технология обработки аудио- и видеоинформации (SIMD). Эта технология позволяет ускорить обработку звуковой и графической информации.

Термин «*суперскалярная архитектура*» означает, что процессор имеет более одного вычислительного блока. Эти вычислительные блоки называются ***конвейерами***[3]. Наличие в процессоре двух и более конвейеров позволяет ему выполнять одновременно несколько команд. Каждый конвейер разделяет процесс выполнения команды на несколько этапов:

выборка (считывание) команды из ОЗУ или кэш-памяти;

декодирование (дешифрация) команды;

выполнение команды;

обращение к памяти;

запоминание полученных результатов в памяти.

При конвейерной обработке информации на выполнение каждого этапа отводится один такт синхронизирующей частоты. В каждом новом такте заканчивается выполнение одной команды и начинается выполнение другой команды. Длительность выполнения команды определяется при этом временем на выполнение самого продолжительного этапа, а не временем выполнения всей команды.

[3] Конвейерный принцип обработки команд впервые был реализован в ЭВМ БЭСМ-2, разработанной под руководством С. А. Лебедева.

Разделение кэш-памяти на две части позволяет избежать структурных конфликтов, когда две команды одновременно обращаются к одному и тому же блоку памяти.

Развитие микропроцессоров идет не только по пути повышения разрядности и тактовой частоты, но и путем совершенствования *технологии обработки инструкций*, например, в микропроцессоре 386 операция сложения выполнялась за 6 тактов, а в 486-м – за два такта. Микропроцессоры следующих поколений выполняют операцию сложения за один такт. Для работы с видео-, аудио- и графическими данными Фирмой Intel разработана технология MMX – *технология обработки множественных данных одной инструкцией* (SIMD). Существенную роль играет технология *умножения тактовой частоты*, при которой скорость работы внутренних блоков в несколько раз выше скорости работы остальных частей компьютера.

В процессорах 80386 и 80486 для ускорения вычислений с плавающей точкой встраивался специальный математический сопроцессор. В последующих моделях, в связи с увеличением плотности размещения элементов на кристалле, математический сопроцессор реализован непосредственно в процессоре. В процессоре Pentium имеется два пятиступенчатых конвейера для выполнения операций с фиксированной точкой и один восьмиступенчатый конвейер для выполнения операций с плавающей точкой. Кроме выполнения математических расчетов этот конвейер используется также для быстрой обработки динамических трехмерных изображений.

Одним из путей повышения производительности является использование блока предсказания адреса. При выполнении разветвляющихся процессов и циклов можно заранее предсказать направление перехода и записать адрес перехода в специальный буфер предсказания адреса. Если результат выполнения текущей операции не совпал с содержанием буфера, то адрес считывается из памяти. То есть процессор практически ничего не теряет. Если же адрес перехода записан правильно, то он выбирается из буфера, имеющего малое время доступа, а не из памяти, имеющей значительно большее время доступа. Для реализации этой идеи в процессоре используется два буфера предварительной выборки. Один из буферов выбирает команды последовательно, а другой – согласно предсказаниям блока предсказания адреса.

Управление памятью

Микропроцессоры, начиная с 80386, имеют аппаратные средства для управления памятью и могут функционировать в двух основных

режимах: *реальной адресации* и *защищенной виртуальной*[4] *адресации*. В режиме реальной адресации процессор работает как быстрый 16- разрядный процессор 8086 с расширенным набором команд и может выполнять только одну задачу. В защищенном режиме процессор использует весь механизм 32-х разрядной организации памяти, в том числе механизм поддержки виртуальной памяти и механизм переключения задач. При этом процессор может выполнять одновременно несколько задач. Для каждой задачи процессор i386 эмулирует виртуальные процессоры i86. Каждый виртуальный процессор управляет назначенным ему процессом независимо от других процессов. В защищенном режиме (этот режим обозначают V86) процессор i386 поддерживает страничную организацию памяти и средства многозадачности. При этом используются те же средства межзадачной защиты и защиты операционной системы от пользовательских задач, которые предусмотрены в процессоре i386. Максимальный размер виртуального адресного пространства для каждого виртуального процессора i86 составляет 1 Мб, в то время как физическое адресное пространство для процессора i386 составляет 4 Гб (2^{32}). Возможность микропроцессора работать в защищенном режиме позволила реализовать в операционных системах многозадачный режим работы.

В последующих моделях микропроцессоров фирмы Intel используются аппаратные средства поддержки многопроцессорных систем. Микропроцессор Merced имеет 64-разрядную архитектуру, содержит ряд новаторских решений, повышающих эффективность вычислений, таких как предсказание адреса и спекулятивное выполнение.

Новые технологии в микропроцессорной технике

2006 год можно считать новым этапом в развитии микропроцессорной техники. Произошла очередная технологическая революция. Она охарактеризовалась появлением многоядерных процессоров с технологическим процессом 65 нм и менее (табл. 1.6, 1.7).

Процессоры на ядре Conroe, официально называемые Intel Core 2 Duo, появились на рынке в 2006 году и представляют собой продолжение Pentium III, адаптированное под актуальные потребности, – с широкой шиной, поддержкой новых наборов инструкций и направленностью на минимальное энергопотребление. Таким образом, можно построить следующий эволюционный ряд: Pentium Pro, Pentium II, Pentium III, Pentium M, Core, Core 2.

[4] Виртуальный – вымышленный, не существующий реально, физически.

Таблица 1.6

Основные характеристики двухъядерных процессоров

Процессор	Число ядер	Техпроцесс, нм	Тактовая частота, ГГц	Частота системной шины, МГц	Энергопотребление, Вт
Core 2 Duo E6600	2	65	2,1	1066	70
Core 2 Extreme X6800	2	65	2,9	1066	80
Core 2 Due T7700	2	65	2,4	800	35
Core 2 Extreme Quadro QX 6800	4	65	2,93	1066	90
Core Penryn E8300	2	45	2,83	1333	65
Core Penryn E8500	2	45	3,16	1333	65
Core Penryn QX9550	4	45	2,83	1333	95

Таблица 1.7

Факты, опровергающие закон Мура

Процессор	Площадь кристалла, мм^2	Число транзисторов, млн шт.
Core 2 E6600/E6700/X6800	143	291
Core 2 Duo E6300/E6400	111	167
Pentium D 900	280	376
Athlon 64 FX-62	230	

Intel Core 2 – двухъядерный процессор. Теперь вместо двух практически самостоятельных процессоров в одном сокете, со всеми вытекающими отсюда недостатками, мы имеем полноценный двухъядерный процессор. В Core 2 Duo используется общий кэш второго уровня объемом 4 Мбайт, к которому оба ядра имеют равноправный доступ (рис. 1.8).

Кроме того, в Core 2 Duo реализованы следующие технологии.

Технология Intel Wide Dynamic Execution – повышает производительность и эффективность работы процессора, позволяя каждому ядру исполнять до четырех инструкций за такт с использованием эффективного 14-этапного конвейера.

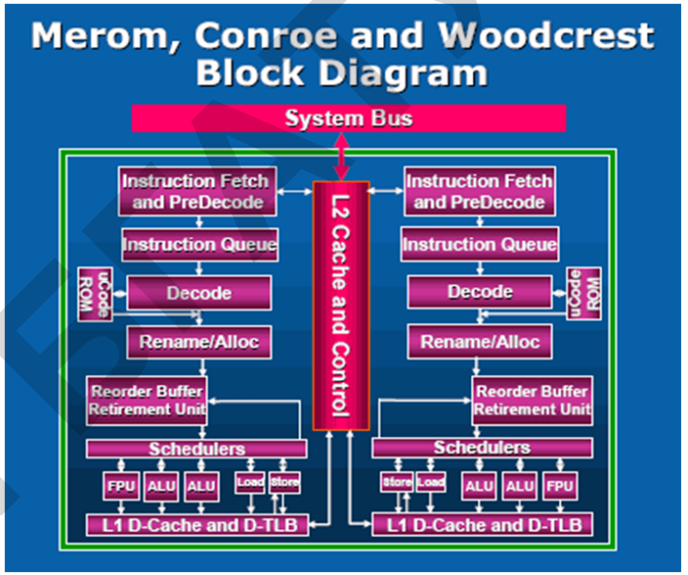


Рис. 1.8. Структура двухъядерного процессора

Технология Intel Smart Memory Access – повышает производительность системы путем снижения задержек при доступе к памяти и таким образом оптимизирует использование доступной пропускной способности, благодаря чему процессор получает данные тогда, когда они требуются.

Технология Intel Advanced Smart Cache – общая кэш-память 2-го уровня сокращает энергопотребление, сводя к минимуму объем «трафика» в подсистеме памяти, и повышает производительность системы, обеспечивая одному из ядер доступ ко всей кэш-памяти при простое другого ядра.

Технология Intel Advanced Digital Media Boost – удваивает скорость выполнения команд, часто используемых в мультимедийных и графических приложениях.

Технология Intel 64 Technology – обеспечивает поддержку 64-разрядных вычислений, предоставляя, например, процессору доступ к большему объему памяти.

В процессоре Core 2 Duo сделано немало улучшений по сравнению с прародителем Core Duo. Наиболее важно здесь следующее: увеличение скорости исполнения инструкций, оптимизация работы

с памятью, введение поддержки 64-разрядных вычислений и, наконец, нового набора инструкций SSE4. Как результат – большая эффективность даже при равной тактовой частоте.

Использование техпроцесса 65 нм привело к снижению энергопотребления. Применяются и другие технологии снижения энергопотребления:

поддержка технологии Intel Enhanced SpeedStep – динамическое изменение тактовой частоты процессора в зависимости от текущих потребностей в вычислительной мощности;

технология Ultra Fine Grained Power Control – возможность выключения тех блоков процессора, которые в данный момент не используются;

снижение разрядности шины. В легких режимах большая часть пропускной способности шин не используется, поэтому разрядность их можно снизить без ущерба для текущих потребностей в производительности.

В результате получился самый «холодный» двухъядерный процессор из тех, что используются в настольных компьютерах.

Примененные в Core 2 Duo технологические решения опровергли известный закон Мура, сделанный ученым в 1965 году на основе анализа тенденций развития микропроцессорной техники, согласно которому каждые два года количество транзисторов на кристалле будет удваиваться. В Core 2 Duo количество транзисторов на кристалле гораздо меньше, чем у конкурирующих процессоров от AMD, и вдвое меньше, чем у Pentium D 900.

На день публикации существовало пять моделей процессоров Core 2 Duo с тактовыми частотами от 1,86 ГГц до 2,93 ГГц. В зависимости от потребляемой мощности также различают 5 моделей (U – 14 Вт, L – от 15 до 24 Вт, T – от 25 до 49 Вт, E – более 50 Вт, X – более 75 Вт).

По производительности Core 2 Duo выигрывает у всех ранее выпущенных моделей процессоров не менее 10–15 % даже в самых тяжелых режимах тестирования, например, 3D-моделирования.

Не успели утихнуть страсти вокруг Conroe, как на смену им появились 4-х ядерные процессоры, а в настоящее время фирмы Intel и AMD уже представили потребителям шестиядерные процессоры.

Фирма AMD готовится выпустить 6-ядерный процессор Thuban (дракон) по 45 нм техпроцессу. Процессор будет иметь кэш второго уровня объемом 3 Мб (по 512 кб на каждое ядро) и кэш третьего уровня объемом 6 Мб, тактовая частота более 3 ГГц.

Не отстает от конкурента и корпорация Intel, начавшая продажи своих шестиядерных процессоров под кодовым названием Gulftown в 2010 году. Образец работал на частоте 2,4 ГГц и имел 12 Мбайт кэша третьего уровня. Gulftown также поддерживает технологию HyperThreading, и каждое из 6 его ядер может обрабатывать два потока данных одновременно. Изготавливается процессор по 32 нм техпроцессу.

Отметим еще одну особенность микропроцессоров – внешний интерфейс (слот подключения). В зависимости от того, какой слот подключения имеет материнская плата, она может работать с определенным типом процессора. Например, на процессорах до Pentium III использовался интерфейс (форм-фактор) Slot 1, на последующих моделях – Slot 2. Pentium Pro имеет интерфейс Socket 8, который позволяет поддерживать до четырех микропроцессоров в симметричной мультипроцессорной системе. Во втором поколении процессоров архитектуры IA-64 применяется физический интерфейс Slot M.

Память ЭВМ

Классификация памяти персонального компьютера

Различают внутреннюю и внешнюю память. *Внутренняя память* размещается на системной плате, вблизи от процессора. *Внешняя память* размещается, как правило, внутри системного блока или в дополнительных внешних устройствах.

Внутренняя память, кроме постоянного запоминающего устройства (ПЗУ), энергозависимая, то есть при выключении питания информация теряется. Основными характеристиками памяти являются время доступа или максимальная частота, с которой может работать эта память, а также ее объем. *Время доступа* представляет собой промежуток времени между формированием запроса на чтение информации из памяти и моментом поступления из памяти запрошенного машинного слова (операнда). *Длительность цикла* – величина, обратная максимальной частоте, определяется минимально допустимым временем между двумя последовательными обращениями к памяти. Время доступа достигает 1,7 наносекунд (нс), а тактовая частота достигает 1066 МГц и выше. *Объем* одного модуля памяти может составлять от 1 до 2048 Мбайт. На практике в настоящее время используются модули памяти 32, 64, 128, 512 Мбайт, 1 Гб и даже 2 Гб, например, DDR – 2 Gb PC3200 фирмы Kingston. Обычно на материнской плате можно размещать несколько модулей памяти, общий объем которых должен находиться в пределах доступного объема ОЗУ.

Внешняя память энергонезависимая. Она предназначена для длительного хранения программ и данных. В персональных компьютерах находит применение память на *магнитных дисках*, *магнитных лентах* и *оптических дисках*, а также на *электронных* внешних запоминающих устройствах.

Внутренняя память

Внутренняя память делится на: *постоянную* – постоянное запоминающее устройство (ПЗУ), *оперативную* – оперативное запоминающее устройство (ОЗУ), *перепрограммируемую* – перепрограммируемое запоминающее устройство (ППЗУ), другое ее название – КМОП-память (такое название она получила по технологии изготовления), *кэш-память* второго уровня.

Постоянная память (ROM) обычно имеет объем 8–256 кбайт. Микросхемы постоянной памяти устанавливаются на системной плате. Данные записываются в ПЗУ при изготовлении компьютера и хранятся в нем без изменений. В ПЗУ хранятся вспомогательные программы, используемые микропроцессором при выполнении функций управления компьютером, а также программы инициализации и проверки оборудования, которые автоматически запускаются при включении электропитания системного блока компьютера. В функции этих программ входит проверка исправности ОЗУ, клавиатуры, наличия подключенных внешних устройств и загрузка операционной системы (ОС). Кроме того, в ПЗУ имеется программа Setup, которая позволяет изменять параметры, определяющие конфигурацию компьютерной системы и необходимые для работы программ ПЗУ. ПЗУ выпускается в виде отдельной микросхемы, например, AMI BIOS.

Перепрограммируемая память (КМОП) – предназначена для хранения сведений о конфигурации и настройках компьютера, дате, времени, пароле. Это энергозависимая память. Для ее питания на материнской плате установлен специальный литиевый аккумулятор.

Оперативная память (RAM). Микросхемы ОЗУ расположены на системной и дополнительной платах компьютера. Объем памяти ОЗУ составляет от 8 Мбайт до 4 Гбайт. В ОЗУ информация загружается с началом работы ПК. Сначала туда записываются файлы операционной системы, которые приводят компьютер в состояние готовности к взаимодействию с пользователем. Память регенерации экрана монитора (видеопамять) размещается в адаптере видеомонитора.

Различают два вида оперативной памяти: статическую и динамическую. Эти два вида памяти отличаются быстродействием и удельной плотностью (объемом) хранимой информации. *Статическая память* (Static RAM, SRAM) в качестве элементарной ячейки использует так называемый статический триггер, схема которого состоит из 4–6 транзисторов. Состояние статического триггера может сохраняться как угодно долго. Данный тип памяти имеет высокое быстродействие, но низкую плотность размещения хранимых данных. *Динамическая память* (Dinamic RAM, DRAM) представляет собой набор конденсаторов, выполненных в структуре полупроводникового кристалла. Для построения элемента динамической памяти требуется 1–2 транзистора, которые используются для подключения и отключения конденсатора. Эта память обладает меньшим быстродействием по сравнению со статической памятью, требует периодической регенерации (это связано с разрядом конденсаторов), но позволяет создавать блоки памяти большой емкости.

В настоящее время в персональных компьютерах в качестве модулей оперативной памяти и видеопамяти чаще всего используются модули памяти **DDR2 SDRAM** – синхронная динамическая память с произвольным доступом и удвоенной скоростью передачи данных, второе поколение. Чипы этого вида памяти имеют несколько стандартов, отличающихся тактовой частотой. Тактовая частота работы этой памяти находится в диапазоне от 100 МГц до 266 МГц, а эффективная частота – в диапазоне от 400 МГц до 1066 МГц. Эффективная частота зависит как от частоты микросхемы, так и от частоты шины.

На смену DDR2 уже пришла более быстродействующая память **DDR3 SDRAM** (англ. *double-data-rate three synchronous dynamic random access memory* – синхронная динамическая память с произвольным доступом и удвоенной скоростью передачи данных, третье поколение). DDR3 имеет меньшее по сравнению с модулями DDR2 энергопотребление (на 40 %), что обусловлено понижением напряжения питания ячеек памяти с 1,8 В до 1,5 В. Снижение напряжения питания, в свою очередь, достигается за счет использования 90 нм (вначале, в дальнейшем – 65, 50, 40 нм) техпроцесса при производстве микросхем и применения транзисторов с двойным затвором *Dual-gate* (что способствует снижению токов утечки). Тактовая частота DDR3 находится в диапазоне от 100 МГц до 300 МГц, а эффективная частота находится в диапазоне от 800 МГц до 2400 МГц.

Модули памяти DDR2 и DDR3 не совместимы электрически и механически.

Распределение оперативной и постоянной памяти компьютера

Постоянное запоминающее устройство и оперативное запоминающее устройство имеют единое адресное пространство, поэтому эту память принято называть ***основной*** памятью. Определенные области памяти компьютера имеют специальное назначение и используются отдельными командами и программами операционной системы.

Выделяют три основные области памяти: область таблицы векторов прерывания, область программ ОС, используемых при загрузке компьютера, рабочая область.

Область таблицы векторов прерывания определяет место расположения подпрограмм для обработки прерываний. Прерывания – это запросы активных процессов и устройств ввода-вывода к процессору на обслуживание. Первые 1024 байта этой области резервируются для таблиц векторов прерываний, определяя тем самым 255 различных прерываний. Остальная часть этой области используется для запоминания состояния прерванного процесса. Эта область имеет абсолютные адреса памяти от 00000 до 00400 в шестнадцатеричном исчислении.

Область программ ОС, используемых при загрузке компьютера. Эта область является местом хранения программ постоянного модуля базовой системы ввода-вывода (BIOS). Так как эта система ввода-вывода контролирует основную работу компьютера и его частей, ей необходим некоторый объем памяти для собственных записей, для ведения учета состояния системы. Среди этой информации имеется и область данных, удерживающая в памяти сигналы, посылаемые с клавиатуры, до тех пор, пока программа не начнет их обработку. Здесь же хранятся сведения об объеме памяти, характеристиках основного оборудования, установленного на компьютере, индикатор режима дисплея и многое другое. Область объемом в 256 байт предназначена специально для данных BIOS. Абсолютные адреса памяти этой области составляют диапазон от 00400 до 00500 в шестнадцатеричном исчислении.

Рабочая область хранит рабочие программы и данные.

Процессор также может обращаться к памяти за пределами основной памяти компьютера. Эту область принято называть *виртуальной* памятью.

Виртуальная память

Виртуальная память – это совокупность программно-аппаратных средств, позволяющих пользователю писать программы, размер которых превышает размеры имеющейся оперативной памяти.

Виртуальная память решает следующие задачи:

размещение данных в запоминающих устройствах разных типов;

перемещение данных между запоминающими устройствами разного типа;

преобразование виртуальных адресов в реальные адреса.

Способностью работать с виртуальной памятью обладают все процессоры, начиная с микропроцессора *Intel 80386*. Такое свойство микропроцессора, как способность управлять памятью, заставляет реальную память машины иметь рабочий адрес, отличающийся от истинного, физического адреса. Принцип использования виртуальной памяти приведен на рисунке 1.9.

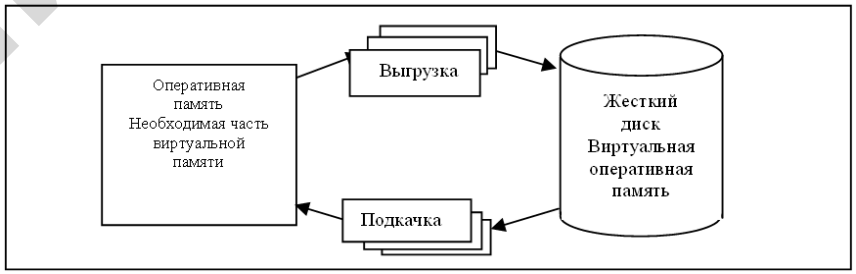


Рис. 1.9. Принцип использования виртуальной памяти

Если программа не помещается в оперативной памяти, ее разбивают на блоки, которые называют страницами, присваивают им адреса, которых нет в реальной памяти компьютера, и размещают на разных носителях. Часть программы остается в оперативной памяти компьютера, а другая часть помещается на жесткий диск.

Программа начинает работу с отображения некоторой части большего объема виртуальной памяти в определенную часть меньшей по объему физической памяти компьютера. Пока программа работает лишь с частью своей виртуальной памяти, она этого не замечает. Программа на самом деле использует другие адреса памяти, но это для нее не имеет никакого значения. Когда программа пытается использовать ту, большую часть виртуальной памяти, ко-

торой не было отведено места в реальной памяти, меньшей по объему, таблица управления памятью микропроцессора обнаруживает, что программа пытается использовать не существующий в данный момент адрес, и выдает команду «отсутствие страницы». Получив эту команду, специальная служебная программа виртуальной памяти начинает работу. Она временно приостанавливает действие программы, выбирает ту часть виртуальной памяти, которая находится в данный момент в физической памяти компьютера, и записывает ее на диск (это называется *выгрузкой* или *откачкой*). Освобожденная часть реальной памяти начинает действовать как необходимая часть виртуальной памяти. Когда же возникает необходимость в выгруженной части памяти, она подкачивается назад, то есть переписывается с диска в ОЗУ. Этот процесс называется *подкачкой*.

Как следует из описанного, диск компьютера используется в качестве склада для хранения частей виртуальной памяти, которые не используются в данный момент.

На основании изложенного выше можно дать ряд практических советов:

- Чем больше объем оперативной памяти процессора, тем реже ему придется обращаться к дисковой памяти, и, следовательно, быстродействие компьютера будет выше.
- Не следует запускать программы, в которых нет необходимости, так как они занимают часть оперативной памяти и снижают доступную оперативную память для рабочей программы.

Кэш-память

Для повышения производительности компьютера на нем устанавливается до трех уровней кэш-памяти: L1, L2, L3. L1 – память первого уровня, расположена непосредственно в кристалле микропроцессора, объем ее достигает 1 Мбайт, L2 – кэш-память второго уровня, объем ее достигает 2 Мбайт. Кэш-память второго уровня статическая. Эта память устанавливается вблизи процессора на системной плате и связана с ним скоростной шиной, работающей на более высокой частоте, чем остальная память, либо внутри картриджа микропроцессора (начиная с Pentium II, Pentium Pro). L3 – кэш-память третьего уровня. Она образована выделением и использованием некоторой части оперативной памяти специальными системными программами. Этот вид памяти используется для буферизации (предварительного хранения) данных при работе с жестким диском, CD-ROM и другими устройствами. Особенностью кэш-памяти является то, что она не доступна программистам. Управляет ей непосредственно микропроцессор.

Внешняя память

Накопители на магнитных дисках

В настоящее время наиболее широкое распространение в качестве внешних запоминающих устройств имеют ***накопители*** на магнитных дисках и магнитных лентах. Эти накопители предназначены для хранения, записи и чтения информации. ***Носителями*** информации в этих накопителях являются магнитные диски и магнитные ленты. В последнее время большую популярность приобрели полупроводниковые накопители, или флеш-диски.

До конца 20-го столетия принцип действия носителей на магнитных лентах и дисках был основан на использовании явления *электромагнитной индукции*. В основе процесса магнитной записи лежал процесс взаимодействия магнитного носителя информации (запоминающей среды) и магнитных головок – миниатюрных электромагнитов, располагаемых у поверхности движущегося магнитного носителя с небольшим зазором, при бесконтактной записи, или без зазора, при контактной записи.

Составными частями устройств записи/чтения являются носитель информации – лента или диск, покрытый порошком из ферромагнитного сплава, имеющего узкую петлю гистерезиса, и электромагнит, расположенный у поверхности движущегося носителя или на небольшом расстоянии от него (рис. 1.10). При записи информации электрический ток, проходящий по обмотке электромагнита записывающей головки, создает в зазоре электромагнита магнитный поток, который взаимодействует с магнитным покрытием движущейся ленты или диска. Под действием этого магнитного потока, в зависимости от направления тока, происходит намагничивание или размагничивание магнитного покрытия носителя. Разные направления намагниченности соответствуют логическому нулю или логической единице информации. При чтении информации с магнитного носителя движущиеся элементарные магнитики наводят в обмотке считывающей головки электродвижущую силу, направление которой зависит от намагниченности носителя информации.

В конце 20-го столетия фирма IBM изобрела головки с магниторезистивным эффектом[5] (GMR – Giant Magnitic Resistance). Это привело к повышению плотности записи до 300 Гбайт и более на одну пластину.

[5] **Магниторезистивный эффект (эффект Гаусса)** – изменение удельного сопротивления проводниковых и полупроводниковых материалов в магнитном поле.

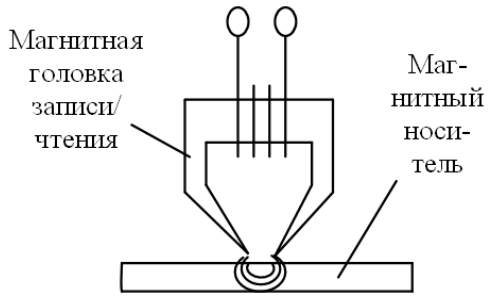


Рис. 1.10. Принцип записи информации на магнитный носитель

На современных компьютерах устанавливается, как правило, только накопитель на жестком диске. В недалеком прошлом устанавливался также накопитель для дискет размером 89 мм (3,5 дюйма).

Жесткий диск представляет собой пакет из нескольких пластин, выполненных из алюминия, латуни, керамики или стекла, толщиной примерно 2 мм, покрытых ферромагнитным материалом. Емкость жесткого диска достигает 2-х терабайт. Скорость вращения диска составляет обычно 3600, 4500, 5400, 7200, 10 000, 12 000 оборотов в минуту. Для чтения информации имеется блок головок чтения-записи информации (рис. 1.11). Вращение дисков и перемещение головок осуществляется с помощью двух электродвигателей.

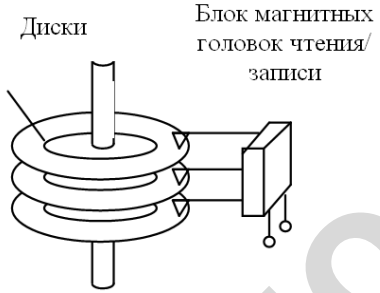


Рис. 1.11. Принцип устройства жесткого диска

Для обеспечения работы диск разбивается на сектора и дорожки. Число дорожек для дискет равно 80, а для жестких дисков – примерно 2500. Число секторов для дискет равно 18, а для жестких

дисков – 63. Подготовка дисков к работе осуществляется в процессе *форматирования*. При форматировании дисков кроме разбиения их на сектора и дорожки на них наносится также служебная информация. Дорожки с одним номером, расположенные на разных дисках, образуют своеобразный информационный цилиндр. Подведя головки чтения к нужной дорожке, можно прочитать информацию со всех дорожек этого цилиндра, что наряду с высокой скоростью вращения диска способствует уменьшению времени доступа.

Накопители на жестких дисках выпускаются различных размеров (форм-факторов) 1,8", 2,5", 3,5", 5,25". Емкость каждого сектора независимо от номера дорожки постоянна. Поэтому на внешних дорожках плотность записи ниже, а на внутренних выше. На жестких дисках форм-факторов 3,5" и 5,25" с целью повышения емкости диск разбивается на зоны, в пределах которых число секторов постоянно, чем дальше зона от центра, тем больше она содержит секторов.

Среди основных параметров накопителей на магнитных дисках, определяющих их производительность, выделяют следующие: среднее время доступа, которое определяется временем позиционирования магнитных головок на дорожке и временем ожидания сектора, и скорость обмена данными, которая зависит от используемого интерфейса. Время доступа складывается из среднего времени перемещения между двумя случайно выбранными дорожками (8–20 мс) и времени ожидания нужного сектора (4–8 мс). Максимальное значение скорости передачи данных зависит от используемого интерфейса. Интерфейс на основе новых технологий SATA II обеспечивает скорость обмена данными до 300 Мбайт/с. Ожидается, что скорость обмена данными в скором времени будет доведена до 600 Мбайт/с. Для повышения скорости обмена данными на жестких дисках создана кэш-память, в которой хранятся данные, наиболее часто используемые процессором. Размер кэш-памяти на жестком диске составляет от 2 до 8 Мбайт, на терабайтных дисках размер кэш-памяти значительно выше.

Накопители на сменных жестких дисках

Накопители на сменных жестких дисках по внешнему виду мало чем отличаются от накопителей на гибких магнитных дисках. Размеры дисков – 3,5 дюйма. Емкость носителя от 230 Мбайт до 2 Гбайт. Сменный магнитный носитель заключен в пластмассовый конверт, имеет металлическую основу и выполнен по технологии жестких дисков.

RAID-массивы

Для увеличения объема хранимой информации и повышения надежности промышленностью выпускаются RAID-массивы (Избыточный Массив на Дешевых Дисках). RAID-массив – устройство, состоящее из нескольких жестких дисков и контроллера. Такое устройство обладает большим объемом дискового пространства, повышенной скоростью обмена информацией и значительной надежностью хранения информации. Надежность повышается за счет дублирования информации на нескольких дисках.

Накопители на магнитных лентах

Накопители на магнитных лентах – *стримеры* – позволяют сохранять большие объемы информации при создании архивных копий. Они используют специальные кассеты (картриджи) с магнитной лентой. Стримеры имеют, как правило, собственные средства сжатия информации при записи на ленту. Емкость картриджей различна – от сотен Мбайт до десятков Гбайт. Для повышения плотности записи используется технология спирального сканирования (в других источниках эту технологию называют *поперечная запись*). Стримеры имеют разные стандарты, определяющие интерфейс с пользователем, формат магнитной ленты, методы кодирования и сжатия.

Накопители на оптических и магнитооптических дисках

Устройства для работы с оптическими дисками получили название CD-ROM. Они различаются по скорости считывания информации и возможности записи информации. Накопители имеют следующую систему обозначений: CD – устройства для чтения оптических дисков; CD-R – устройства для чтения и однократной записи оптических дисков; CD-RW – устройства для записи и чтения оптических и магнитооптических дисков. Основным показателем CD-ROM является скорость считывания информации. За единицу скорости считывания принята скорость чтения звуковых компакт-дисков – 150 кбайт/с. У современных CD-ROM скорость чтения значительно выше, она указывается в обозначении диска. Например, Creative 24xMX означает 24-скоростная. Однако цифра 24 – это не средняя, а максимальная скорость считывания. Средняя скорость считывания информации будет раза в полтора меньше.

В качестве носителей информации для оптических накопителей используются в настоящее время оптические (или лазерные) и магнитооптические диски. Они имеют размер 133 мм и 89 мм и емкость до 700 Мбайт.

По способу организации чтения/записи информации эти диски можно разделить на три группы: только для чтения (CD); с однократной записью и многократным считыванием (CD-R), перезапись таких дисков невозможна; перезаписываемые диски (CD-RW). Эти диски имеют разную технологию изготовления.

Оптические диски

Первыми на свет появились *оптические диски только для чтения*. Носителем информации на CD-диске является подложка из поликарбоната, на которую нанесен тонкий слой отражающего металла (обычно алюминия). В основе записи информации на эти диски с помощью лазера лежит модуляция интенсивности излучения лазера дискретными значениями нуля и единицы. Вначале создается матрица. При записи логической единицы луч лазера прожигает в матрице микроскопическое отверстие. Запись начинается с внутренних дорожек и ведется по спирали с большой плотностью – 630 дорожек на миллиметр. Таким способом создается первичный «мастер-диск». Затем осуществляется тиражирование этого диска методом литья под давлением или штамповкой. Выступы теперь соответствуют логической единице, а впадины – логическому нулю. Получившаяся рельефная основа металлизируется и покрывается лаком, защищающим тонкую металлическую поверхность. Чтение информации также осуществляется лучом лазера, но меньшей интенсивности. Поверхность диска по-разному отражает луч при наличии на нем выступов и вмятин. От выступов получается более мощный отраженный сигнал, а от впадин – более слабый.

В *оптических дисках с однократной записью* и многократным считыванием основу диска составляет алюминиевый или пластмассовый диск, на который нанесен тонкий слой металла – теллура. При записи информации в этом слое металла создается матрица хранимой информации лучом лазера.

В *перезаписываемых носителях* используется свойство рабочего слоя переходить под воздействием луча лазера в кристаллическое или аморфное состояние, имеющее разную отражающую способность. Эти диски могут не читаться на некоторых устаревших приводах CD-ROM.

Магнитооптические диски

Магнитооптические диски с однократной записью и многократным считыванием выполняются из стекла, на поверхность которого наносится сплав из тербия, железа и кобальта, обладающий магнитными свойствами. Этот сплав имеет низкую *темпера-*

туру Кюри – температура, при которой возможно перемагнитить сплав. При записи информации с помощью лазера производится разогрев металла на ограниченном участке диска до точки Кюри и прикладывается магнитное поле нужного направления. После остывания данный участок запоминает направление намагниченности. При нормальной температуре информация на диске не подвержена воздействию внешних магнитных полей. При считывании информации используется *эффект Керра*, который проявляется в изменении направления поляризации лазерного луча, отраженного от намагниченной поверхности. Такие диски читаются на любом приводе CD-ROM.

Известны накопители на магнитооптических дисках (APEX компании Pinnacle Micro) емкостью 4,6 Гбайт, использующие диски размером 5,25".

Основными достоинствами накопителей на оптических и магнитооптических дисках являются: высокая плотность записи информации; универсальность, т. е. пригодность для хранения информации, заданной в различной форме; возможность быстрой перезаписи больших объемов информации и длительного ее хранения; высокая надежность сохранности информации.

Накопитель DVD

Накопители DVD предназначены для хранения и воспроизведения видео-, аудиозаписей высокого качества, компьютерной информации. Носителем информации является цифровой универсальный DVD-диск, но DVD-накопители могут читать и обычные CD-ROM-диски. Двухсторонние накопители DVD могут читать и CD-R- и CD-RW-диски. Цифровой универсальный диск по внешнему виду мало чем отличается от CD-дисков. DVD-диск имеет диаметр 120 мм. Он может быть односторонним или двухсторонним, на каждой стороне может быть один или два рабочих слоя. По этой причине эти диски имеют большую емкость. Односторонние однослойные DVD-диски имеют емкость 4,7 Гбайт, двухслойные – 8,5 Гбайт, двухсторонние однослойные – 9,4 Гбайт, а двухсторонние двухслойные носители могут вмещать до 17 Гбайт информации.

Скорость чтения DVD вполне сравнима со скоростями 20 и более скоростных CD-ROM-приводов. Для двухскоростных приводов DVD (скорость 2xDVD = 20xCD) максимальная скорость передачи данных составляет 2700 кбит/с (в режиме DVD) и 3000 кбит/с (в режиме CD).

По внешнему виду DVD-ROM-диск, как и DVD-ROM-привод, почти ничем не отличаются от CD-ROM-диска и CD-ROM-привода. Но произошли значительные изменения в технологии. Расстояние между дорожками, как и минимальный размер единичной насечки, уменьшено в два раза.

Лазер в обычном устройстве CD-ROM имеет длину волны 780 нм, устройства DVD используют лазер с длиной волны 650 нм или 635 нм, что позволяет различать в два раза больше 1000 засечек на одном треке (дорожке). Более совершенная система позиционирования головки и фокусировки луча у DVD дает возможность в два раза снизить расстояние между треками. В результате плотность записи существенно возрастает. Объем DVD-дисков удваивается за счет использования обеих сторон диска. Увеличение объема записываемой информации на DVD-дисках не ограничивается лишь удвоением сторон – удваивается и количество слоев.

Наружный слой двухслойного диска полупрозрачный. Он несет собственную информацию, но при этом позволяет считывать через свою поверхность информацию на внутреннем слое. Подобная операция чтения обеспечивается за счет различной фокусировки луча лазера для каждого из слоев.

Твердотельные полупроводниковые накопители

Развитие полупроводниковых внешних запоминающих устройств (электронных дисков) определяется быстрым ростом степени интеграции полупроводниковых больших интегральных схем (БИС) и, как следствие, снижением стоимости полупроводниковой памяти в расчете на 1 бит. В то же время повышение разрядности компьютера (до 32) обеспечивает прямую адресацию к полупроводниковой внешней памяти как физическому расширению ОЗУ, рассматриваемому операционной системой в качестве накопителя на магнитных дисках или накопителя на гибких магнитных дисках. Флеш-диски уже находят применение в ноутбуках в качестве альтернативы жестким дискам. Электронный диск включает три компоненты: флеш-память, интерфейс и контроллер.

Флеш-диск (Flash – Drive) – новый вид внешнего запоминающего устройства, приобретающий все большую популярность. Другое название, встречающееся в литературе и соответствующее маркировке ED, – EasyDisk (буквально – легкий диск), например, ED 17, ED 80. Представляет собой полупроводниковое перепрограммируемое запоминающее устройство, предназначенное для хранения и переноса данных с одного компьютера на другой. Ком-

пактный, легкий, удобный и простой в эксплуатации. Как правило, имеет защиту от записи в виде переключателя на корпусе и программного пароля. Для его работы не нужны ни соединительные кабели, ни источники питания (включая батарейки), ни дополнительное программное обеспечение. Изготавливается в виде брелка с размерами в пределах до 90×25×13 мм. Объем памяти составляет от 16 Мбайт до десятков Гбайт. Подключается к контроллеру последовательной шины USB. В объемах до 32 Гбайт флеш-память обходится недорого. По отношению Объем памяти/Стоимость флеш-память значительно уступает жесткому диску.

Флеш-память представляет собой перепрограммируемое запоминающее устройство, основу которого составляют статический триггер[6] с плавающим затвором (рис. 1.12). При записи информации на управляющий электрод подается положительное напряжение. Если на линии бит (сток) имеется отрицательный потенциал, то на плавающем затворе скапливается отрицательный заряд. Заряд на затворе может храниться несколько лет. При чтении информации проверяется величина этого заряда. Если величина заряда больше некоторой пороговой величины, то считается, что сигнал равен логическому нулю, в противном случае – логической единице. При подаче на управляющий электрод большого отрицательного напряжения заряд с затвора стирается.

Различают флеш-память с одноуровневыми ячейками (SLC) и многоуровневыми ячейками (MLC). Флеш-память с многоуровневыми ячейками обеспечивает большую скорость считывания, но имеет меньшую скорость записи по сравнению с одноуровневыми ячейками. Флеш-память на одноуровневых ячейках стоит дороже.

Недостатками флеш-памяти, сдерживающими ее широкое применение, являются низкая скорость записи, ограничение на число циклов записи/чтения, а также неопределенные показатели по надежности. Тем не менее, в настоящее время эти накопители информации полностью вытеснили накопители на гибких магнитных дисках для временного хранения и переноса информации с одного компьютера на другой. Ученые и разработчики ищут пути построения надежных устройств памяти на основе использования других материалов и основанных на других физических принципах.

[6] Триггер – электронное запоминающее устройство, имеющее два устойчивых состояния (0 и 1) и способное под воздействием управляющих сигналов переходить из одного устойчивого состояния в другое.

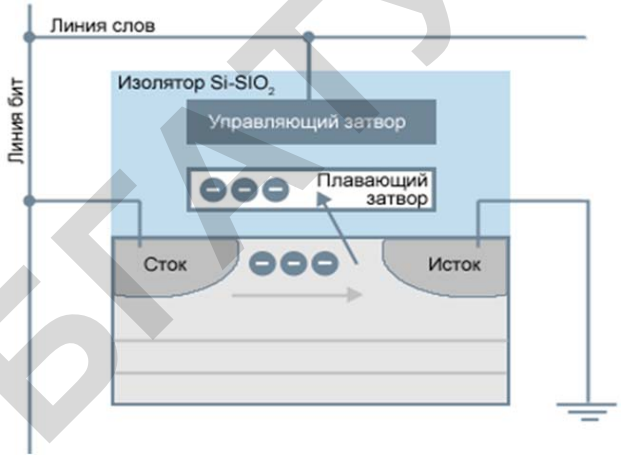


Рис. 1.12. Структура ячейки флеш-памяти

Адресация памяти

Интерфейсная часть микропроцессора предназначена для связи и согласования микропроцессора с системной шиной компьютера, а также приема и предварительного анализа команд выполняемой программы и формирования полных адресов операндов и команд. Важным в этом плане является понятие порта ввода-вывода. ***Порт*** ввода-вывода – это пункты системного интерфейса компьютера, через которые микропроцессор обменивается информацией с другими устройствами. Число портов у микропроцессора определяется разрядностью шины адреса. При 16-разрядной шине адреса число портов может быть 65 536 (2^{16}), при 32-х разрядной шине адреса – 4 294 967 296 (2^{32}). Каждый порт имеет адрес – номер порта, соответствующий адресу ячейки памяти. При этом эта ячейка памяти может быть или частью основной памяти компьютера, или частью устройства ввода-вывода, использующего данный порт. Многие устройства (диски, клавиатура, принтер и др.) имеют постоянно закрепленные за ними порты ввода-вывода.

Внутренний интерфейс

Внутренние интерфейсы расположены внутри корпуса компьютера и предназначены для подключения различных устройств (IDE-устройств) к материнской плате:

- SATA (Serial ATA) – это новая технология, которая приходит на смену ATA. Данный интерфейс используется для подключения

накопителей на жестком диске и обеспечивает высокую скорость передачи данных – 150 Мбайт/с. Интерфейс SATA-II обеспечивает скорость передачи данных до 300 Мбайт/с. Разработчики обещают довести скорость передачи данных с помощью данной технологии до 600 Мбайт/с.

- Разъемы AGP-слот[7] и PCI-E используются для подключения видеокарт. Слоты AGP и PCI-E не взаимозаменяемые.
- Для подключения плат памяти и других плат расширения используются слоты PCI-E, которые выпускаются двух видов: длинные PCI-E 16x и короткие PCI-E 1x.

Внешний интерфейс

Внешние интерфейсы используются для подключения периферийных устройств – устройств, внешних по отношению к компьютеру:

– интерфейсы VGA и DVI для подключения мониторов;

– интерфейсы последовательные универсальные USB, IEEE-1394 / FireWire / i.Link, PS/2;

– интерфейсы для подключения модемов ISDN, RJ11;

– интерфейс RJ45 для подключения сети;

– интерфейсы для подключения аудио, видео: S-Video, SCFRT, HDMI, «Тюльпан», HDTV.

Для каждого устройства в компьютере имеется электронная схема, которая управляет им, – контроллер. Все контроллеры взаимодействуют с процессором и оперативной памятью через системную шину. При вставке в слоты расширения материнской платы контроллеры подключаются к шине. Особо следует выделить контроллер DMA, который обеспечивает прямой доступ к памяти. Этот контроллер выполняет операции передачи данных, не загружая процессор и системную шину.

В последнее время большую популярность приобрел интерфейс *USB*. Он характерен тем, что позволяет подключать последовательно до 256 различных устройств, имеющих последовательный интерфейс. Поддержка стандарта Hi-Speed USB или USB 2.0 для современных системных плат стала обязательной. Hi-Speed USB – последний стандарт в этой области. Он обеспечивает скорость передачи данных до 480 Мбит/с, поддерживается также дополнительный подканал со скоростью обмена данными в 1,5 Мбит/с. Удобство шины

[7] Слот (от английского slot – «паз», «щель») – узкий разъем для установки модулей памяти или карт расширения.

состоит в том, что она практически исключает конфликты между различным оборудованием, позволяет подключать и отключать устройства без выключения компьютера и объединять компьютеры в локальную сеть без применения специального оборудования и программного обеспечения. Компания Intel сообщила о разработке интерфейса USB-3.0 с пропускной способностью 5 Гбит/с.

Клавиатура

В настоящее время в эксплуатации находится значительное число типов клавиатуры, но все они построены по одному принципу. На клавиатуре можно условно выделить четыре поля (рис. 1.13):

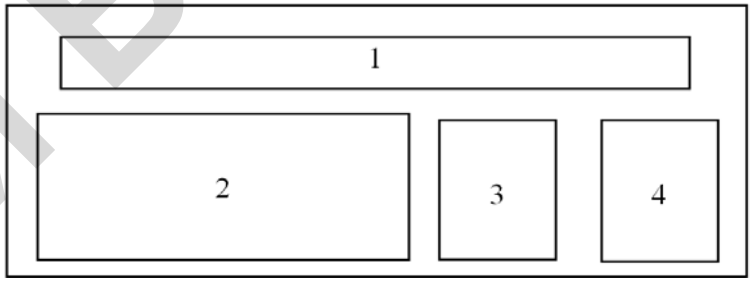


Рис. 1.13. Поля типовой клавиатур

1) *функциональное* – поле с функциональными и управляющими клавишами;

2) *алфавитно-цифровое* – поле, где расположены клавиши с буквами русского и латинского алфавита, цифрами, специальными символами и некоторые управляющие клавиши;

3) *управления курсором* – поле с клавишами для управления курсором, вставки и удаления текста;

4) *дополнительное* – поле для ввода цифровой или символьной информации, а также для управления курсором.

Назначение клавиш приведено в таблице 1.8.

Таблица 1.8

Назначение основных клавиш на клавиатуре IBM PC/AT

Клавиатура IBM PC/AT	Назначение клавиш
F1...F12	Функциональные клавиши
Caps Lock	Фиксация прописных букв

Окончание табл. 1.8

Клавиатура IBM PC/AT	Назначение клавиш
⇑ Shift	Смена регистра. Переключение на ввод прописных букв без фиксации или на ввод спецсимволов
Tab	Табуляция, перемещение курсора на фиксированное число позиций
Ctrl	Ввод дополнительного кода. Используется в комбинации с другими клавишами
Alt	Альтернативный набор. Используется в комбинации с другими клавишами, а также для ввода символов с кодами 128–255
← Backspace	Возврат на символ. Стирает последний введенный символ
Spacebar	Ввод пробела
Enter	Конец ввода команды, набора текста
Esc	Выход в вызвавшую программу, отказ от операции
Prt Sc (Print Screen)	Печать экрана
Scroll Lock	Блокировка прокрутки
Pause/Break	Пауза, приостанавливает вывод файла на экран. Для продолжения просмотра нажать любую клавишу
Num Lock	Переключение дополнительного поля на ввод цифр или управляющих символов
Ins (Insert)	Включение/выключение режима вставки символов
Del (Delete)	Удаление символа в позиции курсора
End	Перемещение курсора в конец файла
Home	Перемещение курсора в начало файла
PgUp	Перемещение курсора на страницу вверх
PgDn	Перемещение курсора на страницу вниз
← ↑ → ↓	Направление перемещения курсора

Клавиши первого ряда основного поля (2) имеют два ряда символов. Символы нижнего ряда вводятся непосредственно, символы верхнего ряда – при нажатии клавиши Shift. На клавиатуре зарубежных компьютеров переключение на ввод символов русского алфавита выполняется специальной программой-драйвером клавиатуры. Эта программа запускается, как правило, в начале работы с компьютером и затем постоянно находится в памяти ПК. Способы переключения на ввод русских символов зависят при этом от используемого драйвера, чаще всего применяются комбинации клавиш **Shift + Shift**, **Ctrl + Alt**, **Ctrl + Shift** правый, **Ctrl + Shift** левый, а также клавиша **Caps Lock**.

Обозначение вида [Ctrl + Alt] означает: нажать первую клавишу и, не отпуская ее, нажать вторую клавишу.

Наиболее часто при работе на компьютере используются следующие комбинации клавиш:

Ctrl + Alt + Del – вызов диспетчера задач, который позволяет снять задачу, перейти к другой задаче, сменить пользователя или выйти из системы;

Alt + PrScr – в операционной системе Windows эта комбинация клавиш помещает в буфер памяти копию активного окна Windows. При нажатии клавиши PrScr в буфер помещается копия всего рабочего окна;

Ctrl + Break – завершение работы выполняемой программы или команды, если это предусмотрено в программе;

Shift + PrScr – печать графической копии экрана;

Ctrl + NumLock – приостанавливает выполнение программы, для продолжения выполнения нажать любую клавишу.

Несмотря на то, что основное управление в графических операционных системах осуществляется с помощью мыши, полезно знать некоторые сочетания клавиш для управления компьютером, например, при выходе из строя мыши.

Сочетания клавиш для управления приложениями и меню:

Ctrl + Esc – открывает меню Пуск;

Alt + Tab – переключение между запущенными приложениями;

Alt + F4 – закрывает текущее окно;

F10 – активизирует строку меню;

Enter – завершает ввод команды, эквивалентна щелчку мыши по выделенному объекту;

Пробел – для выделения значков и меню, заменяет левую клавишу мыши.

Сочетания клавиш для управления документом:

Ctrl + O – открыть документ (в любой программе);

Ctrl + W – закрыть документ (в любой программе);

Ctrl + S – сохраняет документ (в любой программе);

Ctrl + P – печать документа (в любой программе);

Ctrl + A – выделить весь документ (в любой программе);

Ctrl + C – скопировать выделенную часть документа или файл в Буфер обмена;

Ctrl + V – вставить часть документа или файл из буфера обмена;

На современных клавиатурах число комбинаций клавиш значительно увеличилось, появились новые клавиши для управления работой в Интернете, работы со средствами мультимедиа. Назначение всех клавиш и их сочетаний следует смотреть в описании каждой конкретной клавиатуры. Отметим три новые клавиши на основном поле клавиатуры:

Windows (клавиша с логотипом операционной системы Windows) – открывает главное меню Windows;

Fn – дополнительная функциональная клавиша, используется в сочетании с другими функциональными клавишами для управления компьютером;

AltGr – обеспечивает выполнение команд настраиваемой панели инструментов приложений Windows в сочетании с номером этой клавиши.

Мышь

Мышь служит для управления перемещением курсора по экрану, выбора пунктов меню, прокрутки экрана, запуска исполняемых файлов. Различают механические и оптические мыши. Механическая мышь имеет шарик, встроенный в дно корпуса, две или три клавиши. При перемещении мыши по ровной поверхности программа считывает положение шарика и преобразует его координаты в положение указателя мыши на экране. Нажатие клавиш фиксируется программой и преобразуется в команды. Оптическая мышь использует другие способы преобразования координат положения манипулятора. Некоторые прикладные программы работают только с манипулятором мышь, но большинство программ могут работать как с манипулятором, так и с клавиатурой компьютера. В графических операционных системах манипулятор мышь является основным средством управления компьютером, а клавиатура выполняет вспомогательную роль.

В последнее время на рынке появились беспроводные клавиатура и мышь. Передача сигналов в этом случае осуществляется с использованием инфракрасных датчиков и приемников информации.

Видеомонитор

Видеомонитор, наряду с клавиатурой, является основным устройством, обеспечивающим общение пользователя с компьютером. На его экран выводится вся информация: и текстовая, и графическая.

В настоящее время преобладают мониторы двух типов: мониторы на основе электронно-лучевой трубки (CRT-мониторы) и жидкокристаллические мониторы (LCD-мониторы).

CRT-мониторы

Основу CRT-мониторов составляет электронно-лучевая трубка (ЭЛТ). Упрощенная структура ЭЛТ приведена на рисунке 1.14. Она состоит из стеклянной колбы (*1*), трех электронных пушек (*2*), системы сведения и отклонения луча (*3*), маски (*4*), экрана (*5*). С внутренней стороны экран монитора покрыт слоем люминофора – вещества, способного светиться при бомбардировке его заряженными частицами. Электронные пушки предназначены для излучения и фокусировки потока электронов на экране видеомонитора. В цветных мониторах имеется три электронных пушки. Отклоняющая система служит для управления электронным лучом. В состав отклоняющей системы входит генератор строчной развертки, перемещающий луч в горизонтальном направлении, и генератор кадровой развертки, перемещающий луч в вертикальном направлении. В процессе работы луч пробегает по экрану по строкам, с верхнего левого края до правого нижнего края, и создает изображение на экране. Во время обратного хода изображение на экране не меняется, а луч перемещается снова в левый верхний угол экрана. В цветных мониторах экран состоит из точек трех цветов, расположенных группами по три (триады): красного, синего и зеленого. Каждая из этих точек освещается своим лучом. От интенсивности луча зависит яркость ее свечения. Эта триада зерен люминофора образует точку на экране – *пиксел*. Наш глаз воспринимает эти три точки как одну точку с определенным цветом. При воспроизведении изображения точки могут группироваться как по вертикали, так и по горизонтали, образуя более крупные точки. Для обеспечения точного попадания лучей на одну триаду перед экраном устанавливают маски. От технологии изготовления маски, а также от качества электронной схемы управления сведением лучей зависит качество ЭЛТ.

Режим работы монитора. Монитор может работать в двух режимах: текстовом и графическом. При воспроизведении текста эк-

ран условно разбит на строки и столбцы. Число строк и столбцов зависит от разрешающей способности экрана. Нумерация строк и столбцов начинается с нуля. Для каждого символа отводится одно знакоместо. Имеется возможность программно изменять число строк и столбцов на экране. В графическом режиме экран представляется в виде набора точек – пикселей. Число точек по горизонтали и вертикали определяется разрешающей способностью экрана, которая может настраиваться пользователем.

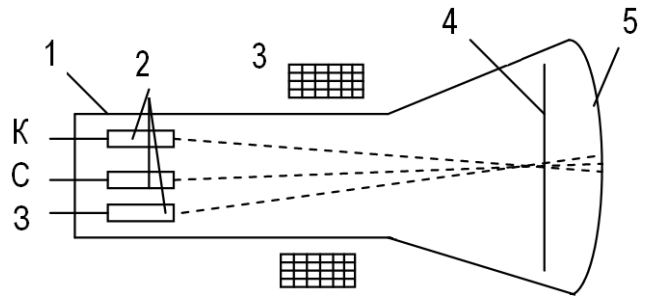


Рис. 1.14. Электронно-лучевая трубка

Основными характеристиками мониторов, определяющими их потребительские качества, являются четкость, частота строчной и кадровой развертки, размер экрана, разрешающая способность, емкость видеопамяти, глубина цвета.

Четкость изображения зависит от *шага точки* – минимального расстояния между люминофорными элементами одинакового цвета. Эта величина находится в интервале 0,28–0,24 мм. На практике этот параметр называют размером зерна. Чем меньше шаг точки, тем выше качество изображения при прочих равных условиях.

Размер экрана – это размер экрана монитора по диагонали, его принято измерять в дюймах. Установилось четыре основных стандарта размеров экрана для бытовых компьютеров: 15", 17", 19", 21".

Разрешающая способность определяется числом точек на экране по горизонтали и по вертикали. Современные мониторы позволяют получать различные значения разрешающей способности: 320×200; 640×480, 800×600, 1024×768, 1280×1024, 1600×1200 и др. При работе в операционной системе Windows предпочтительное значение разрешающей способности для комфортной работы зависит от размера экрана монитора (табл. 1.9).

Таблица 1.9

Рекомендуемые значения разрешающей способности монитора

Размер экрана	800×600	1024×768	1152×864	1280×960	1280×1024	1600×1200
15"	+	+				
17"		+	+	+		
19"			+	+	+	
21"					+	+

Частота строчной развертки и частота кадровой развертки взаимосвязаны. Они зависят также от размера монитора и установленного разрешения экрана. Частота строчной развертки должна обеспечить воспроизведение требуемого количества линий на экране за время прямого хода кадровой развертки. Например, при разрешении экрана 1280×1024 частота строчной развертки должна быть не менее 70 КГц. Частота кадровой развертки непосредственно влияет на качество изображения, комфортность работы, утомляемость пользователя. При низкой частоте кадровой развертки становится заметным мерцание экрана. Рекомендуемые значения кадровой развертки – 75–100 Гц (не все мониторы поддерживают частоту 100 Гц). Установлено, что при частоте 110 Гц глаз человека не может заметить никакого мерцания. Установлено также, что при больших углах обзора мерцание заметнее при больших частотах кадров.

Глубина цвета, или количество воспроизводимых одновременно цветов.

Современные мониторы способны воспроизводить практически не ограниченное число цветов. Чаще всего устанавливаются палитры из 256 цветов, а также 16-и и 24-х битные палитры. Теоретически, выделив для управления каждым цветом (синим, зеленым и красным) один байт, можно получить 256×256×256 различных цветов и оттенков. Число воспроизводимых цветов ограничивается практически только размером видеопамяти. Например, при разрешении 640×480 и 256 цветах для хранения информации о состоянии экрана необходимо 307 200 байт видеопамяти, при разрешении 1200×1024 и 256 цветах потребуется уже 1 228 800 байт видеопамяти.

Видеопамять. Видеопамять размещается в видеоадаптере. Обычно она организуется в виде страниц. Видеомонитор имеет как минимум две страницы памяти: одна страница – видимая, отобра-

жается на экране, другая страница в это время заполняется. Смена кадра осуществляется во время обратного хода луча. Для качественного воспроизведения мультипликации компьютер должен иметь 4 страницы видеопамяти. Объем видеопамяти современных компьютеров составляет 4, 8, 16, 128 Мбайт и более.

Существенными недостатками CRT-мониторов являются большое потребление электрической энергии и наличие излучений. Для обеспечения безопасной эксплуатации мониторов к их изготовлению предъявляют высокие требования. Эти требования изложены в стандартах TCO 92, TCO 95 и TCO 99 – цифры означают год принятия стандарта. Названные стандарты устанавливают требования по максимально допустимому значению электромагнитных излучений, шума и тепла при работе мониторов, функции энергосбережения мониторов, эргономике, экологии, пожарной безопасности и электробезопасности.

LCD-мониторы

Принцип работы LCD-мониторов (жидкокристаллические мониторы) основан на свойстве некоторых веществ, находящихся в жидком состоянии и обладающих свойствами, присущими кристаллическим телам, изменять плоскость поляризации света, проходящего через жидкость или отражающегося от нее, при воздействии на нее электрического поля.

Экран в жидкокристаллических мониторах представляет собой две прозрачные пластины-подложки, пространство между которыми заполнено жидкими кристаллами. На подложках проделаны параллельные бороздки, определяющие ориентацию кристаллов. Бороздки подложек взаимно перпендикулярны. Узлы пересечения бороздок и образуют точки экрана. Молекулы жидких кристаллов при отсутствии напряжения под воздействием источника проходящего или падающего света поворачивают плоскость поляризации на 90 градусов. Поворот плоскости поляризации светового луча не заметен для глаза, поэтому на панели устанавливают несколько поляризационных фильтров, которые пропускают только ту компоненту светового потока, у которой ось поляризации соответствует заданной. В отсутствие напряжения на сегменте углы поляризации света после прохождения ЖК-ячеек и второй подложки совпадают, и поэтому пиксел выглядит прозрачным. Экран разделен на отдельные ячейки, к которым подведены электроды, создающие электрическое поле. Для вывода цветного изображения делается подсветка монитора сзади так, чтобы свет порождался в задней части

LCD-монитора. Цвет получается в результате использования трех фильтров, которые выделяют из излучения источника белого цвета три основных цвета. Комбинируя три основных цвета для каждой точки, можно воспроизвести любой цвет.

Разрешение LCD-мониторов одно, оно соответствует максимальному физическому разрешению CRT-мониторов. Однако имеется возможность выводить изображение на экран с меньшим разрешением либо путем использования части экрана (центрирование), либо растягиванием изображения на весь экран. Другими важными параметрами, определяющими комфортность работы с ЖК-мониторами, являются яркость и контрастность.

Несомненными достоинствами LCD-мониторов являются низкое потребление электрической энергии и отсутствие вредных излучений.

LCD-мониторы изготовляются по различным технологиям, что сказывается на их характеристиках.

Другие технологии изготовления мониторов

Плазменные мониторы PDP

Принцип работы плазменных мониторов основан на явлениях электрического разряда в газах. Плазменные экраны создаются путем заполнения пространства между двумя стеклянными поверхностями инертным газом, например, аргоном или неоном. На стеклянную поверхность помещают маленькие прозрачные электроды, на которые подают высокое напряжение. Под воздействием этого напряжения в прилегающей к электроду газовой области возникает электрический разряд. Плазма газового разряда излучает свет в ультрафиолетовом диапазоне, который вызывает свечение частиц люминофора в диапазоне, видимом человеком. Каждый пиксел работает при этом как обычная флуоресцентная лампа (лампа дневного света). Высокие яркость и контрастность, отсутствие дрожания изображения являются достоинствами этих мониторов.

FED-мониторы

Принцип работы FED-мониторов похож на принцип работы CRT-мониторов – свечение люминофора при бомбардировке электронами. В FED-мониторах каждый пиксел представляет собой, по существу, миниатюрную электронную трубку. За каждым элементом экрана располагается источник электронов. Каждый источник электронов управляется отдельным электронным элементом, и каждый пиксел затем излучает свет благодаря воздействию электронов на люминофорные элементы, как в традиционных CRT-мо-

ниторах. Основное достоинство FED-мониторов – очень малая толщина.

Видеокарта

Видеокарта отвечает за связь между процессором и монитором. Видеокарта (она же видеоадаптер, графический адаптер) представляет собой специальное устройство, сконструированное в виде отдельной платы расширения, устанавливаемой в слоты материнской платы. В функции видеокарты входят быстрая и качественная обработка двумерной графики (2D-графика) и поддержка объемного, трехмерного изображения (3D-графика). Современные видеокарты выполняют и другие функции: прием изображения с внешнего источника – видеокамеры, видеомагнитофона или телевизионной антенны (для этой цели компьютер должен иметь видеовход и TV-тюнер); вывод изображения на внешние устройства – телевизор, видеомагнитофон. Современные видеокарты – это графические сопроцессоры, значительно снижающие нагрузку на процессор, что приводит к повышению производительности всей системы. По этой причине их иногда называют графическими ускорителями.

Основными элементами видеокарты являются графический процессор, видеопамять и цифро-аналоговый преобразователь памяти с произвольным доступом.

Графический процессор обрабатывает команды центрального процессора и создает картинку экрана, состоящую из совокупности координат и цвета каждой точки. Эта информация передается в видеопамять. Видеопамять предназначена для временного хранения информации перед выводом ее на экран. Цифро-аналоговый преобразователь преобразует цифровую информацию из памяти в аналоговое напряжение, управляющее отображением цветов на экране монитора. В мониторах с интерфейсом DVI цифровой образ передается на монитор, минуя цифро-аналоговый преобразователь.

Характеристики видеосистемы зависят как от параметров используемого монитора, так и от установленного в компьютере видеоадаптера. Основными характеристиками видеокарт являются: разрешающая способность, цветовой режим, объем видеопамяти, частота кадровой развертки.

Разрешающая способность определяется количеством точек по вертикали и горизонтали: 640×480, 800×600, 1024×768, 1152×864, 1600×1200, 1792×1344 и т. д.

Цветовой режим – это количество используемых цветов. Различают несколько цветовых режимов: «грубый» – 16 цветов, Low

Color – режим 245 цветов, High Color – режим высококачественного цвета – 65 532 цвета, True Color – режим реального цвета – 16 млн цветов.

Указанные два режима: разрешающая способность и цветовой режим – вместе называют *видеорежимом* и обозначают, например, следующим образом: 800×600×65K (разрешение 800×600 при 65 тыс. цветов).

Способность карты поддерживать тот или иной видеорежим определяется *объемом видеопамяти*. Современные видеокарты имеют значительный объем видеопамяти, не менее 256 Мбайт.

Частота кадровой развертки зависит от используемого монитора. Для мониторов на основе электронно-лучевой трубки она должна находиться в пределах 75–100 Гц, для ЖК-мониторов – 59– 60 Гц.

Способности видеокарты определяет установленный на ней набор микросхем – *чипсет*. Именно на этот параметр рекомендуют обращать внимание при выборе видеокарты. Наиболее известны чипсеты GeForce компании nVidia и Radeon компании AMD.

Печатающие устройства

Печатающие устройства (ПУ), или, как их еще называют, принтеры, предназначены для получения «твердой копии» документа, которая может храниться длительное время и удобна для чтения человеком.

Все печатающие устройства могут выводить текстовую информацию, многие из них могут выводить также графики и рисунки, в том числе и цветные.

Существует множество моделей принтеров, но наибольшее практическое применение получили матричные, струйные и лазерные принтеры.

Матричные принтеры являются наиболее дешевым типом принтеров для персональных компьютеров. Принцип печати этих принтеров следующий: печатающая головка перемещается относительно неподвижного листа бумаги вдоль строки и наносит текст или рисунок с помощью игл, ударяющих по красящей ленте. Печатающая головка содержит 9, 24 или 48 игл. Скорость печати матричного принтера – от 10 до 60 секунд на страницу.

Струйные принтеры обеспечивают изображение на бумаге с помощью сопел, через которые выдуваются мельчайшие капельки чернил. Этот способ печати дает более высокое качество печати по сравнению с матричными принтерами и очень удобен для

цветной печати. Струйные принтеры содержат 24 или 64 сопла. Скорость печати у них в несколько раз выше, чем у матричных принтеров. Основной недостаток струйных принтеров заключается в том, что при случайном попадании влаги на готовый отпечаток он размазывается. Скорость печати струйных принтеров составляет до 20 страниц за минуту.

Лазерные принтеры обеспечивают в настоящее время наилучшее (близкое к типографскому) качество печати. В них используется электрографический принцип создания изображения. Основными элементами лазерного принтера являются специальный барабан, источник лазерного излучения, электронная схема управления и сухой порошок – *тонер.* Тонер представляет собой кусочки железа, покрытые пластиком. Принцип работы лазерного принтера заключается в следующем. Изображение формируется построчно. Полупроводниковый барабан предварительно электризуется. Затем под управлением электронной схемы луч лазера пробегает по барабану. При наличии информации луч включается, при отсутствии – выключается. Там, где луч лазера коснулся барабана, электрический заряд стекает с барабана. Таким образом на поверхности барабана оказывается нанесенным скрытое изображение. На следующем этапе осуществляется «проявление» изображения – частички тонера притягиваются к тем участкам барабана, которых коснулся луч лазера, под действием сил электростатического взаимодействия зарядов разного знака. Когда видимое изображение на барабане построено, то есть он покрыт тонером в соответствии с изображением оригинала, лист бумаги заряжается таким образом, что частички тонера с барабана притягиваются к нему. Прилипший порошок закрепляется на бумаге за счет нагрева частиц тонера до температуры плавления пластмассы, покрывающей частицы железа. В результате этих операций формируется несмываемый водоупорный отпечаток.

Цветные лазерные принтеры формируют изображение, последовательно накладывая голубой, пурпурный, желтый и черный тонеры на фоточувствительный барабан. Принтер работает в четырехпроходном режиме, поэтому скорость печати цветного принтера существенно меньше скорости печати черно-белого принтера.

Скорость печати лазерных принтеров достигает 30 страниц за минуту.

К потребительским качествам принтеров относятся следующие показатели:

качество и скорость печати;

наличие русского шрифта в ПЗУ (аппаратная русификация) или возможность его загрузки из программы компьютера;

надежность;

количество шрифтов, которые поддерживает принтер;

возможность и удобство смены красящего элемента;

совместимость с имеющимися программами по управляющим кодам.

Дополнительные устройства компьютера

Дополнительные устройства обеспечивают ввод в компьютер и вывод из него текстовой и графической информации.

Джойстик – это специальное устройство для ввода информации в компьютер. Он представляет собой стержень (ручку), свободно перемещаемый в двух измерениях, и две кнопки-переключатели. Программа считывает положение стержня в виде двух координат X и Y и преобразует их в положение курсора на экране. Нажатие кнопок-переключателей также может быть зафиксировано программой. Используется джойстик чаще всего как манипулятор в игровых программах.

Световое перо предназначено для считывания информации с экрана монитора, а именно координат точки экрана, к которой оно прикасается. Световое перо используется чаще всего в системах автоматизированного проектирования. При использовании светового пера программа позволяет «рисовать» на экране, «брать» и перемещать части рисунка по экрану, производить выбор из меню, прикасаясь пером к элементу данных, и так далее.

Графопостроитель (плоттер) представляет устройство для вывода чертежей на бумагу. Графопостроители бывают барабанного и планшетного типа, используются, как правило, в системах автоматизированного проектирования (САПР) для вывода графических изображений.

Графический планшет – это устройство для ввода контурных изображений в ЭВМ. Используется в САПР для ввода чертежей в компьютер.

Сканер (читающий автомат) – оптико-электронное устройство, предназначенное для считывания графической и текстовой информации в компьютер. Текст вводится при этом в компьютер в виде графических символов. Распознавание текста осуществляется специальными программами, поставляемыми со сканером. При наличии соответствующего программного обеспечения можно распознавать и рукописный текст. Сканеры бывают планшетные и руч-

ные. Настольные сканеры обрабатывают весь лист бумаги целиком. Ручные сканеры необходимо проводить над нужным текстом или рисунком вручную.

Сетевой адаптер – используется для подключения компьютера в локальную вычислительную сеть. При этом пользователь получает доступ к данным, находящимся на другом компьютере.

Контрольные вопросы

1. Назовите основные классификационные признаки ЭВМ.
2. Приведите структурную схему ЭВМ и поясните принцип ее работы.
3. Приведите основные характеристики персональных компьютеров.
4. Поясните назначение и основные типы микропроцессоров.
5. Какие технические приемы и технологии применяются для повышения быстродействия процессоров?
6. Приведите классификацию памяти современного компьютера и поясните ее.
7. В чем заключается принцип работы накопителей на магнитных дисках?
8. В чем заключается принцип работы накопителей на оптических дисках?
9. На чем основан принцип действия твердотельных полупроводниковых накопителей?
10. Назовите условные поля клавиатуры и поясните их назначение.
11. Поясните принцип работы мониторов на электронно-лучевых трубках (CRT-мониторов).
12. Поясните принцип действия жидкокристаллических мониторов.
13. Поясните принцип работы матричных печатающих устройств.
14. Поясните принцип работы лазерных печатающих устройств.

Заключение

Персональные компьютеры реализуют в основном фон-неймановский принцип программного управления, имеют открытую архитектуру, что позволяет пользователям самостоятельно менять конфигурацию компьютера путем подключения дополнительных устройств. Ядром вычислительной системы является процессор, который наряду с памятью определяет основные потребительские качества компьютера.

1.3. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ КОМПЬЮТЕРА

Ключевые слова: каталог, операционная система, папка, программное обеспечение, файл, файловая система.

1.3.1. Классификация программного обеспечения

Программным обеспечением (ПО) называется комплекс программ, описаний и инструкций, позволяющих автоматизировать отладку программ и решение задач на ЭВМ.

В зависимости от назначения программное обеспечение ПК делится на две большие группы: базовое и прикладное.

Базовое программное обеспечение

Базовое программное обеспечение включает комплекс программ, обеспечивающих управление ресурсами вычислительной системы, подготовку программ и техническое обслуживание компьютера и делится на четыре группы:

операционная система (ОС) и ее окружение;
системы программирования;
средства контроля и диагностики;
средства телекоммуникации и сетевой поддержки.

Основными компонентами ОС являются управляющие программы, программы-интерпретаторы, файловая система, утилиты.

Управляющая программа планирует ресурсы ПК, обеспечивает взаимодействие его с внешней средой, оперативно контролирует исправность технических средств и организует восстановление процессов обработки данных после появления неисправности.

Интерпретаторы обеспечивают диалог пользователей с системой, воспринимают и расшифровывают (интерпретируют) его команды, обеспечивают их выполнение.

Файловая система представляет совокупность средств ОС, обеспечивающих выполнение операции поиска и ввода/вывода данных, создания, копирования, удаления, переименования файлов, обработку прерываний.

Утилиты – специальные сервисные программы, расширяющие возможности ОС или облегчающие использование ее команд с помощью программных оболочек. Примерами таких программ являются нортоновские утилиты, программы *Total Commander, Far Manager* и другие.

Системы программирования образуют языки программирования, в состав которых входят трансляторы, компиляторы, интерпретаторы программ, представленных на этих языках, соответствующая программная документация, а также вспомогательные средства для подготовки программ к выполнению.

Средства контроля и диагностики предназначены для обеспечения процедур контроля и диагностики неисправностей, проверки и восстановления работоспособности системы. После включения ПК обычно сразу же запускается программа проверки состояния внешних устройств, оперативной памяти и микропроцессора. В случае обнаружения неисправности на экран видеомонитора выдается сообщение о характере обнаруженной неисправности, рекомендации по дальнейшей работе. Вместе с компьютером поставляются обычно на дисках программы – драйверы всех основных устройств.

Драйверы – специальные программы, управляющие устройствами ввода-вывода и оперативной памятью, обеспечивающие подключение новых внешних устройств и нестандартное использование уже имеющихся устройств (драйверы видеокарты, дисководов, мыши, клавиатуры, принтера и т. д.). Как правило, при покупке нового оборудования для компьютера покупатель получает и диск с драйвером данного устройства.

Средства телекоммуникации и сетевой поддержки обеспечивают возможность передачи данных между компьютерами, объединение их в локальные или глобальные вычислительные сети.

Прикладное программное обеспечение

Прикладное программное обеспечение осуществляет ориентацию персонального компьютера на конкретные области применения. Оно включает пакеты прикладных программ (ППП) и специальные программные продукты (оригинальные программы).

В настоящее время выработана следующая классификация прикладного программного обеспечения: по способу реализации и принципам функционирования, по области применения и классам решаемых задач, по ориентации на определенный метод или процедуру обработки данных.

По способу реализации и принципам функционирования различают библиотеки прикладных программ (БПП), пакеты прикладных программ (ППП), интегрированные пакеты.

Библиотеки прикладных программ являются простейшей формой организации программного обеспечения и представляют собой

набор программ в совокупности с системой их вызова, хранения и настройки на соответствующие параметры, предназначенные для решения задач определенного класса. Программы, входящие в библиотеку, написаны, как правило, на одном языке программирования.

Пакеты прикладных программ – более совершенная форма организации программного обеспечения. ППП – это комплекс взаимосвязанных программ и системных средств, работающих под управлением головной (организующей) программы пакета – программы монитора. ППП ориентирован, как правило, на решение определенного класса задач, близких по содержанию или применяемым математическим методам, например, математические ППП Matematica, Mathcad, Matlab.

Интегрированные пакеты – это более высокий уровень организации программного обеспечения, вызванный расширением технических возможностей ПК. В отличие от наборов ППП, не связанных между собой, интегрированный пакет представляет собой единый программный комплекс с возможностью обмена данными между его компонентами. В интегрированной операционной среде вместо сложной системы различных режимов и командных строк реализуется простое и эффективное взаимодействие с ПК с использованием перекрывающихся окон и манипулятора «мышь». При работе в интегрированной среде не требуется обращение к другим программам. Но работа с интегрированными пакетами требует наличия значительного объема оперативной и внешней памяти компьютера. *Интегрированные пакеты* объединяют функции многих систем. Характерной особенностью интегрированных систем является многооконный интерфейс, позволяющий отображать, например, в одном окне данные, выбираемые из базы данных, во втором окне – данные, записанные на электронном бланке, в третьем – графическое представление данных и так далее. Можно обмениваться данными между программами, вызванными в разные окна. Интегрированные системы предоставляют также специальный язык, с помощью которого можно запрограммировать работу прикладной системы обработки данных.

Все современные приложения ОС *Windows*: *Word*, *Excel*, *Access* и др. являются интегрированными пакетами.

По области применения и классам решаемых задач прикладное программное обеспечение делится на программы, расширяющие возможности операционных систем, общего назначения, спе-

циального применения, для решения инженерных и научно-технических задач, для решения экономических задач и задач автоматизированных систем управления.

К пакетам прикладных программ общего назначения относятся редакторы текстов (текстовые процессоры), электронные таблицы (табличные процессоры), системы управления базами данных (СУБД), графические редакторы, интегрированные системы.

Редакторы текста позволяют создавать на персональном компьютере различного рода документы, печатать их в заданном формате и требуемом количестве экземпляров. Редакторы текстов отличаются по сложности и объему выполняемых функций. Одни используются для подготовки небольших текстов и документов, другие позволяют обрабатывать документы больших объемов и готовить их к изданию с помощью настольных издательских систем.

Электронные таблицы позволяют решать широкий круг научно-технических, планово-экономических, учетных и других задач, для которых исходные данные и результаты могут быть представлены в табличной форме. При этом обеспечивается хранение в памяти компьютера и просмотр на экране монитора таблиц большого размера.

Системы управления базами данных. Эта группа средств позволяет создавать на базе персонального компьютера автоматизированные информационно-справочные системы различного назначения, обеспечивающие хранение во внешней памяти компьютера на магнитных дисках большого объема информации, поиск и выборку этой информации по запросам, оформление выходных данных в виде документов и отчетов определенной формы.

Графические редакторы обеспечивают вывод на экран дисплея графических изображений при наличии в составе компьютера графического дисплея. Они позволяют наглядно отображать результаты вычислений в виде круговых, линейных, столбиковых и иных диаграмм, а также графиков функций.

Средства презентации позволяют создавать сложные документы с использованием мультимедийных средств: звука, анимации, таблиц, графики и других изобразительных средств – для представления результатов выполненной работы, докладов, обучения, рекламы и т. п.

По ориентации на определенный метод или процедуру обработки программы делятся на методо-ориентированные, проблемно-ориентированные и технологически ориентированные.

Методо-ориентированные ППП предназначены для решения различных научных и инженерных задач, реализуя определенные методы вычислительной математики, в том числе задач математической статистики, сетевого планирования и управления, многокритериальной оптимизации и так далее.

Проблемно-ориентированные ППП предназначены для решения конкретных задач в различных областях применения и создания на базе персональных компьютеров автоматизированных рабочих мест (АРМ) для различных видов профессиональной деятельности.

Технологически-ориентированные ППП предназначены для управления технологическими процессами на фабриках, заводах, в цехах.

1.3.2. Операционная система

Классификация операционных систем

Операционная система – это совокупность программ, управляющих функционированием всех компонентов компьютера.

Операционная система обеспечивает:

автоматический запуск в работу вычислительной системы с проверкой аппаратных и программных средств;

распределение ресурсов системы в соответствии с решаемой задачей и управление работой процессора, памяти, устройств ввода/вывода;

запуск на выполнение прикладных программ и их взаимодействие с аппаратными средствами;

обработку прерываний (аппаратных прерываний, прерываний по запросам программ пользователя);

создание и ведение каталогов, организацию и управление файлами;

управление аппаратными средствами;

диалог с пользователем о ходе обработки информации и работе аппаратных средств.

Операционные системы можно классифицировать по ряду признаков: *количеству одновременно работающих пользователей, числу одновременно решаемых задач, количеству используемых процессоров, типу пользовательского интерфейса, способу использования общих аппаратных средств и программных ресурсов, разрядности*.

По количеству одновременно работающих пользователей ОС делятся на однопользовательские и многопользовательские:

• В *однопользовательских* ОС все ресурсы вычислительной системы находятся в распоряжении одного пользователя.

• В *многопользовательских* ОС ресурсы (в основном процессорное время) распределяются между несколькими пользователями по определенным правилам. При этом ввиду высокого быстродействия процессора пользователю, работающему на *рабочей станции* или *терминале*, кажется, что процессор полностью находится в его распоряжении. И только при выполнении длительных операций, связанных, например, с вводом/выводом информации, пользователь может заметить замедление в работе.

По числу задач, одновременно выполняемых под управлением ОС, они делятся на однозадачные и многозадачные:

• *Однозадачные ОС* предоставляют пользователю виртуальную машину и включают средства управления файлами, периферийными устройствами и средствами общения с пользователями. К выполнению новой задачи ОС может приступить только после завершения выполнения текущей задачи.

• *Многозадачные ОС* дополнительно управляют разделением между задачами совместно используемых ресурсов: времени работы процессора, оперативной и дисковой памяти. При этом в зависимости от способа управления распределением процессорного времени различают ОС с невытесняющей и вытесняющей многозадачностью. *Невытесняющая* многозадачность (Windows 3.x, NetWare) характеризуется тем, что активный процесс выполняется до тех пор, пока он сам, по собственной инициативе, не передаст управление операционной системе. При *вытесняющей* многозадачности (Windows 9.x, Windows NT, OS/2, UNIX) решение о переключении процессора с одного процесса на другой принимается операционной системой, а не самим процессом. Многозадачные ОС позволяют, например, редактировать текст и выводить на печать другую программу в фоновом режиме, включить для комфорта музыку и работать в графическом редакторе или в электронной таблице и т. п.

Одним из важнейших свойств ОС является возможность распараллеливания вычислений в рамках одной задачи. Это свойство получило название *многонитевидность*. Многонитевидная ОС распределяет время не между задачами, а между отдельными ветвями (нитями) текущей задачи.

По количеству используемых процессоров ОС делятся на *однопроцессорные* и *многопроцессорные*. Алгоритмы работы однопроцессорных ОС соответствуют принципам работы ЭВМ фон-неймановской

архитектуры. *Мультипроцессирование* является другой важной характеристикой ОС. Мультипроцессирование приводит к усложнению всех алгоритмов управления ресурсами.

По типу пользовательского интерфейса ОС делятся на *командные* (текстовые) и *объектно-ориентированные* (графические). В настоящее время в связи с ростом быстродействия процессоров и объемов оперативной и дисковой памяти преимущественное распространение имеют графические ОС, так как они предоставляют пользователю значительно более удобный интерфейс и средства управления компьютером с помощью мыши.

По способу использования общих аппаратных средств и программных ресурсов ОС делятся на сетевые и локальные.

Локальные ОС обеспечивают управление только данным компьютером.

Сетевые ОС обеспечивают работу компьютера в локальных и глобальных вычислительных сетях. Сетевые ОС предназначены для эффективного решения задач распределенной обработки данных. Такая обработка ведется не на отдельном компьютере, а на нескольких компьютерах, объединенных сетью. Сетевые ОС поддерживают распределенное выполнение процессов, их взаимодействие, обмен данными между ЭВМ, доступ пользователей к общим ресурсам и ряд других функций. Все сетевые ОС делятся на две группы: одноранговые ОС и ОС с выделенным сервером. В *одноранговых* сетях каждая ЭВМ может выполнять функции как сервера, так и рабочей станции. В *сетях с выделенным сервером* рабочие станции не предоставляют свои ресурсы другим ЭВМ.

По разрядности ОС делятся на 8-, 16-, 32- и 64-разрядные.

Устройства компьютера

Операционная система осуществляет ввод и вывод информации на различные устройства. Такими устройствами являются, как известно, дисководы, монитор, клавиатура, порты параллельного и последовательного ввода/вывода информации. Операционная система обращается к этим устройствам по их логическим именам. Каждая операционная система имеет свои правила доступа к устройствам.

Дисководы как физические устройства имеют номера. Имя диска обозначается буквой латинского алфавита с двоеточием. Например: первый диск обозначается *A*:, второй – *B*:, жесткий диск – *C*: Ввиду того, что емкость жестких дисков в настоящее время очень большая, их с помощью программных средств разбивают на логи-

ческие диски. На жестком диске в этом случае могут быть расположены логические диски *D*:, *E*:, *F*:, *G*:, *H*: и т. д.

Для других устройств установлены следующие логические имена:

PRN – печатающее устройство (принтер);

CON – консоль. При вводе информации под консолью понимается клавиатура, при выводе информации — экран видеомонитора;

NUL – пустое устройство: все операции ввода-вывода для этого устройства игнорируются. Используется при отладке программ;

LPT1–LPT3 – устройства, присоединяемые к параллельным портам 1...3 (обычно это принтеры);

COM1–COM3 – устройства, присоединяемые к асинхронному последовательному порту 1, 2, 3;

AUX – дополнительное устройство, присоединяемое к асинхронному последовательному порту 1.

Наиболее часто используются устройства PRN и CON. Имена устройств используются в командах ОС для обращения к ним.

На современных компьютерах число разъемов значительно больше: имеются специальные разъемы для подключения звуковых колонок, устройств мультимедиа, игровых приставок, внешней сети, телефона, микрофона, наушников.

Файловая система

Файловая система – это часть операционной системы, которая обеспечивает пользователю удобный интерфейс при работе с данными, хранящимися на диске, совместное использование файлов несколькими пользователями и процессами. Основными задачами файловой системы являются организация хранения, поиска, переименования, копирования, пересылки и удаления файлов и каталогов.

Файлы

Информация на дисках или других машинных носителях, а также в памяти компьютера хранится в файлах.

Файл – это организованный, поименованный набор данных на диске или другом носителе информации.

Файлы бывают разных типов: обычные файлы, специальные файлы, файлы-каталоги. В файлах могут храниться тексты программ, документы, данные, сведения о файлах и других каталогах.

Обычные файлы в свою очередь подразделяются на текстовые файлы и двоичные файлы. Текстовые файлы состоят из строк символов, представленных в ASCII-коде. Это могут быть документы, исходные тексты программ, данные и т. п. Текстовые файлы можно прочитать на экране и распечатать на принтере. Двоичные файлы не

используют ASCII-коды, они часто имеют сложную внутреннюю структуру, например, объектный код программы или архивный файл.

Специальные файлы – это файлы, предназначенные для управления устройствами ввода-вывода.

Файлы-каталоги – хранят сведения о файлах и каталогах.

Имя файла

Каждый файл для возможности его идентификации на носителе информации имеет обозначение, состоящее из двух частей: имени и расширения.

Имя файла в дисковой операционной системе может содержать от 1 до 8 символов. Имя должно начинаться с буквы и не должно содержать знаков пунктуации и пробелов, может содержать специальные символы: _ - $ # & @ ! % ( ) {} `~ ^. В операционной системе Windows для обозначения имен файлов можно использовать длинные имена до 255 символов. Это позволяет присваивать файлам вполне понятные имена, характеризующие их содержание. В имени файла могут использоваться любые символы, имеющиеся на клавиатуре, кроме символов \ / : * ? < > |. В имени файла допускается использование пробела и точки, при этом последняя точка считается началом расширения имени файла.

Расширение имени файла

Расширение имени файла начинается с точки, за которой следует несколько символов. В дисковой операционной системе длина расширения имени файла фиксирована и не должна превышать трех символов. В операционной системе Windows расширение имени файла может содержать и большее число символов. Расширение имени файла не является обязательным, однако оно является очень полезным, так как характеризует вид хранимой в нем информации. Некоторые расширения имен файлов приведены в таблице 1.10. Примеры имен файлов: *command.com*, *expert1.bas*, *autoexec.bat*. В имени фала *autoexec.bat*:

autoexec – имя файла;

.bat – расширение имени файла;

autoexec.bat – полное имя файла.

Таблица 1.10

Примеры использования типовых расширений имен файлов

Расширение	Вид файла
.COM, .EXE	программные файлы в машинных кодах
.BAS	программный файл на языке БЕЙСИК

Окончание табл. 1.10

Расширение	Вид файла
.PAS	программный файл на языке ПАСКАЛЬ
.FOR	программный файл на языке ФОРТРАН
.BAK	файл-копия
.BAT	командный файл
.TXT	текстовый файл, документ
.DOC	документ, подготовленный в редакторе Microsoft Word
.XLS	документ электронной таблицы Excel

Расширения имен файлов, приведенные в таблице 1.10, являются зарезервированными, то есть используются по умолчанию при работе с соответствующими программами.

Файлы, имеющие расширение .EXE, .COM или .BAT, считаются внешними командами ОС. При вызове внешней команды можно вводить только имя файла без расширения. Если используется несколько файлов, имеющих одинаковые имена, но разные расширения, то при вводе имени этой команды ОС выполнит только одну программу в соответствии с приоритетом: .COM, .EXE, .BAT. Файлы с указанными расширениями называются *исполняемыми*. Причем файлы типа .COM и .EXE хранятся в двоичных кодах, а файлы с расширением .BAT – в символьном виде и содержат последовательность команд, которые должны выполняться так же, как и при вводе с клавиатуры.

Атрибуты файлов

Файл может иметь один из четырех атрибутов: R – только для чтения, A – не архивирован, S – системный, H – скрытый. Если файл имеет атрибут R, то в него нельзя внести изменения. Данный атрибут позволяет защитить файл от несанкционированного изменения пользователем. Атрибут «системный» предназначен для использования операционной системой. Такой файл имеет защиту от несанкционированного удаления. При попытке удаления системного файла операционная система выдает предупреждение о том, что это опасная операция, и требует подтверждения на удаление. Атрибут «скрытый» позволяет скрыть файл от посторонних глаз.

Маска

При выполнении некоторых операций с файлами: копировании, переименовании, удалении, поиске файлов – возникает необходимость выделить группу файлов, имеющих, например, одинаковые

имена или расширения имен. В этом случае для выделения файлов применяются маски.

Маска – это символ, который заменяет все слово, его часть или один символ.

В качестве маски используются символы * и ?. Символ * заменяет все имя файла, расширение имени файла или их часть, символ ? означает любой символ в месте расположения данного знака. Например:

. – все файлы на текущем диске;

*.com – все файлы с расширением .com;

a*.sys – все файлы с расширением .sys, имя которых начинается с символа «a»;

contr?.bas – все файлы с именем contr и расширением .bas, отличающиеся последним символом в имени файла (contr1.bas, contr2.bas и т. д.).

Каталог (папка)

Имена файлов регистрируются на магнитных дисках в каталогах (или папках). Понятие «каталог» использовалось в дисковой операционной системе. В операционной системе Windows вместо понятия «каталог» введено понятие *папка.* В дальнейшем не будем делать различия между этими понятиями.

Каталог – это, с одной стороны, файл, в котором хранятся сведения о файлах и подчиненных каталогах. С другой стороны, каталог – это группа файлов или каталогов, объединенных по какому-либо признаку: принадлежность одной программе, одному пользователю, имеющих один тип и т. п.

Каталог – это файл, в котором хранятся сведения о файлах и подчиненных каталогах.

К сведениям, которые хранятся в каталогах, относятся: имя и расширение имени файла, размер файла в байтах, дата и время его создания или последней модификации, *атрибуты* файла, начальный адрес файла на диске. На каждом диске всегда имеется один каталог, который называется корневым и обозначается символом \ – обратный слэш. Все остальные каталоги размещаются в корневом каталоге. На диске может быть несколько каталогов. Наличие каталогов позволяет сгруппировать файлы по назначению, теме или пользователю, что облегчает их поиск на диске. Структуру каталогов на диске принято называть деревом каталогов (рис. 1.15).

Имена каталогов (папок) образуются по тем же правилам, что и имена файлов. Каталог, с которым работает пользователь в на-

стоящие момент, называется текущим. Если в команде указать имя файла, то он будет отыскиваться или создаваться в текущем каталоге. Для отыскания или создания файла в другом каталоге необходимо перейти в этот каталог или указать путь (маршрут) к этому каталогу.

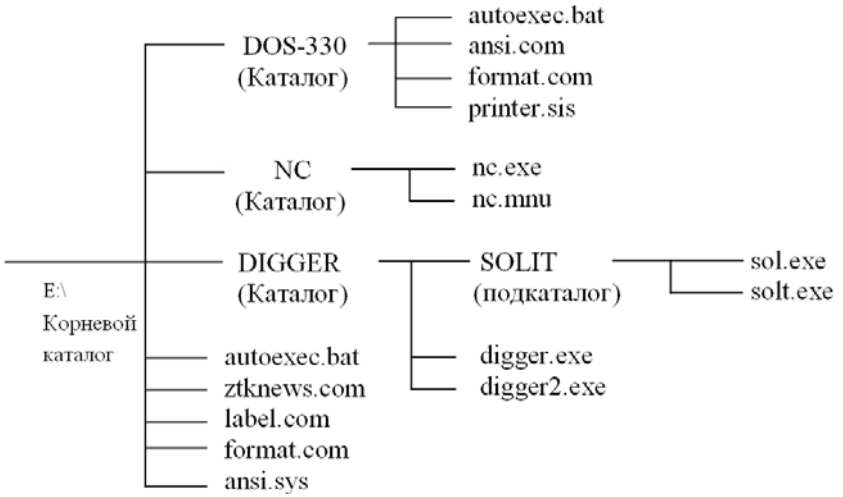


Рис. 1.15. Дерево каталогов

Путь (маршрут) – последовательность каталогов, разделенных символом \, ведущая к файлу.

Часто файл находится не на том диске, с которым работает пользователь. В этом случае при указании пути поиска необходимо указать еще и имя диска, например:

E:\DOS330\ansi.sys

E:\DIGGER\COLIT\col.exe

Здесь E: – имя диска; DOS330, DIGGER, COLIT – имена каталогов и подкаталогов.

Таким образом, ***полное имя файла*** включает имя диска, путь, имя и расширение имени файла:

[дисковод:] [путь] имя файла [расширение]

Обозначения, указанные в квадратных скобках, необязательны. Если дисковод не указан, то подразумевается текущий диск. Если каталог (подкаталог) является текущим, то путь также не указывается, например:

shrift.com

A:chrdsk.com

A:\UCH\sessia.bas

Полное имя файла называют также ***спецификацией***.

Число символов в пути не должно превышать 64, а длина полного имени файла не должна превышать 255 символов.

Таблица размещения файлов

Сведения о файлах и каталогах хранятся в специальной области диска. В операционной системе Windows 9x (общее обозначение семейства ОС Windows 95, Windows 98) применяются файловые системы FAT-16, FAT-32. В операционной системе Windows NT, Windows XP и старших версиях ОС Windows применяется файловая система NTFS.

В ОС Windows 95 используется FAT-16, в Windows 98 – FAT-16 и FAT-32. В этих файловых системах сведения о каталогах хранятся в корневом каталоге, а сведения о файлах – в таблице размещения файлов – FAT. Для обеспечения надежности на диске создается копия таблицы размещения файлов. Операционная система тщательно «следит» за тем, чтобы информация в обеих копиях была идентична.

Минимальным физическим объемом файла на дискетах является сектор одной дорожки. Его размер равен 512 байтам. Это минимальный размер адресуемого дискового пространства. На жестких дисках минимальный размер адресуемого дискового пространства может составлять несколько секторов. Эти группы секторов называют ***кластерами***. Размер кластера зависит от размера диска и от типа используемой файловой системы. В операционной системе MS-DOS применяется файловая система FAT-16, то есть она имеет 16 разрядов для адресации файлов и папок. Имея 16 разрядов, можно задать 65 536 адресов. Эта файловая система не может работать с дисками больших размеров из-за ограниченности количества адресов (устанавливать FAT-16 в разделы дисков больше 2 Гбайт не рекомендуется). Например, при объеме диска более 512 Мбайт размер кластера составит 64 кбайт. Это значит, что если создать файл, содержащий всего один байт (логический размер), то на диске он займет 64 кбайт (физический размер), то есть будет иметь место неэффективное использование дискового пространства. Другим ограничением является число секторов в кластере. Это объясняется следующим: для хранения числа секторов в кластере в FAT-16 отведен только один байт, следовательно, можно адресовать 255 секторов, но так как число секторов в кластере должно быть кратно 2,

то максимальное число секторов в кластере ограничено 128 (64 кбайт). Еще одним ограничением FAT-16 было то, что размер главного каталога был фиксирован и занимал на диске строго определенное место.

Файловая система FAT-32 успешно работает с дисками больших размеров (до 2 терабайт). Но и в этой операционной системе при емкости диска 8 Гбайт размер кластера согласно спецификации составляет 4 кбайт. В файловой системе FAT-32 сняты также ограничения на размер главного каталога. Одним из недостатков FAT-32 является то, что в оперативную память загружаются обе копии файловой системы и занимают достаточно большой объем памяти, при 2-х гигабайтном диске – 4 Мбайт.

В связи с указанными недостатками файловых систем FAT-16 и FAT-32 в операционных системах Windows NT и Windows XP, Windows 7 применяется файловая система NTFS. Она поддерживает диски объемом до 16 777 216 терабайт. Эта файловая система обладает большими преимуществами перед файловыми системами FAT-16 и FAT-32. К достоинствам файловой системы NTFS следует отнести следующие:

- Каждый элемент NTFS – это файл. Самый главный файл, который имеет имя $MFT, представляет собой централизованный *каталог* всех остальных файлов диска. В системе NTFS нет понятия «главный каталог», как это было в FAT-16 и FAT-32. Отсутствуют ограничения на размер каталога. Файл $MFT – это и есть главный каталог.
- Общая таблица файлов (MFT) имеет копию. Основная таблица размещается в начале диска, а копия – в середине. Это обеспечивает повышение надежности файловой системы при возникновении сбойных участков на диске в области размещения MFT-зоны. MTF-зона – это область диска, которую операционная система зарезервировала для размещения файла файловой системы.
- Поддерживает кластеры разных размеров: от 512 байт до 64 кбайт. Стандартом считается файл размером 4 кбайт.
- Обеспечивает более высокий уровень отказоустойчивости. Отказоустойчивость обеспечивается использованием ***транзакции*** (запрос на изменение данных). Механизм транзакции состоит в том, что при выполнении операций записи/чтения действие должно быть совершено целиком или вообще не совершено. Если, например, при записи информации на диск обнаружено повреждение поверхности диска, то операция копирования откладывается

полностью, неисправный участок помечается как сбойный, а данные записываются в другое место. То же произойдет при недостатке места на диске, например, при сохранении файла, уже имеющегося на диске. Операция будет отложена, данные на диске сохранятся в прежнем виде.

- Обеспечивается высокая степень безопасности за счет разграничения прав доступа к файлам и папкам. Права доступа могут быть установлены для каждого пользователя, группам пользователей или и отдельным пользователям, и большим группам.
- Имеет встроенную поддержку сжатия дисков. Любой файл или каталог может храниться на диске в сжатом виде. Сжатие файлов имеет большую скорость. При этом часть файла может храниться в несжатом виде, а часть – в сжатом.
- Позволяет зашифровать файлы или каталоги, обеспечивая безопасность важных данных, применяется 128-битовое шифрование файлов и папок.

Отличительной особенностью NTFS является то, что с увеличением размеров дисков эффективность хранения файлов на дисках при использовании NTFS не снижается. Файловая система NTFS обеспечивает большее быстродействие за счет кэширования файлов, а также большую эффективность использования дискового пространства. Кэширование файлов поддерживается для сетевых и совместно используемых дисков. Кэширование ускоряет доступ к файлам, а также позволяет пользователям работать с ними в автономном режиме.

Файловая система NTFS не совместима с файловыми системами FAT-16 и FAT-32. При установке на одном компьютере нескольких операционных систем допускается использовать две файловые системы, например, FAT-32 и NTFS. Нельзя устанавливать на компьютер операционные системы, использующие разные версии файловой системы NTFS.

При работе с разделами жесткого диска, превышающими 32 Гбайт, рекомендуется использовать файловую систему NTFS.

1.3.3. Включение компьютера и выключение компьютера

Включение напряжения питания

Включение современных компьютеров осуществляется предельно просто: достаточно нажать кнопку включения питания на лицевой панели системного блока и на панели управления монитора.

Бытовые компьютеры имеют, как правило, одну кнопку включения питания на лицевой панели системного блока, но могут иметь также выключатель (тумблер), расположенный на задней стенке системного блока. В этом случае для включения компьютера необходимо включить вначале выключатель, а затем нажать кнопку включения питания.

Загрузка операционной системы

После включения питания процессор под управлением базовой системы ввода-вывода (BIOS), хранящейся в ПЗУ, проверяет наличие установленных внешних устройств, памяти и проводит их инициализацию. Затем проверяются платы расширения, в том числе и видеоадаптера. После этого проверяется ОЗУ. Эта проверка может занять значительное время, до нескольких минут. Если будет обнаружена критическая ошибка: неисправность блоков памяти, неисправность клавиатуры, – то на экран выводится код ошибки и работа программы завершается. При неисправности монитора выдается последовательность звуковых сигналов. Если проверка прошла успешно, то BIOS пытается загрузить ОС с первого (обычно жесткого) диска. Если на жестком диске не обнаружено блока начальной загрузки[8], то осуществляется попытка загрузить ОС с компакт-диска. *Последовательность опроса дисков может быть изменена путем настройки BIOS*.

Если на первом диске обнаружена неисправность, то выдается сообщение об ошибке.

При обнаружении в первом секторе нулевой дорожки блока начальной загрузки BIOS загружает его в оперативную память компьютера и передает ему управление. Блок начальной загрузки загружает остальную часть операционной системы в ОЗУ компьютера. После загрузки ОС Windows на экране появляется рабочий стол с элементами управления.

Если на компьютере установлено несколько операционных систем, то после обнаружения главной загрузочной записи на экран выводится меню, из которого надо выбрать ту операционную систему, которая требуется. Если операционная система не выбрана пользователем, то загружается операционная система, установленная по умолчанию.

[8] Блок начальной загрузки – небольшая программа, расположенная в первом секторе нулевой дорожки системного диска и предназначенная для загрузки операционной системы с диска в оперативную память компьютера.

В некоторых случаях, например, при «зависании» компьютера, требуется перезагрузка компьютера. Для перезагрузки компьютера следует нажать клавишу RESET (перезапуск), установленную на лицевой панели системного блока.

Завершение работы. После окончания работы необходимо закрыть все приложения. Выход из программы Windows следует осуществлять только через команду главного меню ***Завершение работы***. В этом случае будет обеспечено сохранение всех настроек пользователя и сохранность данных. При выходе иным способом возможны нарушение целостности файловой системы и потеря данных. В таких случаях при повторном включении компьютера автоматически включается проверка дисков и восстановление файловой системы. Если пренебрегать проверкой дисков, то ошибки будут накапливаться и может наступить день, когда компьютер откажется нормально загружаться и нормально работать. После ввода команды *Завершение работы* на экране монитора появляется окно диалога, в котором необходимо отметить мышью флажок «Выключить компьютер» и щелкнуть по кнопке ДА (ОК). Программа приступит к выполнению операций по подготовке к выключению питания.

Выключение компьютера

После завершения всех подготовительных операций к выключению питания на экран монитора выводится сообщение «Теперь питание компьютера можно отключить». В некоторых операционных системах выключение питания происходит автоматически. Мониторы часто включаются и выключаются независимо от системного блока, поэтому не забудьте выключить питание видеомонитора.

Контрольные вопросы

1. Что включает в себя базовое программное обеспечение?
2. Какие программные средства относятся к прикладному программному обеспечению?
3. Назовите классификационные признаки операционных систем.
4. Назовите наиболее известные операционные системы, применяемые на персональных компьютерах.
5. Что такое файловая система, для чего она предназначена?
6. Что такое файл?
7. Что такое имя, расширение и спецификация файла? Приведите примеры записи спецификации файла.
8. Назовите наиболее распространенные расширения имен файлов. Что они означают?

9. Поясните, что такое маска. Приведите примеры использования масок.

10. Что такое атрибут файла, какие атрибуты имеет файл?

11. Что такое каталог? Какая информация в нем содержится?

12. Что такое спецификация файла? Приведите примеры.

Заключение

Программное обеспечение является второй составляющей, определяющей потребительские качества персонального компьютера. Оно определяет функциональные возможности компьютера по обработке данных. Ядром программного обеспечения является операционная система. В настоящее время доминирующее положение занимают графические операционные системы, которые обеспечивают пользователю дружественный интерфейс и берут на себя все функции по организации взаимодействия с внешними устройствами.

1.4. ОПЕРАЦИОННАЯ СИСТЕМА WINDOWS

Ключевые слова: Total Commander, Windows 7, меню, объект, окно, проводник, рабочий стол, системный реестр.

1.4.1. Общие сведения

Windows 7 – новая операционная система для персональных компьютеров от фирмы Microsoft Office. Она пришла на смену операционной системе Windows Vista, на которую было много нареканий у пользователей. Windows 7 относится к семейству ОС Windows NT, серийный номер 6.1. Операционная система имеет закрытый исходный код, базируется на надежном и защищенном ядре, интерфейс Aero. Первый выпуск операционной системы состоялся 22 октября 2009 года.

Программа имеет шесть редакций, отличающихся набором приложений и возможностей: Начальная, Домашняя базовая, Домашняя расширенная, Профессиональная, Корпоративная, Максимальная. Для рядового пользователя вполне подойдет Профессиональная. ОС выпускается как 32-х, так и 64-х разрядная версия. Максимальный объем оперативной памяти для 32-х разрядной версии ограничен 4 Гбайт, 64-х разрядная версия способна поддерживать оперативную память объемом в 192 Гбайт.

По классификации Windows 7 является объектно-ориентированной, многозадачной, многопроцессорной, многонитевой, однопользовательской операционной системой.

Особенности операционной системы Windows 7

Поддержка нескольких процессоров.

Поддержка нескольких мониторов.

Быстрое переключение между пользователями.

Возможность смены фонового рисунка рабочего стола.

Наличие диспетчера рабочего стола.

Улучшенное распознавание рукописного текста (только для английского языка).

Имеется эмулятор Windows XP для обеспечения совместимости с программами для Windows XP.

Имеется система шифрования данных – EFS.

Используется технология ***Мультитач*** (англ. multitouch) – технология, по которой сенсорный экран или *тачпад* отслеживает одновременно несколько точек нажатия. Например, сближая пальцы рук, можно уменьшить картинку на дисплее, а раздвигая – увеличить. Кроме того, мультитач-экраны позволяют работать с устройством одновременно нескольким пользователям.

Единый графический интерфейс

Пользовательские программы, разработанные для работы в среде Windows, принято называть ***приложениями***. Пользовательский интерфейс приложений Windows имеет одинаковое исполнение, единые приемы управления. Научившись работать в одном из приложений, можно достаточно быстро освоиться в другом приложении. В состав пользовательского интерфейса входят такие элементы, как рабочий стол, окна, меню, контекстные меню, панели инструментов, линейки прокрутки, средства навигации, приемы управления элементами интерфейса.

Объектно-ориентированная платформа операционной системы

Windows 7 – это объектно-ориентированная графическая операционная система. Все элементы операционной системы – это, по сути, объекты со своими свойствами. Вычисления в компьютере осуществляются путем обмена данными между объектами. Один объект требует, чтобы другой объект выполнил некоторые действия. Объекты взаимодействуют, посылая сообщения. *Сообщение* – это запрос на выполнение действия. Каждый объект имеет независимую память, которая состоит из других объектов. Каждый объект является представителем класса, который выражает свойства принадлежащих ему объектов. В классе задается поведение объекта, поэтому все объекты, принадлежащие к данному классу, могут выполнять одинаковые действия. Например, кнопка в текстовом про-

цессоре, табличном процессоре или в системе управления базами данных выполняет одинаковые функции. Все классы образуют иерархическую древовидную структуру, отражающую иерархию наследования. Память и поведение, связанные с экземпляром определенного класса, могут использоваться любым классом, расположенным ниже в иерархической структуре.

Использование новых компьютерных технологий

Развитие графических операционных систем привело к возникновению понятия *компьютерные технологии* – это алгоритмы и их конкретная реализация для передачи, хранения и обработки информации в цифровом виде.

> *Компьютерные технологии – это алгоритмы и их конкретная реализация для передачи, хранения и обработки информации в цифровом виде.*

При разработке ОС фирмой Microsoft использованы различные технологии, которые обеспечивают возможность повторного использования кода, упрощения взаимодействия приложений, позволяют использовать программы, написанные на различных языках программирования, обмениваться данными между программами, обнаруживать новые программы и объекты, создавать пользовательские макропрограммы, окна диалога, меню и т. д. на базе компонентов, имеющихся в Windows, приложениях или загруженных из сети Интернет.

Технология COM (модель компонентных элементов) устанавливает открытый стандарт, который определяет принципы взаимодействия программ. COM – это архитектура программного обеспечения, позволяющая приложениям пользователя взаимодействовать с другими компонентами, в том числе поставляемыми другими разработчиками и содержащимися в коммерческих приложениях. COM-компоненты обычно представляют собой библиотеки классов – файлы с расширением .exe, .ocx, .dll. В роли COM-компонентов могут выступать и такие приложения Windows, как Word, Excel, PowerPoint, Internet Explorer. В то же время стандартные приложения Windows (Word, Excel и др.) являются COM-серверами[9], поддерживающими ***Автоматизацию***.

[9] Сервер – объект (программа), предоставляющий свои ресурсы другим объектам (программам). Клиент – объект (программа), запрашивающий ресурсы. Клиент-сервер – способ организации взаимодействия компонентов и программ. Клиент может инициировать выполнение процедур сервером и получать результаты.

Разработка программ на базе компонентов ускоряет программирование, позволяет собирать приложения из уже проверенных и стандартизированных компонентов, что значительно сокращает сроки разработки программных продуктов и повышает их качество. На основе COM-компонентов можно создавать ActiveX-компоненты.

ActiveX – это технология, основанная на COM, а ActiveX-компонент – это модуль исполняемого кода (например, файлы с расширением .exe, .dll или .ocx), созданный в соответствии со спецификацией ActiveX на повторно используемые объекты. Несмотря на схожесть понятий, имеется существенная разница между объектно-ориентированным программированием (ООП) и разработкой программного обеспечения на основе ActiveX-технологии. ООП – это способ разработки программных компонентов на базе объектов только для одной среды разработки. А технология ActiveX позволяет комбинировать объекты независимо от того, на каком языке они написаны. ActiveX и COM позволяют использовать компоненты, созданные на основе ООП во многих средах.

Автоматизация – это технология, позволяющая программам предоставлять свои объекты средствам разработки, макроязыкам и другим программам. Например, редактор электронной таблицы может предоставлять разработчику такие объекты, как таблица, ячейка, группа ячеек, а текстовый процессор – объекты в виде документа, абзаца, предложения, выделенного фрагмента. Любое клиентское приложение способно использовать внешний COM-компонент через автоматизацию, независимо от того, в каком файле он содержится – *.exe или *.dll.

Технология drag-and-drop позволяет перемещать объекты в окнах и на рабочем столе при помощи мыши.

Обмен данными между приложениями

Для реализации этой возможности в ОС Windows предусмотрено три механизма:

- Буфер обмена (ClipBoard);
- DDE (Dinamic Data Exchange) – динамический обмен данными;
- OLE (Object Linking and Embedding) – механизм связи и внедрения объектов.

В основе указанных механизмов обмена информацией лежит автоматизация.

Буфер обмена (ClipBoard) – это специальным образом организованное динамическое пространство оперативной памяти для временного размещения данных. Программа хранит не только данные, но и сведения о программном приложении, которому оно принад-

лежит. Буфер обмена – это буфер *межпрограммного обмена* данными. После того как данные помещены в буфер, они могут быть вставлены из него в текущий или в любой другой документ, подготовленный данным приложением или другим приложением. При помещении в буфер новых данных старые данные уничтожаются. Буфер можно просмотреть и при необходимости очистить.

Технология DDE представляет собой набор системных процедур, позволяющих обращаться из одного приложения (DDE-клиента) в процессе его выполнения к другому, активному на тот момент, программному приложению (DDE-серверу). По запросу клиента сервер обрабатывает запрос и возвращает результаты в той или иной форме приложению-клиенту. Имеется возможность установить привязку внедренного объекта к источнику данных, благодаря этой привязке все изменения, проведенные в приложении-сервере, будут отражены во внедренном объекте.

Технология OLE – *OLE 1* – это усовершенствованная технология DDE. Она не только позволяет внедрить объект, но и обеспечивает возможность редактирования внедренного объекта средствами программы-сервера. Для этого достаточно дважды щелкнуть по внедренному объекту мышкой.

Усовершенствованная технология OLE – *OLE 2* позволяет редактировать внедренный объект в месте вставки, не передавая его в документ-родитель. В OLE 2 улучшена технология *drag-and-drop* за счет введения межоконного перемещения объектов, усовершенствования связывания объектов. При перемещении объектов связь между ними не теряется, как было ранее, а отслеживается. Технологии DDE, OLE 1 и OLE 2 совместимы сверху вниз: приложение-сервер и приложение-клиент обмениваются по наиболее новой технологии, доступной им обоим. Если, например, приложение-сервер поддерживает технологию OLE 2, а приложение-клиент поддерживает технологию OLE 1, то активизировать вставленный объект в приложении-клиенте окажется невозможным.

Поддержка масштабируемых шрифтов

В Windows встроен совершенный механизм поддержки масштабируемых шрифтов *True Type*. Эти шрифты содержат не растровые (поточечные) изображения символов, а описания контуров символов, позволяющие строить символы любого нужного размера.

Установка оборудования

Установка оборудования основана на технологии *Plug and Play*. Поскольку общение прикладных Windows-программ с внешними

устройствами осуществляется через посредство Windows, для подключения к компьютеру любого нового устройства достаточно установить драйвер этого устройства, предназначенного для Windows. Технология Plug and Play обеспечивает автоматическое распознавание устройств для их установки и настройки. Она обеспечивает динамический контроль состояния системы и автоматическое уведомление об этом программных приложений, обеспечивает интеграцию драйверов устройств, системных компонентов и пользовательского интерфейса. Windows обеспечивает динамическое изменение конфигурации системы, построенной на базе данной технологии. Эта технология позволяет работать также с устройствами, не подчиняющимися спецификации Plug and Play, упрощая их настройку и управление оборудованием.

Оборудование, выполненное в соответствии со стандартом Plug and Play, считается самоустанавливающимся. Это значит, что его достаточно подключить только физически, операционная система сама найдет его, определит, что это такое, подберет в своей базе подходящий драйвер и пропишет его в своем реестре. А в тех случаях, когда устройство не соответствует стандарту Plug and Play или когда операционная система не может его распознать или не имеет в своей базе данных подходящего драйвера, установку можно выполнить с помощью программного обеспечения, полученного вместе с устройством при его покупке.

Сетевые возможности

Операционная система Windows обеспечивает пользователя всем комплексом услуг по работе во всемирной информационной сети Интернет: поиск информации, ведение списков посещаемых Web-узлов, поддержка всех основных стандартов Интернета; подготовка Web-страниц; передача информации по электронной почте, факс-модему и др. Сетевые средства Windows обеспечивают включение компьютера во всемирную сеть Интернет (при наличии в компьютере сетевой карты), или в локальную одноранговую компьютерную сеть, или в локальную компьютерную сеть с выделенным сервером. В одноранговой компьютерной сети все компьютеры равноправны и могут выступать как в роли сервера, так и в роли клиента. В локальной компьютерной сети с выделенным сервером все управление работой сети, распределением ресурсов, обработку запросов других компьютеров-клиентов осуществляет один компьютер – сервер. В операционной системе Windows 7 сетевые возможности персонального компьютера значительно расширены. Для

использования всех возможностей сети, например, автоматическое обновление операционной системы, офисных программ, получение и отправка писем, получение различных справок, компьютер должен быть постоянно включен и подключен к сети.

Обеспечение бесперебойной и бесконфликтной работы внешних устройств

Так как к компьютеру может подключаться одновременно несколько внешних устройств, то приняты меры, исключающие конфликты между программами. Для этой цели используются *прерывания* и *каналы прямого доступа к памяти*.

Прерывания. В архитектуре компьютеров PC предусмотрено 15 аппаратных прерываний. Прерывание указывает компьютеру, что какой-то процесс требует его внимания. Прерывания передаются по специальным шинам IQR. Таких шин 15. Распределение номеров линий прерываний зависит от используемых системных шин. Большинство устройств не могут совместно использовать одинаковые прерывания. Если два устройства используют одинаковые номера прерываний, то это может привести к зависанию компьютера или к тому, что не произойдет никакой передачи сообщения. Нехватку номеров запросов прерываний решают в операционной системе Windows двумя способами:

- Первый способ – создание нескольких аппаратных профилей. Каждый из этих профилей можно настроить на решение определенного типа задач. Выбрав при загрузке соответствующий профиль, можно получить доступ к тому или иному устройству.
- Второй способ – использование шин последовательного доступа USB, IEEE 1394 или SCSI. Шина USB, как уже отмечалось, позволяет подключить последовательно 256 устройств, шина IEEE 1394 – 63, а SCSI – 7.

Каналы прямого доступа к памяти. Каналы прямого доступа к памяти (каналы DMA) предназначены для быстрой передачи данных к периферийному устройству. При передаче данных по каналу прямого доступа к памяти процессор не задействуется и может выполнять другие задачи. Обычный компьютер имеет 8 каналов прямого доступа, при этом каналы, назначенные для сетевых карт или контроллера жесткого диска, не могут использоваться другими устройствами, так как это может привести к потере важных данных. Некоторые устройства используют одновременно несколько каналов DMA, например, адаптер Media Vision ProAudio Spectrum 16 использует два канала DMA. Каждому каналу прямого доступа выделяется определенная область оперативной памяти.

Распределение портов ввода-вывода. Чтобы программа не перепутала данные, относящиеся к тому или иному процессу, каждому процессу назначается определенная область памяти – *порт*. В архитектуре PC порты ввода-вывода распределяются в памяти. Это позволяет микропроцессору получить к ним доступ, указав соответствующий адрес. Таким образом, каждое устройство, подключенное к порту ввода-вывода, должно знать адрес этого порта, иначе информация попадет не по назначению.

Доступ к памяти компьютера в сети Интернет. Для доступа к определенной области памяти компьютера в сети Интернет кроме адреса порта необходимо знать IP-адрес компьютера (IP адрес см. раздел 7). Совокупность номера порта и IP-адреса называют ***сокетом***. За некоторыми процессами номера портов закреплены. Так, например, порт 21 закреплен за службой удаленного доступа к файлам FTP, порт 23 – за службой управления telnet.

Адресация памяти. В памяти компьютера, в зависимости от способа адресации, различают две области (рис. 1.16): область ***непосредственной адресации***, занимающую первые 1024 кбайта с адресами от 0 до 1024 кбайт – 1, и ***расширенную память*** с адресами больше 1024 кбайт. Доступ к ячейкам этой памяти возможен при использовании специальных программ-драйверов или в защищенном режиме работы микропроцессора. Основная память в соответствии с методами обращения и адресации делится на отдельные области, имеющие общепринятые названия. Логическая структура основной памяти ПК общей емкостью 64 Мбайт представлена на рисунке 1.16.

Непосредственно адресуемая память		Расширенная память	
Стандартная память	Верхняя память	Высокая память	
0 640 кб	640 кб 1024 кб	1024 кб 1086 кб	1086 кб 64 Мб

Рис. 1.16. Логическая структура основной памяти

Область памяти до 1 Мбайта, то есть область непосредственной адресации, тоже делится на две части: *стандартную область* – 640 кбайт (64 кбайт используется для служебных программ и данных операционной системы, остальные 576 кбайт используется для программ и данных пользователя) и *верхняя память* – 384 кбайт (256 кбайт используется для области видеопамяти дисплея и слу-

жебных программ, а 128 кбайт используется для программ начальной загрузки ОС (ПЗУ) и др.) (рис.1.17).

<table>
<tr><td colspan="4">Непосредственно адресуемая память 1 Мбайт</td></tr>
<tr><td colspan="2">Стандартная память 640 кбайт</td><td colspan="2">Верхняя память 384 кбайт</td></tr>
<tr><td>64 кбайт, область системных программ и данных ОС</td><td>576 кбайт, область программ и данных пользователя</td><td>256 кбайт, область видеопамяти дисплея и служебных программ</td><td>128 кбайт, область программ начальной загрузки и ОС и др.</td></tr>
<tr><td colspan="3">ОЗУ</td><td>ПЗУ</td></tr>
</table>

Рис. 1.17. Логическая структура непосредственно адресуемой памяти

Стандартная область памяти используется различными устройствами для связи с процессором, для некоторых устройств базовые адреса памяти устанавливают с помощью переключателей или специальных программ-утилит. При выделении области памяти для сетевой карты указывается не только базовый (начальный) адрес, но и размер адресуемой памяти. При выделении адреса памяти для сетевой карты операционная система резервирует под него место. Эта область не будет использоваться процессором. Естественно, что при этом уменьшается объем системной памяти.

Большинство более поздних ISA-карт могут использовать верхние адреса памяти. Многие устройства для шин VLB, PC, а также некоторые ISA-карты могут работать с областями памяти выше 1 Мбайт. Предпочтительны карты, работающие с верхними адресами памяти, так как их использование снижает вероятность конфликтов с операционной системой. Расширенная память используется главным образом для хранения данных и некоторых программ ОС, расширенную память часто используют для организации *виртуальных* (электронных) дисков. При этом небольшой объем памяти, до 64 кбайт – высокая память, используется обычно для хранения программ и данных операционной системы.

Системный реестр

В операционной системе MS-DOS настройка операционной системы осуществляется с помощью файлов config.sys и autoexec.bat. В операционной системе Windows в этих файлах принципиальная необходимость отпала, хотя они по-прежнему могут присутствовать.

Вместо этих двух небольших файлов в ОС Windows для хранения сведений об операционной системе, установленном оборудовании и настройках используется большая база данных, называемая *системным реестром*. В системном реестре хранятся сведения не только об операционной системе, но и обо всех установленных программах и оборудовании. Все управление занесением или удалением данных из системного реестра осуществляется операционной системой автоматически. Однако возможна и ручная настройка системного реестра. Организация системного реестра напоминает структуру жесткого диска – реестр имеет несколько разделов, а его иерархическая структура ничем не отличается от структуры жесткого диска с папками и файлами. Разделы реестра называют ***ветвями***. В каждой ветви имеются разделы. Разделы, в свою очередь, могут содержать подразделы и *параметры*, в которых хранится такая информация, как числа и текст.

Реестр можно редактировать, но использовать эту возможность рекомендуется только *опытным пользователям*, вмешательство в настройку реестра неопытного пользователя может привести к фатальным последствиям – операционная система не будет загружаться.

Наличие средств мультимедиа

Операционная система Windows обеспечивает интерактивную работу с высококачественным звуком и видео при помощи специальных аппаратных и программных средств. В группу мультимедийных программ входят четыре стандартные программы: проигрыватель Windows Media, киностудия Windows Live, DVD-студия Windows, фотоальбом Windows Live. Кроме того, имеется специальный медиацентр, обладающий многими функциональными возможностями по работе с графическими объектами, звуком, музыкой, просмотру кинофильмов, телепередач (при наличии в компьютере TV-тюнера).

Средства автоматизации

Средства автоматизации позволяют автоматически запускать программы при включении компьютера, настраивать режим копирования файлов на нужное устройство, настраивать главное меню программы, автоматически рассылать документы по электронной почте, принимать и обновлять файлы с каналов Интернет, на которые оформлена подписка, осуществлять дистанционное техническое обслуживание компьютера в случае неполадок в его работе, автоматическое обслуживание операционной системы (дефрагмен-

тация жесткого диска, проверка диска, очистка жесткого диска, запуск программ по расписанию).

1.4.2. Основы работы в Windows

Объекты

Операционная система *Windows* является многооконной объектно-ориентированной программой.

> К *объектам* в *Windows* относится все, что может быть различимо средствами операционной системы.

Все объекты, с которыми она работает, представляются графически в виде значков: файлы, папки, диски и т. д. Многие значки в Windows стандартизированы, например, файлы текстовых, звуковых, графических и других документов.

Каждый объект обладает свойствами. Свойства – это параметры или характеристики объектов. Например, диск имеет такие параметры, как тип, имя, размер; характеристиками папки являются: место размещения, имя, размер, дата и время создания; свойствами файлов являются: имя, тип, размер, дата и время создания или последнего изменения, форма значка, который связан с файлом, и адрес, где этот значок хранится, атрибуты файла, имя и адрес приложения, с помощью которого он может просматриваться и редактироваться, распечатываться, воспроизводиться и т. д.

Окна

Одним из важных элементов *Windows* являются окна.

> *Окно Windows* – прямоугольная экранная область, внутри которой размещаются графическая информация и элементы управления.

В зависимости от назначения окна делятся на окна программ, окна документов, окна диалога (запроса), окна сообщений. Все они отличаются составом элементов управления, размещаемых в них.

На рисунке 1.18 представлено окно программы Проводник операционной системы *Windows XP*. В верхней части окна размещается строка заголовка *2*, в которую выводится наименование открытой папки. Если окно активно, то строка заголовка закрашивается в синий цвет. В левой части строки заголовка размещается кнопка системного меню *1*, а справа – кнопки свертывания *6*, развертывания *7* и закрытия окна *8*. Если окно развернуто, то на месте кнопки развертывания окна появится кнопка восстановления предыдущего состояния окна. Ниже строки заголовка размещается строка главного

меню *3*. При выборе пункта главного меню открывается всплывающее меню второго уровня. Меню второго уровня могут иметь подменю третьего уровня. Признаком наличия подменю у меню второго уровня является наличие стрелки ▶ в правой части строки меню. Некоторые пункты меню второго уровня выделены черным цветом, а некоторые – серым. Пункты меню, выделенные черным цветом, являются активными, а серые пункты меню – пассивными, они недоступны для управления программой. Состав активных пунктов меню второго уровня зависит от состояния программы. Меню позволяет ввести любую команду, предусмотренную для управления программой.

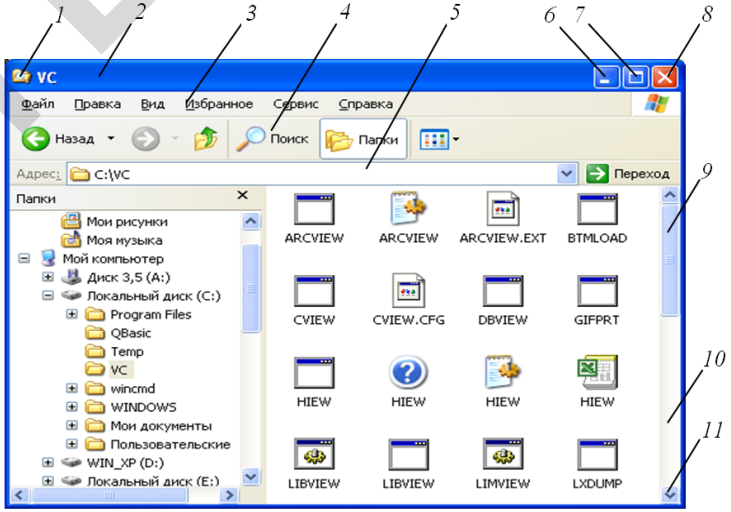


Рис. 1.18. Окно программы Проводник ОС Windows XP

Ниже строки главного меню располагается, как правило, панель инструментов *4*, на которую выводятся основные команды управления программой.

В рабочей части окна могут располагаться строки ввода, раскрывающиеся списки *5*, кнопки, линейки прокрутки *10* и другие элементы управления. Линейки прокрутки появляются в окне программы в том случае, когда информация не умещается в открытой области окна. Линейки прокрутки могут быть горизонтальными и вертикальными. На них размещаются элементы управления: кноп-

ки прокрутки *11*, ползунок *9*. Щелчок мышью по кнопке прокрутки перемещает экран на строку вверх (вниз) или на колонку вправо (влево). Для быстрого перемещения по окну служит ползунок, который иногда называют «лифт». Зацепив мышью за ползунок и перемещая его, можно быстро перейти к нужной области просмотра. Управлять просмотром можно также щелчком мыши по линейке прокрутки ниже или выше ползунка. При каждом щелчке окно просмотра перемещается на фиксированное число строк (столбцов).

На рисунке 1.19 приведены основные элементы управления, используемые в окнах *Windows*. К таким элементам относятся:

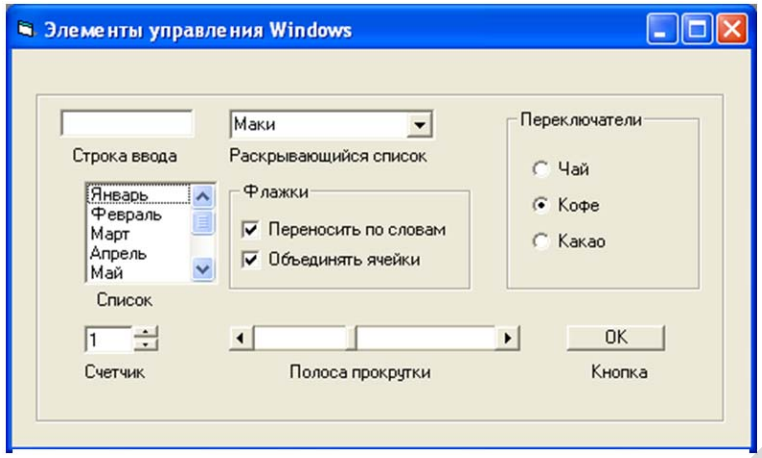


Рис. 1.19. Элементы управления *Windows*

строка ввода – горизонтальная полоса белого цвета, служит для ввода буквенно-цифровой информации. Для активизации строки ввода необходимо щелкнуть по ней мышью. При этом в строке ввода появляется курсор – вертикальная мигающая черта;

списки – перечень элементов. Если весь список не умещается в окне, то справа появляется линейка прокрутки. Для выбора элемента списка необходимо щелкнуть по нему мышью;

раскрывающийся список – список, который для экономии места в окне свернут. На экране присутствует лишь заголовок списка, содержащий строку ввода и кнопку раскрытия окна. Для выбора нужного элемента списка его необходимо открыть щелчком мыши по кнопке раскрытия окна, найти в списке нужный элемент и щелкнуть

по нему мышью. Выбранный элемент списка записывается в строку ввода заголовка, и список сворачивается;

флажок – элемент управления в виде маленького прямоугольника. Флажок можно установить или снять. Если флажок установлен, то внутри прямоугольника появляется отметка в виде крестика или галочки. Снятие и установка флажка осуществляются щелчком мыши. Если флажок установлен, то действие, указанное в текстовом сообщении справа от флажка, выполняется. В окне диалога может быть несколько флажков, объединенных в группу. Все флажки независимы друг от друга. Поэтому установка или снятие какого-либо флажка не влияет на состояние других флажков;

переключатель – переключатель представлен на экране кругом с пояснительной надписью справа. Переключатель может быть включен или выключен. Если переключатель включен, то он отмечается черной точкой внутри круга. Все переключатели объединяются в группу по умолчанию. При этом только один переключатель может быть включен, а остальные переключатели автоматически выключаются;

счетчик – предназначен для ввода чисел. Ввод числа можно либо осуществлять вручную в строке ввода, либо установить с помощью кнопок, расположенных справа (слева, сверху или снизу) от строки ввода.

Управление окнами

Окна можно перемещать, а также можно изменять их размеры. Для перемещения окна зацепите его мышью за заголовок и переместите в требуемое положение. Для изменения размеров окон зацепите мышью за одну из сторон (курсор принимает вид двунаправленной стрелки) и протяните мышь в нужном направлении. Если зацепить мышью за угол окна, то можно изменять одновременно оба размера.

Контекстное меню

Одним из эффективных способов управления программами является использование контекстного меню. Для вызова контекстного меню щелкните правой кнопкой мыши по объекту. В контекстном меню отображаются все задачи, которые можно выполнить с данным объектом, а также с его помощью можно узнать свойства объекта.

Технология работы с мышью

Основным средством управления при работе в Windows является графический манипулятор мышь. Конечно, можно использовать для управления и клавиатуру, но это не эффективно. Поэтому на-

чинающим пользователям полезно ознакомиться с основными приемами работы с мышью.

Мышь имеет две или три кнопки, на последних моделях на месте средней кнопки размещают ролик. Для управления используются только две кнопки: левая и правая. По умолчанию всегда подразумевается левая кнопка. Если для управления применяется правая кнопка, то это специально оговаривается. В Windows имеется возможность поменять настройку кнопок мыши для левшей. Ролик применяется для «прокрутки» документа.

При загруженном драйвере мыши на экране монитора появляется ее указатель. Форма указателя может изменяться в зависимости от того, какой объект выделен: заголовок формы, граница объекта, разделительная линия и тому подобное.

При работе с мышью используется определенная технология, или, иначе, приемы работы. Этой технологии соответствуют определенные понятия.

Зависание мыши – установите указатель мыши на объект и задержите его на некоторое время.

Щелкнуть мышью – кратковременно нажмите и отпустите кнопку мыши.

Дважды щелкнуть мышью – два раза кратковременно нажмите и отпустите кнопку мыши. Интервал времени между нажатиями кнопки, необходимый для того, чтобы программа распознала нажатия кнопки как двойной щелчок, может настраиваться средствами Windows.

Выделить объект – для этого щелкните по нему мышью. Выделенный объект закрашивается, обычно синим цветом.

Зацепить объект – установите указатель мыши на объект, нажмите и удерживайте клавишу мыши.

Протягивание мышью – зацепите объект так, чтобы курсор изменил форму, например, на двунаправленную стрелку, и протяните указатель мыши в нужном направлении. Этот прием применяется при изменении размеров объекта.

Перетащить мышью – зацепите объект, убедитесь, что возле указателя появился маленький прямоугольник с крестиком или без него и перенесите объект. Перетаскивание объектов левой или правой клавишами мыши используют для перемещения или копирования объектов.

Перемещение объектов мышью. Для перемещения окон зацепите объект за заголовок и перенесите в другое место, отпустите

клавишу мыши. Для перемещения других объектов зацепите их мышью и переместите в требуемое положение.

Изменение размеров объектов. У некоторых объектов, имеющих рамку, можно изменять размеры. Для изменения размера объекта зацепите его за границу так, чтобы курсор изменил форму на двунаправленную стрелку, например, ←||→, и переместите границу в требуемом направлении.

Выделение группы объектов. Для выделения непрерывной группы объектов мышью выделите первый объект, нажмите клавишу Shift и, не отпуская ее, щелкните мышью по последнему объекту в группе. Отпустите клавишу мыши, отпустите клавишу Shift.

Выделение группы разрозненных объектов или групп объектов: выделите первый объект (или группу объектов), нажмите клавишу Ctrl и, не отпуская ее, щелкните мышью по выделяемым объектам (или выделите последующие группы объектов).

Перемещение и копирование объекта с помощью мыши. Выделите копируемый объект, зацепите его мышью и перетащите в требуемое положение. Во время перетаскивания объекта к указателю мыши прикрепляется маленький прямоугольник. При перетаскивании объектов при нажатой клавише ***Ctrl*** осуществляется копирование объектов. При перетаскивании объекта с нажатой клавишей Ctrl к указателю мыши прицепляется маленький прямоугольник с крестиком. *Перетаскивание файлов правой* клавишей отличается от перетаскивания левой клавишей тем, что при ее отпускании в новом положении объекта появляется меню, которое предлагает выбрать операцию копирования или перемещения.

Рабочий стол

Запуск операционной системы

После включения питания программа проверяет исправность оперативной памяти и наличие подключенных внешних устройств, и затем осуществляется загрузка операционной системы в оперативную память компьютера. Если на компьютере установлено несколько операционных систем, то на экран выводится список установленных операционных систем и пользователю предлагается выбрать нужную.

Если выключение компьютера было выполнено некорректно, то проводится проверка состояния дисков. Эти операции могут занять значительное время.

После загрузки операционной системы на экране появляется рабочий стол программы Windows (рис. 1.20).

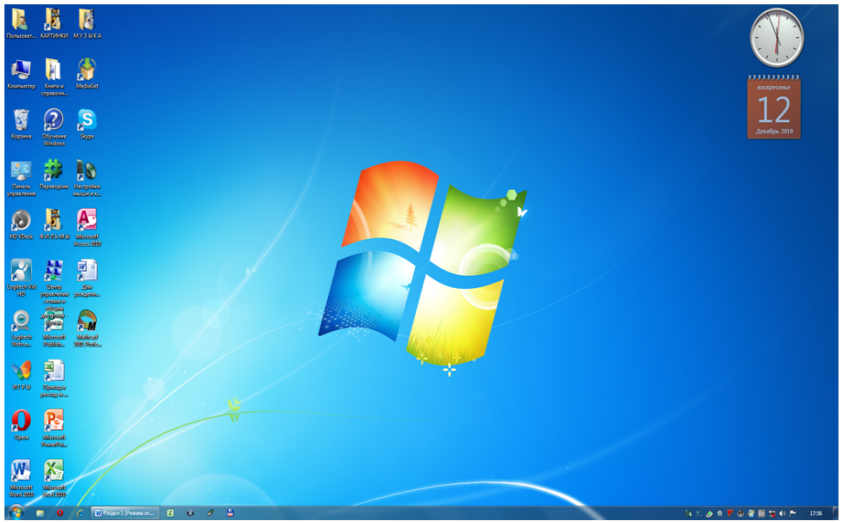


Рис. 1.20. Рабочий стол операционной системы Windows 7

На рабочем столе Windows размещаются объекты: значки и ярлычки программ и папок, панель задач и *гаджеты* – мини-приложения Windows, выполняющие определенные полезные функции, например, часы, календарь.

Каждый объект в операционной системе Windows обозначается специальным значком. Кроме того, для каждого объекта могут быть созданы ярлычки.

Значки программ связаны непосредственно с файлом, программой, установленной на компьютере. Для запуска программы необходимо дважды щелкнуть мышкой по значку программы. При удалении значка удаляется и соответствующий файл, поэтому значки программ можно перемещать, переименовывать, но нельзя удалять.

Ярлычки программ, в отличие от значков, имеют стрелку в левом нижнем углу значка. Они являются представителем программы и содержат сведения о ее местонахождении. Ярлычков у программы может быть несколько, они могут быть размещены в разных папках. Ярлычки, так же, как и значки программ, используются для запуска программ, но, в отличие от значков, удаление ярлычков никак не сказывается на программе.

Среди значков отметим, прежде всего, «Мой компьютер», «Корзину» и «Центр управления». «Мой компьютер» – это программа-

навигатор, предназначенная для поиска нужных объектов на компьютере. «Корзина» – это программа, предназначенная для хранения удаленных файлов и папок. При необходимости удаленный файл может быть восстановлен. Корзину периодически следует очищать для удаления накопившегося «мусора». «Центр управления» обеспечивает доступ к инструментам настройки параметров компьютера и рабочей среды.

Панель задач

Панель задач обычно размещается внизу рабочего стола, но может быть установлена пользователем в любое положение. В левой части панели задач расположена круглая кнопка *ПУСК*, которую закамуфлировали под логотип Windows, справа от нее расположена панель быстрого запуска, куда помещаются значки наиболее часто используемых программ и значки запущенных программ (рис. 1.21). В правой части панели задач размещается *панель уведомлений* (рис. 1.22). Назначение значка, размещенного на панели уведомления, можно узнать, если подвести к нему указатель мыши. На панель уведомления выводятся значки программ, которые автоматически запускаются при включении компьютера. Если часть значков скрыта, то слева на панели уведомлений присутствует кнопка *Отображать скрытые значки*. Если щелкнуть по ней, то откроется панель инструментов, позволяющая выполнить настройку панели уведомления по своему усмотрению.

На панель задач могут выводиться также и другие панели управления. Возможности управления представлением информации можно узнать, открыв контекстное меню панели задач.

Кнопка ПУСК открывает доступ к командам *Главного меню* (рис. 1.21). Чтобы изучить возможности Windows, достаточно просмотреть внимательно пункты меню. Главное меню состоит из двух колонок. В левой колонке размещаются значки некоторых служебных программ и наиболее часто используемых программ. Справа от некоторых пунктов меню левой колонки расположены черные стрелки, которые открывают список последних программ, с которыми работал пользователь. В правой колонке размещен список ряда полезных папок.

В нижней части левой колонки помещен раскрывающийся список программ, установленных на компьютере, – *Все Программы,* который позволяет найти и запустить любую программу. Запуск программ осуществляется двойным щелчком мыши по соответствующему пункту меню. Чтобы поместить значок программы на панель

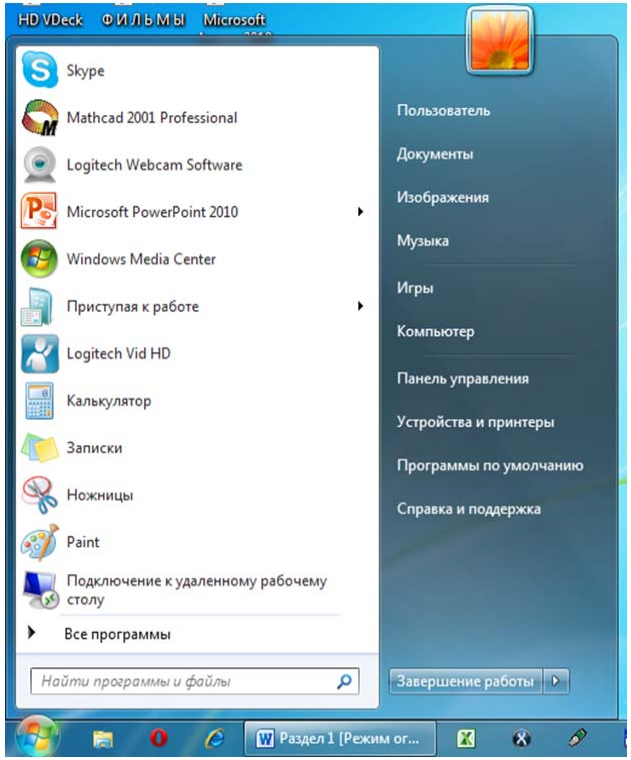


Рис. 1.21. Левая часть панели задач: кнопка ПУСК и панель быстрого запуска

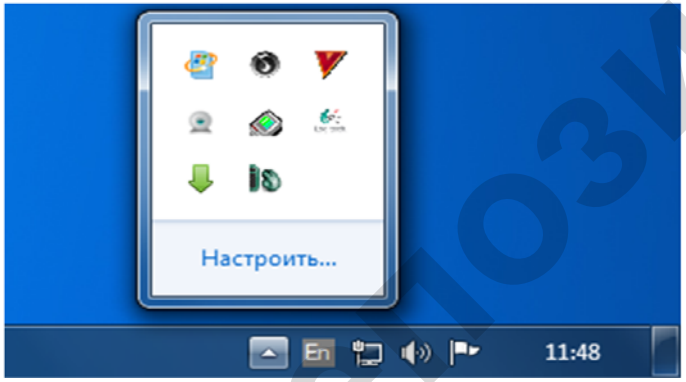


Рис. 1.22. Правая часть панели задач (панель уведомлений)

быстрого запуска, найдите ее в списке *Все программы*, откройте контекстное меню этой программы и щелкните команду «Закрепить на панели задач».

Завершение работы с компьютером

Для завершения работы с компьютером достаточно нажать кнопку выключения питания на лицевой панели компьютера. Windows самостоятельно закроет все запущенные программы, сохранит все важные файлы и завершит работу.

Однако правильнее не полагаться на умную программу, а самому закрыть все открытые программы и завершить работу, воспользовавшись командой ***Пуск, Завершение работы***.

При завершении работы с помощью данной команды имеется возможность выбрать один из предложенных вариантов выключения компьютера: спящий режим, гибернация, блокировка, смена пользователя, перезагрузка, выход из системы.

Спящий режим. Компьютер работает в режиме экономии энергопотребления. Сохраняются данные о всех открытых документах и запущенных программах. В спящий режим программа переходит автоматически при длительных перерывах в работе.

Гибернация. Компьютер запоминает состояние оперативной памяти на жестком диске и выключается. При последующем включении компьютер восстанавливает состояние оперативной памяти.

Блокировка. В этом режиме все программы продолжают работать, но рабочий стол прячется под заставкой, которая не позволяет выполнять какие-либо операции. Для продолжения работы необходимо ввести пароль.

Настройка параметров спящего режима и установка пароля при выходе из спящего режима устанавливаются на ***Панели управления*** в разделе ***Система и безопасность***, ***электропитание***.

1.4.3. Средства навигации в Windows

Одной из главных задач Windows является обеспечение работы файловой системы: создание папок, поиск файлов и папок, переименование, копирование, пересылка файлов и папок.

Для выполнения этих функций в Windows имеются средства навигации: Internet Explorer, Проводник, Мой компьютер.

Internet Explorer – программа, предназначенная для навигации во всемирной информационной сети, в том числе и по компьютеру. Для навигации только по компьютеру предназначены программы

Проводник и Мой компьютер. Какой из этих программ отдать предпочтение – дело вкуса и привычки.

Программа Проводник

Вызов программы

Вызов программы можно произвести несколькими способами. Самый простой – через контекстное меню кнопки ПУСК. А лучше всего поместить значок программы Проводник на рабочий стол или на панель быстрого запуска (по умолчанию программа Проводник выведена на панель быстрого запуска).

Эту операцию выполняют следующим образом:

- Найдите программу в главном меню: ***Пуск, Все Программы, Стандартные***.
- Вызовите контекстное меню программы Проводник и щелкните мышью по пункту меню ***Отправить на панель задач***.

Окно программы Проводник

Окно программы Проводник является типичным окном Windows (рис. 1.23). Оно во многом похоже на окно программы Проводник операционной системы Windows XP (рис. 1.18). В верхней части окна выводится строка заголовка с кнопками управления свертыванием в значок, свертыванием/развертыванием и закрытием окна. Ниже строки заголовка располагаются строка адреса и окно поиска файлов. Здесь же расположены кнопки *Вперед* и *Назад* для перемещения по уже пройденному маршруту. Расположение команд меню и панелей инструментов в программе Проводник ОС Windows 7 несколько отличается по сравнению с прежними версиями Windows (см. рис. 1.18). Ниже панели адресов располагается панель меню и панель инструментов, где размещены раскрывающиеся списки команд и кнопок управления, состав которых зависит от выбранной папки.

На **панели Меню** размещены команды ***Файл, Правка, Вид, Сервис*** и ***Справка***.

В меню ***Файл*** наше внимание может привлечь команда ***Создать***, которая позволяет создавать папки, ярлыки и файлы разных типов.

Меню ***Правка*** обеспечивает управление файлами: выделение, копирование, вставка, удаление, переименование и др.

Меню ***Вид*** отвечает за внешний вид программы Проводник, в том числе позволяет управлять представлением файлов (огромные, крупные, обычные, мелкие, список, таблица, плитка, содержимое) и сортировать файлы и папки (по возрастанию или убыванию имени, типа, размера, даты создания).

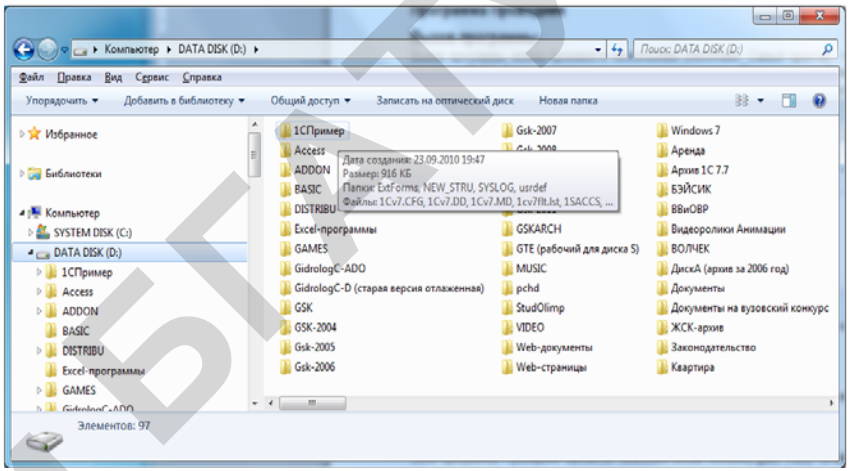


Рис. 1.23. Программа Проводник ОС *Windows 7*

Меню ***Сервис*** позволяет управлять сетевыми дисками и изменять параметры папок.

Меню ***Справка*** позволяет получить полноценную информацию о работе с программой Проводник (команда ***Просмотреть справку***).

На панели инструментов размещены:

список команд ***Упорядочить*** – обеспечивает выполнение операций копирования, вставки, удаления, управления представлением кнопок панели инструментов. Здесь следует обратить внимание на команду ***Представление***, которая позволяет управлять выводом на экран строки меню, области предварительного просмотра, области сведений и области переходов (дерева папок);

список ***Добавить в библиотеку*** – позволяет добавить файлы в одну из стандартных библиотек: Документы, Музыка, Видео, Изображения или создать новую библиотеку;

список ***Общий доступ*** – позволяет предоставить доступ к ресурсам компьютера другим пользователям сети;

меню ***Запись на оптический диск*** – позволяет производить запись информации на оптический диск. В Windows 7 отпала необходимость в использовании дополнительных программ для записи информации на оптические диски. Эти функции выполняет встроенная программа, вызываемая данной командой;

меню ***Новая папка*** – позволяет создать новую папку на текущем диске;

кнопка панели инструментов ***Изменить представление*** с расположенной рядом кнопкой раскрывающегося списка позволяет изменить представление папок и файлов в правом окне программы Проводник. Эта кнопка дублирует команды представления файлов меню Вид;

кнопка ***Показать область предварительного просмотра*** позволяет просматривать содержимое текстовых файлов;

кнопка ***Справка*** аналогична команде Справка меню.

Ниже панели инструментов располагаются панели программы Проводник.

Панели Проводника

Программа Проводник имеет две панели: левую и правую. В левую панель (область переходов) выводится дерево папок, а в правую – содержание активного диска или папки.

Левая панель содержит дерево папок. Корневой папкой считается Рабочий стол. В качестве объектов второго уровня выступают папки «Избранное», «Библиотеки», «Компьютер» и «Сеть».

Чтобы открыть папку, необходимо щелкнуть по ней мышью. Содержание открытой папки отображается в правой панели. Чтобы закрыть папку, необходимо открыть другую папку, или подняться на уровень вверх соответствующей кнопкой на панели инструментов, или воспользоваться кнопкой Назад.

Развертывание структуры папок осуществляется двойным щелчком мыши по имени папки или щелчком по кнопке, расположенной слева от имени папки.

Чтобы скрыть структуру папки, необходимо щелкнуть по черной кнопке слева от имени папки.

Правая панель

Правая панель содержит папки и файлы, имеющиеся в родительской папке. Выделение папок и файлов в правой панели осуществляется, как обычно, щелчком мыши по объекту. Выделенный объект отмечается синим цветом. В некоторых случаях, например, при выполнении операций копирования, перемещения, удаления, необходимо выделить группу файлов. Для этого есть несколько приемов:

- ***выделение непрерывной группы:***

а) выделение протягиванием мыши;

установить указатель мыши выше и левее первого файла группы;

нажать клавишу мыши и протянуть указатель к правому нижнему углу группы. Выделяемая область помечается сплошной линией и синим цветом;

б) выделение с использованием клавиши Shift:

выделить первый файл группы;

нажать и удерживать клавишу Shift;

щелкнуть мышью по последнему файлу группы;

отпустить клавишу Shift;

- ***выделение произвольной группы файлов:***

а) использование клавиши Ctrl:

выделить первый файл группы;

нажать и удерживать клавишу Ctrl;

выделить щелчком мыши файлы, включаемые в группу;

отпустить клавишу Ctrl;

б) использование клавиш Ctrl и Shift:

выделить первый файл группы;

нажать и удерживать клавишу Ctrl;

выделить щелчком мыши файлы, включаемые в группу;

при необходимости выделить непрерывную подгруппу файлов в пределах выделяемой группы, продолжая удерживать клавишу Ctrl, необходимо нажать и удерживать на время выделения подгруппы клавишу Shift;

отпустить клавиши Ctrl и Shift.

Управление представлением файлов в панели

Для управления представлением файлов в панели используется, как упоминалось ранее, кнопка панели инструментов ***Изменить представление*** с расположенной рядом кнопкой раскрывающегося списка или контекстное меню правой панели программы Проводник.

При работе с файлами предпочтительным является режим ***Таблица***, в котором на экран выводятся основные свойства файла: Имя, Дата изменения, Тип, Размер. Кроме того, в этом режиме удобно выполнять сортировку файлов и папок щелчком мыши по соответствующему заголовку таблицы. Например, для сортировки файлов по наименованию в порядке возрастания алфавита достаточно щелкнуть мышью по заголовку графы Имя. Повторный щелчок мышью по данному заголовку приводит к сортировке содержимого панели в обратном порядке. При большом числе файлов в папке целесообразно использовать для просмотра *список*. В этом случае на экран выводятся только значки, определяющие тип файла, и их имена.

Полные сведения о каждой папке и файле можно получить с помощью команды *Свойства* контекстного меню папки или файла.

Сортировка файлов и папок

Сортировку файлов и папок в панели можно выполнить с помощью команды ***Вид, Сортировать***. Сортировку можно выполнить по имени файла, типу, размеру или дате изменения. Сортировка выполняется по возрастанию кода символов в кодовой таблице компьютера: цифры 0...9, буквы латинского алфавита A...Z, a...z, буквы русского алфавита А...Я, а...я. Команда *автоматически* используется для выравнивания значков при представлении файлов в виде крупных или мелких значков.

Создание папок

Для создания новой папки щелкните мышкой по кнопке ***Новая папка*** на панели инструментов или выберите команду ***Создать, Папку*** в меню ***Файл*** или в ***Контекстном меню***. В панели появится папка с именем ***Новая папка***. Присвойте папке другое, оригинальное имя.

Копирование и перенос файлов и папок

Копирование – двуместная операция. При копировании всегда присутствуют источник информации и приемник информации. Алгоритм копирования зависит от используемого способа копирования и адреса источника и приемника.

Копирование с помощью меню (Главного или Контекстного) или панели инструментов:

выделить файл или группу файлов;

выбрать команду ***Правка, Копировать***. Информация помещается в буфер;

перейти в папку-место назначения;

ввести команду ***Правка, Вставка***.

Копирование левой клавишей мыши:

вывести в правую панель папку-источник данных;

вывести в левую панель диск и папку-назначение, раскрывая папки щелчком мыши по кнопкам +;

выделить файл в правой панели;

зацепить файл и перенести его в папку-назначение. Во время переноса к указателю мыши прикрепляется маленький прямоугольник.

Копирование правой клавишей мыши

При копировании с помощью правой клавиши мыши после отпускания клавиши мыши в месте назначения на экран выводится запрос на подтверждение выполняемой операции: *копировать*, *перемещать*, *создать ярлык*.

При копировании файлов с помощью мыши из одной папки в другую в пределах одного диска исходный файл остается на прежнем месте, а имя файла переносится в указанную папку. Из исходной папки имя файла удаляется, при этом на экран выводится предупреждение: *Переместить*.

Если хотите запускать существующий файл из другой папки, создайте ярлык для данного файла и поместите его в нужную папку.

Перенос файлов

Перенос файлов выполняется так же, как и копирование, но после выполнения копирования *исходный файл удаляется* с диска.

Переименование файлов и папок

Для переименования файла или папки необходимо:

выделить файл или папку;

ввести команду ***Правка, Переименовать***;

записать в строке ввода новое имя папки или файла.

Поиск файлов

Для поиска файлов в программе Проводник введите имя файла в строке ***Поиск данных***. Если имя файла неизвестно, но известно его расширение, укажите маску. Например, для поиска всех файлов, подготовленных в текстовом процессоре Microsoft Word, надо ввести маску *.docx или *.doc. Для сужения области поиска применяется фильтр по дате создания или последнего изменения файла, а также типа файла (рис. 1.24, 1.25). Для ввода фильтра щелкните мышью по полю ввода и выберите требуемый режим. Поиск файлов ведется в активной папке. Для изменения области поиска следует выбрать другой диск или папку.

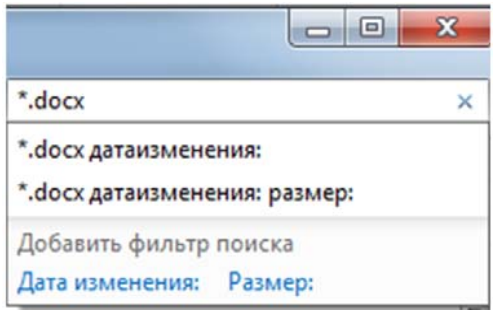


Рис. 1.24. Ввод маски для поиска

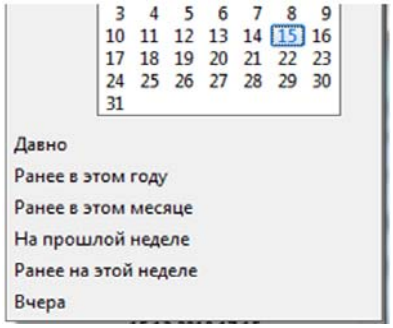


Рис. 1.25. Изменение даты поиска

1.4.4. Настройка рабочего стола

Windows имеет средства для выполнения различных настроек, позволяющих учитывать индивидуальные особенности пользователя, создавать удобную и приятную пользовательскую среду.

Рассмотрим некоторые, наиболее часто используемые настройки.

Имеется несколько путей доступа к настройкам операционной системы Windows 7: контекстное меню рабочего стола, кнопки **Приступая к работе** и **Панель управления** в меню кнопки **Пуск**, кнопка **Центр управления**, размещенная на рабочем столе. Меню команды **Приступая к работе** приведено на рисунке 1.26, а окно диалога **Панель управления** – на рисунке 1.27.

Все команды на панели управления объединены в группы по назначению.

Система и безопасность. Эта группа содержит команды для настройки Сети и Интернета, управления оборудованием и звуком, установкой и удалением программ, созданием учетных записей и семейной безопасностью, оформлением рабочего стола и персонализацией, настройкой часов, языков и региона, специальными возможностями.

Сеть и Интернет. Обеспечивает подключение к Интернету и локальной сети, если она имеется, просмотр состояния сети и задач, выбор домашней группы и общего доступа к данным.

Оборудование и звук. Содержит команды для настройки принтеров, мыши, автозапуска программ, изменения параметров энергосбережения, управления звуком, экраном.

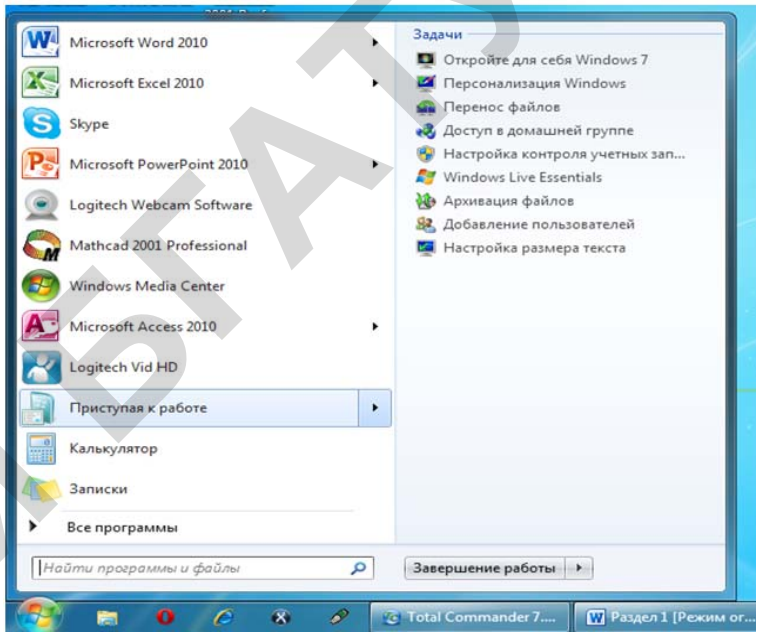


Рис. 1.26. Меню команды Приступая к работе

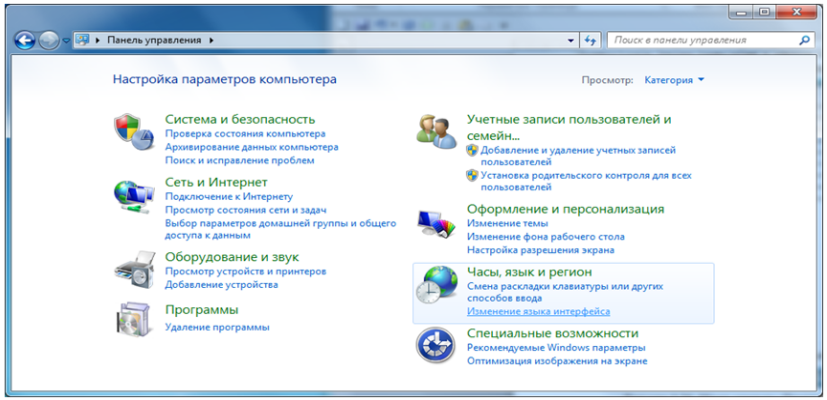


Рис. 1.27. Панель управления

Программы. Эта группа команд обеспечивает управление установкой и удалением программ из компьютера. Удалять программы

из компьютера можно только с использованием данной группы команд. В этом случае программа удаляется полностью, в том числе удаляются все ссылки на эту программу и из реестра. Здесь же расположены и программы управления гаджетами рабочего стола.

Учетные записи пользователей. Обеспечивает настройку многопользовательского режима работы Windows и управление пользовательскими конфигурациями.

Оформление и персонализация. Здесь возможно производить настройку параметров рабочего стола и его элементов.

Часы, язык и регион. Позволяет производить настройку часов, часовых поясов, форматы представления даты, времени, чисел, устанавливать раскладку клавиатуры, добавлять и удалять языки.

Специальные возможности – настройка системы для пользователей с ограниченными двигательными возможностями.

Настройка интерфейса: оформление и персонализация

Windows 7 легко позволяет настроить рабочий стол по своему вкусу, что и называется *персонализацией*. Для этой цели имеются в запасе экранные темы, оформление и цветовые схемы. Для установки новой темы выполните следующее:

- Введите команду ***Пуск, Приступая к работе, Персонализация Windows***: откроется окно ***Изменение изображения и звука на компьютере***.
- Выберите понравившуюся тему и щелкните по ней мышью (рис. 1.28).

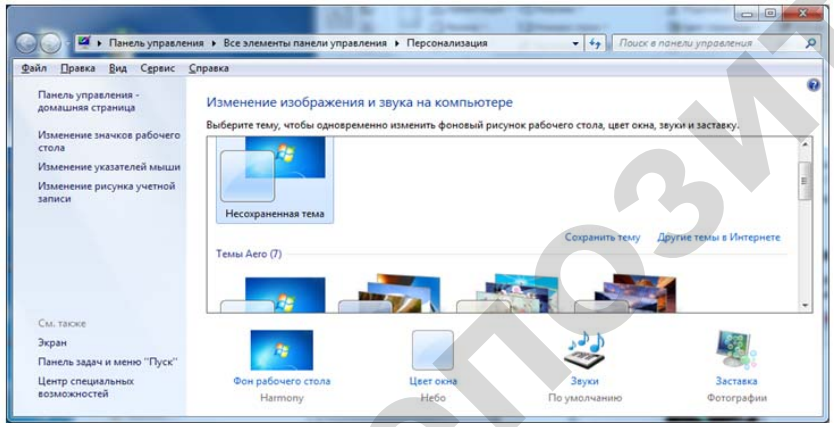


Рис. 1.28. Выбор темы рабочего стола

Новые темы можно скачать из сети Интернет по ссылке. Здесь же можно выбрать и тему ***заставки*** (рис. 1.29).

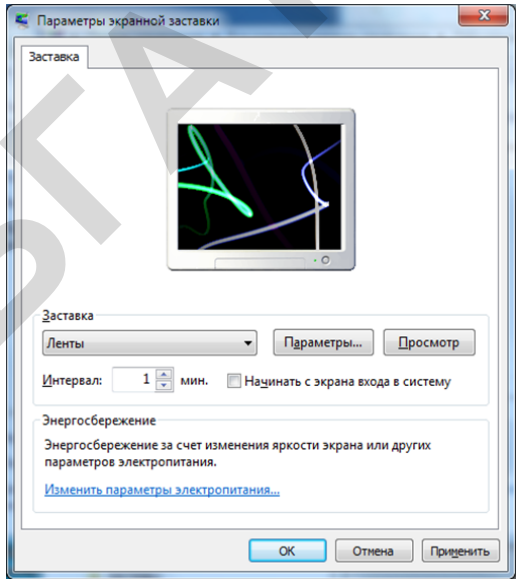


Рис. 1.29. Настройка экранной заставки

Заставка появляется, когда пользователь длительное время не вводит никакие команды. Интервал времени, через который появляется заставка, можно устанавливать пользователю. Здесь же в списке можно выбрать тему и при необходимости настроить параметры заставки.

Флажок ***Начинать с экрана входа в систему*** позволяет защитить ваш компьютер от вмешательства посторонних. Однако этот эффект сработает только в том случае, если Вы при установке Windows выбрали режим входа по паролю.

Пароль можно установить в свойствах Учетной записи: ***Панель управления, Учетные записи пользователей, Изменение пароля учетной записи***.

Настройка разрешения экрана

При работе с графическими объектами может понадобиться изменить разрешение экрана. На мониторах с электронно-лучевыми трубками можно устанавливать различные разрешения из стан-

дартного набора. Возможности мониторов на жидкокристаллических элементах по изменению разрешения экрана ограничены, и качество изображения значительно ухудшается. Поэтому на ЖК-мониторах рекомендуется работать при наивысшем разрешении экрана. Самый простой доступ к команде настройки разрешения экрана – через контекстное меню рабочего стола. Вызовите ***Контекстное меню*** рабочего стола, щелкните по команде ***Разрешение экрана***, в открывающемся списке ***Разрешение*** установите ползунок на нужный режим и щелкните по кнопке ОК.

Настройка языка ввода

Для настройки языка ввода, добавления языка ввода рекомендуется воспользоваться Панелью управления. Чтобы настроить или добавить язык ввода, введите команду ***Пуск, Панель управления, Языки и региональные стандарты*** (или *Центр управления, Все элементы управления, Языки и региональные стандарты*), на вкладке ***Языки и клавиатура*** щелкните кнопку ***Изменить клавиатуру*** (рис. 1.30).

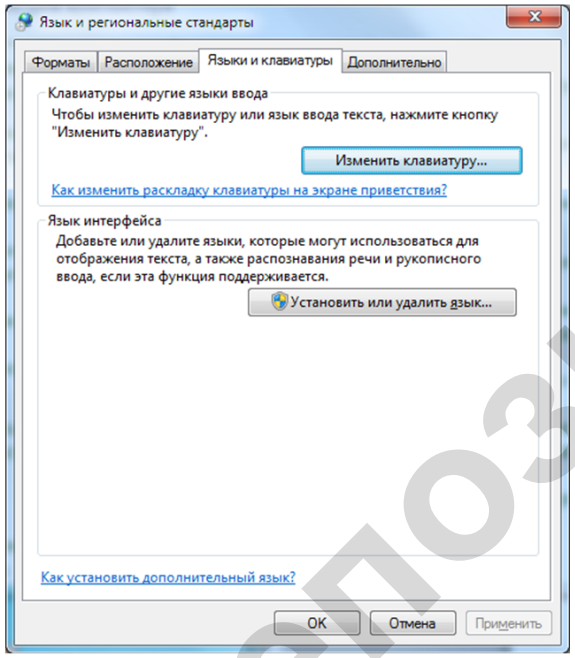


Рис. 1.30. Добавление языка ввода

В окне диалога ***Языки и службы текстового ввода*** выберите вкладку ***Общие***. В группе ***Язык ввода по умолчанию*** установите требуемый вам язык, например, Белорусский. Если требуемого языка в списке нет, щелкните кнопку ***Добавить*** в группе ***Установленные службы*** (рис. 1.31). Настройка комбинации клавиш для переключения языка ввода осуществляется в окне диалога ***Языки и службы текстового ввода*** (рис. 1.32) на вкладке ***Переключение клавиатуры***.

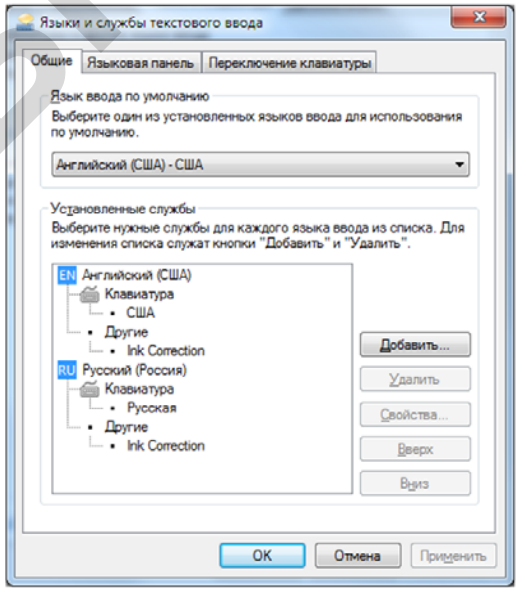


Рис. 1.31. Добавление языка ввода

При необходимости изменить сочетание клавиш откройте вкладку ***Переключение клавиатуры*** и щелкните по кнопке ***Сменить сочетание клавиш***. Активизируйте нужный переключатель сочетания клавиш щелчком мыши.

Настройка параметров мыши

Параметры мыши редко требуется настраивать. Чаще всего необходимость в этом возникает, когда надо сменить назначение левой и правой кнопок мыши, например, для левшей.

Кроме того, может понадобиться настроить под себя скорость двойного щелчка мыши.

Для настройки мыши откройте ***Панель управления*** и выберите ***Мышь***. В окне диалога ***Свойства мыши*** выберите вкладку ***Кнопки мыши.*** Для смены назначения левой и правой кнопок мыши активизируйте щелчком мыши флажок ***Обменять назначение кнопок*** в группе ***Конфигурация кнопок***. Для изменения скорости двойного нажатия передвиньте ***ползунковый регулятор*** в группе ***Скорость выполнения двойного щелчка*** в нужную сторону. Проверьте правильность настройки, щелкнув дважды по значку папки справа от регулятора. Если скорость нажатия выбрана правильно, папка откроется. При повторном двойном щелчке по значку папки папка закроется.

Настройка принтера

Настройка принтера может понадобиться в случае, если к компьютеру подключено несколько печатающих устройств, а также при необходимость очистить очередь печати.

Для настройки принтера введите команду ***Пуск, Устройства и принтеры*** – откроется одноименное окно диалога (рис. 1.32).

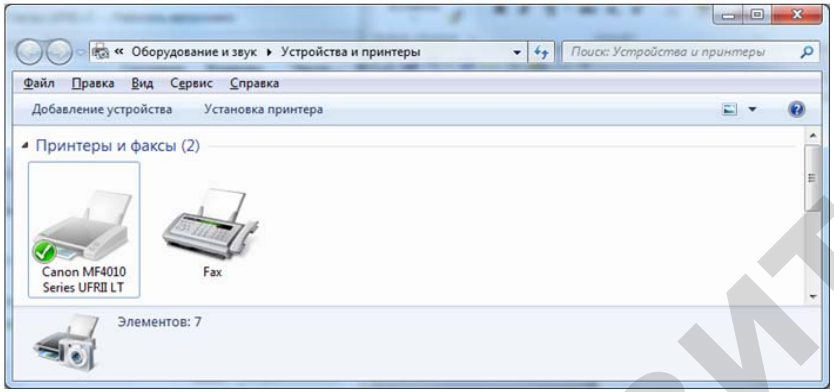


Рис. 1.32. Окно диалога Устройства и принтеры

Выберите нужный тип принтера из числа подключенных к компьютеру двойным щелчком мыши – открывается окно диалога управления печатью принтера (рис. 1.33). В этом окне приводятся сведения о документах, поставленных в очередь на печать, и ходе печати: наименование документа, состояние (ожидание очереди или печатается), владелец (логическое имя пользователя), число страниц печатаемого документа, размер в байтах и время постановки

в очередь. Документы из очереди можно удалить. Для этого выделите нужный документ и нажмите клавишу ***Delete*** на клавиатуре.

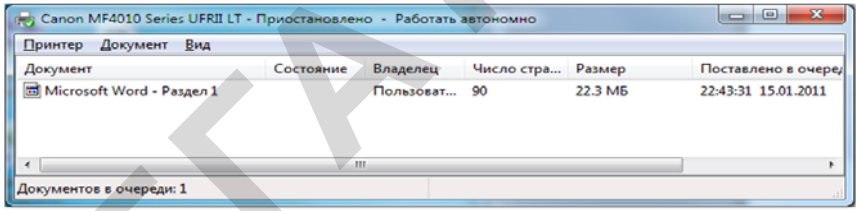


Рис. 1.33. Окно управления печатью принтера

Пункт меню ***Принтер*** (рис. 1.34) позволяет управлять печатью.

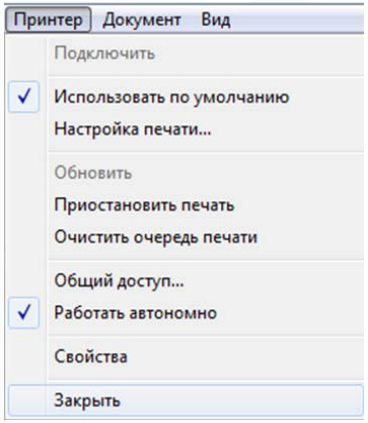


Рис. 1.34. Меню Принтер

Чтобы выбранный принтер был активным по умолчанию, щелкните по соответствующему пункту меню. Активный принтер отмечается в списке принтеров галочкой (рис. 1.32). Для приостановки печати следует щелкнуть по команде ***Приостановить печать*** в меню Принтер. Для продолжения печати повторно щелкните мышью по команде ***Приостановить печать***.

Команда ***Свойства*** позволяет производить настройку свойств принтера и режимов печати. Однако свойства документа целесообразно настраивать до отправки документа на печать.

1.4.5. Сервисная оболочка Total Commander

Одной из популярных программ для работы с файловой системой является сервисная оболочка *Total Commander* (предыдущие версии этой программы *Windows Commander*, *Far Manager* и др.). Первоначально она задумывалась, очевидно, для облегчения работы с программами, разработанными для операционной системы *MS-DOS*. Другой не менее важной причиной было то, что многие пользователи привыкли работать с сервисными оболочками операционной системы *MS-DOS* и хотели видеть похожий интерфейс и в среде *Windows*. А с течением времени она настолько прижилась в среде *Windows*, что многие пользователи и не мыслят работать в другой среде. К достоинствам *Total Commander* относится, во-первых, двухоконный интерфейс, что весьма удобно при копировании или перемещении файлов.

Во-вторых, она позволяет легко работать с архивными файлами, чего не позволяла ранее делать программа Проводник. В архивы можно входить, как в каталоги. Для управления в среде *Total Commander* одинаково удобно пользоваться как мышью, так и клавиатурой.

Рабочее окно программы *Total Commander* версии 7.55a приведено на рисунке 1.35. Окно программы оформлено в соответствии с принципами Windows: строка заголовка, главное меню, панель инструментов. Ниже панели инструментов расположены два открывающихся списка для выбора диска, выводимого в соответствующую панель, рядом со списком выведена метка диска, его размер и размер свободного пространства на диске. Под списками расположены две панели: левая и правая. Ниже панелей имеется строка ввода команд и далее строка помощи, в которую выведено назначение функциональных клавиш.

Total Commander версии 7.55a имеет три панели инструментов: основную, системную и пользовательскую. Вызов соответствующей панели осуществляется кнопками (синяя, оранжевая и красная), расположенными слева от панелей инструментов.

Главное меню

Главное меню постоянно присутствует на экране.

Меню Файл содержит команды для работы с файлами: запуск файлов, изменение атрибутов файла, упаковка, распаковка и проверка архива, печать файлов, разбивка файлов и их сборка, кодирование и декодирование файлов, проверка контрольной суммы, работа с контекстным меню.

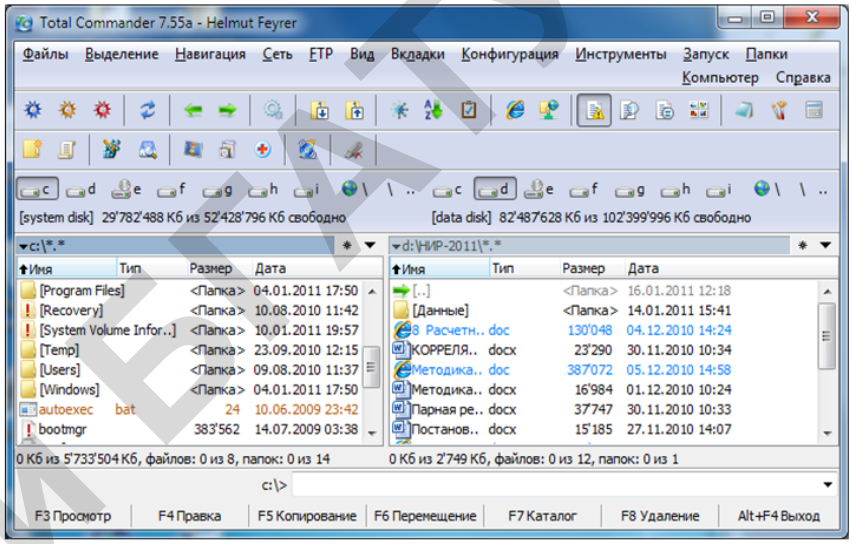


Рис. 1.35. Сервисная оболочка *Total commander*

Команда ***Открыть с помощью…*** позволяет указать, какой программой можно просматривать и редактировать файлы с указанным расширением.

Команда ***Упаковать*** позволяет упаковать выделенный файл с помощью одного из архиваторов (рис. 1.36). Архиваторы должны быть на диске, и маршруты их поиска должны быть прописаны в файле autoexec.bat. Аналогично выполняется и распаковка файлов. При этом, по возможности, используется внутренний распаковщик.

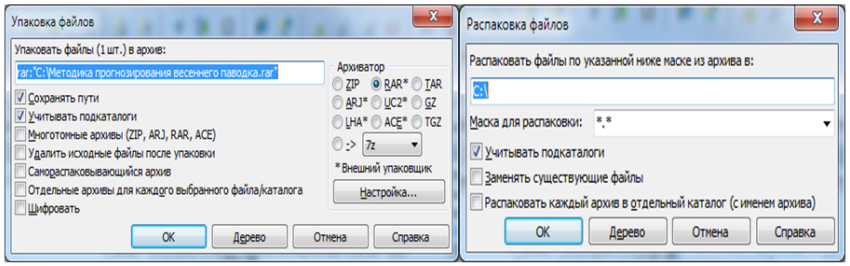


Рис. 1.36. Упаковка и распаковка файлов

Команда ***Разбить*** позволяет разбить длинный файл на части (рис. 1.37). При этом создается заголовочный файл с расширением .CRC и его части с размером, указанном в окне диалога. Части разбитого файла имеют то же имя, что и заголовочный файл, и расширения .001, .002 и т. д. *Части разбитого файла не редактируются.* Из рисунка 1.37 можно догадаться, что эта команда предназначалась для сохранения больших файлов на дискетах 3,5". В настоящее время, в связи с повсеместным использованием флеш-карт большого объема, эта операция потеряла свою актуальность.

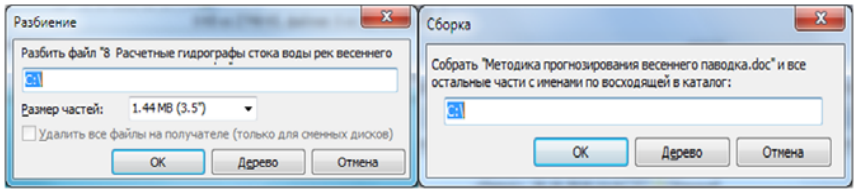


Рис. 1.37. Разбивка и сборка файла

Для сборки файла необходимо выделить заголовочный файл и ввести команду ***Файл, Собрать файл*** или просто щелкнуть дважды по имени заголовочного файла, а затем указать маршрут для сохранения результирующего файла. Для этой цели можно использовать команду *Дерево* в окне диалога.

Меню Выделение обеспечивает групповое выделение файлов, а также сравнение каталогов.

Меню Навигация позволяет перемещаться по каталогам и сохранять цепочки перемещений при поиске файлов. Возникают серьезные сомнения, что кто-то когда-то воспользуется всем этим многообразием возможностей.

Меню Сеть предназначено для работы с сетевыми дисками и управления файловой системой NTFS, с которой работает операционная система *Windows*.

Меню FTP обеспечивает работу с файловым сервером. FTP-сервер позволяет сделать обмен данными с друзьями, знакомыми, деловыми партнерами более быстрым, безопасным и удобным. С помощью FTP-сервера можно не только открывать доступ к определенным папкам на собственном компьютере, но и гибко управлять объемами трафика, а также списками доступных файлов и пользователей. При этом пользователям, для которых предназначены

материалы, также станет гораздо удобнее их получить, так как они смогут использовать для скачивания файлов привычные FTP-клиенты, обеспечивающие докачку и умеющие проводить скачивание в несколько потоков.

Меню Вид позволяет управлять выводом информации в панели. Все вводимые параметры относятся к текущей панели. Меню Вид позволяет выводить информацию о файлах в краткой или полной форме, устанавливать типы файлов, информация о которых будет выводиться в панель, сортировать файлы и изменять порядок сортировки, управлять окнами.

Меню Вкладки позволяет создавать дополнительные вкладки для папок и управлять ими. Вновь создаваемая вкладка содержит все файлы текущей папки. В ней можно создавать новые папки.

Меню Конфигурация позволяет осуществлять разнообразные настройки рабочей среды. Отметим лишь одну из них – управление выводом скрытых или системных файлов: ввести команду ***Конфигурация, Настройка, Содержимое панелей*** и в группе ***Отображение файлов*** установить или снять флажок ***Показывать скрытые/системные файлы***. Этой командой не следует пользоваться без особой надобности.

Меню Инструменты содержит список всех команд Total Commander, команды установки метки диска, поиска файлов, а также команды для копирования файлов, имен файлов, содержимого панелей, управления строкой ввода команд.

Меню Запуск позволяет поместить список имен выделенных файлов в блокнот, производить настройку меню запуска. Команда ***Изменить меню запуска*** позволяет добавлять или удалять команды из меню запуска. Команда ***Изменить главное меню*** содержит команды для перевода описания интерфейса на национальные языки.

Меню Папки. Команды этого меню обеспечивают быстрый переход к системным папкам: Рабочий стол, Мой компьютер и др.

Меню Компьютер служит для управления завершением работы, перезагрузки компьютера, настройки энергосбережения.

Для любителей работать с клавишами следует помнить, что для большинства команд зарезервированы комбинации клавиш, которые указаны в соответствующем меню.

Панели

Панели предназначены для вывода списков папок и файлов на дисках. В заголовке панели указано имя диска, папки и маска,

в соответствии с которой выводится информация в панель. Справа в этой строке имеется два списка. Первый список * содержит список избранных папок. Второй список ▼ содержит список наиболее часто посещаемых папок. В нижней части панели имеется строка состояния, в которой приведены сведения о выделенных файлах и папках. Выделение папок и файлов осуществляется клавишей ***Insert*** или щелчком мыши при нажатой клавише ***Ctrl***.

Одна из панелей активная, заголовок активной панели выделен более темным цветом. В активной панели находится курсор панели – прямоугольник синего цвета. Другая панель – нетекущая. Переход из одной панели в другую осуществляется либо щелчком мыши по нетекущей панели, либо клавишей ***Tab***.

Информация в панели может выводиться в полной или краткой форме. В краткой форме на экран выводится только имя и тип файла (расширение имени файла), в полной форме на экран дополнительно выводятся размер, дата и время создания.

В первых строках панели выводятся имена папок диска. Папки отмечаются значками в виде желтого прямоугольника, справа от имени каталога изображается <Папка>. Самую верхнюю строку занимает ссылка на родительский каталог. Для корневого каталога ссылки на родительский каталог нет. В поле имени для родительского каталога изображается стрелка и символ [..], а справа от него <Папка>.

Имена файлов выводятся строчными буквами. Для файлов с атрибутами «скрытый» или «системный» на значке файла отображается восклицательный знак красного цвета.

Общие принципы управления

Информация в панелях и меню представляется в виде списка. Для перемещения по списку используются клавиши управления перемещением курсора ←, →, ↑, ↓, клавиша ***Home*** устанавливает курсор в начало списка, а клавиша ***End*** – в конец списка. Для быстрого перемещения по списку можно использовать также клавиши прокрутки PgUp и PgDn.

Для ввода команды ее необходимо выделить ***курсором*** и нажать клавишу ***Enter***. Нажатие клавиши ***Enter*** завершает ввод команды. В дальнейшем изложении материала, когда речь будет идти о вводе команд, клавиша Enter может не упоминаться.

Для завершения ввода команды нажмите клавишу Enter.

Ввод команд можно производить также двойным щелчком мыши. При необходимости указать при вводе команды дополнительные параметры необходимо воспользоваться строкой ввода: запишите команду (или имя файла) в строку ввода, допишите необходимые параметры и нажмите Enter.

Ввод команд главного меню осуществляется щелчком мыши. Для ввода команд главного меню можно использовать также ***горячие клавиши*** – символы, выделенные в команде символом подчеркивания, в комбинации с клавишей Alt. При управлении с помощью клавиатуры для перехода в главное меню нажмите клавишу Alt. Для перемещения по командам меню используются клавиши управления перемещением курсора. Для вывода меню второго уровня нажмите клавишу Enter или Пробел. Для выхода из меню нажмите клавишу Esc.

Выделение файлов

Все операции в среде Total Commander выполняются над выделенным файлом или папкой. Для выделения файла (папки) щелкните по нему мышью. Выделение группы файлов выполняется так же, как и в программе Проводник, с использованием клавиш *Shift* и *Ctrl*. Для выделения группы файлов можно использовать также клавишу *Insert*: установите курсор на имя файла и нажмите клавишу Insert – файл выделяется красным цветом. Повторное нажатие клавиши Insert на выделенном файле отменяет выделение. Для группового выделения однотипных файлов используется маска. Чтобы выделить файлы по маске, выберите команду ***Выделение, Выделить группу***, укажите в окне диалога маску или выберите тип файлов по шаблону и щелкните по кнопке ОК. Для отмены выделения введите команду ***Выделение, Снять выделение***.

Работа с файлами

Файлы можно создавать, просматривать, копировать, перемещать, переименовывать и удалять.

Создание файлов

Для создания небольшого файла можно воспользоваться блокнотом, значок которого находятся на панели инструментов ***Системная***.

Просмотр и правка файлов

Для просмотра выделенного файла нажмите клавишу F3. Параметры просмотра/правки можно изменить командой ***Конфигурация, Настройка***, ***Правка/Просмотр***.

Правка выделенных файлов осуществляется нажатием клавиши F4. По умолчанию для правки файлов используется редактор дво-

ичных файлов. Имеется возможность изменить настройку и указать другой редактор для просмотра файлов, как указано выше.

Копирование файлов

Для копирования файлов можно использовать технологию перетаскивания файлов:

1. Выведите в одну из панелей, например, левую, диск и папку, где находится копируемый файл (источник данных).

2. Выведите в другую, нетекущую панель папку назначения.

3. Зацепите копируемый файл (файлы) в левой панели и переместите его в правую панель.

Аналогично выполняется копирование с использованием клавиши F5. После выполнения подготовительных операций по пп. 1, 2 выделите копируемые файлы, нажмите клавишу ***F5***, а затем ***Enter***. По умолчанию файл копируется в нетекущую панель. При необходимости можно изменить маршрут, используя кнопку Дерево окна диалога.

Пересылка и переименование файлов

Пересылка и переименование файлов осуществляются с помощью клавиши ***F6***. При переименовании в окне диалога необходимо указать только новое имя файла, так как файл остается на своем месте. При пересылке указываются маршрут и новое имя файла. Операция пересылки выполняется так же, как и операция копирования, но после ее выполнения исходный файл удаляется с диска.

Удаление файлов

Для удаления файла выделите его и нажмите клавишу ***F8***. При удалении системных файлов выдается предупреждение пользователю об опасной операции.

Поиск файлов

Total Commander предоставляет пользователю, с точки зрения автора, более удобные средства для поиска файлов, чем программа Проводник. Вызов окна для настройки режима поиска осуществляется командой ***Инструменты, Поиск файлов*** или комбинацией клавиш ***[Alt + F7]***. После ввода команды появляется окно диалога (рис. 1.38), в котором в строке ***Искать файл*** необходимо указать ***Имя файла*** или ***маску*** для поиска файла, в строке ***Место поиска*** указать диск и папку. При необходимости можно указать текст в строке ***С текстом***. Для сужения области поиска на вкладке ***Дополнительно*** указать диапазон дат или конкретную дату создания файла. Тип файла для поиска можно задать на вкладке ***Шаблоны***

поиска. Для расширения возможностей программы по поиску файлов можно выбрать ***плагин***[10], а также установить с помощью флажков другие ограничения на область поиска.

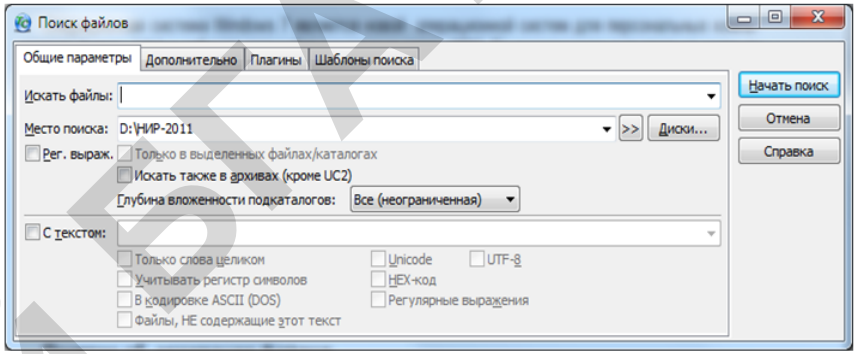


Рис. 1.38. Поиск файлов в *Total Commander*

Управление папками (каталогами)

Для открытия папки (входа в каталог) щелкните по ней дважды мышью или нажмите клавишу Enter. Для выхода из папки (каталога) щелкните дважды мышью по имени родительского каталога или установите курсор на имя родительского каталога и нажмите клавишу Enter. Для создания папки на текущем диске нажмите клавишу ***F7*** и введите в окне диалога имя папки. Для переименования папки выделите ее курсором, нажмите клавишу ***F6*** и введите в окне диалога новое имя, удалив предварительно маршрут, предлагаемый программой по умолчанию. При сохранении маршрута будет выполняться пересылка файла с изменением имени в новую папку. Для удаления папки выделите ее курсором и нажмите клавишу ***F8***. На экран будет выведено окно с предупреждением. Можно отказаться от удаления или подтвердить удаление папки.

Контрольные вопросы

1. Какая информация о файле выводится при полной и краткой формах представления каталога?

2. Какая панель является текущей?

[10] **Плаги́н** (от англ. plug-in) – независимо компилируемый программный модуль, динамически подключаемый к основной программе, предназначенный для расширения и/или использования ее возможностей.

3. Как перевести курсор в другую панель?

4. Что означают выражения «выделить файл», «выделить каталог»?

5. Как «войти» в каталог? Как «выйти» из каталога?

6. Поясните порядок выполнения команд из среды *Total Commander*.

7. Поясните порядок запуска исполняемых и командных файлов.

8. Выведите для редактирования файл autoexec.bat.

9. Выведите в правую панель оглавление диска C:.

10. Выведите в левую панель каталог в полной форме по алфавиту, по расширению имени файла, по дате изменения.

11. Выведите в правую панель каталог в краткой форме.

12. Как осуществляется архивация файлов в *Total Commander*?

13. Как установить атрибуты файлов в Total Commander?

14. Как просмотреть скрытые или системные файлы в *Total Commander*?

Заключение

Операционная система *Windows 7* является новой операционной системой для персональных компьютеров фирмы *Intel* и рабочих станций в локальных сетях ЭВМ. Ее возможности значительно шире, чем у ее предшественников. Программа имеет дружественный и привлекательный интерфейс, позволяющий реализовать практически любые эстетические запросы пользователя. Значительно расширены возможности операционной системы по работе в сетях. Навигация в компьютере осуществляется с помощью встроенных программ Проводник, Мой компьютер, сервисной оболочки *Total Commander*.

1.5. ДОПОЛНИТЕЛЬНЫЕ СВЕДЕНИЯ О РАБОТЕ НА ПЕРСОНАЛЬНОМ КОМПЬЮТЕРЕ

Ключевые слова: архивация, восстановление, компьютерный вирус, степень сжатия.

1.5.1. Архивация файлов

Понятие об архивации файлов

При эксплуатации компьютера по самым разным причинам возможна порча или потеря информации на магнитных дисках:

неправильная корректировка или случайное уничтожение файлов, разрушение информации компьютерным вирусом и тому подобное. Для полного или частичного восстановления потерянной информации необходимо иметь копии используемых файлов и систематически обновлять копии изменяемых файлов. Кроме того, многие файлы являются избыточными. Избыточность при передаче и хранении данных является отрицательным фактором, увеличивающим объемы и приводящим к повышению стоимости процессов хранения и передачи данных. Копии файлов создаются на дискетах или магнитных лентах, при этом они могут храниться в обычном или сжатом виде.

Для создания копий файлов или дисков используются команды операционной системы *Copy*, *XCopy* и др. Для хранения копий файлов в сжатом виде используются специальные ***программы архивации*** файлов.

Архивный файл представляет собой набор из одного или нескольких файлов, помещенных в сжатом виде в единый файл, из которого их можно извлечь при необходимости в первоначальном виде. Архивный файл содержит оглавление, позволяющее узнать, какие файлы содержатся в архиве, имя файла, сведения о каталоге, в котором содержится файл, дату и время последней модификации, размер файла на диске и в архиве, а также код циклического контроля для каждого файла, позволяющий проверить целостность архива.

Процесс создания архива называется ***архивированием*** (или упаковкой), процесс восстановления файла в первоначальном виде – ***разархивированием*** (или восстановлением). Упакованный (сжатый) файл принято называть ***архивом***.

Архивация информации – это такое преобразование информации, при котором объем информации в файле уменьшается при неизменном количестве информации.

Степень сжатия информации зависит от типа архивируемого файла и от используемого метода сжатия. Степень сжатия характеризуется коэффициентом сжатия $\mathbf{K_c}$, который определяется как отношение объема сжатого файла к объему исходного файла, выраженное в процентах.

Все используемые методы сжатия можно поделить на две группы: методы сжатия без потери информации и методы сжатия с потерей информации. При использовании первой группы методов информация восстанавливается полностью, при использовании второй группы методов при восстановлении информации она не может быть восстановлены в первоначальном виде. Очевидно, что

для сжатия текстовых файлов можно использовать только методы сжатия без потери информации. При архивации графических файлов можно использовать любые методы, так как даже в случае потери части информации это может отразиться только на качестве воспроизведенного рисунка.

Для архивации без потери качества используют разные методы: метод Хаффмана, RLE-кодирование, LZW-кодирование.

Метод Хаффмана основан на том, что частота повторения символов в тексте различна. Поэтому можно заменить символы вспомогательным кодом. Чем больше частота повторения символов, тем меньше должна быть длина кода. Кодовая таблица, образовавшаяся при архивации, хранится вместе с архивным файлом.

RLE-кодировка основана на замене повторяющихся последовательностей байтов одной последовательностью с указанием числа повторений.

LZW-кодирование – словарный метод, наиболее распространенный метод архивации. Используется словарь, состоящий из последовательностей данных или слов. При сжатии эти слова заменяются на их коды из словаря. В наиболее распространенном варианте реализации в качестве словаря выступает сам исходный блок данных. Основным параметром словарного метода является размер словаря. Чем больше словарь, тем больше эффективность. Для эффективной работы данного метода при сжатии требуется дополнительная память, приблизительно на порядок больше, чем нужно для исходных данных словаря. Существенным преимуществом словарного метода является простая и быстрая процедура распаковки. Дополнительная память при этом не требуется. Это особенно важно, если необходим оперативный доступ к данным.

Программы для архивации файлов

Существует много программ для архивации файлов. Как правило, эти программы позволяют помещать копии файлов в сжатом виде в архивный файл на диск, извлекать файлы из архива, просматривать оглавление архива и так далее. Разные программы отличаются форматом архивных файлов, скоростью работы, степенью сжатия файлов при помещении в архив, удобством использования. В операционной системе *Windows* наиболее часто применяются программы *WinZip* и *WinRar*. Программа *WinZip* обладает высокой скоростью сжатия, но степень сжатия достаточно низкая. Программа *WinRar* обеспечивает большую степень сжатия, но проигрывает *WinZip* в скорости сжатия данных.

Все без исключения современные архиваторы используют один и тот же алгоритм сжатия без потери информации – словарный LZW.

Программа архивации *WinRAR*

Для архивации файлов в среде Windows разработаны две программы: *WinZip* и *WinRAR*. Программа *WinRAR* имеет более удобный интерфейс (рис. 1.39).

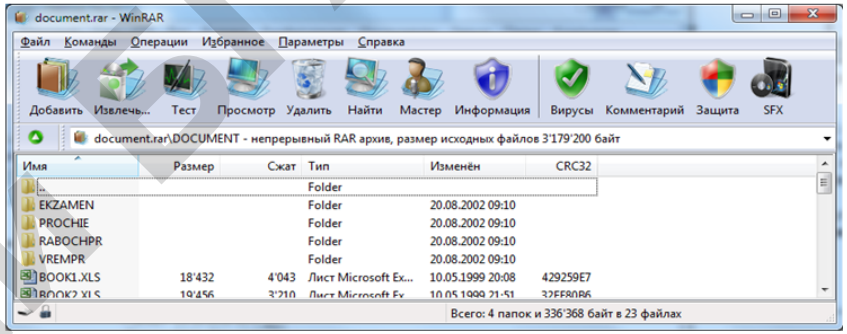


Рис. 1.39. Архиватор *WinRar*

Алгоритм работы при архивации и разархивации файлов также предельно прост.

Алгоритм архивации файлов:

- Выделите в окне диалога архивируемый файл (папку), используя раскрывающийся список адресов.
- Щелкните по кнопке ***Добавить*** – открывается окно диалога *Имя и параметры архива* (рис. 1.40).
- Установите требуемые атрибуты архивации, метод обновления и укажите маршрут, куда следует поместить архивный файл, используя строку ввода или кнопку *Обзор*. Имя архивного файла можно принять по умолчанию.
- При создании архива можно добавить электронную подпись, а также пароль (команда ***Дополнительно, Установить пароль***).
- После настройки параметров архива и маршрута щелкните по клавише *OK*.

Программа *WinRar* позволяет создавать ***самораспаковывающиеся архивы***. Это так называемые SFX-архивы. Обычный архив с помощью команды ***SFX*** может быть преобразован в самораскрывающийся архив. Программа WinRar позволяет также просматри-

вать архив и проверять его целостность (команда ***Тест***), а также проверять на вирусы (команда ***Вирусы***).

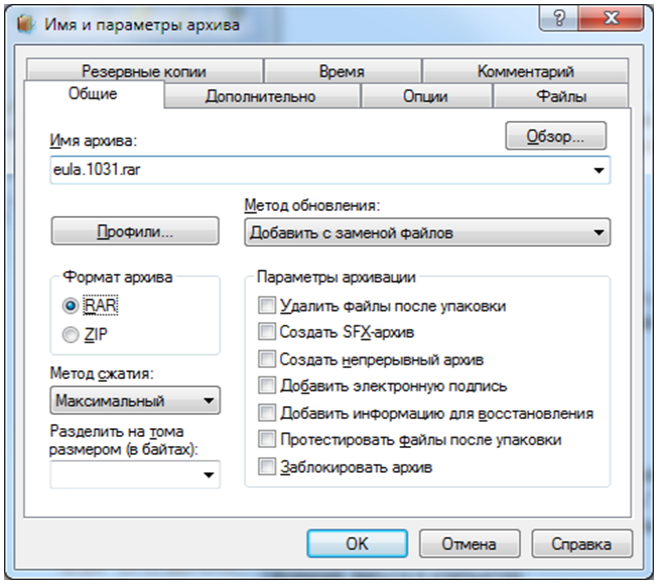


Рис. 1.40. Настройка параметров архива

Алгоритм извлечения файлов

Алгоритм извлечения файлов из архива также прост:

выделите в окне диалога (рис. 1.39) архивный файл;

щелкните мышью по кнопке ***Извлечь в...***;

в открывшемся окне диалога ***Путь и параметры извлечения*** укажите маршрут, куда следует извлечь файлы из архива, и настройте параметры для извлечения файла из архива;

щелкните по кнопке ОК.

1.5.2. Защита от компьютерного вируса

Понятие о компьютерном вирусе

Компьютерный вирус – это специально написанная небольшая по размерам программа, которая может «приписывать» себя к другим программам (т. е. «заражать» их), создавать свои копии и внедрять их в файлы, системные области компьютера и в вычислительные

сети, а также выполнять различные нежелательные действия на компьютере.

Основные пути проникновения вируса в компьютер:

загрузка компьютера с диска, содержащего вирус;

загрузка (скачивание) файлов из сети;

запись файлов с других компьютеров;

Заражение диска может произойти при записи на него зараженного файла.

Зараженный диск – это диск, в загрузочном секторе которого находится программа-вирус.

Зараженная программа – это программа, содержащая внедренную в нее программу-вирус. Когда такая программа начинает работу, то сначала управление получает вирус. Вирус находит и «заражает» другие программы, а также выполняет какие-нибудь вредные действия, например, портит файлы или таблицу размещения файлов на диске, «засоряет» оперативную память и так далее. Для маскировки вируса действия по заражению других программ и нанесению вреда могут выполняться не всегда, а при определенных условиях. После того как вирус выполнит свои действия, он передает управление той программе, в которой находится, и она работает так же, как обычно. В результате внешне работа зараженной программы выглядит так же, как и незараженной. Последствия заражения программ вирусом могут быть очень серьезными, вплоть до разрушения баз данных (баз знаний), нарушения работы сложных вычислительных систем.

Внешними проявлениями зараженности программы вирусом могут быть: замедление времени работы, сообщения о нехватке памяти, программа не находит нужный файл, могут быть и другие непонятные явления.

Заражение компьютера вирусом происходит, как правило, при использовании «чужих» дискет или через компьютерную сеть. Каждая разновидность вируса может заражать только один или два типа файлов. Чаще всего встречаются вирусы, заражающие COM-файлы, на втором месте по распространенности находятся вирусы, заражающие EXE-файлы, и вирусы, заражающие COM-файлы и EXE-файлы. Иногда компьютеры заражаются вирусами, распространяющимися через загрузочные сектора дискет.

Классификация программ-вирусов

Программы-вирусы классифицируют по следующим признакам:

по «среде обитания»: сетевые, файловые, макровирусы, загрузочные, файловозагрузочные;

по способу заражения программ: резидентные и нерезидентные;

по степени воздействия: неопасные, опасные, очень опасные;

по особенностям алгоритмов: паразитические, репликаторы, невидимки, мутанты, троянские.

Сетевые вирусы распространяются по различным компьютерным сетям.

Файловые вирусы внедряются главным образом в исполняемые модули, т. е. файлы, имеющие расширение *COM* и *EXE*. Файловые вирусы могут внедряться и в другие типы файлов, но, как правило, записанные в таких файлах, они никогда не получают управление и, следовательно, теряют способность к размножению.

Макровирусы – это файловые вирусы, использующие особенности документов подготовленных редакторами текстов, электронных таблиц, систем управления базами данных. В этих файлах могут содержаться программы на макроязыках, которые позволяют создавать программы-вирусы.

Загрузочные вирусы внедряются в загрузочный сектор дисков или в сектор, содержащий программу загрузки системного диска.

Файловозагрузочные вирусы заражают как файлы, так и загрузочные сектора дисков.

Резидентные вирусы при заражении (инфицировании) компьютера оставляют в ОЗУ свою резидентную часть, которая потом перехватывает обращения операционной системы к объектам заражения (файлам, загрузочным секторам дисков и т. п.) и внедряется в них. Резидентные вирусы находятся постоянно в оперативной памяти и остаются активными до момента выключения питания компьютера.

Нерезидентные вирусы не заражают оперативную память компьютера и являются активными ограниченное время, время работы зараженной программы.

Неопасные вирусы не мешают нормальной работе компьютера, но уменьшают объем свободной оперативной памяти на дисках. Действие таких вирусов проявляется в появлении каких-нибудь графических или визуальных эффектов.

Опасные вирусы могут приводить к различным нарушениям в работе компьютера.

Очень опасные вирусы приводят к порче программы, уничтожению данных, стиранию информации в системных областях дисков.

По особенностям алгоритма вирусы трудно классифицировать из-за большого разнообразия.

Паразитические вирусы – простейшие. Они изменяют содержимое файлов и секторов дисков. Эти вирусы легко обнаруживаются и уничтожаются.

Вирусы-репликаторы (черви) распространяются по компьютерным сетям. Они вычисляют адреса сетевых компьютеров и записывают по этим адресам свои копии.

Вирусы-невидимки (стелс-вирусы) очень трудно обнаружить и уничтожить, так как они перехватывают обращения операционной системы к пораженным файлам и секторам дисков и подставляют вместо своего тела незараженные участки диска.

Вирусы-мутанты содержат алгоритмы шифровки-расшифровки, благодаря которым копии одного и того же вируса не имеют ни одной повторяющейся цепочки байтов. Из-за этого свойства они очень трудно обнаруживаются.

Троянские программы-вирусы очень опасны, хотя и не способны к саморазмножению. Такие вирусы, маскируясь под полезную программу, разрушают загрузочные сектора дисков и файловую систему дисков.

Профилактика заражения вирусом

Для предотвращения заражения компьютера вирусом, а также облегчения процедуры «лечения» зараженного компьютера необходимо выполнять некоторые профилактические мероприятия:

создание архивных копий информации и дискет с программными продуктами. Необходимо периодически архивировать те файлы, которые Вы создали или изменили. Перед архивацией файлов целесообразно выполнять программу для ранней диагностики наличия вируса, чтобы убедиться в отсутствии вируса в компьютере и избежать помещения испорченных или зараженных файлов в архив;

разграничение доступа к данным. На жестком диске целесообразно создать логический диск, защищенный от записи, и поместить в него программы и данные, которые надо только использовать, но не изменять;

защита дискет от записи. Все архивные дискеты с программами и данными необходимо защищать от записи;

использование для перезагрузки компьютера только защищенной от записи эталонной дискеты с операционной системой для предотвращения заражения вирусом, распространяющимся через загрузочные сектора дискет;

использование резидентных программ-фильтров для защиты от вируса либо постоянно, либо тогда, когда это возможно;

проверка целостности программ и данных каждый раз в начале работы с компьютером, что позволит выявить наличие вируса на раннем этапе. Для этого следует установить на компьютер антивирусную программу Касперского (или другую антивирусную программу). После установки она будет запускаться при каждой загрузке компьютера. Установка программы несколько замедлит работу компьютера;

проверка чужих носителей информации на отсутствие вируса перед их загрузкой в компьютер.

Программы для борьбы с компьютерным вирусом

Число вирусов постоянно растет. Появление самих вирусов и их рост (как количественный, так и качественный) связаны с несовершенством как человеческой личности, так и самого общества, общественно-экономических отношений, продуктом которых является человек как личность.

Наличие яда, как известно, всегда вызывает появление противоядия, в данном случае – программ для борьбы с вирусом, число которых также растет. Имеется несколько групп программ для борьбы с компьютерным вирусом: программы-детекторы, программы-доктора, программы-ревизоры, доктора-ревизоры, программы-фильтры, программы-вакцины (иммунизаторы).

Программы-детекторы позволяют обнаруживать файлы, зараженные каким-либо одним или несколькими известными вирусами. Эти программы проверяют, имеется ли в ОЗУ и файлах на указанном пользователем диске специфическая для данного вируса комбинация битов (сигнатуры вирусов). При ее обнаружении на экран выводится соответствующее сообщение. Многие программы-детекторы могут не только обнаруживать, но и «лечить» зараженные файлы, настраиваться на новые типы вирусов, если указана комбинация битов, присущая данному вирусу. Наиболее известны программы-детекторы Aidstest, Scan, Norton AntiVirus, Droctor Web, AVSP. Такие программы, как Aidstest и AntiVirus, могут обнаруживать более 1000 вирусов.

Программы-ревизоры – самое надежное средство защиты от вирусов. Они сначала запоминают сведения о состоянии программ и системных областей дисков тогда, когда компьютер не заражен вирусом, а затем периодически или по запросу пользователя сравнивают текущее состояние с исходным. При выявлении несоответствия выдается сообщение пользователю. Программы-ревизоры запускаются при каждом включении компьютера и позволяют

выявить наличие вируса на ранней стадии, когда он не нанес еще большого вреда. Проверка осуществляется путем подсчета контрольной суммы файла и сравнения ее с контрольной суммой исходного файла. Эта проверка занимает много времени, поэтому такой проверке подвергаются наиболее важные файлы. Для остальных программ проверяется их размер, указанный в каталоге. Программы-ревизоры могут обнаруживать вирусы-невидимки.

К программам-ревизорам относятся такие программы, как ADinf фирмы «Диалог-Наука», CRCLIST, CRCTEST.

Программы-доктора «лечат» зараженные программы или диски, выкусывая из зараженных программ тело вируса, т. е. восстанавливая программу в том состоянии, в котором она находилась до заражения вирусом. Основной недостаток программ-докторов – их узкая специализация. Программа-доктор, ориентированная на некоторый фиксированный набор вирусов, не в состоянии будет «вылечить» файлы, зараженные другими типами вирусов, кроме того, программы-доктора лечат не всегда правильно. Некоторые программы, например, AVSP фирмы «Диалог-МГУ», могут обучаться не только способам обнаружения, но и способам «лечения» новых вирусов.

Программы доктора-ревизоры – это гибрид программ ревизоров и докторов, т. е. программы, которые не только обнаруживают изменения в системных файлах, но и могут, в случае изменений, автоматически вернуть файлы в исходное состояние. Доктора-ревизоры обнаруживают «нападение» вируса и лечат зараженные программы, причем, в отличие от программ-докторов, лечат правильно. Доктора-ревизоры обеспечивают защиту от 90–95 % вирусов. К докторам-ревизорам относятся программы ADinf + ADinfExt фирмы «Диалог-Наука» и комплексная антивирусная система AVSP фирмы «Диалог-МГУ».

Программы-фильтры – это резидентные программы для защиты от вирусов. Они перехватывают те обращения к операционной системе, которые используются вирусами для размножения и нанесения вреда, и сообщают об этом пользователю. Пользователь может разрешить или запретить выполнение соответствующей операции. К таким подозрительным операциям относятся, например:

изменения .COM и .EXE файлов;

снятие атрибутов только для чтения;

прямая запись на диск по абсолютному адресу;

запись в загрузочный сектор диска;

загрузка резидентной программы.

Эти программы также не обеспечивают стопроцентной защиты, так как некоторые вирусы используют для своего размножения непосредственное обращение к программам операционной системы, а не стандартные прерывания. Кроме того, программы-фильтры не обнаруживают вирусы, которые распространяются через загрузочные сектора дискет.

Программы-вакцины, или иммунизаторы – это резидентные программы. Они модифицируют программы и диски таким образом, что это не отражается на работе программы, но тот вирус, от которого производится «вакцинация», считает эти программы или диски уже зараженными. Очевидно, что нельзя провести вакцинацию программ сразу от нескольких типов вирусов, поэтому такие программы неэффективны и не получили широкого распространения.

Среди популярных и эффективных антивирусных программ, работающих под управлением операционных систем Windows, выделяют: антивирус Касперского – *AntiViral Toolkit Pro (AVP), F-Secure Anti-Virus, McAfee VirusScan, Norton AntiVirus, Panda Antivirus, Sophos Anti-Virus; Trend Micro PC-Cillin*.

В настоящее время ведутся работы по созданию антивирусных программ на основе искусственного интеллекта.

Действия при заражении компьютера вирусом

При заражении компьютера вирусом или подозрении на заражение вирусом рекомендуется следующий порядок действий:

1. Запустить программу AVP-монитор для проверки и уничтожения вирусов на дисках.

2. После обнаружения вируса следует последовательно обезвредить все дискеты, которые могли подвергнуться заражению вирусом. Если на дискете нет нужных файлов, копий которых нет в архиве, то рекомендуется заново отформатировать диск, а затем восстановить файлы с архивных дисков.

При отсутствии опыта борьбы с вирусами рекомендуется обратиться к специалисту. И не забудьте оповестить всех товарищей, с которыми вы обмениваетесь программами, о возможности заражения их компьютеров вирусами.

Контрольные вопросы

1. Что понимается под архивацией файлов?
2. Что такое архивный файл?
3. Назовите основные программы для архивации/разархивации файлов.

4. Поясните алгоритм архивации файлов.
5. Поясните алгоритм восстановления файлов из архива.
6. Что такое компьютерный вирус?
7. В чем состоит опасность заражения компьютера вирусом?
8. Перечислите основные меры профилактики для защиты от компьютерного вируса.
9. Каково назначение программ-докторов?
10. Какие первоочередные меры необходимо принять при подозрении на заражение компьютера вирусом?

Заключение

Программы архивации обеспечивают сжатие данных при сохранении резервных копий программ и данных на внешних носителях информации и восстановление их, в случае необходимости, в первоначальном виде.

Антивирусные программы обеспечивают защиту компьютера от программных вирусов, распространяющихся через загрузочные сектора дисков, исполняемые файлы или сети ЭВМ. Для обеспечения защиты компьютера от заражения вирусами необходимо соблюдать несложные правила и установить на компьютер одну из популярных антивирусных программ.

2. СИСТЕМА КОМПЬЮТЕРНОЙ МАТЕМАТИКИ MATHCAD

Ключевые слова: система компьютерной математики

2.1. НАЗНАЧЕНИЕ И ВОЗМОЖНОСТИ

Математический пакет Mathcad (система компьютерной математики) предназначен для решения физико-математических задач. По мнению специалистов, он является одним из наиболее удобных математических пакетов для несложных расчетов на персональном компьютере. Имеется уже несколько версий этой программы. В данном пособии описан интерфейс Mathcad 11.

Программа имеет естественный язык представления математических зависимостей и инструменты их выбора. Имеется возможность вводить размерности переменных и автоматически контролировать соответствие размерностей операндов и результата. Mathcad имеет также встроенный символьный процессор, который позволяет, например, производить вычисления в символьном виде или подготовить статью без помощи других средств редактирования текста. Этот редактор обеспечивает экспорт данных, графиков, используя буфер обмена, а также механизм вставки и внедрения (DDE, OLE). Основным достоинством пакета разработчики считают возможность только собственными средствами сформулировать задачу в привычных обозначениях, исследовать ее, обработать исходные данные, выбрать метод решения, получить результаты, задокументировать их и передать по сети. Вместе с пакетом могут использоваться прикладные дополнения (обработка сигналов и изображений, анализ электрических цепей, численный анализ, «продвинутая» математика и статистика, теория очередей). Имеется ресурсный центр, позволяющий получать необходимую информацию и программное обеспечение через Интернет.

Запуск программы можно произвести через главное меню Windows или щелчком мыши по значку программы на рабочем столе.

2.2. ИНТЕРФЕЙС ПОЛЬЗОВАТЕЛЯ

После запуска программы на экране появляется рабочее окно Mathcad (рис. 2.1). При загрузке на экране появляются те окна, которые были на экране при выходе из программы. Рабочее окно Mathcad – типичное окно приложения Windows. Оно содержит строку заголовка с кнопками системного меню, открытия, развертывания/свертывания и закрытия окна, главное меню и панели инструментов, окно редактирования документа, строку состояния.

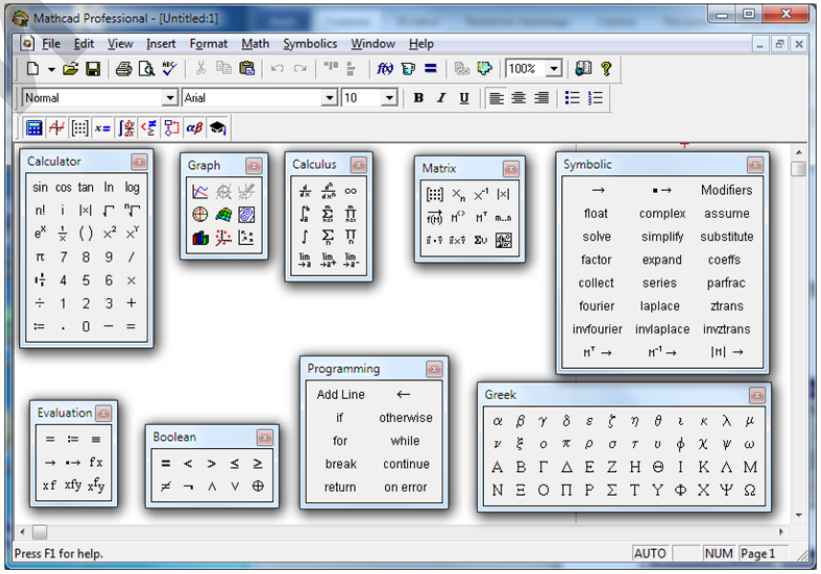


Рис. 2.1. Рабочее окно Mathcad

Главное меню

Главное меню содержит следующие команды.

Файл (File) – создание, открытие, сохранение и печать документов, пересылка их по электронной почте, настройка параметров страниц.

Правка (Edit) – правка текста: копирование, вставка, удаление, поиск, замена текста, переход к указанной странице.

Вид (View) – содержит команды, влияющие на внешний вид документа в окне редактора Mathcad, а также команды для создания файлов анимации.

Вставка (Insert) – содержит команды вставки различных объектов в документы: графиков, функций, рисунков, областей (регионов).

Формат (Format) – команды форматирования текста и графиков, оформления документов, форматирования вывода результатов на экран.

Математика (Math) – содержит команды управления вычислениями, оптимизации и настройки параметров вычислений.

Символика (Symbolics) – команды символьных вычислений, преобразования выражений, матриц, стиля вычислений.

Окно (Window) – управление многооконным интерфейсом.

Справка (Help) – доступ к справочной системе Mathcad.

Панели инструментов

Mathcad имеет три панели инструментов: стандартную панель инструментов (Standard), панель инструментов форматирования (Formating) и панель инструментов Математика (Math) (рис.2.1). Панель инструментов Математика, в свою очередь, содержит ряд панелей, на которых расположены кнопки для ввода команд и операторов языка программирования (на рис. 2.1 открыты все панели инструментов, расположенные на панели инструментов Математика):

арифметика (Calculator) – содержит кнопки для ввода основных математических операторов и функций;

график (Graph) – содержит операторы для построения графиков в разных системах координат и разных типов;

матанализ (Calculus) – содержит операторы для выполнения операций математического анализа (дифференцирования, интегрирования, разложения функций и нахождения пределов);

матрицы (Matrix) – содержит кнопки для вставки матриц и матричных операторов;

вычисления (Evaluation) – для вставки операторов управления вычислениями;

булевы операторы (Boolean) – операторы отношений и булевой алгебры;

программирование (Programming) – содержит операторы, позволяющие выполнять программирование средствами языка Mathcad;

символы (Symbolic) – операторы символьных вычислений и преобразований;

греческие символы (Greek) – для вставки в документы и формулы символов греческого алфавита.

Рабочий документ

На рабочем поле документа располагается *курсор* – красный крестик, который отмечает точку ввода. На рабочем поле видны также сплошные вертикальные и пунктирные горизонтальные линии разметки границ листов. Параметры страницы можно изменить командой ***Файл, Параметры страницы*** (Page, Setup).

Для удобства редактирования может выводиться горизонтальная линейка командой ***Вид, Линейка*** (Ruler).

В документ могут вводиться текст, числа и формулы. Признаком ввода текста является символ парные кавычки (“).

Области рабочего документа

Формулы и текст размещаются в специальных областях (Regions). В Mathcad возможно создание двух областей: области математики и текстовой области. Область математики открыта по умолчанию. Текстовая область создается командой: ***Вставка, Текстовая область***.

Для работы с областями используются команды меню: ***Вставка области*** (Area), ***Математическая область*** (Math Region) или ***Текстовая область*** (Text Region).

Внутри ***Текстовой области*** можно вставлять формулы, используя ***Математическую область*** (команда Вставка, Математическая область):

Существующие области можно показать командой ***Вид, Области*** (View, Regions).

В Mathcad можно вставлять закрытые области, которые могут быть заблокированы от изменения пользователем. Закрытые области создаются командой ***Вставка, Область.***

Блокировка областей осуществляется командой ***Формат, Область, Блокировка***, а разблокировка – командой ***Формат, Область, Разблокировка.***

Закрытые области могут также быть скрыты соответствующими опциями команды Формат, Область. Скрытая область сжимается в одну горизонтальную линию.

Перемещение по рабочему листу

Перемещение по рабочему листу осуществляется с помощью мыши, клавиш управления перемещением курсора, клавиши Пробел, клавиш прокрутки и линеек прокрутки. Перейти к нужной странице можно командой ***Правка, Перейти к странице*** (Edit, Go to Page). При попадании в математическую область курсор принимает вид ломаной линии синего цвета в виде уголка, а при попадании в текстовую область – вертикальной черты красного цвета.

При вводе операторов управления вычислениями, функций или математических операторов автоматически формируется область с маркерами мест заполнения – местозаполнителями, или слотами, в которые должны вводиться числа или другие функции. Ввод данных осуществляется в месте расположения линии ввода. Перемещение линии ввода между слотами осуществляется с помощью клавиш управления перемещением курсора, клавиш ***Home, End, Tab, Пробел***. Для выделения всей формулы необходимо, чтобы нижняя линия ввода подчеркивала всю формулу, это делается с помощью клавиши Пробел. Выделить всю формулу или ее часть можно также с помощью мыши, протаскивая ее по тексту формулы. Для изменения положения курсора ввода используется клавиша ***Insert*** клавиатуры компьютера.

Изменение масштаба осуществляется командой ***Вид, Масштаб*** (View, Zoom).

В процессе работы с документом (вставки, удаления) в документе накапливается «мусор» – не удаленные части формул, рисунков и т. п. Для очистки документа необходимо ввести команду ***Вид, Очистка*** (View, Refresh).

Режимы вычислений

Mathcad имеет два режима вычислений: ручной и автоматический. Для изменения режима вычислений служит команда ***Математика, Автоматические вычисления*** (Math, Automatic Calculation), для установки режима автоматического вычисления следует установить флажок щелчком мыши.

Для прерывания вычислений следует нажать клавишу ***Esc***.

Имеется возможность отключить режим вычисления для выделенной формулы или для всего документа. Для этого служит команда ***Запретить вычисления*** (Disable Evaluation), которая вводится через контекстное меню формулы.

Строка состояния

Строка состояния содержит информацию о состоянии системы:

подсказку о получении контекстно-зависимой помощи – клавиша F1;

режим вычисления – AUTO (автоматический);

раскладка клавиатуры – CAP (нажата клавиша Caps Lock);

состояние дополнительной панели управления – NUM;

номер страницы – Page.

Справочная система

Mathcad имеет мощную систему встроенной помощи, вызываемую кнопкой F1, а также **ресурсный центр**, который позволяет получить подсказку, рассмотреть пример решения и, при необходимости, скопировать пример на рабочее поле.

Содержание выпадающего меню меню Help (Справка) приведено на рис. 2.2. Оно содержит три раздела. Первый раздел содержит справочные системы по вопросам использования Mathcad:

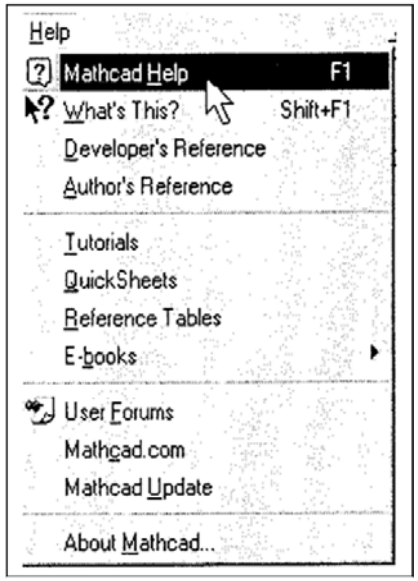


Рис. 2.2. Выпадающее меню Справка

Mathcad Help (Справка) – система справки или технической поддержки;

What's This? (Что это такое) – контекстно-зависимая интерактивная справка;

Developer's Reference (Справка для разработчиков) – дополнительные главы справки для разработчиков собственных самостоятельных приложений на языке Mathcad;

Author's Reference (Справка для авторов) – дополнительные главы справки для пользователей, разрабатывающих собственные электронные книги Mathcad.

Второй раздел – Ресурсы Mathcad. Это дополнительные материалы, организованные в виде электронных книг, содержащих множество математических примеров:

Tutorials (Учебники) – библиотека электронных книг, построенных в форме обучающих курсов;

QuickSheets (Быстрые шпаргалки) – быстрый поиск и справочная таблица, содержащая множество шаблонов решения задач. Например, для получения подсказки о правилах записи алгебраических выражений необходимо войти в раздел и выбрать опцию *Arithmetic and algebra*;

Reference Tables (Справочный стол) – набор физических и инженерных таблиц, включающих перечни физических констант, единицы измерения, параметры веществ;

E-books (Электронные книги) – доступ к существующим электронным книгам пользователя и встроенным электронным книгам, посвященным расширениям Mathcad.

Третий раздел – выход на Интернет-ресурсы (форумы, связь с фирмой, обновления Mathcad).

И последнее – *About Mathcad* – информация о фирме-разработчике программного продукта.

Контрольные вопросы

1. Каковы назначение и функциональные возможности математической системы Mathcad?
2. Перечислите команды главного меню и поясните их назначение.
3. Какие панели инструментов имеются в системе?
4. Какие панели инструментов размещены на панели инструментов Математика?
5. Какие области могут быть созданы на рабочем листе, как можно управлять этими областями?
6. Какие режимы вычислений имеются в Mathcad?
7. Что такое Ресурсный центр и каково его назначение?

2.3. ОРГАНИЗАЦИЯ ВЫЧИСЛЕНИЙ И ТИПЫ ДАННЫХ

Ключевые слова: аргумент, константа, переменная, операнд, функция.

Алфавит

Система компьютерной математики Mathcad допускает использование в командах и именах переменных любых символов латинского и греческого алфавита, чисел и специальных знаков. Русские символы могут использоваться только в текстовых документах и символьных переменных.

Переменные, константы, функции, операторы

В математических выражениях в качестве *операндов* могут использоваться *переменные*, *константы* и *функции*.

Переменные

Переменные – это данные, значения которых могут изменяться в процессе вычислений. Переменная имеет имя (идентификатор), по которому программа обращается к ней. Длина имени переменной не ограничена. В имени переменной допускается использовать все символы латинского языка, числа, греческие символы и ряд специальных символов: штрих, бесконечность, символ подчеркивания, знак процента, нижний индекс.

Редактор Mathcad различает прописные и строчные буквы, а также их начертание. Например, символы **x** и ***x*** или **x** и ***X*** для редактора будут разными символами. При написании имен переменных необходимо соблюдать некоторые требования:

- Имя переменной не может начинаться с цифры.
- Символ ∞ может быть только первым символом в имени переменной.
- Все символы в имени должны иметь один стиль и шрифт.
- Не допускается использовать в качестве имен переменных зарезервированные слова.
- Допускается использовать в качестве имен переменных выражения, например: [a + b] или a + b.

Размерные переменные

Mathcad позволяет создавать размерные переменные. Для создания размерной переменной выполните следующие действия:

- Присвойте переменной значение: u := 10.
- Введите символ * сразу же после цифр.
- Выберите команду ***Вставка, Единицы*** (Insert, Unit) и в списке ***Единицы*** (Unit) выберите требуемую единицу измерения – Volt

(в примере), аналогично введите величину тока – *I*, размерность результата формируется автоматически в соответствии с выбранной системой единиц измерения:

$$U := 10 \cdot V$$
$$I := 5 \cdot A$$
$$R := \frac{U}{I} \qquad R = 2\,kg\,m^2\,s^{-3}\,A^{-2}$$

Система единиц измерения устанавливается командой ***Математика, Параметры***, вкладка ***Единицы измерения*** (Math, Options, Unit System). По умолчанию установлена международная система единиц измерения СИ (SI).

Константы

Mathcad имеет два типа констант: математические и системные.

Математические константы:

∞ может вставляться с панели Матанализ или комбинацией клавиш [Ctrl + Shift + z];

e – основание натурального логарифма, символ e;

π – вводится с панели инструментов Арифметика или комбинацией клавиш [Ctrl + Shift + p];

i, j – мнимая единица. Вводится с клавиатуры или с панели инструментов Арифметика;

% – символ процента, делит число на 100.

Системные константы определяют работу большинства численных алгоритмов, реализованных в Mathcad:

TOL – точность численных методов;

CTOL – точность вычисления выражений, используемых в некоторых численных методах;

ORIGIN – начальный индекс нумерации элементов массива и др.

Функции

В Mathcad используется большое число встроенных функций. Все функции для облегчения их поиска разбиты на категории (30 категорий). Вызов функций осуществляется командой ***Вставка/Функция*** или кнопкой f(x) на панели инструментов Стандартная. При вызове функции открывается окно диалога ***Вставить функцию*** (рис. 2.3).

В левой части окна выводится список категорий, а в правой – список соответствующих функций. Выберите нужную категорию, а в ней требуемую функцию. Для решения вычислительных задач используются категории «***логарифм***», «***тригонометрия***», «***усечение и округление***». Если неизвестно, к какой категории относится функция, то выберите категорию ***Все*** (All). Основные встроенные математические функции приведены в таблице 2.1.

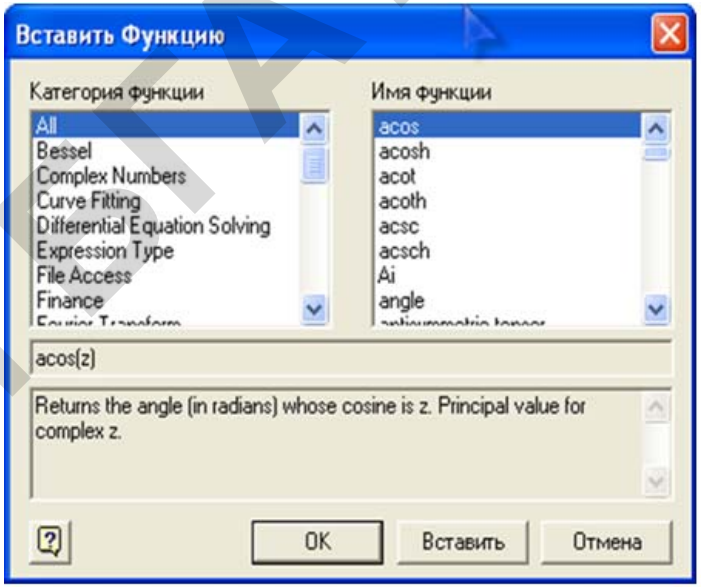


Рис. 2.3. Мастер функций

Таблица 2.1

Встроенные математические функции

Обозначение в математике	Обозначение в Mathcad	Комментарий
Арифметические функции		
|x|	abs(x)	Абсолютная величина числа x
]x[	ceil(x)	Целое, большее, чем x
[x]	int(x)	Целое, меньшее, чем x
e^x	exp(x)	e в степени x, экспонента
Ln(x)	ln(x)	Логарифм по основанию e
Lg(x)	log(x)	Десятичный логарифм
$Log_a(x)$	log(x,a)	Логарифм по произвольному основанию

Окончание табл. 2.1

Обозначение в математике	Обозначение в Mathcad	Комментарий
Условное выражение	if(логич_выраж;x;y)	Функция если. Возвращает значение x, если логическое выражение истинно, и y, если логическое выражение ложно
Тригонометрические и обратные тригонометрические функции		
Cos(x)	cos(x)	
Sin(x)	sin(x)	
Tg(x)	tan(x)	
Ctg(x)	cot(x)	
Arccos(x)	acos(x)	
Arcsin(x)	asin(x)	
Arctg(x)	atan(x)	
Arcctg(x)	acot(x)	
Число □	□	Ввод с панели инструментов Арифметика
Перевод из градусов в радианы	deg*x	Пример: cos(deg*x)

Для ввода некоторых встроенных функций удобно пользоваться панелями инструментов, например: ***Вид, Панели инструментов, Арифметика***, причем некоторые функции имеются только на панелях инструментов и отсутствуют в мастере функций.

Синтаксис функций: ***имя_функции(аргументы)***.

Внимание! Между именем функции и открывающей скобкой не должно быть никаких символов.

Например, в арифметике мы привыкли писать: $\sin^2(x^3)$.

В математических системах следует писать: $\sin(x^3)^2$.

Функции, определяемые пользователем

Кроме встроенных функций Mathcad позволяет создавать и функции пользователя. Синтаксис определения функции пользователя:

<имя функции>(<аргументы>) := <выражение>.

Например: $g(x) := e^x + \sin(bx)$ – функция одной переменной, $u(x,y) := x^2\cos(x + y)$ – функция двух переменных. Параметры x и y –

аргументы функции, формальные параметры. При использовании функций вместо них указываются фактические параметры.

Функцию можно запомнить на текущий сеанс по ***Shift + F9***. Новую функцию можно определить через функции, заданные ранее, а также через функции, встроенные в Mathcad.

Используется функция пользователя так же, как и встроенная функция.

Операторы

Операнды в математических выражениях объединяются операторами. В зависимости от используемого оператора различают *арифметические*, *вычислительные*, *логические*, *символьные* и *матричные* операторы.

Mathcad имеет специфический редактор, в котором многие операторы вводятся с помощью комбинаций клавиш. Запомнить эти комбинации, особенно на первых порах, достаточно сложно. К счастью, разработчики программы предусмотрели возможность ввода всех операторов с панелей инструментов. Предпочтительно вводить операторы с использованием этих панелей, так как иногда операторы, введенные с помощью комбинаций клавиш, работают неправильно.

Арифметические операторы:

+, – – сложение и вычитание;

•, /, ÷ – умножение и деление;

! – вычисление факториала;

$|x|,\ \sqrt{\ },\sqrt[n]{\ },x^y,e^x,\frac{1}{x},x^2$ – вычисление соответствующих математических функций;

( ) – скобки – операторы изменения порядка вычислений;

:= – присваивание значения переменной, функции;

= – численный вывод.

Вычислительные операторы

Вычислительные операторы вводятся с панели инструментов ***Матанализ*** и позволяют вычислять производные, интегралы, суммы и произведение рядов, пределы.

Логические операторы

Логические операторы вводятся с панели инструментов ***Булево***. На этой панели расположены операторы отношения:

=, <, >, <=, >=, ≠ – равно, меньше, больше, меньше или равно, больше или равно, неравно,

а также булевы операторы:

¬ – отрицание, ∨ – логическое сложение ИЛИ, ∧ – логическое умножение И, ⊕ – исключающее ИЛИ.

Матричные операторы

Матричные операторы определяют действия над векторами и матрицами:

$\begin{bmatrix} \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet \end{bmatrix}$ – вставка матрицы или вектора, размер матрицы запрашивается;

$\mathbf{x_n}$ – ввод нижнего индекса;

$\mathbf{x^{-1}}$ – обратная матрица;

$|\mathbf{x}|$ – вычисление определителя квадратной матрицы или модуля вектора;

f(M) – векторизация, или, иначе, умножение элементов одного вектора (матрицы) на элементы другого вектора (матрицы);

$\mathbf{M^{\langle\rangle}}$ – выделение столбца;

$\mathbf{M^T}$ – транспонирование матрицы;

m..n – ввод ранжированной переменной (задать диапазон дискретной величины);

$\vec{x} \cdot \vec{y}$ – скалярное произведение векторов;

$\vec{x} \times \vec{y}$ – векторное произведение векторов;

$\sum \mathbf{v}$ – сумма всех элементов вектора.

Операторы управления вычислением:

= – получение результата в численном виде;

:= – присваивание;

≡ – глобальное присваивание;

→ – оператор символьного вывода результатов;

■→ – символическая оценка.

Следующая группа операторов предназначена для создания пользовательских операторов:

fx – префиксный оператор;

xf – постфиксный оператор;

xfy – инфиксный оператор;

$x^f y$ – древовидный оператор.

Примеры применения префиксного, постфиксного, инфиксного и древовидного операторов приведены на листингах 2.1 и 2.2.

Типы данных

Mathcad может работать с числами, строками символов и массивами.

Листинг 2.1. **Пример создания оператора**

Создадим оператор, в котором выражение зависит только от одной переменной x.

$$Calcx(x) := 5 \cdot x^2 + 2 \cdot x - 3$$

Применим этот оператор для вычисления выражения при x=2

а) префиксный оператор fx: Calcx 2 = 21

б) постфиксный оператор xf: 2 Calcx = 21

Листинг 2.2. **Пример создания оператора**

Создадим аналогично оператор, в котором выражение зависит от двух переменных x и

$$Calcx2(x, y) := x^2 + 2 \cdot y$$

Применим этот оператор для вычисления выражения при x=2 и y=3

а) инфиксный оператор xfy: 2 Calcx2 3 = 10

б) древовидный оператор $x\ ^f y$: Calcx2 = 10 (2, 3)

Числа в Mathcad представляются по умолчанию в виде вещественных чисел с одинарной точностью.

Действительные числа

Действительные числа могут быть представлены (листинг 2.3):

Листинг 2.3. **Форматы представления чисел**

Число 4/3

Обычный (General)	Десятичный (Decimal)	Научный (Scientific)	Инженерный (Engineering)	Дробный (Fracnion)
1.333	1.333	1.333×10^0	1.333×10^0	$\frac{4}{3}$

как целое число (1, 14, 76);

как десятичное число с любым количеством десятичных знаков, по умолчанию три знака после точки (1.763, 23.185). Разделителем целой и дробной части является точка. Изменить число знаков после запятой можно с помощью окна диалога ***Формат результата*** (рис. 2.4), который вызывается командой ***Формат, Результат***;

в виде обыкновенной дроби: 1/5, 5/6;

в экспоненциальной форме ($0.134 * 10^{-5}$, $2.765 * 10^3$).

Для представления чисел в других системах счисления используются суффиксы после числа:

b – двоичное: 10011011b;

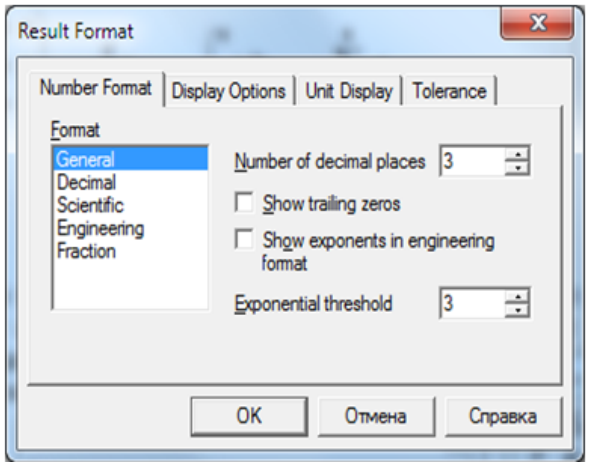


Рис. 2.4. Формат результата

o – восьмеричное: 3751o;

h – шестнадцатеричное: A8C7h.

Комплексные числа

Mathcad успешно работает и с комплексными числами. Задать комплексное число можно следующим образом: Y := 2i + 4.

Для работы с комплексными числами имеется несколько функций:

Im – выделение мнимой части числа;

Re – выделение действительной части числа;

arg(z) – определение аргумента комплексного числа;

|Z| – модуль комплексного числа.

Ввод данных

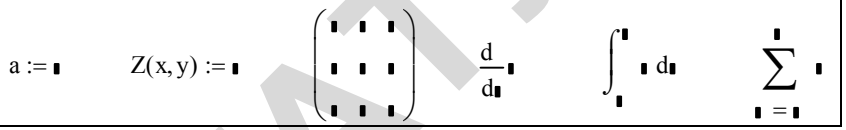
Для присваивания значений переменным используется следующий синтаксис:

<имя_переменной>* := *<выражение>.

Знак логического равенства в выражении, при необходимости, вставляется комбинацией клавиш [Ctrl + =].

При работе с инструментами Mathcad, например, объявлении функции или вводе оператора, на экране в текущей позиции ввода появляется шаблон затребованной конструкции с черными прямоугольниками, которые называют *слотами*, в них следует ввести нужные аргументы (листинг 2.4).

Листинг 2.4. **Шаблоны**



Перемещение между слотами осуществляется с помощью клавиш управления перемещением курсора или мыши. Перемещение между верхними и нижними пределами оператора осуществляется нажатием клавиш *End* и *Home*.

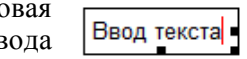
Начинать набор формулы предпочтительнее с выбора операции, формирующей слоты. В процессе набора формулы нужно следить за текущим уровнем слота по синему (уголковому) курсору и при завершении подвыражения нажать клавишу ***Пробел***.

Расширение выделенной части выражения также осуществляется с помощью клавиши ***Пробел***.

Перемещение точки вставки к противоположному концу выделенного выражения выполняется клавишей ***Insert***.

Ввод текста

Ввод текста начинается с ввода символа ", а затем текста. После ввода первого символа появляется текстовая область со слотами и курсором – линией ввода красного цвета.



Форматирование выделенного текста осуществляется с помощью панели инструментов ***Формат***.

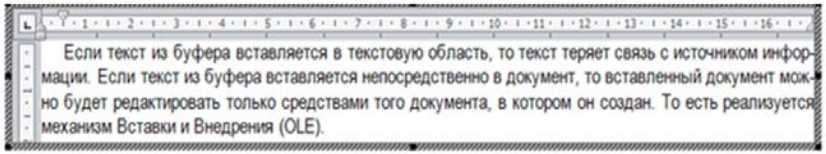
Для импорта текста из других документов, например, Word, необходимо пользоваться буфером обмена. Вставка текста из буфера обмена производится с помощью команды ***Правка/Вставка***, с помощью соответствующей кнопки панели инструментов или комбинации клавиш ***Ctrl + V***.

Если текст из буфера вставляется в текстовую область, то текст теряет связь с источником информации. Если текст из буфера вставляется непосредственно в документ, то вставленный документ можно будет редактировать только средствами того документа, в котором он создан (листинг 2.5). То есть реализуется механизм Вставки и Внедрения (OLE). Для редактирования текста, вставленного в документ, необходимо дважды щелкнуть по нему мышью.

Особенности ввода и редактирования выражений

- Данные вводятся в ***точке ввода***, отмеченной красным крестиком.

Листинг 2.5. **Вставка текста**



- Числа в научной (полулогарифмической) нотации вводятся как произведение мантиссы на 10 в соответствующей степени, например, 0,7364 * 10 + 5.
- Значения углов по умолчанию задаются в радианах. Для перевода градусов в радианы используется функция **deg**, например: **cos(deg * 30)**;
- Мнимая единица записывается как i или j сразу после числового множителя, например, 5i или 5j.
- Латинские буквы, цифры и знаки операций, круглые скобки набираются непосредственно с клавиатуры. С клавиатуры можно вводить также имена встроенных функций.
- Переменные могут иметь имена с индексами (подстрочным текстом), например, x_{min}, переход к набору индексов осуществляется вводом символа "." или "[". Возврат к основному шрифту осуществляется нажатием клавиши *Пробел*.
- Латинские символы могут быть преобразованы в греческие символы с учетом регистра исходного символа нажатием клавиш Ctrl + G сразу после ввода латинского символа. Большинство латинских символов соответствуют греческим символам, исключение составляют пары $c - \chi$, $q - \Theta$, $x - \xi$, $y - \psi$. Греческие символы могут быть введены также с панели инструментов ***Greek***.
- Для Mathcad символы в нижнем и в верхнем регистре разные символы.
- Символ ∞ выбирается из панели с операторами анализа или комбинацией клавиш Ctrl + Z, а число π – комбинацией клавиш Ctrl + P.
- Знаки операций *, /, ^ вводятся с клавиатуры, но при выводе на экран заменяются точкой, обыкновенной дробью и показателем степени соответственно.
- Обратный слэш вызывает шаблон квадратного корня.
- Символ апостроф вставляет выделенное выражение в круглые скобки.
- Вертикальная черта – вызывает шаблон для вычисления абсолютной величины или определителя матрицы.

Ошибочное действие отменяется по нажатию Alt + ←.

Все операции в Mathcad выполняются над выделенными выражениями. Можно выделять часть выражений. Выделение выражений осуществляется мышью путем протаскивания указателя мыши над выделяемым выражением или уголковым курсором синего цвета.

Вывод результатов

Для вывода результатов в числовой форме нажмите клавишу = после выражения.

Для вывода результатов в символьном виде следует ввести оператор →. Оператор символьных вычислений вводится с панели Вычисления или панели Символика.

Матрицы

В Mathcad реализуются два типа массивов:

векторы, матрицы, тензоры (многомерные массивы);

ранжированные переменные.

Для работы с массивами используется панель инструментов ***Матрицы***.

Нумерация массивов по умолчанию начинается с нуля. Для изменения начального индекса используется системная переменная ***ORIGIN*** (команда ***Математика/Параметры/Переменные***, установить значение начального индекса массивов).

Ввод массивов осуществляется с помощью оператора ***Вставить Матрицу*** панели инструментов ***Матрицы***. В открывающемся окне диалога укажите число строк и столбцов – появится шаблон матрицы, заполненный слотами. Введите данные. В местозаполнители (слоты) матрицы можно вводить любые функции.

Элементы массивов обозначаются A_i или A(i). Для ввода нижнего индекса используется оператор X_n панели инструментов Матрицы.

Матрицу можно создать или изменить вводом одного или нескольких элементов матрицы. Например:

$$A_{1,1} := 99 \text{ равносильно } A := \begin{pmatrix} \bullet & \bullet \\ \bullet & 99 \end{pmatrix}.$$

Трехмерную матрицу можно создать следующим образом:

1) определить вектора третьей размерности:

$$A_{0,0} := \begin{pmatrix} 1 \\ 2 \end{pmatrix} A_{1,0} := \begin{pmatrix} 3 \\ 4 \end{pmatrix} A_{0,1} := \begin{pmatrix} 5 \\ 6 \end{pmatrix} A_{1,1} := \begin{pmatrix} 7 \\ 8 \end{pmatrix};$$

2) создать матрицу размерности 2×2:

$$A = \begin{pmatrix} A_{0,0} & A_{0,1} \\ A_{1,0} & A_{1,1} \end{pmatrix};$$

3) для представления матрицы в развернутом виде необходимо ввести команду ***Формат, Результат, Параметры экрана*** и установить флажок ***Развернуть вложенные массивы***. Пример трехмерного массива приведен на листинге 2.6.

Листинг 2.6. **Пример массива**

$$A_{0,0} := \begin{pmatrix} 1 \\ 2 \end{pmatrix} \quad A_{1,0} := \begin{pmatrix} 3 \\ 4 \end{pmatrix} \quad A_{0,1} := \begin{pmatrix} 5 \\ 6 \end{pmatrix} \quad A_{1,1} := \begin{pmatrix} 7 \\ 8 \end{pmatrix}$$

$$A = \begin{pmatrix} \{2,1\} & \{2,1\} \\ \{2,1\} & \{2,1\} \end{pmatrix}$$

Вид массива после установки флажка Развернуть вложенные массивы:

$$A = \left[\begin{pmatrix} 1 \\ 2 \end{pmatrix} \begin{pmatrix} 5 \\ 6 \end{pmatrix} \right] \blacksquare$$

Для обращения к элементам трехмерного массива необходимо использовать следующие конструкции, например:

$$(A_{1,0})_0 = 3 \qquad (A_{1,0})_1 = 4.$$

Создание ранжированных переменных

Для создания ранжированной переменной необходимо указать диапазон изменения переменной. Диапазон указывается горизонтальным двоеточием, которое вводится символом точка с запятой – ; :

S := 3..15 или S := 3,3.2..15,

здесь 3 – начальное значение, 3.2 – следующее значение, 15 – конечное значение.

Шаг ранжированной переменной программа определяет самостоятельно как разницу между следующим и начальным значениями.

В первом примере шаг равен единице. Во втором примере шаг равен 0.2.

Для получения вектора столбца теперь достаточно ввести имя переменной и нажать клавишу «равно».

Ранжированные переменные используются для организации циклических вычислений. Например, табулирования функции одной переменной.

Контрольные вопросы

1. Какие требования предъявляются к именам переменных?
2. Как создать размерную переменную?
3. Как изменить число десятичных знаков в результатах вычислений?
4. Какие константы используются в Mathcad при вычислениях?
5. Перечислите основные арифметические, тригонометрические и обратные тригонометрические функции, используемые в Mathcad?
6. Перечислите арифметические операторы, укажите их приоритет.
7. Перечислите логические операторы. Приведите примеры их использования.
8. Какие типы данных используются в Mathcad?
9. Как осуществляется ввод данных в Mathcad?
10. Как вывести результаты вычислений на экран?
11. Как создать матрицу и ввести в нее данные?
12. Как создать трехмерный массив (тензор)?
13. Что такое ранжированная переменная, как ее создать?

2.4. ВЫЧИСЛЕНИЕ ВЫРАЖЕНИЙ

При числовых расчетах в Mathcad переменные считаются по умолчанию вещественными переменными удвоенной длины.

Для получения результата после ввода выражения достаточно нажать клавишу =. Если выражение заканчивается знаком равенства, то для получения результата достаточно нажать клавишу *Enter*. Данный режим действует при включенном режиме автоматического пересчета. В ином случае для получения результата необходимо нажать *F9*. Прервать процесс вычисления можно клавишей *Esc*. Для продолжения прерванного вычисления повторно нажать *F9*.

При записи выражений в математических системах для исключения ошибок следует строго выполнять правила арифметики. Наиболее часто встречающиеся ошибки приведены в табл. 2.2.

Таблица 2.2

Правила записи выражений в математических системах

Выражение	Правильно	Неправильно
e^x	Exp(x)	Exp^ (x)
$\sin^3(x^2)$	(Sin(x^2)) ^3 или Sin(x^2) ^3	Sin^3(x^2)
$\sqrt[3]{x}$	x^ (1/3)	x^1/3
$\frac{a+b}{u}$	(a + b)/u	a + b/u
$\frac{a+b}{uq}$	(a + b)/(u * q)	(a + b)/u * q

Пример 2.1. Вычислить 5 + 5

Введите 5 + 5, нажмите [=] – ответ 10. Ответ появляется после знака [=].

$$5 + 5 = 10$$

Пример 2.2. Вычислить U = ab/c при a= 5, b = 10, c = 5.675.

Присвойте переменным заданные значения. Команда присваивания := находится на панели инструментов ***Арифметика*** и ***Вычисления***.

Введите a*b/c – появится a*b/c, нажмите [=] – появится ответ 8.811.

Для удаления задачи выделите ее курсором и нажмите клавишу Delete.

$a := 5 \qquad b := 10 \qquad c := 5.675$

$a \cdot \frac{b}{c} = 8.811$ или $U := a \cdot \frac{b}{c} \quad U = 8.811$

Пример 2.3. Вычислить значение выражения

$$y = \frac{\sin^3(x^2) + e^{-2x} + \arcsin(x)}{\sqrt{2x^3 + 5x} - \sqrt[3]{7x - 6}} + \ln(3x^3)$$

при $x = 0{,}5$.

Решение приведено на листинге 2.7.

Листинг 2.7. **Вычисление значения выражения**

$x := 0.5$

$$\frac{\exp(-2\cdot x) + \sin\left(x^2\right)^3 + \operatorname{asin}(x)}{\sqrt{2\cdot x^3 + 5\cdot x - (7\cdot x - 6)^{\left(\frac{1}{3}\right)}}} + \ln\left(3\cdot x^3\right) = -0.601 + 0.455i$$

Пример 2.4. Протабулировать функцию $y = \cos(x) + e^x$ на отрезке $[0, \pi]$ с шагом 0,5.

Решение:

1. Создать ранжированную переменную. Для создания ранжированной переменной используется следующий порядок:

- Ввести символ *z*, нажмите комбинацию клавиш [Shift + :].
- Ввести начальное значение аргумента – 0, следующее значение – 0.5, символ [;], конечное значение аргумента:

z : 0, 0.5; 3.14.

Комбинация клавиш [Shift + :] при вводе заменяется символом присваивания [:=], точка с запятой заменяется горизонтальным двоеточием [..].

2. Создать функцию пользователя:

$$u(z) := \cos(z) + \exp(z).$$

3. Сгенерировать ряд значений аргумента *z*. Ввести *z* =.

4. Сгенерировать ряд значений функции u. Ввести *u*(*z*) =.

После ввода знака «равно» автоматически генерируется ряд значений функции. Функция выводится с четырехразрядной мантиссой (три знака после десятичной точки). Число разрядов можно изменить командой *Формат, Формат числа*. Можно регулировать также порог мнимости, значение нуля, начальный индекс компонент массива.

Результат показан на листинге 2.8.

Контрольные вопросы

1. Каково назначение математической системы *Mathcad*?
2. Назовите функциональные возможности математической системы *Mathcad*.
3. Назовите пункты главного меню математической системы *Mathcad* и их назначение.

Листинг 2.8. **Табулирование функции**

$z := 0, 0.5 .. 3.14$

$u(z) := \cos(z) + \exp(z)$

z =	u(z) =
0	2
0.5	2.526
1	3.259
1.5	4.552
2	6.973
2.5	11.381
3	19.096

4. Какие панели инструментов относятся к семейству «Математические»?

5. Какие панели инструментов относятся к стандартным и по умолчанию активизируются при загрузке математической системы *Mathcad*?

6. Поясните порядок ввода данных и вычислений в *Mathcad*.

7. Как вводятся встроенные функции?

8. Как создать функцию пользователя?

9. Как перевести значение аргумента тригонометрической функции в радианы, если оно задано в градусах?

10. В какой форме могут отображаться числа и где можно указать формат отображения результатов расчетов?

11. Какие встроенные функции используются для работы с комплексными числами?

12. Перечислите математические константы *Mathcad*. Как они обозначаются?

13. Какие переменные *Mathcad* относятся к встроенным системным переменным и для чего они используются?

14. Как создать массив?

15. Как обозначается элемент массива?

16. Как ввести нижний индекс при обозначении элементов массивов?

2.5. РЕШЕНИЕ ТИПОВЫХ МАТЕМАТИЧЕСКИХ ЗАДАЧ

Решение алгебраических и трансцендентных уравнений

Нелинейные уравнения вида $F(x) = 0$ принято называть алгебраическими, если они содержат только алгебраические функции, и трансцендентными, если они содержат другие функции (тригонометрические, показательные, логарифмические и т. д.).

При решении нелинейных уравнений всегда возникают серьезные трудности. В аналитическом виде можно получить решение алгебраического уравнения не выше второй степени. Для решения уравнений большей степени, а также трансцендентных уравнений можно использовать графические и численные методы.

Под численным методом решения уравнений понимают метод нахождения численных значений корней уравнения с заданной точностью.

Пусть дано уравнение $f(x)= 0$, где $f(x)$ – функция, непрерывная на отрезке $[a; b]$ и дифференцируемая на интервале $]a; b[$, т. е. включая и граничные значения.

Всякое число c, принадлежащее отрезку [a; b], такое, что f(c)= 0, называется корнем уравнения на отрезке [a; b].

Или иначе: корнем выражения $f(x) = 0$ называется значение аргумента, при котором выражение обращается в тождество (функция равна нулю).

Процесс нахождения корней уравнения состоит из двух этапов:

1) отделения корней;

2) уточнения значения корней на отрезках отделения с заданной точностью h.

Отделением корней уравнения называется процесс выделения из области определения функции f(x) отрезков $[a_i; b_i]$, в каждом из которых содержится один, и только один, корень уравнения f(x) = 0.

Под уточнением значения корня с заданной точностью понимают сужение границ отрезка отделения корня $[a_i; b_i]$ до длины, не превосходящей заданной точности h.

Отделение корней уравнения

Отделение корней уравнения можно выполнить графически и аналитически.

Пример 2.5. Пусть дано уравнение $x^3 - 3x - 0{,}4 = 0$. Требуется отделить корни уравнения.

Задача может быть решена графически, аналитически, а также численными методами.

Отделение корней уравнения графически

Решение. Запишем уравнение в виде $x^3 = 3x + 0{,}4$ и построим графики функций $y = x^3$ и $y = 3x + 0{,}4$ (рис. 2.5). Абсциссы точек пересечения этих графиков и будут корнями исходного уравнения. Из рисунка следует, что исходное уравнение имеет три действительных корня:

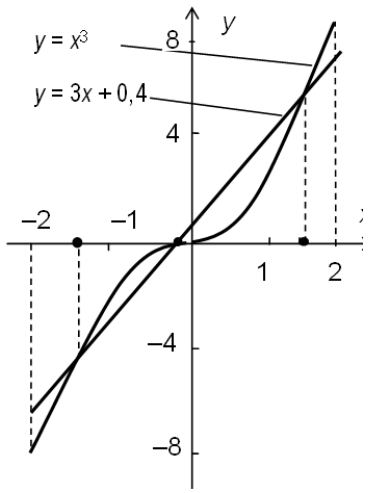


Рис. 2.5. Отделение корней уравнения

с1 – на отрезке [–2; –1];

с2 – на отрезке [–1; 0];

с3 – на отрезке [1; 2].

При графическом отделении корней уравнения результат зависит от точности построения графиков.

Отделение корней уравнения аналитически

В основе аналитического метода лежит утверждение: если функция $f(x)$ непрерывна на отрезке $[a; b]$ и принимает на концах этого отрезка значения противоположных знаков, то внутри отрезка существует корень уравнения и при том единственный, если производная $f'(x)$ на интервале $]a; b[$ сохраняет постоянный знак.

Границами интервалов знакопостоянства производной от функции являются ее критические точки, то есть минимумы и максимумы.

В критических точках производная обращается в ноль. На каждом таком интервале будет находиться только один корень, если значения функции на концах интервала имеют противоположные знаки. В качестве границ интервалов знакопостоянства функции кроме критических точек необходимо рассматривать также и границы отрезка.

Для отделения корней используется следующая схема:

1. Найти производную $f'(x)$:

$$f'(x) = (x^3 - 3x - 0.4)' = 3x^2 - 3 = 3(x^2 - 1) = 3(x + 1)(x - 1).$$

2. Найти критические точки функции. Таких точек две: –1 и +1.

3. Составить таблицу знаков функции $f(x)$ на границах отрезка $[a; b]$ и в критических точках:

x	–2	–1	+1	+2
$f(x)$	–	+	–	+

4. Определить отрезки, на концах которых функция принимает значения противоположных знаков.

Уравнение имеет три корня, так как происходит три перемены знака функции $f(x)$: с1 [–2; –1]; с2 [–1; 1]; с3 [1; 2].

5. Корень находится в интервалах, определяемых пунктом 4. При решении задач необходимо проверять, не являются ли корнями также и критические точки.

Для анализа функции дополнительно можно использовать *свойства второй производной*. В точках минимума функции вторая производная положительная, в точках максимума функции – отрицательная, а в точках перегиба – равна нулю.

Уточнение значения корня

Уточнение значений корней на отрезках отделения осуществляется различными методами одномерной поисковой оптимизации: деления отрезка пополам, простых итераций, касательных и др.

При выборе метода следует исходить из:

а) сложности подготовки задачи к решению;

б) быстродействия алгоритма;

в) времени и точности решения задачи.

Эти методы будут рассмотрены в разделе 3.

Решение алгебраических уравнений в Mathcad

Отделение корней уравнения в Mathcad также можно выполнить графически (см. раздел 2.6).

В некоторых случаях удается получить решение алгебраических уравнений с помощью команды ***Символика, Переменная, Решить***.

Например, при решении уравнения из примера 2.5 данным способом получим следующие результаты:

$$x^3 - 3\cdot x - 0.4 \qquad \begin{pmatrix} -1.6610819035405872883 \\ -.13413784570453647251 \\ 1.7952197492451237608 \end{pmatrix}$$

Пример 2.6. Решить уравнение $y = ax^2 + bx + c$ при $a = 0{,}5$, $b = -5$, $c = 1{,}74$.

Решение. Присвоить значения переменным a, b, c.

Ввести выражение $ax^2 + bx + c$ и введите команду ***Символика, Переменная, Решить***.

На экране появится промежуточный результат в символьном виде. Для получения численного значения следует выделить полученное выражение и ввести команду [=]. Результат приведен на листинге 2.9.

Листинг 2.9. **Решение уравнения**

$$a := 0.5 \qquad b := -5 \qquad c := 1.74$$

$$a\cdot x^2 + b\cdot x + c \qquad \begin{bmatrix} \frac{1}{2\cdot a}\cdot\left[-b + \left(b^2 - 4\cdot a\cdot c\right)^{\frac{1}{2}}\right] \\ \frac{1}{2\cdot a}\cdot\left[-b - \left(b^2 - 4\cdot a\cdot c\right)^{\frac{1}{2}}\right] \end{bmatrix} = \begin{pmatrix} 9.639 \\ 0.361 \end{pmatrix}$$

При решении трансцендентных уравнений данным способом можно найти только один корень уравнения.

Пример 2.7. Найти корни уравнения $\frac{\sin(2x)}{\cos x} + \ln(5x) - 2 = 0$.

Решение. Записать выражение. Выделить аргумент x и ввести команду ***Символика, Переменная, Решить***. Получим следующий результат:

$$\frac{\sin(2x)}{\cos(x)} + \ln(5x) - 2 \qquad .5089710551409455743$$

Уточнение значения корней уравнений в СКМ Mathcad можно выполнить с помощью функции ***root***, а также с помощью блоков ***Given/Find*** или ***Given/Minerr***.

Функция root имеет следующий синтаксис: ***root(f(x),x,a,b)***.

В функции ***root(f(x),x,a,b)*** a и b – границы отрезка отделения корня. Границы отрезка можно не указывать, то есть достаточно записать ***root(f(x),x)***. Однако в обоих случаях перед применением функции root необходимо указать начальное значение аргумента ***x***.

Функция ***root*** реализует метод секущих. Точность поиска задается системной константой TOL, значение которой по умолчанию равно 0.001. Можно изменить требуемую точность вычислений. Введите команду ***Математика/Параметры/Переменные*** и установите значение параметра ***Допуск сходимости***.

Пример 2.8. Найти корни уравнения $\cos(x) + e^x - 1{,}5 = 0$ на отрезке $[-6; 5]$.

Решение. Отделим корни уравнения графически (рис. 2.6).

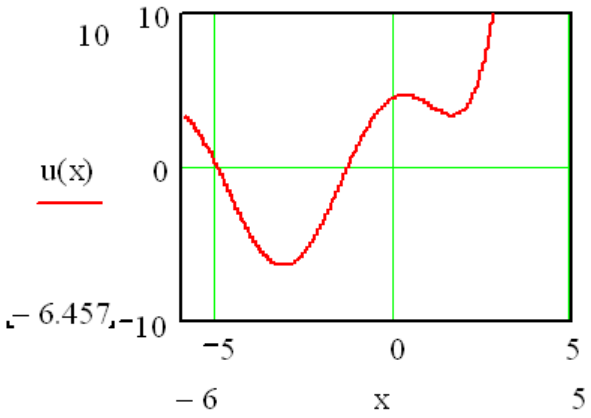


Рис. 2.6. Отделение корней

Функция имеет два корня на заданном отрезке. Первый корень находится между левой границей отрезка исследования функции и локальным минимумом. Второй отрезок находится между точками первого локального минимума и локального максимума.

Уточнение значения корней.

В качестве начального приближения корня при поиске решения с помощью функции root следует выбрать любую точку внутри отрезка отделения корня.

В качестве начального приближения для первого корня примем $x = -6$, а для второго корня укажем начальное приближение $x = 0$

и область определения [–2; 0]. Результат вычисления приведен на листинге 2.10.

Листинг 2.10. **Поиск корней уравнения с помощью функции *root***

$u(x) := 5 \cdot \cos(x) + e^{x} - 1.5$

$x := -5$ $\quad root(u(x), x) = -5.016$

$x := 0$ $\quad root(u(x), x, -2, 0) = -1.322$

Пример поиска корня с помощью блоков Given/Find и Given/Minerr приведен на листинге 2.11.

Листинг 2.11. **Поиск корней уравнения с помощью блока *Given/find* и *Given/minerr***

$x := -5$

Given

$5 \cdot \cos(x) + e^{x} - 1.5 = 0$

$find(x) = -5.016$

$x := -5$

Given

$x^2 - 3x + 2.5 = 0$

$minerr(x) = 1.5$

$ERR = 0.062$

Внимание! На листинге 2.11 в выражении стоит не знак «равно», а знак логического равенства, вводимый с панели инструментов Булево или комбинацией клавиш [Ctrl + =].

Функция ***Minerr*** в блоке ***Given*** применяется в том случае, когда уравнение не имеет корня. Она возвращает значение аргумента, при котором функция имеет минимальное расстояние от оси x. Значение ошибки (невязки) возвращает функция ERR.

Решение полиномиальных уравнений

Выражения вида $a_0x^{n-1} + a_1x^{n-2} + \ldots + a_{n-i}n^i + \ldots + a_{n-2}x^2 + a_{n-1}x + a_n$ называются полиномиальными.

Полиномиальные уравнения решаются в Mathcad с помощью функции ***polyroots(v)***, где ***v*** – вектор-столбец коэффициентов многочлена, расположенных в порядке возрастания степеней.

Пример 2.9. Решить уравнение $5x^2 - 2x - 7 = 0$.

Решение. Введите функцию **polyroots(▪)**. Установите курсор ввода в слот, щелкните мышью в панели Математика по пиктограмме $\begin{bmatrix} \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet \end{bmatrix}$, в окне диалога матрицы щелкните мышью по

пиктограмме матрицы и в открывшемся окне диалога укажите число строк – 3 и число столбцов – 1, введите в слоты значения коэффициентов, начиная со свободного члена. Нажмите клавишу «равно» – ответ готов:

$$polyroots\begin{pmatrix} -7 \\ -2 \\ 5 \end{pmatrix} = \begin{pmatrix} -1 \\ 1.4 \end{pmatrix}$$

Пример 2.10. Решить полиномиальное уравнение $2x^5 - 4x^4 + 3x^2 - 2x - 7 = 0$.

Решение. Порядок решения аналогичен порядку, рассмотренному в предыдущем примере, но вектор коэффициентов объявлен предварительно как строка. Для преобразования вектора-строки в вектор-столбец он транспонирован:

$$v := (-7 \quad -2 \quad 3 \quad 0 \quad -4 \quad 2)^T$$

$$polyroots(v) = \begin{pmatrix} -0.837 + 0.52i \\ -0.837 - 0.52i \\ 0.849 + 1.05i \\ 0.849 - 1.05i \\ 1.975 \end{pmatrix}$$

Для функции polyroots можно изменить расчетный метод с помощью контекстного меню. Методы записаны в верхней части контекстного меню функции.

Поиск экстремумов функций

Поиск локального экстремума осуществляется с помощью функций ***Minimize(f,x₁,..,xₘ)*** – поиск минимума и ***Maximize(f,x₁,..,xₘ)*** – поиск максимума, где ***f*** – имя функции, а ***x₁, …, xₙ*** – аргументы. Поиск экстремумов функции можно выполнить также с помощью блока ***Given***. Блок ***Given*** позволяет указать область поиска локального экстремума. Для поиска локального экстремума необ ходимо, как и при поиске корня уравнения, задать начальное значение аргумента. Функции ***Minimize*** и ***Maximize*** возвращают значение аргумента, при котором функция имеет экстремум. Для определения значения минимума или максимума функции необходимо найденное значение аргумента подставить в функцию.

Пример 2.11. Найти локальные минимум и максимум функции $f(t) = t^3 + 5\cos(t - 0{,}25)^2 - 4$.

$f(t) := t^3 + 5\cos(t - 0.25)^2 - 4$

$t := 2$

Minimize $(f, t) = 1.238$ $\quad f(1.238) = -0.588$

Maximize $(f, t) = 0.272$ $\quad f(0.272) = 1.018$

График функции показан на рисунке 2.7.

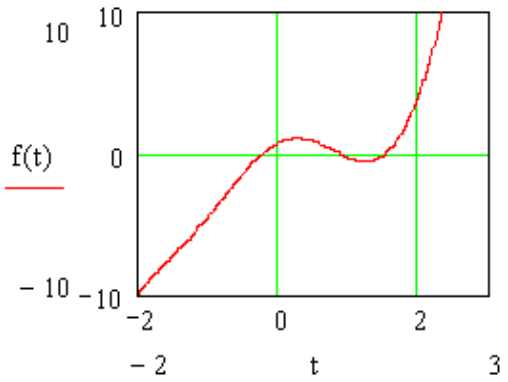


Рис. 2.7. График функции

В этих примерах важно правильно выбрать начальное приближение.

Для поиска экстремумов функции в заданной области следует использовать блок Given в сочетании с функциями Maximize и Minimize.

Пример 2.12. Найти условный экстремум функции с помощью блока Given в условиях примера 2.11.

$t := -1$ $\qquad$ $t := -1$

Given $\qquad$ Given

$0 < t < 2$ $\qquad$ $0 < t < 2$

Minimize$(f, t) = 1.238$ $\qquad$ Maximize $(f, t) = 0.272$

Контрольные вопросы

1. Что называется корнем уравнения?
2. Что значит отделить корни уравнения? Как это осуществляется в Mathcad?
3. Что значит уточнить значение корня на отрезке отделения, каким образом уточняется значение корней уравнения?
4. Каково назначение функции root? Приведите синтаксис функции и поясните назначение ее аргументов.
5. Для чего предназначена функция Find? Приведите синтаксис функции и поясните назначение ее аргументов.
6. Как найти корни полиномиального уравнения?
7. Как найти экстремумы функции?
8. Поясните назначение блоков Given/Find, Given/Minerr, Given/Maximize, Given/Minimize.

2.6. ГРАФИЧЕСКИЕ ВОЗМОЖНОСТИ MATHCAD

Система компьютерной математики Mathcad имеет достаточно развитые возможности для построения графиков функций и диаграмм. Для работы с графиками служит команда ***Формат, График***, а также панель инструментов ***График***. Mathcad позволяет строить семь типов графиков (рис. 2.8):

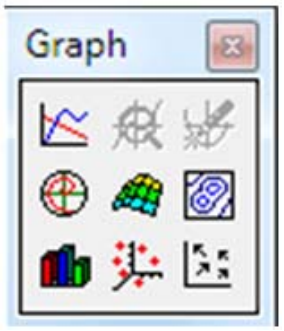


Рис. 2.8. Панель инструментов График

график в декартовой системе координат;
график в полярной системе координат;
трехмерный график;

контурный график поверхностей уровня;
трехмерная столбиковая гистограмма;
трехмерный точечный график;
график векторного поля на плоскости.

Для построения графика необходимо определить функцию и ее аргументы в некотором диапазоне. В Mathcad используется только линейная интерполяция, поэтому для получения плавных кривых шаг табулирования функций должен быть достаточно малым.

Построение графиков функций в декартовой системе координат

Режим построения графика в декартовой системе координат вызывается щелчком мыши по соответствующему рисунку в панели инструментов либо выбором команды из меню. После ввода команды на экране появляется шаблон графика с двумя слотами (рис. 2.9).

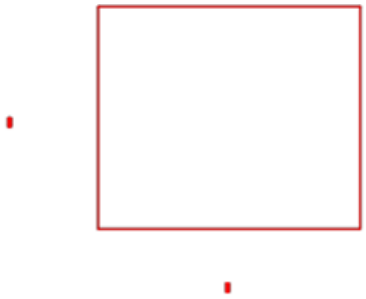


Рис. 2.9. Шаблон для построения графика

В левом слоте следует записать функцию, а в нижнем – аргумент и щелкнуть мышью по полю вне бокса (рис. 2.10).

Если необходимо построить на одном шаблоне несколько графиков, то их следует перечислить через запятую. После ввода запятой курсор ввода переходит на следующую строку. Если функции зависят от разных аргументов, то их также необходимо перечислить в нижнем слоте через запятую. При построении графика программа по умолчанию присваивает значения аргументов в диапазоне от –10 до +10. При необходимости изменить область просмотра графика следует ввести нужные значения в нижних слотах графика. Аналогично можно редактировать и значения функции по оси Y, изменяя значения в левых слотах графика.

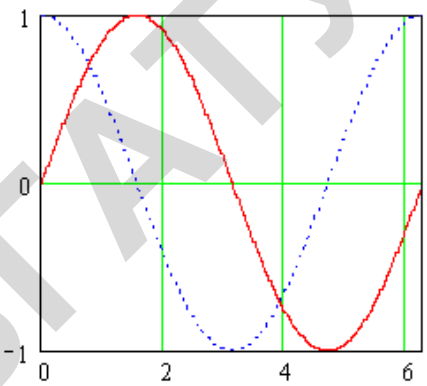


Рис. 2.10. График функции

Пример 2.13. Протабулировать функции трехфазного тока и построить их графики на отрезке [0; 2π] (рис. 2.11). В данном примере для построения графиков будет использована ранжированная переменная.

$a := 0 \quad b := 2\pi \quad n := 20 \quad i := 1..n \quad h := \frac{b-a}{n}$

$x_i := a + i \cdot h \quad y1_i := \sin(x_i) \quad y2_i := \sin\left(x_i + 2\frac{\pi}{3}\right) \quad y3_i := \sin\left(x_i + 4\frac{\pi}{3}\right)$

$x =$

	0
0	0
1	0.314
2	0.628
3	0.942
4	1.257
5	1.571
6	1.885
7	2.199
8	2.513
9	2.827
10	3.142
11	3.456
12	3.77
13	4.084
14	4.398
15	4.712

$y1 =$

	0
0	0
1	0.309
2	0.588
3	0.809
4	0.951
5	1
6	0.951
7	0.809
8	0.588
9	0.309
10	0
11	-0.309
12	-0.588
13	-0.809
14	-0.951
15	-1

$y2 =$

	0
0	0
1	0.669
2	0.407
3	0.105
4	-0.208
5	-0.5
6	-0.743
7	-0.914
8	-0.995
9	-0.978
10	-0.866
11	-0.669
12	-0.407
13	-0.105
14	0.208
15	0.5

$y3 =$

	0
0	0
1	-0.978
2	-0.995
3	-0.914
4	-0.743
5	-0.5
6	-0.208
7	0.105
8	0.407
9	0.669
10	0.866
11	0.978
12	0.995
13	0.914
14	0.743
15	0.5

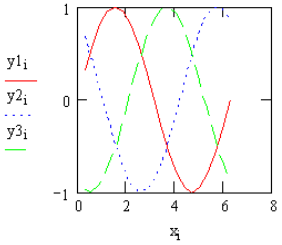


Рис. 2.11. Графики трехфазного тока

Решение:

- Определите начало и конец отрезка табулирования функции a и b, задайте число отрезков n.
- Вычислите шаг табулирования функции h.
- Задайте интервал изменения индекса i и запишите формулу для расчета текущего значения аргумента x_i. Для перехода в режим ввода индекса переменной нажмите клавишу [, для отмены режима ввода индекса нажмите клавишу *Пробел*.
- Определите функции $y1_i = \sin(x_i)$; $y2_i = \sin(x_i + 2\pi/3)$; $y3_i = \sin(x_i + 4\pi/3)$.
- Сгенерируйте ряды значений x, $y1$, $y2$, $y3$: установите точку вставки в требуемое место экрана, введите имя переменной и нажмите клавишу [=];
- Постройте графики функций, как описано выше.

При построении графиков по ранее вычисленным массивам необходимо задать диапазон изменения индексов, выбрать тип графика и указать на осях соответственно x_i, y_i. Эта задача уже похожа на программирование.

Для настройки графика введите команду ***Формат, Графики*** и выберите тип редактируемого графика или дважды щелкните по графику мышью, откроется окно диалога для изменения описания графика (рис. 2.12). Перечислим основные параметры настройки графика.

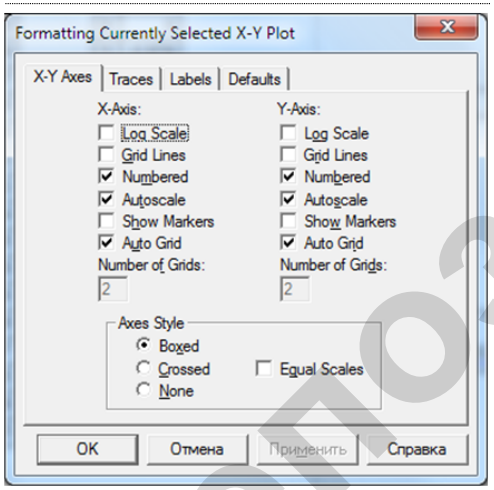


Рис. 2.12. Настройка X–Y графика

Вкладка ***X-Y Axes*** – форматирование графика:

Log Scale – логарифмическая шкала. Применяется, когда данные разнятся на несколько порядков.

Grid Lines – показать линии сетки.

Numbered – показать нумерацию шкалы, выводит числовые метки на ось.

Autoscale – выбор диапазона оси производится автоматически.

Show Markers – показать маркеры, выделение значений на осях.

Auto Grid – автоматическая шкала. Если флажок снят, то в поле ниже него необходимо указать желаемое количество меток шкалы.

Equal Scales – одинаковый масштаб по обеим осям.

Axes Style – позволяет выбрать один из трех видов координат: прямоугольная (Boxed) – график заключен в прямоугольную рамку, пересечение (Crossed) – изображаются только линии координат без рамки, нет осей (None).

Вкладка ***Traces*** – форматирование рядов данных.

Legend Label – описание меток графиков.

Symbol – позволяет выбрать символ, которым обозначаются отдельные точки данных: простая линия, линия с маркерами – пять видов. Следовательно, на черно-белом графике можно отобразить шесть различных графиков.

Line – стиль линии: сплошная, пунктирная, штрих, штрих-пунктирная.

Color – цвет линии и точек данных.

Weight – толщина линии и точек данных.

Type – тип графика: линия, точка, столбик, уступ, штрих, закрашенный столбик.

Вкладка ***Labels*** позволяет указать заголовок и название осей, а также положение заголовка: сверху или снизу.

Вкладка ***Defaults*** позволяет восстановить настройки по умолчанию.

Построение графиков, заданных параметрически

График строится в декартовой системе координат. В данном случае и y, и x являются функциями от некоторой независимой переменной t. Наиболее наглядным примером будет, очевидно, построение эллипса: $x = a\cos(\varphi)$; $y = b\sin(\varphi)$. Здесь a – длина большой полуоси эллипса, b – длина малой полуоси эллипса, φ – угол, изменяющийся от 0 до 2π. Пример построения приведен на рисунке 2.13.

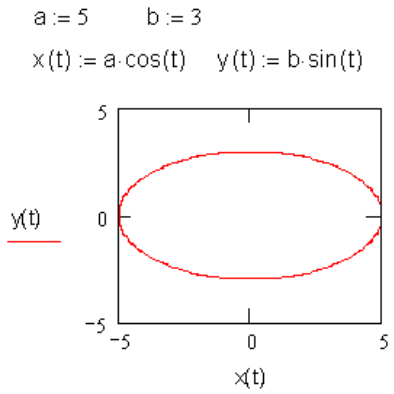


Рис. 2.13. Построение графика, заданного параметрически

Построение графика в полярных координатах

В полярной системе координат при активизации шаблона графика рабочее поле представлено окружностью. В нижнем слоте шаблона задается имя угловой переменной, в левом слоте – имя функции, определяющей радиус как функцию угла. В правой верхней части расположены два поля для задания нижнего и верхнего значений радиуса. При построении графика эти значения устанавливаются программой по умолчанию. Но их можно изменять для анализа функции.

Построение графика в полярных координатах мало отличается от построения графика в декартовых координатах:

- Объявите функцию: $r(\alpha) := 5\cos(\alpha) - 5$.
- Вызовите шаблон для построения графика в полярных координатах.
- Укажите на графике функцию и аргументы.

Пример построения графика в полярных координатах приведен на рисунке 2.14.

Построение графиков поверхности

Для построения графика поверхности необходимо:

- Объявить функцию двух переменных $z(x,y)$.
- Вызвать шаблон для построения графика. На шаблоне только один слот.
- Указать в слоте имя функции (без аргументов).

Пример 2.14. Построить график функции $F(x,y) = (x^2 + y^2)/3$ (рис. 2.15).

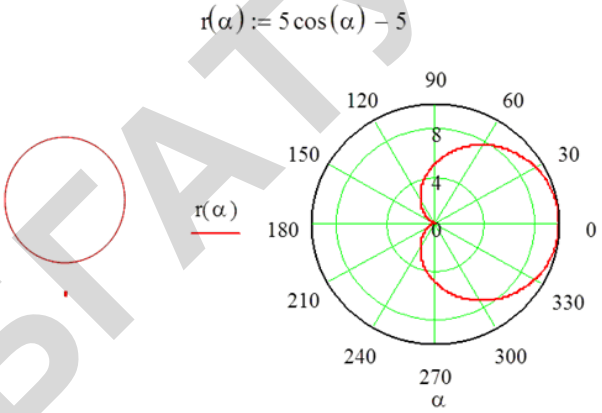


Рис. 2.14. График в полярной системе координат

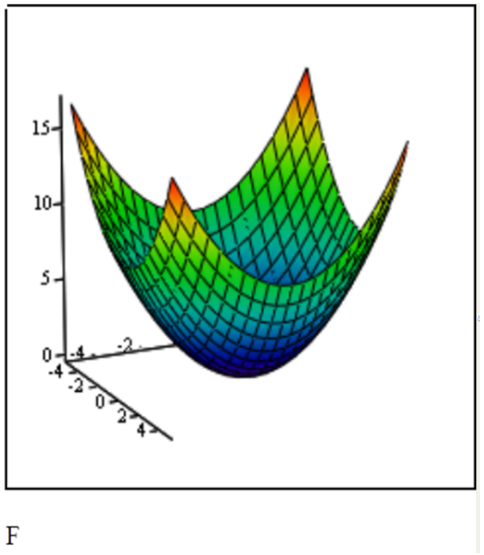


Рис. 2.15. График поверхности

Решение.

$$F(x,y) := \frac{x^2 + y^2}{3}$$

Настройка параметров трехмерного графика

Форматирование осуществляется с помощью окна диалога 3-D Plot Format, который вызывается двойным щелчком мыши по графику.

Заголовок графика определяется на вкладке Title и может быть расположен сверху или снизу графика.

Изменение типа графика. Выберите тип графика в закладке General (Общие), Display As: *поверхность* (Surface Plot), *точечная* (Data Points), *столбиковая* (Bar Plot), *контурная* (Contour Plot), *векторные поля* (Vector Field Plot), *смешанный* (Patch Plot) – и щелкните кнопку ***Применить*** или Ок.

Вращение графика. Вращать график можно с помощью левой клавиши мыши: зацепите график за нужную точку и перемещайте. Более точное вращение осуществляется с помощью опций поля View: *Rotazion* (Вращение), *Tilt* (Наклон), *Twist* (Поворот) на вкладке *Общие*.

Здесь же осуществляется ***масштабирование*** – поле Zoom.

Стиль осей. Стиль осей настраивается с помощью переключателей в группе Axes Style. Можно задать три стиля: *Perimeter* (Периметр), *Corner* (Углом), *None* (Отсутствуют).

Форматирование осей. Вкладка *Backplanes* (Плоскость заднего плана) задает показ проекции координат сетки на три скрытые плоскости трехмерного графика. Для каждой плоскости можно указать цвет (флажок *Fill Backplane* и *Color*), границы областей (*Backplane Border*) и стиль линий разметки (*Draw Lines*), меток (*Draw Ticks*) и их толщину (*Line Weight*). Вкладка ***Axes*** (Оси) задает параметры осей: толщину линий, меток, цвет, масштаб шкалы.

Стиль заливки и линий устанавливается с помощью закладки *Appearance* (Появление): параметры заливки поверхности, параметры линий и точек данных. При выборе опции *Colormap* будет иметь место сложная заливка, а *Solid Color* – однотонная заливка.

Спецэффекты создаются с помощью вкладки *Advanced*. Они позволяют красочно оформить график. Основные эффекты: *Shininess* (Сияние), *Fog* (Туман), *Transparency* (Прозрачность), *Perspective* (Перспектива). Еще один спецэффект *Lighting* (Подсветка) устанавливается с одноименной вкладки.

Примеры поверхностей:

эллипсоид $\frac{x^2}{a^2}+\frac{y^2}{b^2}+\frac{z^2}{c^2}=1;$

гиперболоид однополостной $\frac{x^2}{a^2}+\frac{y^2}{b^2}-\frac{z^2}{c^2}=1;$

гиперболоид двуполостной $\frac{x^2}{a^2}+\frac{y^2}{b^2}-\frac{z^2}{c^2}=-1;$

конус $\frac{x^2}{a^2}+\frac{y^2}{b^2}-\frac{z^2}{c^2}=0;$

параболоид $z=\frac{x^2}{a^2}-\frac{y^2}{b^2};$

эллиптический цилиндр $\frac{x^2}{a^2}+\frac{y^2}{b^2}=1;$

параболический цилиндр $y = 2px$.

Построение контурных графиков

Примером контурного графика является топографическая карта с отметками высот. В общем случае он представляет собой проекцию графика поверхности на плоскость X–Y. Поэтому принцип построения контурного графика не отличается от построения графика поверхности. Порядок построения контурного графика может быть следующим:

- Записать выражение для функции двух переменных $Z(x,y)$.
- Выбрать тип графика – контурный.
- Записать в слоте имя функции.

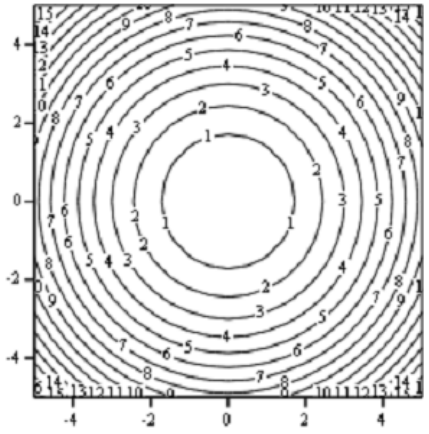
Для вывода на график числовых значений вызовите окно диалога настройки графика двойным щелчком мыши, откройте закладку ***Специальные*** и в группе ***Параметры контура*** установите флажок ***Нумерация***. Пример построения контурного графика для условий примера 2.14 приведен на рисунке 2.16.

Построение сечений функции двух переменных

Построение сечений функции двух переменных выполняется в декартовой системе координат. При этом одному из аргументов присваивается фиксированное значение. Следовательно, функция двух переменных превращается в функцию одной переменной.

Контрольные вопросы

1. Какие типы графиков позволяет строить Mathcad?
2. Как построить на одной форме несколько графиков одной переменной?
3. Как построить график функции ранжированной переменной?
4. Как настроить параметры графика?



F

Рис. 2.16. График в полярной системе координат

5. Как построить график в полярных координатах?
6. Как построить график функции двух переменных?
7. Что такое контурный график? Как его построить?
8. Как построить сечение графика функции двух переменных на плоскость X–Y?

2.7. МАТРИЧНЫЕ ВЫЧИСЛЕНИЯ

Массивом называется совокупность элементов, имеющих одно имя и отличающихся индексом.

Операции, выполняемые над векторами и матрицами, можно разбить на две большие группы.

К *первой группе* относятся операции, которые применяются к отдельным векторам и матрицам. Например, транспонирование матрицы или вычисление обратной матрицы.

Ко *второй группе* относятся операции, которые выполняются над группой векторов и матриц. Как правило, они выполняются над двумя матрицами или матрицей и вектором или скаляром. Например, сложение, вычитание матриц, перемножение матриц или умножение матрицы и вектора, умножение и деление матрицы на скаляр. К векторам и матрицам при выполнении операций над ними

могут предъявляться определенные требования в соответствии с требованиями классической математики.

Линейные операции (сложение, вычитание, умножение и деление на число, прибавление и вычитание числа) выполняются над матрицами так же, как и с простыми переменными (листинг 2.12).

***Листинг 2.12.* Операции над матрицами**

Даны матрицы A и B, а также вектор-столбец C и константа D
Выполнить операции над матрицами и векторами.

$$A := \begin{pmatrix} 1 & 3 & 2 \\ 8 & 4 & 5 \\ 11 & 3 & 6 \end{pmatrix} \quad B := \begin{pmatrix} 2 & 6 & 1 \\ 7 & 3 & 2 \\ 5 & 4 & 8 \end{pmatrix} \quad C := \begin{pmatrix} 0.2 \\ 1.5 \\ 2.7 \end{pmatrix} \quad D := 2.5$$

$$A1 := A + B \quad A2 := A \cdot B \quad A3 := A \cdot C \quad A4 := B + D$$

$$A1 = \begin{pmatrix} 3 & 9 & 3 \\ 15 & 7 & 7 \\ 16 & 7 & 14 \end{pmatrix} \quad A2 = \begin{pmatrix} 33 & 23 & 23 \\ 69 & 80 & 56 \\ 73 & 99 & 65 \end{pmatrix} \quad A3 = \begin{pmatrix} 10.1 \\ 21.1 \\ 22.9 \end{pmatrix} \quad A4 = \begin{pmatrix} 4.5 & 8.5 & 3.5 \\ 9.5 & 5.5 & 4.5 \\ 7.5 & 6.5 & 10.5 \end{pmatrix}$$

Простейшие матричные операции выполняются с помощью операторов панели ***Матрица***: вычисление обратной матрицы, транспонирование матриц, вычисление определителя квадратной матрицы или модуля вектора, скалярное и векторное произведения векторов, сумма элементов вектора (табл. 2.2).

Таблица 2.2

Операции над матрицами и векторами

Операция	Обозначение
Обращение матрицы	M^{-1}
Возведение матрицы в степень	M^n
Определитель матрицы	$\|M\|$
Транспонирование матрицы	M^T
Выделение n-го столбца матрицы	$M^{<n>}$
Выделение i,j-го элемента матрицы	$M_{i,j}$

Примеры вычислений приведены на листинге 2.13.

Mathcad имеет более 50 функций, предназначенных для работы с векторами и матрицами. Все функции можно разбить на группы по их функциональному назначению. Например, функции, предназначенные для создания матриц общего и специаль-

ного вида, редактирования и преобразования матриц, функции, определяющие параметры матриц, и т. д. Рассмотрим некоторые из них.

Листинг 2.13. **Операции над матрицами и векторами**

Исходная матрица

$$B := \begin{pmatrix} 2 & 6 & 1 \\ 7 & 3 & 2 \\ 5 & 4 & 8 \end{pmatrix}$$

Обратная матрица

$$B1 := B^{-1}$$

$$B1 = \begin{pmatrix} -0.069 & 0.19 & -0.039 \\ 0.199 & -0.048 & -0.013 \\ -0.056 & -0.095 & 0.156 \end{pmatrix}$$

Транспонирование матрицы

$$B2 := B^{T}$$

$$B2 = \begin{pmatrix} 2 & 7 & 5 \\ 6 & 3 & 4 \\ 1 & 2 & 8 \end{pmatrix}$$

Выделение столбца

$$C1 := B^{\langle 1 \rangle}$$

$$C1 = \begin{pmatrix} 6 \\ 3 \\ 4 \end{pmatrix}$$

Векторизация, почленное умножение элементов векторов

$$C := \begin{pmatrix} 0.2 \\ 1.5 \\ 2.7 \end{pmatrix}$$

$$C3 := C \cdot C1$$

$$C3 := \overrightarrow{(C \cdot C1)}$$

$$C3 = \begin{pmatrix} 1.2 \\ 4.5 \\ 10.8 \end{pmatrix}$$

Определитель матрицы

$$|B| = -231$$

Скалярное произведение векторов

$$C \cdot C1 = 16.5$$

Векторное произведение векторов

$$C \times C1 = \begin{pmatrix} -2.1 \\ 15.4 \\ -8.4 \end{pmatrix}$$

Сумма элементов вектора

$$\sum C1 = 13$$

matrix(M,N,f) – определение элементов матрицы через функцию, здесь *f* – функция.

Пример 2.15. Создание матрицы размером *m*×*n*, каждый элемент которой является функцией от индексов *i* и *j*, при *m* = 2 и *n* = 3:

$$f(i, j) := i + 0.5 \cdot j \qquad A := \text{matrix}(2, 3, f) \qquad A = \begin{pmatrix} 0 & 0.5 & 1 \\ 1 & 1.5 & 2 \end{pmatrix}$$

identity(n) – создает единичную матрицу порядка n.

Следующие три функции позволяют создавать новые матрицы из существующих матриц, объединяя их по строкам и столбцам:

augment(A, B) – формирует матрицу, где в первых столбцах располагается элементы матрицы *A*, а в последующих – *B*;

stack(A, B) – формирует матрицу, где в первых строках располагается элементы матрицы *A*, а в последующих – *B*;

submatrix(A, ir, jr, ic, jc) – формирует матрицу, которая является блоком матрицы *A*, расположенным в строках от *ir* до *jr* и в столбцах от *ic* до *jc*;

length(v) – вычисляет количество элементов в векторе *v*;

rows(A) – вычисляет количество строк в матрице *A*;

cols(A) – вычисляет количество столбцов в матрице *A*;

max(A) – вычисляет максимальный элемент в матрице *A*;

min(A) – вычисляет минимальный элемент в матрице *A*;

mean(A) – вычисляет среднее значение элементов в матрице *A*;

tr(A) – след матрицы (сумма диагональных элементов);

diag(V) – получение диагональной матрицы, на диагонали которой находятся элементы вектора *V*;

rref(A) – приведение матрицы к ступенчатому виду с единичным базисным минором («прямой» и «обратный» ход метода Гаусса);

norm1(A), ***norm2(A)***, ***norme(A)***, ***normi(A)*** – вычисление норм матрицы *A* при различных пространствах (L1, L2, евклидова норма, max-норма, или infinity norm). Норма матрицы отражает порядок величины матричных элементов.

Для выражения меры близости в виде числа используется какая-либо норма вектора, например, евклидова норма или длина вектора в n-мерном пространстве (другое определение – это корень квадратный из скалярного произведения вектора на себя):

$$\|\Delta\| = \sqrt{\Delta_1^2 + \Delta_2^2 + \ldots + \Delta_n^2} = \sqrt{((\Delta),(\Delta))}.$$

Иногда используются другие векторные нормы: норма-максимум (равна наибольшей по абсолютной величине компоненте вектора) или норма-сумма (равна сумме абсолютных величин компонентов вектора):

$$\|\Delta\|_\infty = \max|\Delta_i|;$$

$$\|\Delta\|_1 = \sum_{i=1}^{n} \max|\Delta_i|;$$

cond1(A), ***cond2(A)***, ***conde(A)***, ***condi(A)*** – определение числа обусловленности матрицы *A* при различных базисах.

Число обусловленности матрицы является мерой чувствительности системы линейных уравнений $Ax = b$, определяемых матрицей *A*, к погрешностям задания вектора свободных членов *b* для соответствующей нормы. При небольших изменениях значений коэффициентов погрешность определения значений переменных оказывается существенно больше, чем погрешность коэффициента.

Задачи, в которых малое изменение исходных параметров кардинально сказывается на результате, называются плохо обусловленными.

Величина $\|A\|\|A^{-1}\|$ называется числом (мерой) обусловленности матрицы A. От этой величины зависит степень влияния погрешности коэффициентов системы уравнений на погрешность полученного решения. Если это число невелико, то относительная погрешность решения будет не сильно отличаться от относительной погрешности коэффициентов. Чем больше число обусловленности, тем больше будет влияние погрешности коэффициентов на погрешность решения;

rank(A) – вычисляет ранг матрицы A;

eigenvals(A) – вычисляет собственные значения квадратной матрицы A;

eigenvecs(A) – вычисляет собственные вектора квадратной матрицы A;

eigenvec(A, λ) – вычисляет собственный вектор квадратной матрицы A, отвечающий собственному значению λ;

lsolve(A, b) – решение системы линейных алгебраических уравнений вида $Ax = b$.

Ранг матрицы – наибольшее натуральное число k, для которого существует не равный нулю определитель k-го порядка подматрицы, составленный из любого пересечения k столбцов и k строк матрицы (листинг 2.14).

Листинг 2.14. **Ранг матрицы**

$$A := \begin{pmatrix} 1 & 4 & 5 \\ 5 & 8 & 7 \\ 4 & 7 & 9 \end{pmatrix} \qquad B := \begin{pmatrix} 1 & 3 & 5 \\ 2 & 6 & 10 \\ 3 & 7 & 9 \end{pmatrix}$$

$$\text{rank}(A) = 3 \qquad \text{rank}(B) = 2$$

В матрице B две строки эквивалентны, поэтому независимыми являются только две строки, и, следовательно, ранг матрицы B равен двум.

Собственные значения

Пусть имеется выражение $Ax = \lambda x$. Всякий вектор $\vec{x}$, не равный нулю, называется собственным вектором данного выражения, если он обращает данное выражение в тождество, а параметр λ называется

собственным значением данного линейного преобразования. Уравнение может иметь не одно, а несколько собственных значений, которым соответствуют собственные векторы $\vec{x}$.

Собственные векторы находят из уравнения:

$$Ax - \lambda E = 0,$$

где E – единичная матрица.

Уравнение

$$A \cdot x - \lambda \cdot E = 0 \qquad \begin{pmatrix} A_{0,0} - \lambda & A_{0,1} \\ A_{1,0} & A_{1,1} - \lambda \end{pmatrix} \cdot x = 0$$

называется ***характеристическим*** уравнением (листинг 2.15). Ненулевое решение данной системы существует тогда и только тогда, когда определитель системы равен нулю. Решая данное уравнение, находим собственный вектор уравнения.

Листинг 2.15. **Пример решения характеристического уравнения**

$$a_{0,0} := 2 \qquad a_{0,1} := 4 \qquad a_{1,0} := 5 \qquad a_{1,1} := 3$$

$$\lambda^2 - \left(a_{0,0} + a_{1,1}\right) \cdot \lambda + \left(a_{0,0} \cdot a_{1,1} - a_{1,0} \cdot a_{0,1}\right) \rightarrow \lambda^2 - 5 \cdot \lambda - 14$$

Для решения полученного характеристического уравнения выделим переменную в выражении $\lambda^2 - 5\lambda - 14$ и введем команду ***Символика, Переменные, Решить***.

При решении данного характеристического уравнения получаем вектор собственных значений λ, которому соответствуют два собственных вектора:

$$\lambda^2 - 5 \cdot \lambda - 14 \qquad \begin{pmatrix} -2 \\ 7 \end{pmatrix} \qquad \text{eigenvecs}(a) = \begin{pmatrix} 0.707 & 0.625 \\ -0.707 & 0.781 \end{pmatrix}$$

Сортировка матриц

Сортировка матриц осуществляется функциями ***sort(v)***, ***reverse(v)***, ***csort(A,n)***, ***rsort(A,n)***.

Функция ***sort(v)*** сортирует матрицу по возрастанию значения аргумента, функция ***reverse(v)*** – по убыванию. Функция ***csort(A,n)*** сортирует матрицу по столбцу, а функция ***rsort(A,n)*** – по строке.

Решение систем линейных алгебраических уравнений (СЛАУ)

Пусть дана система линейных алгебраических уравнений:

$$\begin{cases} 6x - 4y + 7z = 3, \\ -6x + 8y + z = 8, \\ 2x + 5y - 4z = -3. \end{cases}$$

Представим ее в матричном виде: $\boldsymbol{Ax = b}$, где A – матрица коэффициентов при неизвестных, b – вектор свободных членов. A и b должны быть предварительно сформированы как матрица и как вектор-столбец:

$$A := \begin{pmatrix} 6 & -4 & 7 \\ -6 & 8 & 1 \\ 2 & 5 & -4 \end{pmatrix} \qquad b := \begin{pmatrix} 3 \\ 8 \\ -3 \end{pmatrix}$$

Решение систем линейных алгебраических уравнений в Mathcad может осуществляться несколькими способами:

с помощью обратной матрицы $\boldsymbol{x = A^{-1}b}$. Здесь A^{-1} – обратная матрица;

с помощью функции ***lsolve(A,b)***;

с помощью блока ***Given/Find***;

методом ***Крамера;***

методом ***Гаусса***.

Примеры решений приведены на листинге 2.16.

Листинг 2.16. **Решение СЛАУ матричным методом и с помощью функции *lsolve***

$$A := \begin{pmatrix} 6 & -4 & 7 \\ -6 & 8 & 1 \\ 2 & 5 & -4 \end{pmatrix} \qquad b := \begin{pmatrix} 3 \\ 8 \\ -3 \end{pmatrix}$$

$$\begin{pmatrix} x_0 \\ x_1 \\ x_2 \end{pmatrix} := A^{-1} b \qquad \begin{pmatrix} x_0 \\ x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} -0.485 \\ 0.496 \\ 1.127 \end{pmatrix} \qquad x := \mathrm{lsolve}(A, b) \qquad x = \begin{pmatrix} -0.485 \\ 0.496 \\ 1.127 \end{pmatrix}$$

При использовании блока ***Given/Find*** достигается наглядность представления хода решения. Однако в этом случае неизвестным необходимо присвоить начальные значения. Пример решения СЛАУ с помощью блока Given/Find приведен на листинге 2.17.

Листинг 2.17. **Использование блока Given/Find для решения СЛАУ**

$x_0 := 0 \quad x_1 := 0 \quad x_2 := 0$

Given

$6x_0 - 4x_1 + 7x_2 = 3$

$-6x_0 + 8x_1 + x_2 = 8$

$2x_0 + 5x_1 - 4x_2 = -3$

$x := \mathrm{Find}(x_0, x_1, x_2)$

$$x = \begin{pmatrix} -0.485 \\ 0.496 \\ 1.127 \end{pmatrix}$$

Определим точность решения СЛАУ данным методом.

Подставим значения вектора x в систему уравнений и вычислим вектор $b1$. Разница между значением полученного вектора $b1$ и исходным вектором b позволит получить ошибку вычисления (вектор невязки) системы уравнений.

$$b1 := A \cdot x \qquad D := b1 - b \qquad D = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

В данном случае вычисления выполнены с высокой точностью и вектор невязки равен 0.

Число обусловленности матрицы также позволяет оценить влияние точности исходных данных (коэффициентов при неизвестных) на результаты вычислений. Произведение матрицы A на обратную матрицу A^{-1} дает единичную матрицу: $AA^{-1} = E$, cond1 = 4.921.

Внесем погрешность в матрицу коэффициентов и оценим результат. Изменим коэффициент $A_{1,1}$ в матрице A на 8.01 и вычислим невязку и обусловленность матрицы (листинг 2.18).

Cond1 = 4.925. Результаты существенно не изменились, следовательно, матрица хорошо обусловлена.

Алгоритм решения СЛАУ методом Гаусса

Решение СЛАУ методом Гаусса осуществляется с помощью функции ***rref***. Для решения задачи данным методом необходимо

составит расширенную матрицу, то есть дополнить матрицу коэффициентов матрицей свободных членов, и применить к данной матрице функцию ***rref.*** Результат выводится на месте вектора свободных членов:

$$A := \begin{pmatrix} 6 & -4 & 7 & 3 \\ -6 & 8 & 1 & 8 \\ 2 & 5 & -4 & -3 \end{pmatrix} \qquad \mathrm{rref}(A) = \begin{pmatrix} 1 & 0 & 0 & -0.485 \\ 0 & 1 & 0 & 0.496 \\ 0 & 0 & 1 & 1.127 \end{pmatrix}$$

Листинг 2.18. **Исследование влияния точности задания матрицы коэффициентов на результаты**

$$A1 := \begin{pmatrix} 6 & -4 & 7 \\ -6 & 8.01 & 1 \\ 2 & 5 & -4 \end{pmatrix} \qquad x := A^{-1} b \qquad b1 := A1 \cdot x \qquad b1 - b = \begin{pmatrix} 0 \\ 4.956 \times 10^{-3} \\ 0 \end{pmatrix}$$

Решения СЛАУ другими методами более громоздки и могут применяться только в учебных целях для закрепления материала по курсу математика.

Алгоритм решения СЛАУ методом Крамера

Для решения СЛАУ методом Крамера необходимо выполнить следующие операции:

сформировать дополнительные матрицы, по числу неизвестных, заменяя вектор при неизвестных на вектор свободных членов;

вычислить главный определитель и дополнительные определители матриц;

вычислить неизвестные путем деления определителей дополнительных матриц на главный определитель.

Пример решения системы уравнений методом Крамера приведен на листинге 2.19.

В первой строке программы Системной переменной ORIGIN присвоено значение 1, теперь нумерация строк и столбцов матрицы будет начинаться с единицы.

Во второй строке программы введены матрица *A* и вектор *B*, вычислен определитель матрицы.

В третьей строке сформирован дополнительный определитель: сначала создается дополнительный массив *A*1, в этом массиве первый столбец заменяется на вектор свободных членов и вычисляется дополнительный определитель. Аналогично вычисляются второй и третий дополнительный определители.

Затем вычисляются значения элементов массива.

Листинг 2.19. **Решение системы линейных алгебраических уравнений методом Крамера**

$$\mathrm{ORIGIN} := 1$$

$$A := \begin{pmatrix} 2 & 3 & 6 \\ 6 & 7 & 3 \\ 5 & 8 & 4 \end{pmatrix} \qquad B := \begin{pmatrix} -3 \\ 2 \\ 5 \end{pmatrix} \qquad \Delta := |A|$$

$$A1 := A \quad A1^{\langle 1 \rangle} := B \qquad A1 = \begin{pmatrix} -3 & 3 & 6 \\ 2 & 7 & 3 \\ 5 & 8 & 4 \end{pmatrix} \qquad \Delta 1 := |A1|$$

$$A2 := A \quad A2^{\langle 2 \rangle} := B \qquad A2 = \begin{pmatrix} 2 & -3 & 6 \\ 6 & 2 & 3 \\ 5 & 5 & 4 \end{pmatrix} \qquad \Delta 2 := |A2|$$

$$A3 := A \quad A3^{\langle 3 \rangle} := B \qquad A3 = \begin{pmatrix} 2 & 3 & -3 \\ 6 & 7 & 2 \\ 5 & 8 & 5 \end{pmatrix} \qquad \Delta 3 := |A3|$$

$$x_1 := \frac{\Delta 1}{\Delta} \qquad x_2 := \frac{\Delta 2}{\Delta} \qquad x_3 := \frac{\Delta 3}{\Delta} \qquad x = \begin{pmatrix} -1.78 \\ 2.254 \\ -1.034 \end{pmatrix} \qquad \mathrm{lsolve}\ (A, B) = \begin{pmatrix} -1.78 \\ 2.254 \\ -1.034 \end{pmatrix}$$

$$\text{Невязка} := A \cdot x - B \qquad \text{Невязка} = \begin{pmatrix} 0 \\ 6.217 \times 10^{-15} \\ 4.441 \times 10^{-15} \end{pmatrix}$$

Результаты вычислений методом Крамера сравнены с результатом вычисления с помощью функции *lsolve*, а также вычислена ошибка вычислений неизвестных – невязка – как разница между произведением матрицы *A* на вектор *X* и вектором свободных членов. Эта разница практически равна нулю.

Решение систем нелинейных уравнений

Для решения систем нелинейных уравнений используются ключевое слово *Given* и функция *Find*. Функция Find использует градиентные методы поиска решения. Переменным присваиваются начальные значения, уравнения записываются, как обычно принято в математических системах. Для вставки знака [=] используется комбинация клавиш [*Ctrl* + =]. После записи выражения Find(*x*,*y*) следует нажать клавишу [=]. В примере на листинге 2.20 результат присваивается переменной v, что позволяет осуществить проверку.

При решении сложных систем нелинейных уравнений может случиться, что программа откажется от решения какой-либо из сис-

тем. Это означает лишь, что лежащий в ее основе алгоритм недостаточно универсальный. В этом случае можно рекомендовать алгоритмы из монографии Деннис, Дж. Численные методы безусловной оптимизации и решения нелинейных уравнений. Москва, 1988.

***Листинг 2.20*. Решение системы нелинейных уравнений**

$x := 1 \qquad y := 1$

Given

$x^2 + y^2 = 6$

$x + y = 0$

$v := \text{Find}(x, y)$

$v = \begin{pmatrix} 1.732 \\ -1.732 \end{pmatrix}$

Проверка

$(v_0)^2 + (v_1)^2 - 6 = -2.085 \times 10^{-6}$

$v_0 + v_1 = 0$

Точность вычислений задается системной константой CTOL, значение которой по умолчанию равно 0.001. Изменение значения константы CTOL осуществляется так же, как и константы TOL.

Приближенное решение уравнений

Для приближенных вычислений систем нелинейных уравнений, так же, как и для решения уравнений с одной неизвестной, может использоваться функция ***Minerr*(x_1, …, x_n)**. Синтаксис этой функции такой же, как и у функции Find. С помощью функции Minerr можно решать несовместные системы уравнений и неравенств. Решение, выдаваемое функцией Minerr, минимизирует невязку данной системы.

Контрольные вопросы

1. Опишите порядок выполнения операций над матрицами.
2. Перечислите основные матричные операции.
3. Какими методами можно решить систему линейных алгебраических уравнений в Mathcad?
4. Что такое обусловленность матрицы, как вычисляется обусловленность матрицы?
5. Что такое ранг матрицы, как его вычислить?
6. Как решить систему линейных алгебраических уравнений матричным способом?
7. С помощью каких функций можно сформировать новую матрицу из существующих матриц?
8. Как решить систему неравенств?

2.8. ВЫЧИСЛЕНИЕ ОПРЕДЕЛЕННОГО ИНТЕГРАЛА, ПРОИЗВОДНЫХ, СУММ, ПРЕДЕЛОВ

Для решения задач математического анализа необходимо воспользоваться панелью инструментов Математический анализ (***Вид, Панели инструментов, Математика*** и щелкнуть по значку ***Матанализ*** – изображение интеграла) (рис. 2.1).

При вычислении пределов и сумм рядов для получения результата необходимо нажать комбинацию клавиш *Shift + F9*. Результат появится ниже формулы. Чтобы результат выводился в той же строке, необходимо выполнить настройку: введите команду ***Символика, Изменение стиля*** и в открывшемся окне диалога активизируйте переключатель ***Горизонтально***.

Вычисления могут производиться с помощью команд меню или операторов панелей инструментов. При этом необходимо учитывать, что при выполнении операций с помощью меню команда применяется только к выделенному выражению. А при выполнении операций с помощью операторов панелей инструментов учитываются выполненные ранее вычисления. Поэтому результаты вычислений окажутся разными.

Вычисление производных и интегралов

Для вычисления производной или интеграла с помощью команд меню выполните следующее:

- Запишите выражение.
- Выделите переменную, относительно которой выполняется вычисление.
- Введите команду ***Символика, Переменная, Дифференцирование*** (Интегрирование).

Пример вычисления приведен на листинге 2.21.

***Листинг 2.21*. Пример вычисления производной и интеграла**

$\sin(2x) + 3\cos\left(x^2\right)^3 \qquad 2\cdot\cos(2\cdot x) - 18\cdot\cos\left(x^2\right)^2\cdot\sin\left(x^2\right)\cdot x$

Пример интегрирования

$2\cdot\cos(2\cdot x) - 18\cdot\cos\left(x^2\right)^2\cdot\sin\left(x^2\right)\cdot x \qquad \sin(2\cdot x) + 3\cdot\cos\left(x^2\right)^3$

Для выполнения операций с помощью панели инструментов порядок вычислений будет несколько иной:

- Выберите щелчком мыши нужный оператор на панели инструментов *Матанализ*.

- Впишите в слоты значения аргумента и функции.
- Введите после выражения оператор символьного вывода с панели инструментов *Вычисления* или *Символика*.
- Нажмите клавишу Enter.

При вычислении производной высшего порядка выберите соответствующий оператор. Ввод данных начинайте с указания переменной дифференцирования и ее порядка.

При вычислении определенного интеграла дополнительно необходимо указать значения нижнего и верхнего пределов интегрирования. Примеры вычисления производных и интегралов с помощью панелей инструментов приведены на листинге 2.22.

Листинг 2.23. **Символьные и численные расчеты рядов**

$x := 1.5 \qquad \sum_{i=0}^{10} (-1)^i \cdot \frac{x^i}{i!} = -0.223 \qquad \sum_{i=0}^{10} \text{if}\left[(-1)^i \cdot \frac{\left(x^i\right)}{i!} > 0, (-1)^i \cdot \frac{\left(x^i\right)}{i!}, 0\right] = 2.129$

$\sum_{i=0}^{10} \frac{x^i}{i!} = 4.482 \qquad \prod_{i=1}^{\infty} \frac{1}{i^3 + 1} \to 0 \qquad \prod_{i=1}^{\infty} \sqrt{i} \to \infty$

Вычисление сумм и произведений рядов

Принцип вычисления рядов аналогичен порядку вычисления производных и интегралов. Вычисления можно выполнять как численно, так и с помощью оператора символьного вывода. Если численное решение не дает результата, попробуйте вычислить выражение с помощью оператора символьного вывода. Пример решения приведен на листинге 2.23. В одном из операторов применяется функция ***if*** (см. раздел 2.3), которая позволяет проверять знак функции. В сумму знакопеременного ряда включаются только положительные числа.

Операторы $\sum_{\blacksquare}$ $\prod_{\blacksquare}$ применяются для работы с ранжированными переменными, например:

$i := 1..10 \qquad \sum_{i} \frac{x^i}{i!} = 3.482 \qquad \prod_{i} i = 3.629 \times 10^6$

Вычисление пределов

Mathcad позволяет вычислять пределы трех видов: предел, правый предел, левый предел (листинг 24):

$\lim_{\blacksquare \to \blacksquare} \blacksquare \qquad \lim_{\blacksquare \to \blacksquare^+} \blacksquare \qquad \lim_{\blacksquare \to \blacksquare^-} \blacksquare$

Листинг 2.24. **Примеры вычисления пределов**

$\lim_{x \to 3} \frac{2x+5}{x-3} \to \text{undefined} \qquad \lim_{x \to 3} \frac{x-3}{2x+5} \to 0 \qquad \lim_{x \to \infty} \left(\frac{x+5}{x-3}\right)^{2x+7} \to \exp(16)$

$\lim_{x \to 3^-} \frac{2x+5}{x-3} \to -\infty \qquad \lim_{x \to 3^+} \frac{2x+5}{x-3} \to \infty$

Из примера на листинге 24 видно, что при *x*, стремящемся к 3, функция не определена. В то же время при *x*, стремящемся к 3 слева или справа, эта же функция стремится к бесконечности.

Контрольные вопросы

1. Как вычислить производную?
2. В чем разница в вычислении производной (интеграла) с помощью панели инструментов и с помощью меню Символика?
3. Как вычислить сумму (произведение) ранжированной переменной?
4. Какие виды пределов позволяет вычислять Mathcad?

2.9. СИМВОЛИЧЕСКИЕ ВЫЧИСЛЕНИЯ

В пакет Mathcad включены средства символических вычислений: алгебраических операций, операций математического анализа, символического решения уравнений и дискретных преобразований. Они могут применяться и к выделенным подвыражениям.

Символьные вычисления можно проводить в двух режимах:

с помощью команд меню;

с помощью операторов символьного вывода, ключевых слов символьного процессора и обычных формул.

Основное отличие этих способов состоит в том, что в первом случае преобразования касаются только выделенного выражения и не учитывают предыдущее состояние документа.

Во втором случае учитываются все предыдущие преобразования и результаты. Если на момент вычисления были известны значения аргументов, то будет получен численный результат.

Для *символьных* вычислений служит команда ***Символика, Расчеты, Символические*** (Symbolics, Evaluate, Symbolically) и другие опции команды меню.

Упростить (Simplify). Перед вычислением программа пытается упростить выражение, это позволяет ускорить вычисления. При выполнении данной операции раскрываются скобки и приводятся подобные члены. Например:

$$(x + 2\cdot y)\cdot z - z^2\cdot(x + 5\cdot y) + z \text{ simplify } \rightarrow z\cdot x + 2\cdot z\cdot y - z^2\cdot x - 5\cdot z^2\cdot y + z$$

$$\frac{x^2 + x - 20}{x + 5} + 2x + 3 \text{ simplify } \rightarrow 3\cdot x - 1$$

При упрощении выражений с числами результат будет зависеть от того, в какой форме представлено число: целое или вещественное, например:

$$\sqrt{5} \text{ simplify } \rightarrow \sqrt{5} \qquad \sqrt{5.03} \text{ simplify } \rightarrow 2.24276614920058038$$

Расширить (Expand). Программа раскрывает все суммы и произведения, а сложные тригонометрические зависимости разлагаются с помощью тригонометрических тождеств:

$$\sin(2x) \text{ expand}, x \rightarrow 2\cdot\sin(x)\cdot\cos(x) \quad \text{или} \quad \sin(2x) \quad 2\cdot\sin(x)\cdot\cos(x)$$

Фактор (Factor). Эта команда раскладывает выражение на простые множители, полиномы разлагаются на более простые сомножители:

$$\left(x^4 - 16\right) \text{ factor } \rightarrow (x - 2)\cdot(x + 2)\cdot\left(x^2 + 4\right) \quad \text{или} \quad x^4 - 16 \quad (x - 2)\cdot(x + 2)\cdot\left(x^2 + 4\right)$$

Подобные (Collect). Данная команда приводит подобные члены выражения:

$$5x^2 - 3x + 3\left(y - x^2\right) + 8x - y \text{ collect}, x \rightarrow 2\cdot x^2 + 5\cdot x + 2\cdot y$$

Коэффициенты полинома (Polynomial Coefficients) – вычисляет коэффициенты полиномов:

$$(x + 2y) + z^3 - 5(x\cdot z)^2 + 2y\cdot z \text{ coeffs}, z \rightarrow \begin{pmatrix} x + 2\cdot y \\ 2\cdot y \\ -5\cdot x^2 \\ 1 \end{pmatrix}$$

При выполнении операций Упростить и Фактор необходимо выделить все выражение. При выполнении операций Расширить, Подобные и Коэффициенты полиномов необходимо выделить переменную, относительно которой будут выполняться преобразования, или указать ее в слоте оператора, вводимого с панели инструментов Символика.

Разложение на элементарные дроби (Convert to Partial Fractions). Этой команде меню соответствует команда ***parfrac*** панели инструментов. Пример:

$$\frac{11\cdot x^2 + 9\cdot x + 1}{x^2 - 3\cdot x + 2} \text{ convert}, \text{parfrac}, x \rightarrow 11 - \frac{21}{(x - 1)} + \frac{63}{(x - 2)}$$

Подстановка переменной (Substitute). Данная команда позволяет заменить переменную или выражение другим выражением. Например, в заданном выражении заменить константу k на ax^2:

$$\sin\left(k\cdot x^2 + b\cdot x\right) + \cos\left(k^2 x\right) \text{ substitute}, k = a\cdot x^2 \rightarrow \sin\left(a\cdot x^4 + b\cdot x\right) + \cos\left(a^2\cdot x^5\right)$$

Разложение в ряд (Expend to Series). С помощью данной команды можно разложить выражение в ряд Тейлора по любой переменной x в точке $x = 0$. После ввода команды необходимо на запрос программы указать имя переменной и порядок аппроксимации. Приведем простейшие примеры:

$$\sin(x) \text{ series}, x, 6 \rightarrow x - \frac{1}{6}\cdot x^3 + \frac{1}{120}\cdot x^5$$

$$e^x \text{ series}, x, 6 \rightarrow 1 + x + \frac{1}{2}\cdot x^2 + \frac{1}{6}\cdot x^3 + \frac{1}{24}\cdot x^4 + \frac{1}{120}\cdot x^5$$

Символьное решение уравнений

С помощью символьного процессора можно вычислить аналитическое значение переменной, при которой выражение обращается в ноль. Для этой цели используется команда ***Symbolics, Variable, Solve*** (Символика, Переменная, Решить).

Символьный процессор позволяет также выполнять операции с матрицами и решать системы линейных алгебраических уравнений в символьном виде, а также выполнять преобразования с помощью матричных операторов на панели инструментов символика: транспонирование матриц, вычисление обратной матрицы и определителя матрицы. Примеры приведены на листинге 2.25.

Листинг 2.25. **Символьные операции с матрицами**

$$\begin{pmatrix} a & b \\ c & d \\ t & f \end{pmatrix} \cdot \begin{pmatrix} o & p & q \\ r & g & t \end{pmatrix} \rightarrow \begin{pmatrix} a\cdot o + b\cdot r & a\cdot p + b\cdot g & a\cdot q + b\cdot t \\ c\cdot o + d\cdot r & c\cdot p + d\cdot g & c\cdot q + d\cdot t \\ t\cdot o + f\cdot r & t\cdot p + f\cdot g & t\cdot q + f\cdot t \end{pmatrix}$$

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} \rightarrow \begin{bmatrix} \frac{d}{(a\cdot d - b\cdot c)} & \frac{-b}{(a\cdot d - b\cdot c)} \\ \frac{-c}{(a\cdot d - b\cdot c)} & \frac{a}{(a\cdot d - b\cdot c)} \end{bmatrix}$$

$$\begin{pmatrix} a \\ b \\ c \end{pmatrix} \times \begin{pmatrix} r \\ s \\ t \end{pmatrix} \rightarrow \begin{pmatrix} b\cdot t - c\cdot s \\ c\cdot r - a\cdot t \\ a\cdot s - b\cdot r \end{pmatrix}$$

Контрольные вопросы

1. Какие виды символьных преобразований позволяет выполнять Mathcad?
2. Как решить систему линейных уравнений в символьном виде?

2.10. РЕШЕНИЕ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ

Решение дифференциальных уравнений первого порядка

Постановка задачи

Инженеру-исследователю постоянно приходится в своей деятельности сталкиваться с дифференциальными уравнениями. Многие задачи механики, физики, химии и других отраслей науки и техники при их математическом моделировании сводятся к дифференциальным уравнениям. В связи с этим решение дифференциальных уравнений является одной из важнейших математических задач. В вычислительной математике изучаются численные методы решения дифференциальных уравнений, которые особенно эффективны в сочетании с использованием вычислительной техники.

В зависимости от числа независимых переменных дифференциальные уравнения делятся на две существенно разные категории: обыкновенные дифференциальные уравнения (ОДУ), содержащие одну независимую переменную, и уравнения с частными производными, содержащие несколько независимых переменных.

Мы ограничимся рассмотрением обыкновенных дифференциальных уравнений.

Обыкновенными дифференциальными уравнениями называются такие уравнения, которые содержат одну или несколько производных от искомой функции $y = y(x)$. Их можно записать в виде:

$$F(x, y, y', ..., y^{(n)}) = 0,$$

где x – независимая переменная.

Наивысший порядок n производной, входящей в уравнение, называется порядком дифференциального уравнения, например:

$F(x, y, y') = 0$ – уравнение первого порядка;

$F(x, y, y', y'') = 0$ – уравнение второго порядка.

Решением дифференциального уравнения называется всякая функция $y = \varphi(x)$, которая после подстановки превращает его в тождество. Или иначе, решить дифференциальное уравнение – это значит определить неизвестную функцию на определенном интервале изменения ее переменных.

Дифференциальные уравнения имеют единственное решение, если известны некоторые начальные условия.

В общем виде задача формулируется следующим образом: найти решение дифференциального уравнения

$$y^{(n)} = f(x, y, y', ..., y^{(n-1)})$$

в виде функции $y = y(x)$, которая удовлетворяет заданным начальным условиям

$$y(x_0) = y_0, y'(x_0) = y'_0, \ldots, y^{(n-1)}(x_0) = y_0^{(n-1)},$$

где $y_0, y'_0, \ldots, y_0^{(n-1)}$ – заданные значения функции и ее производных до $(n - 1)$-го порядка включительно.

Методы решения

В Mathcad имеется возможность решать задачи двух типов:

задачи Коши – дифференциальные уравнения, для которых определены начальные условия на искомые функции в начальной точке интервала интегрирования уравнения;

краевые задачи – дифференциальные уравнения, для которых заданы определенные соотношения сразу на обеих границах интервала.

Задачи Коши

Для решения задачи Коши разработаны эффективные методы решения.

Пусть требуется найти функцию $Y = Y(x)$, удовлетворяющую уравнению:

$$\frac{dY}{dx} = f(x, Y). \tag{2.1}$$

Решение дифференциальных уравнений сводится к замене точного решения приближенным. Чаще всего используются разностные методы. Введем последовательность точек x^0, x^1, ..., x^n и шаги $h_i = x_{i+1} - x_i$. В каждой точке x_i, называемой ***узлом***, вместо точного значения $Y(x)$ вводятся числа y_i, аппроксимирующее точное решение Y на данном множестве точек. Функцию y, заданную в виде таблицы (x_i, y_i) ($i = 1, 2, ...$), называют ***сеточной функцией***. Далее, заменяя значение производной в уравнении (2.1) отношением конечных разностей, осуществляют переход от дифференциальной задачи (2.1) относительно функции Y к разностной задаче относительно сеточной функции y:

$$y_{i+1} = F(x_i, h_i, y_{i+1}, y_i, ..., y_{i-k+1}),\ i = 1, 2, ...; \tag{2.2}$$

$$y_0 = Y_0. \tag{2.3}$$

Разностное уравнение записано в общем виде, а конкретное выражение его правой части зависит от выбранного способа аппроксимации производной. Для каждого численного метода получается свой вид уравнения (2.2).

Методы решения задачи Коши можно разделить на два вида: одношаговые и многошаговые.

К ***одношаговым методам*** относится метод Эйлера. Это простейший метод. Он основан на разложении искомой функции $Y(x)$ в ряд Тейлора в окрестностях узлов $x = x_i (i = 0, 1, ...)$, в котором отбрасываются все члены, содержащие производные второго и более высоких порядков. Это разложение запишется в виде

$$Y(x_i + \Delta x_i) = Y(x_i) + Y'(x_i)\Delta x_i + O(\Delta x_i^2), \tag{2.4}$$

Заменяя значения функции Y в узлах x_i значениями сеточной функции y_i и учитывая выражение (2.1), получаем:

$$y_{i+1} = y_i + hf(x_i, y_i).$$

Погрешность метода Эйлера определяется отброшенными членами разложения в ряде Тейлора. Эта ошибка имеет порядок h^2. При

суммировании значений функции на каждом шаге ошибки складываются, следовательно, суммарная ошибка будет иметь порядок nh^2. Поэтому для уменьшения ошибки следует уменьшать шаг. Однако при значительном уменьшении шага эта ошибка может сравняться с погрешностями вычисления $|y_{i+1} - y_i|$. Для контроля величины шага рекомендуется использовать выражение $\frac{|y_{i+1} - y_i}{|y_i + 1|} > 0,01$.

Наиболее распространенным одношаговым методом решения дифференциальных уравнений является метод Рунге–Кутты. Этот метод более сложен с точки зрения вычислений по сравнению с методом Эйлера. Но этот недостаток покрывается его преимуществами. Преимуществом метода Рунге–Кутты перед методом Эйлера является то, что он позволяет получить более точный результат при большем шаге вычислений.

При решении данного дифференциального уравнения методом Рунге–Кутты точное значение y заменяют его приближенным значением:

$$Y_{i+1} = Y_i + \frac{1}{6}\left(K_1 + 2K_2 + 2K_3 + K_4\right), \tag{2.5}$$

где значения коэффициентов на i-м шаге вычисляются по формулам:

$$\begin{aligned} K_1 &= hf(x_i, y_i), \\ K_2 &= hf(x_i + 0,5h, y_i + 0,5K_1), \\ K_3 &= hf(x_i + 0,5h, y_i + 0,5K_2), \\ K_4 &= hf(x_i + h, y_i + K_3). \end{aligned} \tag{2.6}$$

Здесь $x_{i+1} = x_i + h$.

Для проверки точности необходимо сделать второй проход с шагом $h/2$. Если разница между Y_i на $k + 1$ и k-м проходах будет меньше требуемой точности, то процесс вычисления прекращается. Значение точности выбирают в интервале от 0,001 до 0,00001. Итерационный процесс обычно быстро сходится.

Значение шага при переходе к следующей точке можно изменять. Правильность выбора шага проверяется по формуле:

$$T = \left|\frac{K_2 - K_3}{K_1 - K_2}\right|, \tag{2.7}$$

величина T не должна превышать нескольких сотых.

Грубую оценку погрешности метода проводят с помощью выражения:

$$Y_k - Y(x_i) = \frac{|Y_{k+1} - Y_k|}{15}, \qquad (2.8)$$

где $Y(x_i)$ – значение точного решения уравнения.

Для уточнения решения можно применять метод двойного прохода с разным шагом. Уточненное значение функции в узлах определяется по формуле:

$$y_h = 2y_{h/2} - y_h + O(h). \qquad (2.9)$$

Необходимо отметить, что имеется целый класс дифференциальных уравнений, которые не поддаются решению названными методами. Это так называемые жесткие задачи. Для них разработаны специальные методы.

Краевые задачи

Краевые задачи возникают при решений дифференциальных уравнений высших порядков или систем уравнений.

Для решения краевых задач применяются разные методы. Одни из них основаны на сведении краевой задачи к решению задачи Коши, другие на прямом применении разностных схем.

Метод стрельбы основан на сведении краевой задачи к задаче Коши. Полагают, что решение задачи Коши зависит от некоторого параметра α: $Y = Y(x, \alpha)$. Решают задачу Коши при некотором значении α и проверяют, совпали значения на конце отрезка с заданными значениями или нет. Если решения не совпали, то меняют значение коэффициента α и повторяют решение.

Методы конечных разностей основаны на том, что они сводят решение краевой задачи для дифференциального уравнения к решению системы алгебраических уравнений относительно значений искомой функции на заданном множестве точек.

Функции, предназначенные для решения обыкновенных дифференциальных уравнений

Для решения обыкновенных дифференциальных уравнений первого порядка в Mathcad имеется две возможности: использовать вычислительный блок Given/Odesolve или использовать функции rkfixed, Rkadapt, Bulstoer.

Вычислительный блок Given/Odesolve

Вычислительный блок для решения ОДУ реализует численный метод Рунге–Кутты и состоит из трех частей:

Given – ключевое слово;

ОДУ – записанное с помощью логических операторов, причем начальное условие должно быть в форме $y(t_0) = b$;

odesolve(t,t1) – встроенная функция для решения ОДУ относительно переменной t на интервале (t_0, t_1). Рекомендуется функцию *Odesolve применять* в формате *Odesolve(t,t1,step)*, где step – внутренний параметр численного метода, определяющий количество шагов. Чем больше значение параметра, тем выше будет точность полученного результата. При этом требуется, чтобы конечная точка лежала правее начальной точки, т. е. должно выполняться условие $t_0 < t_1$.

Пример 2.15. Решить дифференциальное уравнение (рис. 2.17):

$$Y' = y - y^2.$$

t := 0 , 0.1 .. 10

t =	y(t) =
0	0.1
0.1	0.109
0.2	0.119
0.3	0.13
0.4	0.142
0.5	0.155
0.6	0.168
0.7	0.183
0.8	0.198
0.9	0.215
1	0.232
1.1	0.25
1.2	0.269
1.3	0.29
1.4	0.311
1.5	0.332

Рис. 2.17. Результат решения уравнения блоком Given

Given

$$\frac{d}{dt}y(t) = y(t) - y(t)^2$$

$$y(0) = 0.1$$

$$y := \text{Odesolve}(t, 10)$$

Функция *Odesolve* имеет две модификации: с фиксированным шагом (Fixed) и адаптивным шагом (Adaptive). По умолчанию применяется фиксированный шаг.

Функции rkfixed, Rkadapt, Bulstoer

Функции имеют одинаковый синтаксис: ***rkfixed(y_0,t_0,t_1,M,D)***. Параметры функции:

y_0 – вектор начальных условий;

t_0 и t_1 – граничные значения отрезка решения задачи;

M – число интервалов разбиения отрезка $[t_0; t_1]$;

$D(t,y)$ – вектор функция, содержащая правые части первых производных, записанных в символьном виде.

Функции rkfixed и Rkadapt реализуют численное решение задачи Коши по методу Рунге–Кутты. Первая функция имеет фиксированный шаг, а вторая – переменный шаг. Вследствие автоматического подбора шага функция Rkadapt, как правило, дает более точный результат по сравнению с другими функциями. Функция Bulstoer реализует метод Булирша–Штера.

Результат получается в виде матрицы, в первом столбце которой записаны значения аргумента, а во втором – соответствующие значения функции (рис. 2.18).

Названные функции проигрывают блоку Given/Odesolve в наглядности, но имеют и некоторые преимущества. А именно, они могут быть использованы в программных модулях и позволяют оперативно пересчитывать результаты при изменении параметров.

Пример 2.16. Использование функции rkfixed.

Решить дифференциальное уравнение:

$$\frac{dy}{dt} = y - y^2$$

на отрезке [0; 10].

$y := 0.1$

$D(t,y) := y - y^2$

y =

	0	1
0	0	0.1
1	0.1	0.109
2	0.2	0.119
3	0.3	0.13
4	0.4	0.142
5	0.5	0.155
6	0.6	0.168
7	0.7	0.183
8	0.8	0.198
9	0.9	0.215
10	1	0.232
11	1.1	0.25
12	1.2	0.269
13	1.3	0.29
14	1.4	0.311
15	1.5	0.332

Рис. 2.18. Результат решения дифференциального уравнения функцией rkfixed

$M := 100$

$y := \text{rkfixed}(y, 0, 10, M, D)$

Обыкновенные дифференциальные уравнения (ОДУ) высших порядков

ОДУ высших порядков содержат в правой части производные от первого до $(n - 1)$-го порядка. В Mathcad такие задачи могут быть решены или с использованием блока Given/Odesolve или путем сведения их к системам уравнений первого порядка.

Внутри вычислительного блока ОДУ должно быть линейно относительно старшей производной, т. е. должно быть представлено в стандартной форме (2.1). Начальные условия должны иметь форму $y(t) = b$ или $y^{(n)}(t) = b$.

Пример 2.17. Решение уравнения гармонического осциллятора.

Given

$$\frac{d^2}{dt^2}y(t) + 0.1\frac{d}{dt}y(t) + 1y(t) = 0$$

$y(0) = 0.1$

$y'(0) = 0$

$y := \text{Odesolve}(t,50)$

Значения функции выводятся на экран так же, как и при решении дифференциальных уравнений первого порядка. График функции приведен на рисунке 2.19.

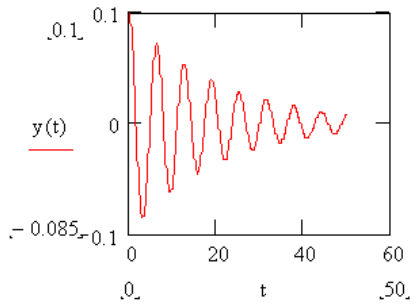


Рис. 2.19. График функции гармонического осциллятор

Системы ОДУ первого порядка

Mathcad требует, чтобы система дифференциальных уравнений была записана в стандартной форме:

$$\begin{cases} y'_0(t) = f_0(y_0(t), y_1(t), ..., y_{n-1}(t), t), \\ y'_1(t) = f_1(y_0(t), y_1(t), ..., y_{n-1}(t), t), \\ ... \\ y'_{n-1}(t) = f_{n-1}(y_0(t), y_1(t), ..., y_{n-1}(t), t). \end{cases}$$

В векторной форме эту систему уравнений можно представить в следующем виде:

$$Y(t) = F(Y(t),t).$$

Кроме системы уравнений необходимо задать вектор начальных условий. В векторной форме это запишется так: $Y(t_0) = B$ (пример 2.18).

***Пример 2.18*.** Пример решения системы двух ОДУ.

$$D(t,y) := \begin{pmatrix} y_1 \\ -y_0 - 0.1\,y_1 \end{pmatrix}$$

$$M := 50$$

$$y_0 := \begin{pmatrix} 0.1 \\ 0 \end{pmatrix}$$

$$u := \text{rkfixed}(y_0, 0, 50, M, D)$$

Для решения систем дифференциальных уравнений первого порядка используются функции Mathcad rkfixed, Rkadapt, Bulstoer. Чтобы получить результат введите имя функции и нажмите клавишу Enter. Результат представляется в виде таблицы значений, в первой колонке которой выводится номер шага, а во второй и третьей колонках значения y_0 и y_1, соответственно.

Решение дифференциальных уравнений второго и более высоких порядков

Дифференциальные уравнения второго и более высоких порядков относятся к краевым задачам.

Отличие краевой задачи от задачи Коши состоит в том, что в краевой задаче начальные условия задаются на концах интервала поиска решения. Для решения подобных задач в системе Mathcad используется метод пристрелки, который начальное условие в правой точке интервала преобразует в дополнительное начальное условие для левой точки интервала. После чего краевая задача трансформируется в задачу Коши. Для реализации метода пристрелки в Mathcad существует функция ***sbval***. Данная функция определяет недостающие условия в начальной точке для двухточечных краевых задач. Функция имеет следующий синтаксис: ***sbval(z,x0,x1,D,load,score)***, где z – вектор приближений недостающих начальных условий на левой границе; x_0, x_1 – левая и правая границы интервала решений; $D(x,y)$ – вектор-функция, содержащая правые части первых производных, записанная в символьном виде; ***load(x0,z)*** – вектор-функция, описывающая начальные условия на левой границе интервала; ***score(x1,y)*** – вектор-функция для задания правых граничных условий.

Пример 2.19. Пример решения задачи с краевыми условиями, заданными на концах интервала методом стрельбы.

Решить систему дифференциальных уравнений на отрезке [0; 1], число шагов $M = 10$.

Дано:

$r(x) := 0.1$, $a(x) := 1$ – коэффициенты дифференциального уравнения;

$R = 1$ – начальное условие на правой границе;

$I0 = 100$ – начальное условие на левой границе интервала.

Система дифференциальных уравнений:

$$\begin{cases} y'(x) = -a(x)\,y_0 + r(x)\,y_1 \\ y'(x) = a(x)\,y_1 - r(x)\,y_0 \end{cases}$$

Порядок решения:

1. Задать вектор z – вектор недостающих начальных значений на левой границе, для начала положим $z_0 = 10$.

2. Присвоить функции Load(x_0,z) начальные значения условий на левой границе:

Load(x_0,z) = $(I0,z_0)^T$.

3. Присвоить функции score(x1,y) начальные значения условий на правой границе:

score(x_1,y) = $Ry_0 - y_1$.

4. Вычислить вектор недостающих начальных условий на левой границе с помощью функции sbval(z,0,1,load,score).

5. Решить задачу с помощью функции rkfixed.

Пример решения приведен на листинге 2.26.

***Листинг 2.26*. Пример решения краевой задачи**

$a(x) := 1 \quad r(x) := 0.1 \quad R := 1 \quad I0 := 100$

$$D(x, y) := \begin{pmatrix} -a(x)\, y_0 + r(x)\, y_1 \\ a(x)\, y_1 - r(x)\, y_0 \end{pmatrix}$$

$z_0 := 10$

$$load(x0, z) := \begin{pmatrix} I0 \\ z_0 \end{pmatrix}$$

$score(x1, y) := R \cdot y_0 - y_1$

$I1 := sbval(z, 0, 1, D, load, score) \qquad I1 = (18.555)$

$$S := rkfixed\left[\begin{pmatrix} I0 \\ I1_0 \end{pmatrix}, 0, 1, 10, D\right] \qquad S = \begin{pmatrix} 0 & 100 & 18.555 \\ 0.1 & 90.665 & 19.504 \\ 0.2 & 82.228 & 20.646 \\ 0.3 & 74.606 & 21.993 \\ 0.4 & 67.723 & 23.558 \\ 0.5 & 61.511 & 25.356 \\ 0.6 & 55.908 & 27.405 \\ 0.7 & 50.86 & 29.726 \\ 0.8 & 46.315 & 32.341 \\ 0.9 & 42.229 & 35.277 \\ 1 & 38.562 & 38.562 \end{pmatrix}$$

Контрольные вопросы

1. Какие виды дифференциальных уравнений позволяет решать Mathcad?

2. Что такое задача Коши?

3. Какие функции используются для решения дифференциальных уравнений первого порядка с начальными условиями?

4. Как решить дифференциальное уравнение методом Эйлера?

5. В чем заключается метод Рунге–Кутты для решения дифференциальных уравнений?

6. Что такое метод стрельбы (пристрелки), для чего он применяется?

7. Как решаются дифференциальные уравнения высшего порядка в Mathcad?

8. Поясните порядок решения системы дифференциальных уравнений в Mathcad.

2.11. ПРОГРАММИРОВАНИЕ В MATHCAD

Общие сведения

Для расширения возможностей Mathcad по решению широкого круга математических задач в него встроен язык программирования, позволяющий пользователю разрабатывать собственные программы. В настоящем разделе мы познакомимся с основными конструкциями этого языка программирования и рассмотрим примеры их применения.

Все операторы, используемые в программах, размещены на панели инструментов Программирование (рис. 2.20). Эти операторы *нельзя* вводить с клавиатуры.

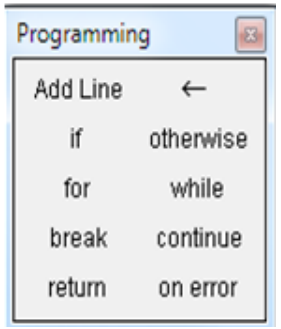


Рис. 2.20. Панель инструментов Программирование

Операторы панели инструментов Программирование имеют следующее назначение.

Оператор (инструкция) ***Add Line*** – выполняет функции создания и расширения программных блоков. По умолчанию оператор содержит две строки (два слота). Строки можно добавлять в оператор, место вставки зависит от положения курсора. Примеры добавления строк приведены на рисунке 2.21.

Для добавления пустой строки необходимо установить курсор как показано в верхней строке рисунка 2.21, и щелкнуть по кнопке Add Line. Если курсор ввода находится слева от выделенной стро-

ки, то новая строка добавляется выше текущей строки, в противном случае новая строка будет вставлена ниже текущей строки.

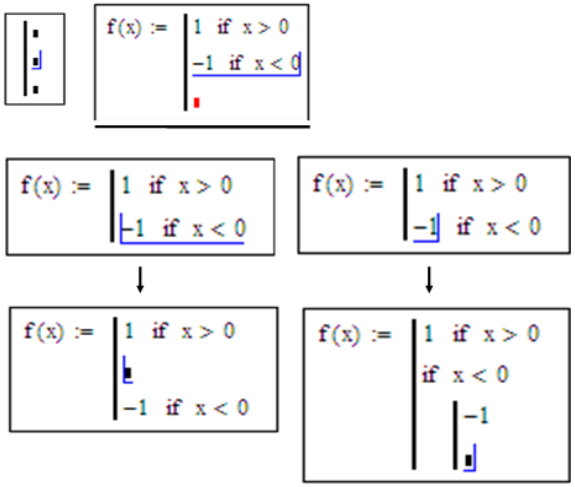


Рис. 2.21. Добавление строк в программный блок

Если курсор выделяет не всю строку, то выделенная часть переходит на новую строку и добавляется новый программный блок. Другой пример добавления программного блока приведен на рисунке 2.22.

Оператор *локального присваивания* создает локальную переменную. Эта переменная не доступна из основной программы.

Оператор *break* вызывает прерывание выполнения программы. Чаще всего этот оператор используется совместно с операторами циклов ***while*** и ***for***, обеспечивая переход в конец цикла.

Оператор *continue* используется для продолжения работы после прерывания программы. Этот оператор используется чаще всего совместно с операторами while, wend, обеспечивая возвращение в точку прерывания и продолжение вычислений.

Оператор *return* прерывает выполнение программы и возвращает значение операнда, стоящего следом за ней.

Оператор *if* – условный оператор, оператор выбора. Принцип его работы такой же, как и у функции ***if*** (раздел 2.3), но синтаксис его отличается от синтаксиса функции if.

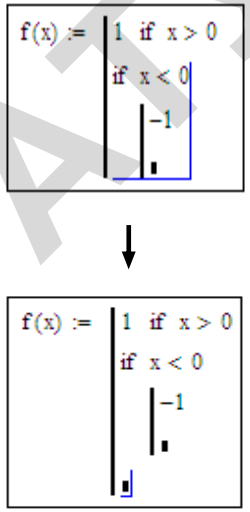


Рис. 2.22. Добавление программного блока

Оператор for – организует цикл с заданным числом повторений.

Оператор While – организует цикл с параметром.

Оператор on error – служит для обработки ошибок периода выполнения (ошибок, возникающих в процессе выполнения программы).

Примеры использования операторов

Условный оператор if

Синтаксис оператора: ***<выражение_истина>if<условие>***.

Пример 2.20. Пример простого оператора ***if***.

Написать программу для вычисления функции:

$$y = \sin(x)^2 - \cos(x),$$

если $x > a$ (листинг 2.27).

Листинг 2.27. **Пример использования простого оператора if**

```
x := 2
y := | a ← 1.5
     | sin(x)^2 − cos(x)  if x > a
y = 1.243
```

Блок схема алгоритма приведена на рисунке 2.23.

Рис. 2.23. Схема алгоритма

Пример 2.21. Пример использования расширенного оператора ***if***.

Написать программу для вычисления функции, $y = \sin(x)^2 - \cos(x)$ если $x > a$ и $y = 1/\sin(x)$ в ином случае. Блок схема алгоритма приведена на рисунке 2.23. Для расширения оператора ***if*** применяется оператор ***otherwise*** (листинг 2.28).

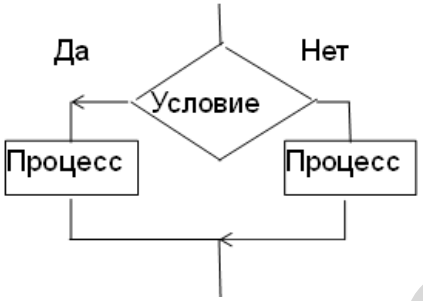


Рис. 2.24. Расширенный оператор

Оператор ***otherwise*** используется совместно с оператором if, активизирует выражение, которое должно быть выполнено, если условие не выполняется (ложно).

Оператор цикла for

Оператор ***for*** служит для организации цикла с заданным числом повторений.

Синтаксис оператора: ***for Var ∈ Nmin . . Nmax***.

Шаг по умолчанию равен 1. Данная запись означает, что выражение, записанное ниже данного оператора, будет выполняться заданное число раз.

Листинг 2.28. **Пример использования расширенного оператора if**

```
x := 1.25
y := | a ← 1.5
     | sin(x)^2 − cos(x)   if  x > a
     | 1/cos(x)            otherwise
y = 3.171
```

Переменная ***Var*** – счетчик циклов. Она может использоваться в выражении. Пример использования оператора и схема алгоритма приведены на листингах 2.29, 2.30, схема алгоритма приведена на рисунке 2.25.

Листинг 2.29. **Примеры использования оператора цикла for**

```
s := | z ← 0                    s1 := | z ← 0
     | for i ∈ 0..10                  | for i ∈ (2  9  7.5  12  2.175)
     |   z ← z + i                    |   z ← z + i
s = 55                          s1 = 32.675

s2 := | z ← 0                   s3 := | z ← 0
      | for i ∈ 0..10                  | for i ∈ 0..10
      |   z ← z + i                    |   z ← z + i
      |   break if i = 5               |   continue
s2 = 15                         s3 = 55
```

Листинг 2.30. **Примеры использования оператора цикла for**

```
u(x) := | y ← x^2                u1(x) := | y ← x^2
        | z ← y + 1                       | z ← y + 1
        | z                               | (x)
                                          | (y)
u(2) = 5                        u1(2) = (2)
                                        (4)
                                u1(2)_0 = 2
                                u1(2)_1 = 4
```

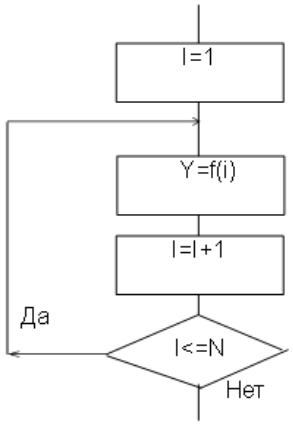


Рис. 2.25. Схема цикла for

В правой части оператора ***for*** может использоваться или ранжированная переменная, или вектор. Совместно с оператором for могут использоваться операторы ***continue*** и ***break***.

При вычислении функции *s* программа повторяет операцию сложения 10 раз и при достижении переменной цикла конечного значения выходит из цикла. Переменная *z* в этом случае выполняет роль контейнера, в который на каждом шаге добавляется очередное значение переменной цикла *i*. Функция *s*1 вычисляется аналогичным образом, но суммируются значения элементов вектора. При вычислении переменной *s*2 программа на каждом шаге проверяет дополнительно текущее значение переменной цикла, и при достижении ей заданного значения цикл прекращается. Процесс вычисления функции *s*3 ничем не отличается от вычисления функции *s*, но дополнительно используется оператор *continue* для возвращения на начало цикла.

В программном блоке можно явно указывать возвращаемое значение. Это может быть переменная или массив. Пример подобного вычисления приведен на листинге 2.29.

При вычислении функции *u*(*x*) возвращаемым значением является переменная *z*, значение которой присваивается функции *u*(*x*). Во втором примере возвращаемым значением является вектор, первый элемент которого равен значению аргумента функции, а вторым элементом – значение *y* (в качестве второго элемента вектора можно было указать также переменную *z*).

На листинге 2.31 приведен пример использования оператора ***return***.

Листинг 2.31. **Использование оператора return**

```
i := 10

u(x) := return "zero"  if x = 0
        return "i"  if x = i
        z ← 1/x^2
        z

u(-1) = 1
u(2) = 0.25
u(0) = "zero"
u(i) = "i"
```

Для сведения: компилятор языка программирования Mathcad не допускает использования символов русского алфавита даже в текстовых константах.

Оператор цикла While

Оператор ***while*** организует цикл с параметром с предусловием, то есть проверка условия окончания цикла производится в начале цикла (рис. 2.26).

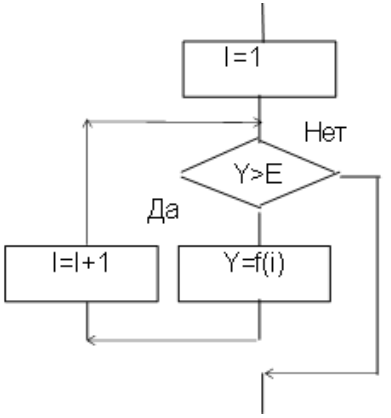


Рис. 2.26. Схема цикла While

Синтаксис оператора: ***while <условие> действия***.

Условием окончания цикла является достижение некоторой переменной, вычисляемой в теле цикла некоторого, наперед заданного значения.

Вычислим для примера значение функции e^x с помощью оператора While/Wend. Функция e^x может быть представлена в виде бесконечного ряда:

$$e^x = 1 + 1\cdot x + \frac{1}{2}\cdot x^2 + \frac{1}{6}\cdot x^3 + \frac{1}{24}\cdot x^4 + \frac{1}{120}\cdot x^5 + O(x^6)$$

Общий член ряда: $\frac{x^i}{i!}$

Пример вычисления приведен на листинге 2.32. На листинге приведен также пример схемы алгоритма и для сравнения вычислено значение функции с помощью встроенной функции e^x.

Листинг 2.32. **Оператор While**

```
e1(x) := | eps ← 0.0001
         | p ← 1
         | i ← 0
         | u ← 1
         | while p > eps
         |     | i ← i + 1
         |     | p ← x^i / i!
         |     | u ← u + p

e1(0.2) = 1.221

x := 0.2    e^x = 1.221
```

Оператор on error

Оператор on error позволяет создавать процедуры обработки ошибок периода выполнения.

Синтаксис оператора: ***Выражение_1 on error Выражение_2***.

Если при выполнении Выражения_1 возникает ошибка, то выполняется Выражение_2. Совместно с оператором *on error* используется функция ***error(«сообщение»)***. Функция *error*(*s*) при возникновении ошибки выводит всплывающую подсказку с сообщением, хранящимся в символьной переменной *s*.

Пример программного блока для обработки ошибок приведен на листинге 2.33. В примере осуществляется контроль деления на ноль. В случае возникновения ошибки выдается сообщение «Ошибка пользователя: не допускается деление на ноль».

Листинг 2.33. **Оператор обработки ошибок on error**

```
f(n) := |  on error

f(n) := | z ← n
        | "user error: can't divide by zero"   on error 1/z

f(2) = 0.5

f(0) = "user error: can't divide by zero"
```

Программирование в символьных расчетах

Программный процессор Mathcad позволяет также производить расчеты и в символьном виде. В качестве примера приведем программу вычисления производной от функции x^{10}. Пример вычислений с помощью данного программного блока в символьном и численном виде приведен на листинге 2.34.

Листинг 2.34. **Программирование в символьных расчетах**

```
u(n,x) := | -1                if n < 0
          | d^n/dx^n x^10     otherwise

u(-1,x) → -1
u(1,x) → 10·x^9
u(2,x) → 90·x^8
u(2,2) → 23040
```

```
x := 1.5

u(n,x) := | -1                if n < 0
          | d^n/dx^n x^10     otherwise

u(-1,x) → -1
u(1,x) → 384.433593750
u(2,x) → 2306.60156250
u(2,2) → 23040
```

Контрольные вопросы

1. Как создать или расширить программный блок в Mathcad?
2. Какие операторы используются для программирования в Mathcad?
3. Приведите синтаксис и схему алгоритма условного оператора.
4. Приведите синтаксис и схему алгоритма цикла For.
5. Приведите синтаксис и схему алгоритма цикла While.

2.12. МАТЕМАТИЧЕСКАЯ СТАТИСТИКА

Понятие о случайных величинах и их характеристиках

В повседневной жизни и работе мы постоянно сталкиваемся с событиями и явлениями, исход которых заранее не известен. Примерами могут служить задачи контроля качества продукции, классификации объектов, резервирования технических систем, диагностика технических систем. Тем не менее, и в этих условиях приходится принимать основанные на опытных данных решения. Чтобы принимаемые решения были научно обоснованы, проводят моделирование ситуаций, поведения технических систем и т. п.

В большинстве явлений присутствуют как закономерная, так и случайная составляющие. Наличие случайной составляющей можно объяснить наличием множества неучтенных факторов. Математиками разработаны различные модели случайных величин и оценки их характеристик, которые могут быть использованы при моделировании. Mathcad имеет развитый аппарат для работы с задачами математической статистики: функции для расчета основных характеристик законов распределения случайных величин; генераторы случайных чисел с соответствующими законами распределений, позволяющие эффективно проводить моделирование методами Монте-Карло (метод статистических испытаний); строить гистограммы и рассчитывать статистические выборки случайных чисел и случайных процессов.

Случайной величиной называется переменная величина, которая в зависимости от случайного исхода испытания принимает какое-то одно из своих возможных значений, неизвестно, какое именно. Числовое значение, которое приняла случайная величина X в каком-либо конкретном испытании, называется *реализацией* этой случайной величины. Множество значений, которые может принимать случайная величина X, называется областью *возможных значений* этой случайной величины.

Случайная величина называется ***дискретной***, если она может принимать только конечное или счетное множество возможных значений.

Случайная величина называется ***непрерывной***, если она может принимать любое значение из некоторого интервала. Она имеет несчетное множество возможных значений.

Основной (исчерпывающей) характеристикой непрерывной случайной величины является ***закон распределения случайной величины – F(x)***. Закон распределения устанавливает связь между возможными значениями случайной величины и вероятностями их появления. Для дискретной случайной величины основной характеристикой является ***ряд распределения***. Ряд распределения – это совокупность различных возможных значений случайной величины, расположенных в порядке возрастания ее значений и вероятностей появления каждого из этих значений. Для непрерывной случайной величины X, распределенной в некотором промежутке $[a; b]$ выполняется равенство $F(x) = \int_{\infty}^{x} f(x)dx$. Здесь функция $f(x)$ называется дифференциальным законом распределения или дифференциальной функцией (или функцией вероятности) непрерывной случайной величины, а $F(x)$ – интегральный закон распределения непрерывной случайной величины или функция распределения непрерывной случайной величины – вероятность того, что случайная величина X будет меньше или равна x ($F(x) = P(X \le x)$). Предел $F(x)$ стремится к 1 при x, стремящемся к бесконечности.

Основными числовыми характеристиками случайных величин являются: математическое ожидание, дисперсия, среднее квадратичное отклонение, асимметрия и эксцесс (табл. 2.3).

Таблица 2.3

Основные числовые характеристики случайных величин

Характеристика	Случайная величина	
	непрерывная	дискретная
Математическое ожидание	$M\xi = \int_{-\infty}^{+\infty} xf(x)dx$	$M\xi = \sum_k x_k p_k$
Дисперсия	$D\xi = M(\xi - M\xi)^2$ или $D\xi = M\xi^2 - (M\xi)^2$	$D\xi = \sum_k (x_i - Mx)^2 p_i$
Среднее квадратичное отклонение	$CKO = \sqrt{D\xi}$	
Асимметрия	$\frac{M(\xi - M\xi)^3}{(D\xi)^{3/2}}$	

Окончание табл. 2.3

Характеристика	Случайная величина	
	непрерывная	дискретная
Эксцесс	$\frac{M(\xi - M\xi)^4}{(D\xi)^2}$	

Математическое ожидание – среднее значение случайной величины.

Дисперсия – характеризует степень рассеяния случайной величины. Дисперсия – математическое ожидание квадрата отклонения случайной величины от центра.

Среднее квадратичное отклонение (СКО) – положительное значение корня квадратного из дисперсии этой случайной величины.

Математическое ожидание и дисперсию называют также первым и вторым ***центральными моментами*** соответственно.

Асимметрия (несимметричность) – третий нормированный момент. Характеризует несимметричность распределения случайной величины.

Эксцесс (островершинность) – четвертый нормированный момент. Эксцесс характеризует поведение случайной величины в окрестности *моды*. Эксцесс нормального распределения равен нулю.

Мода – значение случайной величины, при котором плотность вероятностей достигает максимума.

Для случайных величин, принимающих вещественные значения, часто используют такие характеристики, как ***квантили***. Среди квантилей чаще всего используют медиану и квантиль распределения.

Квантилью x_p случайной величин, имеющей функцию распределения ***F(x)***, называется решение x_p уравнения ***F(x) = p***, где p – заданная вероятность.

Медианой называется квантиль, соответствующая значению $p = 0{,}5$.

Верхней квантилью называется квантиль, соответствующая значению $p = 0{,}75$, а *нижней квантилью* – квантиль, соответствующая $p = 0{,}25$.

Законы распределения случайных величин

Биномиальное распределение

Биномиальное распределение служит моделью для многих явлений. Оно возникает в тех случаях, когда нас интересует, сколько раз происходило некоторое событие в серии из определенного числа

независимых наблюдений. Для дискретной случайной величины вероятность наступления m событий из n испытаний определяется по формуле: $p_n(m) = \frac{n!}{m!(n-m)!} p^m q^{n-m} = C_n^m p^m q^{n-m}$.

Математическое ожидание ***MA = np***, а дисперсия ***DA = np(1 – p)***.

Нормальный закон распределения

Большинство экспериментальных исследований в биологии, медицине, технике и других областях, связанных с измерениями, результаты которых могут принимать различные значения, описываются моделью непрерывных случайных величин, подчиняющихся нормальному закону распределения (гауссово распределение).

Плотность вероятностей описывается формулой:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(x-a)^2}{2\sigma^2}}, \quad -\infty < x < \infty.$$

Здесь a и σ – параметры распределения: a – математическое ожидание, σ – среднеквадратичное отклонение.

Математическое ожидание $M\xi = a$ дисперсия $D\xi = \sigma^2$.

Область на функции распределения [–□; +□] область, вероятность попадания в которую равна 0,683. Вероятность попадания в область $[-2\sigma; 2\sigma]$ равна 0,955, а вероятность попадания в область $[-3\sigma; 3\sigma]$ равна 0,997.

Пример функции, распределенной по нормальному закону, приведен на рисунке 2.27.

Распределение Пуассона

Распределение Пуассона – одно из наиболее распространенных дискретных распределений. Например, отказы оборудования подчиняются закону Пуассона:

$$p(x) = P(X = x) = \frac{\mu^x}{x!} e^{-\mu}, \; x = 0, 1, 2, \ldots, \; \mu > 0.$$

Математическое ожидание равно μ, где $\mu = \lambda\tau$, λ – плотность распределения точек в пространстве, τ – мера соответствующей части пространства (например, λ – плотность потока отказов и τ – время). Дисперсия $D = \mu$.

Имеются и другие законы распределения, например: равномерный, Фишера, Стьюдента, χ^2.

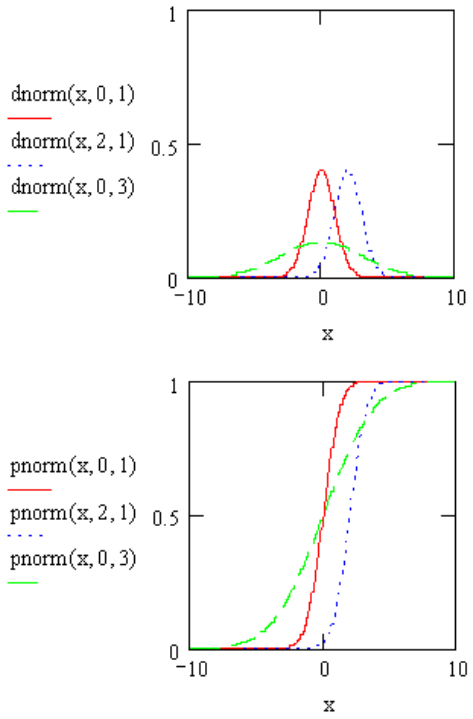


Рис. 2.27. Пример функции распределения и плотности распределения случайной величины, распределенной по нормальному закону

Функции Mathcad для работы со случайными числами

В теории вероятностей доказано, что сумма различных независимых случайных слагаемых (независимо от законов их распределения) оказывается распределенной по нормальному закону.

Для описания нормального закона распределения вероятностей Mathcad имеет следующие функции:

dnorm(x,μ,σ) – плотность вероятностей нормального распределения;

pnorm(x,μ,σ) – функция нормального распределения;

cnorm(x) – функция нормального распределения для μ = 0, σ = 1;

qnorm(p,μ,σ) – обратная функция нормального распределения, по заданному значению *p* вычисляет значение *x*;

rnorm(M,μ,σ) – вектор *M* независимых случайных чисел, каждое из которых имеет нормальное распределение.

Здесь *x* – значение случайной величины, μ – математическое описание, σ – среднеквадратичное отклонение, *p* – значение вероятности.

Для других законов распределения имеются аналогичные функции: d*(x,μ,σ); p*(x,μ,σ); c*(*x*); q*(p,μ,σ); r*(M,μ,σ). Для равномерного распределения вместо звездочки в имя функции следует подставить *unif*, для биномиального – *binom*, для Пуассона – *pois* и т. д.

Пример исследования влияния математического ожидания и дисперсии на плотность распределения и функцию распределения случайной величины, распределенной по нормальному закону, приведен на рисунке 2.27.

Примеры использования функций для вычислений:

- Определить вероятность того, что *x* будет меньше 1,881 (pnorm(1.881,0,1)).
- Вычислить 97 %-ный квантиль нормального распределения (qnorm(0.97,0,1)).
- Определить вероятность того, что *x* будет больше 2 (1 – – pnorm(2,0,1)).
- Определить вероятность того, что *x* будет находиться в интервале [2; 3]. Для вычисления можно воспользоваться формулой (pnorm(3,0,1) – pnorm(2,0,1)) или функцией ошибок $(1/2(\mathrm{erf}(3/\sqrt{2}) - \mathrm{erf}(2/\sqrt{2})))$.

Для генерирования случайной величины, распределенной по какому-либо закону, надо воспользоваться функцией r*. Например, для генерирования случайных чисел, распределенных по нормальному закону, следует ввести функцию rnorm(*N*,μ,σ). Здесь *N* – число испытаний.

Функции для вычисления параметров случайных величин

Mathcad имеет ряд функций для вычисления параметров случайной величины *x*:

выборочное среднее – *mean*(*x*);

выборочная медиана – *median*(*x*);

выборочная дисперсия – *Var*(*x*);

среднеквадратичное отклонение – *stdev*(*x*);

максимальное и минимальное значения выборки – *max*(*x*), *min*(*x*);

мода выборки случайной величины – *mode*(*x*).

Построение гистограмм

Для проверки закона распределения случайной величины по экспериментальным данным можно воспользоваться гистограммой.

Построение гистограмм в Mathcad осуществляется с помощью функций hist(int,*x*) и histogram(bin,*x*). Здесь int – вектор начальных точек построения столбиков на гистограмме, int = <минимальное значение случайного числа в выборке> + <шаг>**j*, bin – число столбиков на гистограмме. Пример построения гистограмм приведен на листинге 2.35.

Листинг 2.35. **Построение гистограммы распределения случайной величины**

$N := 1000$ — Число значений случайной величины

$bin := 30$ — Число столбцов на гистограмме

$x := rnorm(N, 0, 1)$ — Вектор случайных величин

$a := floor(min(x))$ — Минимальное значение случайной величины

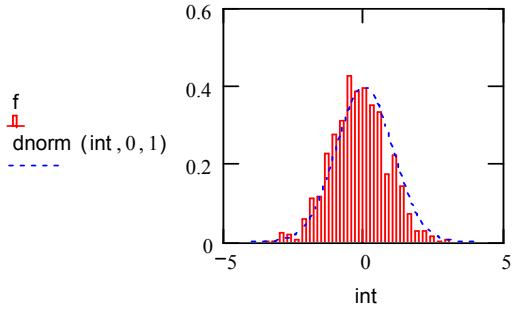
$b := ceil(max(x))$ — Максимальное значение случайной величины

$h := \frac{(b - a)}{bin}$ — Шаг гистограммы

$j := 0 .. bin$ — Переменная цикла

$int_j := a + h \cdot j$ — Вектор начальных значений отрезков разбиения

$f := \frac{1}{N \cdot h} \cdot hist(int, x)$ — Hist - вектор частоты попадания данных в интервалы гистограммы. f - функция нормированная к произведению N*h



Для построения гистограммы необходимо записать программу и вызвать мастера построения гистограмм соответствующей кнопкой панели управления «графики». Чтобы на диаграмме были показаны столбцы, выполните следующее: вызовите контекстное меню, выберите опцию ***Format***, выберите закладку ***Traces*** (***След***) и для опции ***Type*** установите значение ***Bar***.

Функция *histogram* также служит для построения гистограмм, но она проще функции *hist* в применении:

N := 1000 bin := 10 x :=rnorm(N,0,1)

f := histogram(bin, x)

Проверка статистических гипотез

Пусть имеется выборка *N* чисел с нормальным законом распределения и неизвестным значением математического ожидания и дисперсии. В предположении, что математическое ожидание равно 0,2, требуется принять или отклонить эту гипотезу (листинг 2.36).

Листинг 2.36. **Пример проверки гипотезы с помощью распределения Стьюдента**

$N := 50$ $x := rnorm(N, 0, 1)$

$\alpha := 0.1$ $1 - \alpha = 0.9$ $\mu 0 := 0.2$

$t := \frac{mean(x) - \mu 0}{\left(\frac{Stdev(x)}{N}\right)}$ $t = 2.878$

$T := qt\left(1 - \frac{\alpha}{2}, N - 1\right)$ $T = 1.677$

$|t| < T = 0$

Для проверки необходимо задать *уровень критерия* проверки гипотезы – α. Значение α принимают в интервале от 0 до 1. Чем больше значение критерия, тем жестче требования к условию задачи. При очень малом значении критерия гипотеза может быть принята даже в том случае, если она ложная.

Оценка математического ожидания осуществляется с помощью распределения Стьюдента с параметром *N* – 1. Для проверки гипотезы рассчитывается (α/2) – квантиль распределения Стьюдента *T*, который служит критическим значением для принятия или отклонения гипотезы. Если соответствующее выборочное значение меньше *T*, то гипотеза принимается. В противном случае гипотезу следует отклонить.

Пример проверки гипотезы с помощью распределения Стьюдента приведен на листинге 2.36, где *x* – заданный вектор случайных чисел, α – принятый уровень критерия проверки гипотезы, μ_0 – гипотеза о значении математического ожидания.

Если результат в последней строке будет равен 0, то гипотезу следует отклонить, если результат равен 1, то гипотеза принимается. В данном случае гипотезу следует отклонить.

Контрольные вопросы

1. Какие функции Mathcad применяются для работы со случайными числами?
2. Как сгенерировать ряд чисел, распределенных по случайному закону?
3. Изложите алгоритм построения гистограммы распределения случайной величины.
4. Изложите алгоритм проверки статистической гипотезы о значении математического ожидания выборки распределения случайной величины.

2.13. ОБРАБОТКА ЭКСПЕРИМЕНТАЛЬНЫХ ДАННЫХ

При получении экспериментальных данных возможны различные задачи по их обработке и использованию. Обычно эти данные представляются в виде таблицы значений аргумента и соответствующих им значений функции. Значения аргумента, при которых функция определена, называют *узлами интерполяции*.

Интерполяция – выбор функции, совпадающей с экспериментальными данными в узлах интерполяции, и определение значений функции при любых значениях аргумента, не совпадающих с узлами интерполяции.

Экстраполяция (предсказание) – определение значений функции за пределами значений аргумента, представленных в таблице.

Регрессия (сглаживание) – выбор функции, которая наиболее близко подходит к узлам интерполяции, не обязательно проходя через точки $f(x_i, y_i)$.

Фильтрация – подобна задаче сглаживания (сглаживание – частный случай фильтрации), но при этом учитывается, что экспериментальные данные могут содержать шумовую компоненту и даже ошибки (промахи) измерений.

Интерполяция

Постановка задачи

Пусть функция задана в виде таблицы зависимости функции от аргумента. Требуется определить значение функции в точках, отличных от узлов интерполяции.

Очевидно, что можно подобрать бесконечное множество функций, которые будут проходить через узлы интерполяции, имея при этом разное поведение в интервалах между узлами интерполяции, и поэтому результаты будут разными.

Mathcad предоставляет две функции для интерполяции: линейную интерполяцию и кубическую сплайн-интерполяцию.

Линейная интерполяция

При линейной интерполяции узлы интерполяции соединяются прямыми линиями. Линейная интерполяция реализуется посредством функции ***linterp(vx,vy,x)***, где *vx*, *vy* – векторы данных. Для применения данной функции требуется, чтобы шаг между значениями аргументов был постоянный и данные были упорядочены по возрастанию значения аргумента; *x* – аргумент, для которого возвращается значение *y*. Пример использования функции *linterp*:

$$x := (0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6)^T$$

$$y := (-0.121 \quad 0.556 \quad 0.392 \quad 0.809 \quad 0.985 \quad 0.562 \quad 0.916)^T$$

$$a(t) := \text{linterp}(x, y, t) \qquad b := \text{linterp}(x, y, 2.4) \qquad b = 0.559$$

В данном примере *y* и *x* записаны в виде строк, но транспонируются в вектор-столбец.

Кубическая сплайн-интерполяция

Кубическая сплайн-интерполяция позволяет провести через набор точек гладкую кривую так, чтобы в этих точках были непрерывны первая и вторая производные. Интерполяция осуществляется двумя функциями. Вначале вычисляется вектор вторых производных в рассматриваемых точках, затем вычисляется значение функции в точке *x*. Кубическая сплайн-интерполяция реализуется с помощью функции ***interp(vs,vx,vy,x)*** (листинг 2.37). Здесь аргументы функции *vx*, *vy*, *x* имеют тот же смысл, что и для функции ***linterp***, *vs* – вектор вторых производных.

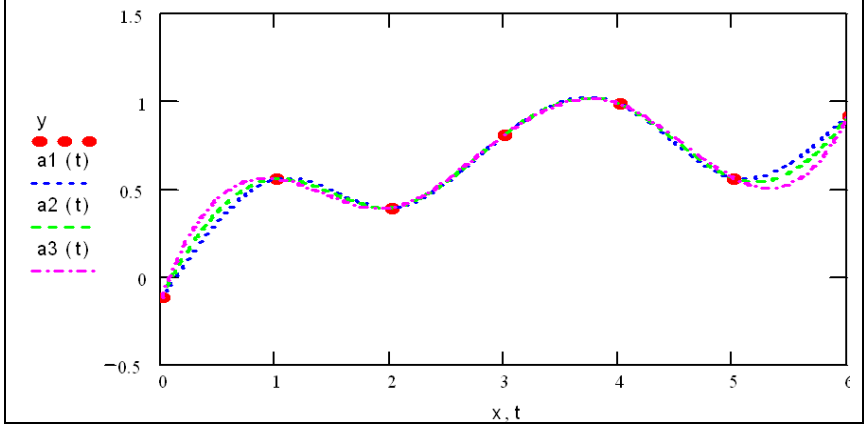
Для построения вектора вторых производных в Mathcad имеется набор из трех функций, которые отличаются лишь граничными условиями: ***lspline(vx,vy)*** генерирует прямую, ***pspline(vx,vy)*** – параболу, ***cspline(vx,vy)*** – кривую, являющуюся кубическим полиномом в граничных точках.

Пример использования кубической сплайн-интерполяции приведен на листинге 2.37.

Примеры рассчитаны на том же наборе данных, что и для линейной интерполяции. В качестве примера можно рассмотреть функцию прогиба балки, закрепленной с двух концов, при постоянной нагрузке, для которой значение прогиба определяется по формуле:

Листинг 2.37. **Пример кубической сплайн интерполяции**

```
s := lspline (x , y)
a1 (t) := interp (s , x , y , t)        b := interp (s , x , y , 2.4)        b = 0.494
s1 := pspline (x , y)
a2 (t) := interp (s1 , x , y , t)       b := interp (s1 , x , y , 2.4)       b = 0.499
s2 := cspline (x , y)
a3 (t) := interp (s2 , x , y , t)       b := interp (s2 , x , y , 2.4)       b = 0.506
```



$$vy_i := F\cdot\frac{(L-a)\cdot\left[\left(vx_i\right)^3-a\cdot(2\cdot L-a)\cdot vx_i\right]-if\left[vx_i>a,L\cdot\left(vx_i-a\right)^3,0\right]}{N\cdot L\cdot E\cdot I}$$

где L – длина балки;

N – число узлов интерполяции;

E – модуль упругости;

I – момент инерции;

vx_i – текущая координата длины балки;

vy_i – текущее значение прогиба балки;

a – точка приложения нагрузки.

$I = 800$, $E = 2 \cdot 10^6$.

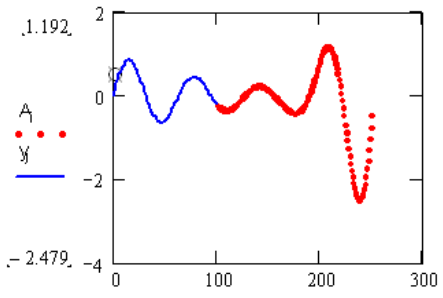
Экстраполяция (предсказание)

Предсказание поведения функции за пределами значений вектора y осуществляется с помощью функции ***predict(y,m,n)***. Здесь y – вектор действительных значений, взятых через равные промежутки, m – количество последовательных элементов вектора y,

согласно которым строится экстраполяция, n – количество элементов вектора предсказаний. Пример предсказания функцией predict приведен на листинге 2.38. Сначала генерируется функция y, моделирующая некоторый известный *наблюдаемый* процесс. На основании функции y функция *predict* вычисляет параметры процесса в будущем.

Листинг 2.38. **Использование функции predict для предсказания**

```
k := 100        j := 0.. k
y_j := e^(-j/100) · sin(j/10)
m := 50         n := 150
A := predict(y , m , n)
i := 0.. n + k
```



Характер функции предсказания будет зависеть от параметра m, то есть того количества точек наблюдения, примыкающих к концу отрезка наблюдения, которое будет учитываться для построения этой функции. При построении графика функции следует обращать внимание на наличие индекса при параметрах A и y.

Регрессионный анализ

При обработке ***экспериментальных*** данных с целью исследования их природы возникает необходимость выразить зависимую переменную в виде некоторой математической функции от одной или нескольких независимых переменных. Данная зависимость получила название «регрессионная модель», или «уравнение регрессии», а методы, позволяющие получить эту зависимость, принято называть методами регрессионного анализа. Методы регрессионного анализа позволяют: производить расчет различного вида регрессионных моделей; проверить гипотезу адекватности модели имеющимся наблюдениям; использовать модель для прогнозирования значений зависимой переменной при новых значениях независимой переменной. В Mathcad существует набор функций, позволяющих рассчитать различные регрессионные модели. В таблице 2.4 представлены функции, используемые при создании регрессионных моделей.

Таблица 2.4

Функции, используемые при создании регрессионных моделей

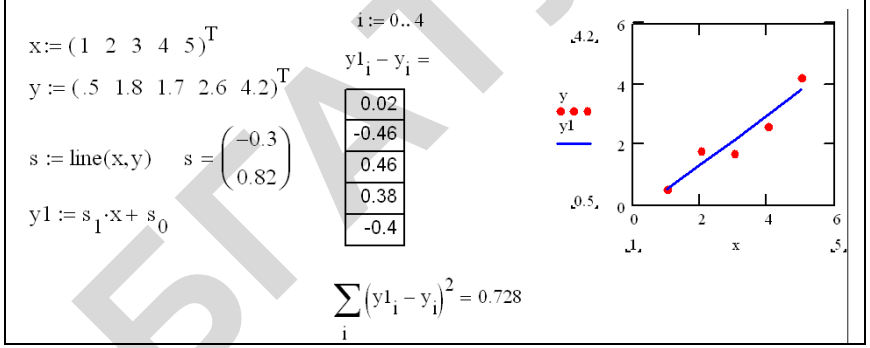
Наименование модели	Вид уравнения регрессии	Функция Mathcad
Линейная	$y(x) = a*x + b$	line(*x*,*y*)
Полиномиальная *n*-й степени	$y(x) = \sum_{i=0}^{n} a_n x^n$	*s* := regress(*x*,*y*,*n*) interp(*s*,*x*,*y*,*t*)
Фрагменты полиномов 2-й степени	$y_i(x) = \sum_{i} a_i x^2 + b_i x + c_i$	*s* := loess(*x*,*y*,*span*) interp(*s*,*x*,*y*,*t*)
Экспоненциальная	$y(x) = a * e^{b*x} + c$	expfit(*x*,*y*,*g*)
Логистическая функция	$y(x) = a/(1 + b * e^{-c*x})$	lgsfit(*x*,*y*,*g*)
Синусоидальная	$y(x) = a * \sin(x + b) + c$	sinfit(*x*,*y*,*g*)
Степенная	$y(x) = a * x^b + c$	pwfit(*x*,*y*,*g*)
Логарифмическая	$y(x) = a * \ln(x + b) + c$	logfit(*x*,*y*,*g*)
Логарифмическая короткая	$y(x) = a * \ln(x) + b$	lnfit(*x*,*y*,*g*)

Рассмотрим суть параметров, используемых в качестве аргументов в функциях. В каждой функции используются два вектора исходных данных: *x* – вектор независимых переменных, *y* – вектор зависимых переменных. Количество элементов векторов *x* и *y* должно быть одинаково. Параметр *g* является вектором начальных приближений для неизвестных функции регрессии и вычисляется специальными функциями, *t* – значение аргумента полиномиальной регрессии.

Линейная регрессия

Линейная регрессия является самым простым видом регрессии. Она представляет на плоскости график прямой линии, проведенной на множестве точек исходных данных (листинг 2.39). Таких прямых на плоскости можно провести бесконечное множество. Однако из них надо выбрать такую прямую, которая наиболее близко расположена ко всем точкам исходных данных. В качестве критерия близости прямой к точкам исходных данных применяют метод наименьших квадратов. С помощью данного метода определяется прямая, для которой сумма квадратов отклонений исходных данных от расчетных, определенная на всем множестве точек, будет наименьшая.

***Листинг 2.39*. Пример линейной регрессии**



Полиномиальная регрессия

Полиномиальная регрессия означает приближение данных (x_i, y_i) полиномом *k*-й степени. При $k = 1$ полином является прямой линией, при $k = 2$ – параболой, при $k = 3$ – кубической параболой.

Полиномиальная регрессия осуществляется функцией *regress(x,y,k)* и *interp(s,x,y,t)*. Функция *regress* вычисляет вектор коэффициентов *s* для построения полиномиальной регрессии данных, а функция *interp* формирует результат полиномиальной регрессии.

Разновидностью полиномиальной регрессии является регрессия отрезками полиномов. В этом случае вспомогательный вектор *s* вычисляется с помощью функции *loess*. Параметр *span* функции *loess* определяет размер отрезков полиномов, на которой строится конкретный фрагмент полинома 2-й степени. Оптимальное значение *span*, предлагаемое справочной системой Mathcad, равно 0,75, но в каждом конкретном случае рекомендуется путем вариантных расчетов подобрать наилучшее значение span. Параметр span задает степень сглаженности данных.

После определения регрессионных зависимостей, актуальным является выбор из их совокупности наилучшей функции, с точки зрения адекватности описания исходных экспериментальных данных. В качестве критерия, позволяющего выбрать наилучшую регрессионную модель, предлагается использовать коэффициент детерминации R^2, численно равный (1 – коэффициент корреляции в квадрате). Значение коэффициента корреляции в Mathcad позволяет рассчитать функция corr(*A*,*B*), где *A* и *B* – два вектора значений.

Кроме рассмотренных функций регрессии Mathcad имеет и ряд функций трехпараметрической регрессии. При использовании этих

функций помимо массива данных требуется задать некоторые начальные значения коэффициентов *a*, *b*, *c*. Каждая из этих функций выдает вектор *g* уточненных параметров a, *b*, *c*. Имеется также возможность осуществлять регрессию в виде линейной комбинации $C_1f(x)_1 + C_2f(x)_2 + ... + C_if(x)_i + ...$, где $f(x)_i$ – любые функции пользователя, а C_i – коэффициенты, подлежащие определению.

Сглаживание и фильтрация

Сглаживание и фильтрация – более общий случай регрессии. Сглаживание и фильтрация позволяют удалить лишние узлы, вызванные помехами или шумами.

Для сглаживания применяются следующие функции:

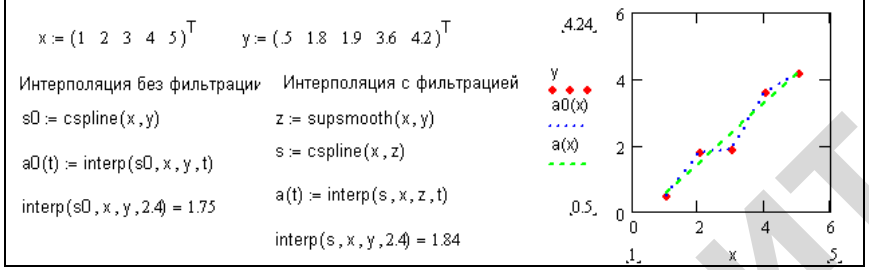
сглаживание алгоритмом «бегущих медиан» – medsmooth(*y*,*b*);

сглаживание на основе функции Гаусса – ksmooth(*x*,*y*,*b*);

локальное сглаживание адаптивным алгоритмом – supsmooth(*x*,*y*).

Часто бывает полезно совместить сглаживание с последующей интерполяцией. В этом случае сначала выполняют сглаживание функции с помощью одной из функций, а затем интерполяцию. Примеры использования функций фильтрации и сглаживания приведены на листингах 2.40, 2.41.

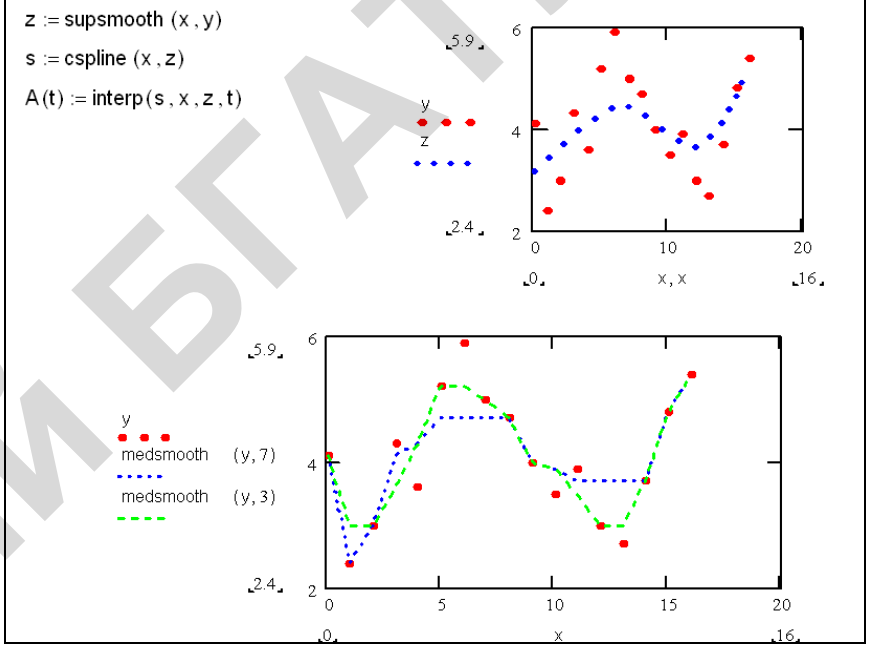
***Листинг 2.40.* Фильтрация**



Контрольные вопросы

1. Поясните, что такое интерполяция, для какой цели она применяется?
2. Какие функции используются в Mathcad для интерполяции табличных данных?
3. Чем отличается кубическая сплайн-интерполяция от линейной интерполяции?
4. Что такое предсказание, какие функции используются для предсказания?
5. Что такое регрессия, чем она отличается от интерполяции?

***Листинг 2.41.* Сглаживание и фильтрация**



6. Для какой цели применяется фильтрация экспериментальных данных?
7. Какие функции применяются для фильтрации и сглаживания в Mathcad?

2.14. РЕШЕНИЕ ЗАДАЧ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

К классу задач линейного программирования относятся задачи, в которых требуется оптимизировать (определить максимум или минимум) целевую функцию вида:

$$z = f(x_1, x_2, \ldots, x_n) = c_1x_1 + c_2x_2 + \ldots + c_nx_n$$

при следующих ограничениях:

$$a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n - b_i \geq 0 \ (i = 1, 2, \ldots, m);$$

$$x_k \geq 0 \ (k = 1, 2, \ldots, n).$$

Первый вид ограничений называют функциональными ограничениями, второй вид ограничений называют ограничениями на значения параметров. Первый вид ограничений устанавливает, как правило, ограничения на значения используемых ресурсов (материальных, финансовых, трудовых). Второй вид ограничений устанавливает требования к значениям переменных (целые, неотрицательные, равные или неравные нулю).

Одним из способов решения подобных задач в Mathcad является использование блока *Given* с функциями *minimize* и *maximize*.

В качестве примера рассмотрим следующую задачу.

Пример 2.22. Задача линейного программирования.

Фабрика выпускает два типа красок – I и E. Для производства красок используются два компонента – A и B.

Максимальные суточные запасы этих компонентов: компонент A – 6 тонн; компонент B – 8 тонн.

Расходы компонентов A и B на производство 1 тонны краски следующие:

на производство краски I расход компонента A – 2 т, компонента B – 1 т;

на производство краски E расход компонента A – 1 т, компонента B – 2 т;

Суточный спрос на краску I никогда не превышает 2 тонн в сутки.

Суточный спрос на краску E не больше спроса на краску I.

Оптовые цены на краску I – 2000 рублей за тонну, а на краску E – 3000 рублей за тонну.

Определить максимальный доход фабрики от продажи краски.

Решение.

Производственный процесс должен обеспечить выпуск всей продукции. Экономико-математическая модель задачи и ее решение приведены на листинге 2.42.

Контрольные вопросы

1. Каким образом решаются в Mathcad задачи оптимизации?
2. Что такое целевая функция?
3. Какие виды ограничений используются в задачах оптимизации?
4. Как выполняется поиск оптимального значения в Mathcad?

Листинг 2.42. **Решение задачи линейного программирования**

ORIGIN:= 1

Задание целевой функции

$F(x) := 2000 \cdot x_1 + 3000 \cdot x_2$

Задание начальных приближений

$x := (1 \quad 1)^T$

Given

Ввод ограничений

$x_1 \cdot 2 + x_2 \le 6$

$x_1 \cdot 1 + x_2 \cdot 2 \le 8$

$x_1 \le 2$

$x_1 \ge x_2$

$x_1 \ge 0$

$x_2 \ge 0$

Решение

$x := \text{Maximize}(F, x) \qquad x = \begin{pmatrix} 2 \\ 2 \end{pmatrix}$

Значение целевой функции $F(x) = 1 \times 10^4$

Заключение

Краткое знакомство с системой компьютерной математики Mathcad показывает, что круг задач, решаемых системой, достаточно обширен. Для полноценного и грамотного использования этой программы пользователь должен обладать определенной математической подготовкой. Для расширения сферы применения программы имеется возможность разрабатывать прикладные программы с использованием встроенного языка программирования. В умелых руках программа позволит эффективно решать различные математические и экономические задачи.

3. АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ

3.1. ЭТАПЫ ЖИЗНЕННОГО ЦИКЛА ПРОГРАММ

Процесс подготовки задачи к решению на ЭВМ распадается на несколько этапов. Конечным результатом этой работы является программа. Программа может находиться в эксплуатации длительное время или короткое время в зависимости от ее назначения, качества. Когда отпадает надобность в использовании данной программы, она снимается с эксплуатации, поэтому принято говорить о жизненном цикле программы. Основными этапами этого цикла являются (рис. 3.1): постановка задачи, разработка математической модели, разработка алгоритма, программирование, отладка программы, передача программы в эксплуатацию и научно-техническое сопровождение (НТС) программы, завершение жизненного цикла.

Рис. 3.1. Этапы жизненного цикла программы

Постановка задачи

Задача – это проблема, подлежащая решению.

Постановка задачи – один из важных и ответственных этапов жизненного цикла. На данном этапе определяются основные цели и функции, выполнение которых должна обеспечивать программа, исходные данные, требования к исходным данным, выходные данные. От качественной проработки всех вопросов на данном этапе зависят в конечном итоге качество программы и сроки ее разработки.

Конечно, в процессе работы над программой многие вопросы могут уточняться, дополняться и т. д., но время разработки программы при этом увеличивается. Различают содержательную и математическую постановку задачи.

Содержательная постановка задачи заключается в формулировке задачи на естественном языке.

Математическая постановка задачи сводится к точному описанию исходных данных, условий задачи и целей ее решения с использованием математических выражений в общем виде. При этом должен применяться системный подход, то есть предмет должен быть исследован всесторонне, учтены все внешние и внутренние связи и их влияние на конечные результаты.

Любую задачу можно представить в виде «черного ящика» (рис. 3.2), на вход которого поступают исходные данные – вектор $\overline{X}$, ограничения на входные параметры – вектор $\overline{Q}$, требования к входным и выходным параметрам – вектор $\overline{T}$, а выходом является вектор $\overline{Y}$. Ни для заказчика, ни для разработчика программы на данном этапе не имеет значения, каким образом будет обрабатываться информация.

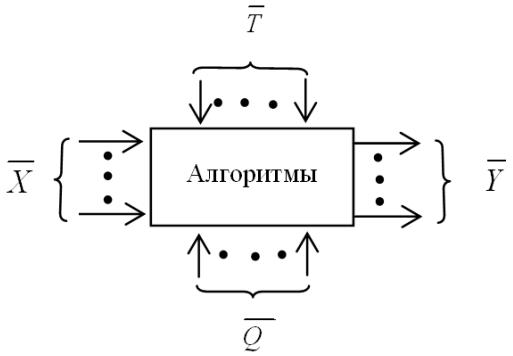


Рис. 3.2. Модель решения задачи

Разработка математической модели

На данном этапе производится декомпозиция задачи, формализация, разработка математической модели, выбор метода решения.

Под ***декомпозицией*** понимается разделение задачи на простые блоки, каждый из которых может разрабатываться самостоятельно

и связан с другими частями программы только входными и выходными данными.

Для деления задачи на блоки чаще всего используется функциональный подход. Например, в каждой вычислительной задаче можно выделить такие блоки, как ввод данных, вычислительный блок, блоки сохранения результатов вычислений на дисках, анализа результатов вычислений, графического представления результатов вычислений, печати результатов.

Под ***формализацией*** в общем случае понимается сведение некоторого содержания к выбранной форме описания. В нашем случае под *формализацией* задачи понимают представление исходных данных и условий решения задачи, сформулированных словесно, в виде, удобном для ввода и последующей обработки на ЭВМ.

Математическая модель задачи представляет собой совокупность математических выражений, описывающих данную задачу. В общем виде математическую модель можно представить следующим образом:

$$\underset{\min}{\text{Функция\_цели}} = F(x_1, x_2, \ldots, x_n, q_1, q_2, \ldots, q_m, t_1, t_2, \ldots, t_p). \qquad (3.1)$$

Функциональные ограничения:

$$q_1 \le \varphi_1(x_1, x_2, \ldots, x_n);$$
$$q_2 \le \varphi_2(x_1, x_2, \ldots, x_n)$$
$$q_m = \varphi_m(x_1, x_2, \ldots, x_n)$$

Ограничения на значения параметров:

$$x_2 \ge T_2;$$
$$\ldots$$
$$x_n \ge T_n.$$

Метод решения задачи выбирается из известных методов. Если для решения данной задачи имеется несколько методов, то выбирается тот метод, который обеспечивает получение решения за меньшее время и с заданной точностью. Если для решения данной задачи нет известных методов, то необходимо разработать такой метод или вернуться на первый этап и уточнить задачу, исходные данные и требования к ним.

Разработка алгоритма программы

На данном этапе разрабатывается алгоритм решения задачи, то есть определяется последовательность действий. Разработка алгоритма предполагает определение состава функциональных модулей

и формирование общей схемы алгоритма, разработку алгоритмов функциональных модулей. В зависимости от сложности задачи алгоритм представляют вначале в общем виде (укрупненном). Затем каждый из блоков алгоритма разбивается на более мелкие задачи таким образом, чтобы на конечном этапе получить базовые схемы алгоритмов. Такой метод проектирования называется *нисходящей* разработкой алгоритма (проектирования).

Основные подходы к разработке алгоритмов и программ:

структурное проектирование;

информационное моделирование предметной области и связанных с ней приложений;

объектно-ориентированное проектирование.

В основе структурного проектирования лежит последовательная декомпозиция, целенаправленное структурирование на отдельные составляющие. Типичными методами структурного проектирования являются:

нисходящее проектирование, кодирование и тестирование программ;

модульное программирование;

структурное программирование.

Нисходящее проектирование предполагает разложение общей функции обработки данных на простые функциональные элементы «сверху вниз» (рис. 3.3). В результате образуется *функциональная схема алгоритма*.

При реализации принципа нисходящего проектирования программа разрабатывается в следующей последовательности:

- На основе анализа задача разбивается на подзадачи, выделяются уровни и подуровни, функции отдельных блоков. В результате составляется иерархическая структура – функциональная структура программы.
- Для каждого уровня:

разрабатывается математическая модель и определяется метод решения задачи;

определяются основные блоки программы и разрабатывается укрупненная схема алгоритма;

определяются входные и выходные переменные, общие для данного уровня;

разрабатываются схемы алгоритмов для реализации основных блоков программы, определяются входные и выходные переменные соответствующего уровня;

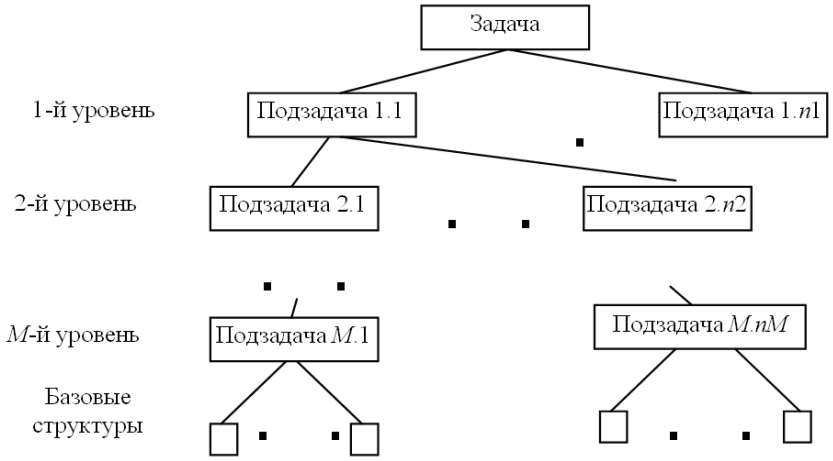


Рис. 3.3. К понятию «нисходящее проектирование»

определяется порядок взаимодействия с другими блоками.

Этот процесс продолжается, пока схема алгоритма не будет доведена до базовых структур соответствующего языка программирования.

Модульное программирование основано на понятии модуля. *Модуль* – логически взаимосвязанная совокупность функциональных элементов, оформленных в виде отдельных программных модулей, имеющих один вход и один выход. Модульное программирование позволяет значительно сократить время на отладку программы.

Структурное программирование основано на модульной структуре программного продукта и типовых управляющих структурах алгоритмов обработки данных различных программных модулей. Структурное программирование применяется в основном при программировании отдельных модулей и заключается в переводе алгоритма программы на алгоритмический язык с использованием определенных конструкций языка программирования.

Информационное моделирование предметной области и связанных с ней приложений предполагает определение состава и способа представления исходных данных и результатов вычислений.

Объектно-ориентированное проектирование основано на использовании при программировании *объектов* – функциональных программных модулей, которые на экране монитора представлены в виде элементов, например, кнопок, списков,

переключателей и т. п., обладающих определенной совокупностью свойств, методов и событий.

Программирование

Программа – упорядоченная последовательность команд (инструкций) компьютера для решения задач.

В общетеоретическом плане *программирование* – это теоретическая и практическая деятельность, связанная с созданием программы. Изучение этих вопросов выходит за рамки настоящего пособия.

В узком смысле под программированием понимается запись алгоритма с использованием команд и операторов одного из языков программирования – *кодирование*.

Отладка программы

Отладка программы заключается в проверке правильности функционирования алгоритма решения задачи с помощью контрольных примеров – тестов, результаты решения которых заранее известны; устранении обнаруженных синтаксических и логических ошибок. Отладка программ – очень важный и ответственный этап. Чем полнее разработаны тестирующие программы, тем больше вероятность, что программа будет безошибочно работать в процессе эксплуатации.

Научно-техническое сопровождение

Научно-техническое сопровождение программы предусматривает контроль над работой программы и устранение ошибок, обнаруженных в процессе эксплуатации, доработку программы и ее совершенствование в соответствии с требованиями заказчика.

3.2. СХЕМА АЛГОРИТМА

Определение и основные свойства

Алгоритм – точное, однозначное предписание последовательности действий (операций), приводящее к решению задач данного класса за конечное число шагов или заданное время.

Основными свойствами алгоритма являются дискретность, определенность, массовость, результативность, эффективность.

Дискретность – данное свойство вытекает из самой сущности цифровой вычислительной техники. Процесс преобразования исходных данных и результата осуществляется дискретно, так что значения величин в каждый следующий момент времени получа-

ются по определенным правилам из значений величин, имевшихся в предшествующие моменты времени.

Определенность *(детерминированность)* – алгоритм не должен допускать различных толкований. Каждое правило алгоритма должно быть четким и однозначным. Действия, которые необходимо произвести, должны быть строго и недвусмысленно определены в каждом конкретном случае.

Массовость – алгоритм должен позволять решать все задачи данного класса. Например, решение квадратного уравнения при любых значениях коэффициентов. Если алгоритм имеет ограничения, то это указывается в его описании. Например, то же квадратное уравнение может быть решено только в области действительных чисел.

Результативность – алгоритм должен приводить к решению задачи за конечное число шагов или заданное время. Если задача не имеет решений, то пользователь должен получить соответствующее сообщение.

Эффективность – это мера качества алгоритма. Критериями эффективности алгоритма являются: простота, точность получаемого результата и время его реализации.

Представление алгоритмов

Для представления алгоритмов используются различные способы:
словесный;
описание на алгоритмическом языке;
представление в виде схем алгоритмов;
представление в предикатной форме записи;
операторная запись алгоритма.

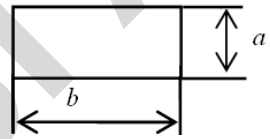
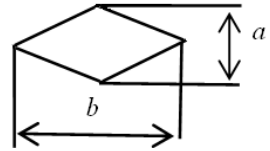
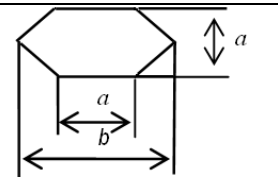
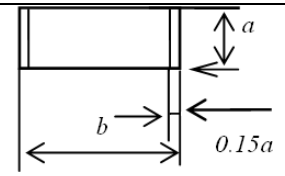
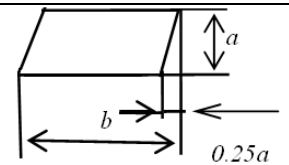
Словесная форма представления алгоритма самая простая. Словесная форма позволяет подробно описать последовательность действий, рассмотреть все возможные варианты. Недостатком данной формы описания является ее громоздкость и невозможность в ряде случаев перейти непосредственно от алгоритма к программе. Тем не менее, при разработке сложных алгоритмов без использования словесной формы нельзя обойтись.

Запись алгоритма на алгоритмическом языке используется, как правило, при обучении. Эта форма записи позволяет описать алгоритм на языке, близком к естественному языку. В качестве примера алгоритмического языка можно назвать язык Мега-Е, используемый в школах и в средних учебных заведениях для обучения программированию. В состав служебных слов входят такие

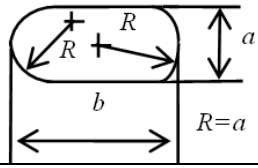
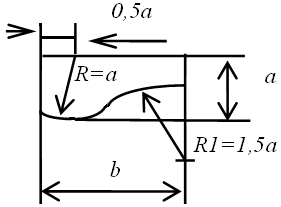
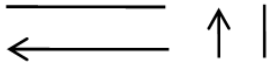
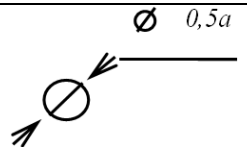
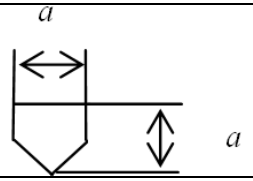
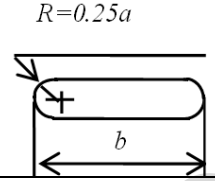
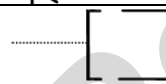
слова, как **алг** – алгоритм, **дано** – ввод данных, **надо** – цель выполнения алгоритма и др. Основные конструкции этого языка приведены в таблице 3.1.

Таблица 3.1

Основные графические элементы схем алгоритмов

Наименование	Обозначение	Функция
Процесс		Выполнение операций или группы операций, в результате которых изменяется значение, форма представления или расположение данных
Решение		Выбор направления выполнения алгоритма или программы в зависимости от некоторых переменных условий
Модификация		Выполнение операций, меняющих команды, или группы команд с целью воздействия на некоторую последующую функцию (установка переключателя, модификация индексного регистра или инициализация программы)
Предопределенный процесс		Использование созданных ранее или отдельно описанных алгоритмов или программ
Ввод – вывод		Преобразование данных в форму, пригодную для обработки (ввод) или отображения результатов обработки (вывод)

Окончание табл. 3.1

Наименование	Обозначение	Функция
Дисплей		Ввод данных с подключенного к компьютеру дисплея или вывод данных на дисплей
Документ		Ввод-вывод данных, носителем которых служит бумага
Линии потока		Указание на последовательность связи между символами. Можно без стрелки, если линия направлена слева направо или сверху вниз, со стрелкой в остальных случаях
Соединитель		Указание на связь между прерванными линиями потока, соединяющими операторы в пределах листа
Соединитель		Указание на связь между прерванными линиями потока, соединяющими операторы на разных листах
Пуск-останов		Начало, конец, прерывание процесса обработки данных или выполнения программы
Комментарий		Связь между элементами схемы с пояснениями

Предикатная форма записи алгоритма удобна при оформлении документации. Она позволяет представить алгоритм в компактной форме, но не обладает наглядностью, например:

$$P(x < y, a, b);\ P1(x, y, g, h).$$

Операторный (язык операторных схем) – похож на предикатный способ записи алгоритма. При использовании данного способа записи алгоритмов вычислительный процесс представляется в виде последовательности символов (операторов). Эти символы обозначают группы стандартных или нестандартных операций, реализующих законченную процедуру с указанием связи между отдельными операторами. Все операторы выполняются последовательно слева направо, стрелки указывают переход от логического оператора (проверки), точка с запятой означает конец варианта или отсутствие связи между операторами, например:

$$C = P_1 \rightarrow A_1 \rightarrow П_{ч1};\ A_2 \rightarrow П_{ч2}.$$

Этот способ записи алгоритма значительно упрощает программирование, при этом вместо операторов подставляют соответствующие команды. Но данный способ имеет малую наглядность.

Схема алгоритма. На практике для представления отчетов и оформления проектной документации чаще используются схемы алгоритмов. Схема алгоритма является одним из способов наглядного представления алгоритма с помощью специальных элементов, предусмотренных единой системой конструкторской документации (ЕСКД). Некоторые из этих элементов приведены в таблице 3.2.

Размер стороны *a* выбирается из ряда 10, 15, 20, … мм с шагом 5 мм. Размер стороны *b* принимается равным 1,5*a*, допускается 2*a*. ГОСТ не накладывает строгих ограничений на размеры элементов схем.

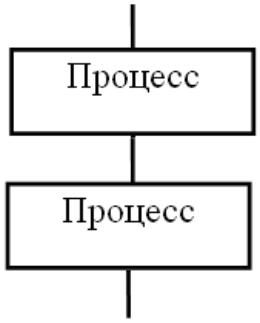
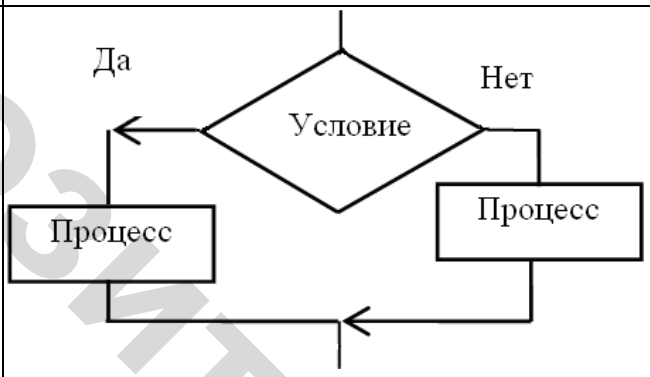
Элементы схемы объединяются линиями потока. Стрелки на линиях потока указываются, если поток направлен справа налево или снизу вверх. Допускается изображать стрелки во всех направлениях. Стрелка проставляется на линии потока в конце потока.

Блок «Начало» имеет только выход. Блок «Конец» имеет только вход. Блок процесса имеет один вход (сверху/снизу) и один выход (снизу/сверху). Блок выбора имеет один вход (сверху или снизу) и два выхода (снизу/сверху и влево или вправо).

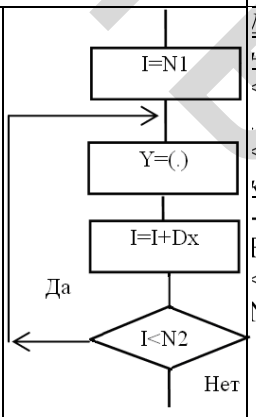
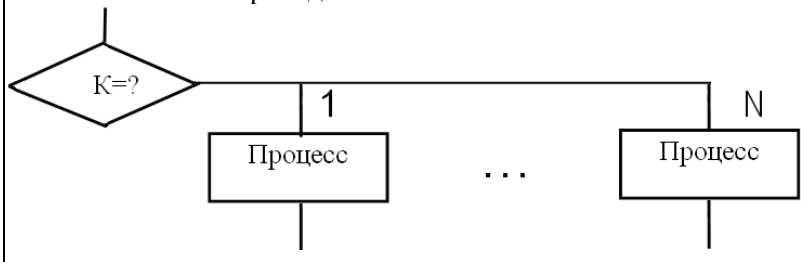
Блоки в схеме алгоритма нумеруются. Блок «Начало» не нумеруется. Номера блоков проставляются у левого верхнего края блока или в разрыве линии.

Таблица 3.2

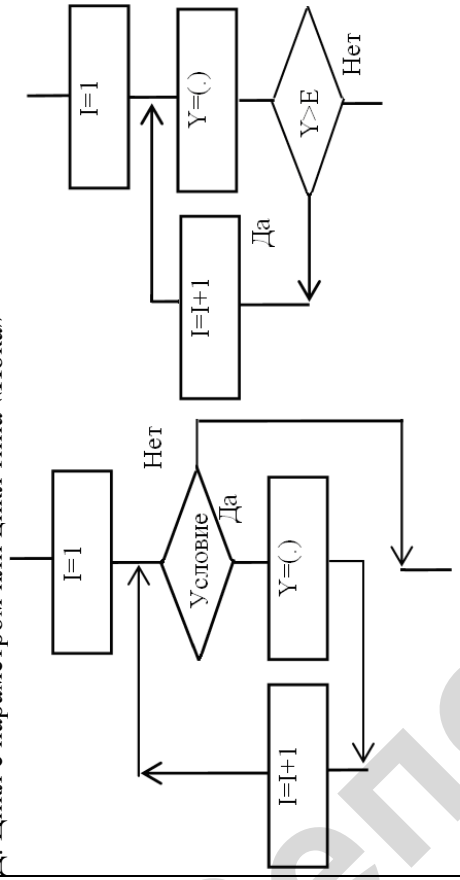
Представление базовых структур алгоритмов

Схема алгоритма		Запись на алгоритмическом языке
А. Следование		*Нач* <выражение> ... <выражение> *Кон*
Б. Выбор		*Если* <условие> *то* <выражение> *иначе* <выражение> *Все*

Продолжение табл. 3.2

Схема алгоритма		Запись на алгоритмическом языке
В. Цикл с заданным числом повторений (цикл «ДО»)		*Для* I от N1 до N2 шаг Dx *цц* <выражение> ... <выражение> *кц* ---------- For I = N1 TO N2 Step Dx <Тело цикла> Next I
Г. Вычисляемый переход		*Выбор* при условии 1:<выраж.> ... при условии N:<выраж.> *Все*

Окончание табл. 3.2

Схема алгоритма	Запись на алгоритмическом языке
Д. Цикл с параметром или цикл типа «Пока»  а) с предусловием	*Пока* <условие> *ци* <выражение> ... <выражение> *ка*
  б) с постусловием	While <условие> <Тело цикла> Wend

Согласно требованиям ГОСТ 19.701–90 «Схемы алгоритмов, программ, данных и систем, условные обозначения и правила выполнения Единой системы программной документации» схемы циклов в программной документации изображаются с помощью двух блоков границ циклов. Примеры блоков границ циклов и их применения для изображения циклов программ приведены на рисунке 3.4. В тексте пособия в учебных целях циклы будут представляться в виде, отражающем их внутреннюю структуру (см. табл. 3.1).

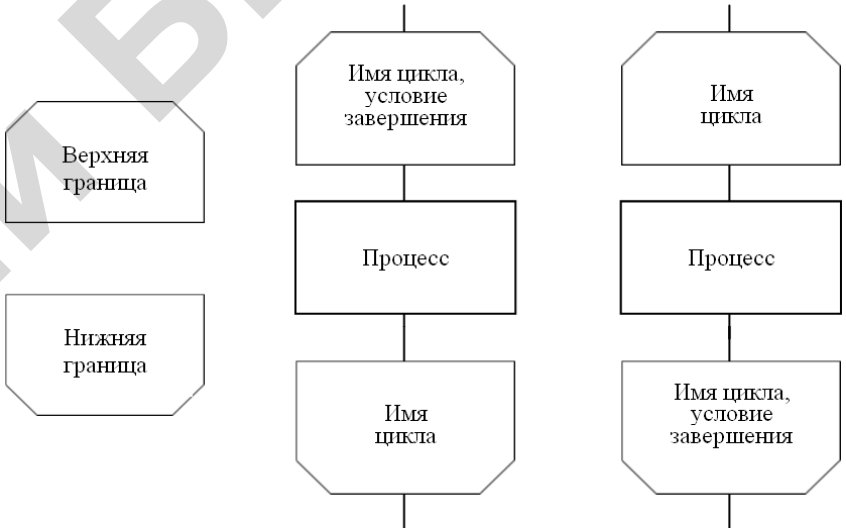


Рис. 3.4. Блоки границ циклов и примеры их применения

При программировании многие структуры алгоритмов повторяются. Различают три основные (базовые) структуры алгоритмов: линейную, выбор и цикл. Эти базовые структуры одинаковы для всех языков программирования. Отличия могут заключаться только в форматах используемых операторов. Базовые структуры схем алгоритмов приведены в таблице 3.1.

Линейная структура представляет собой последовательность операторов, следующих один за другим, ветвления и циклы отсутствуют.

Выбор (или решение) – предполагает выбор одной альтернативы из двух. При этом процесс может быть только в одной ветви –

неполная форма оператора выбора. В этом случае, если условие истинно, то выполняется процесс, в ином случае осуществляется переход на оператор, следующий за оператором выбора. Как разновидность операторов выбора можно указать структуры алгоритмов, позволяющие осуществлять выбор не из двух, а из множества альтернатив. Структура такого алгоритма приведена в таблице 3.1*д*.

Цикл – представляет собой вычислительный процесс, который повторяется многократно при изменении одного параметра – переменной цикла I.

В зависимости от условия окончания цикла циклы делятся на циклы с заданным числом повторений (циклы типа «ДО») и циклы с параметром (бесконечные циклы или циклы типа «Пока»). В циклах типа «ДО» число повторений известно перед началом цикла и условием окончания цикла является выполнение заданного числа повторений. В циклах типа «Пока» число повторений не известно заранее, условием окончания цикла является достижение некоторой переменной, вычисляемой в теле цикла, определенного, наперед заданного значения.

В зависимости от момента, когда проводится проверка условия окончания цикла, циклы делятся на циклы с предусловием и циклы с постусловием. В циклах с предусловием проверка условия окончания цикла проводится в начале цикла, а затем выполняется тело цикла. В циклах с постусловием проверка условия окончания цикла выполняется после выполнения тела цикла.

Контрольные вопросы

1. Назовите основные этапы жизненного цикла программы.
2. Охарактеризуйте основные этапы жизненного цикла программы.
3. Приведите определение алгоритма, назовите и поясните основные свойства алгоритма.

Заключение

В настоящем разделе мы познакомились с основными свойствами алгоритма, способами его представления и базовыми структурами алгоритма. Таких структур четыре: линейная, выбор, цикл, вычисляемый переход. При выполнении схем алгоритмов следует руководствоваться требованиями ГОСТ на выполнение электронных схем.

3.3. ПРОГРАММИРОВАНИЕ

3.3.1. Линейные программы

Линейные программы представляют собой, как уже отмечалось, последовательность операторов; циклов и условных переходов нет. Структурная схема такой программы включает блок ввода данных, блок вычислений и блок вывода результатов (рис. 3.5).

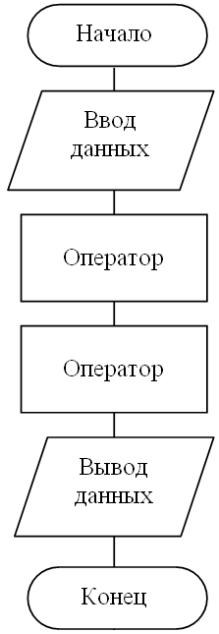


Рис. 3.5. Схема алгоритма линейной программы

Операторы ввода данных

Для ввода данных используются операторы LET.

Оператор ***LET*** служит для присваивания значений переменным. Формат оператора:

LET <имя переменной> = <выражение>.

Оператор LET может быть опущен, поэтому выражения:

$$LET\ A = \exp(x) + \sin(x)$$

и

$$A = \exp(x) + \sin(x)$$

эквивалентны.

Сущность выполнения оператора *LET* состоит в следующем: переменной, имя которой указано слева от знака =, присваивается значение выражения, записанного справа от знака «равно».

Примечание. Операторы, рассматриваемые в примерах, относятся к языку программирования Visual Basic 6.0. Для проверки работы операторов или приводимых фрагментов программ необходимо запустить программу Visual Basic 6.0, открыть окно Программы (Code), поместить текст программы в обработчик события Click формы (или кнопки), запустить программу командой *Run, Start* и после появления формы щелкнуть по ней мышью.

Оператор ***InputBox*** служит для ввода данных с клавиатуры в режиме диалога с пользователем. Простейший формат оператора для ввода переменных имеет следующий вид:

а) при вводе строки символов:

<имя переменной>InputBox(“текстовое сообщение”);

б) при вводе числовой информации:

<имя переменной>Val(InputBox(“текстовое сообщение”)).

Здесь Val – функция для перевода символьного выражения в числовое. При выполнении оператора *InputBox* на экран выводится окно диалога, в котором необходимо ввести требуемую числовую величину; <текстовое сообщение> – сообщение о том, что требуется ввести в строку ввода. При вводе символьных переменных функция Val не используется.

Пример 3.1. Использование оператора InputBox:

C = InputBox(“Введите фамилию автора книги”);

CK = Val(InputBox(“Введите стоимость книги”)).

Оператор вывода данных

Для вывода данных используются операторы ***Print***.

Оператор *Print* выводит данные на форму.

Формат оператора Print:

Print “Комментарий” [;/ /,] <список выражений> [;/,].

Комментарий служит для пояснения выводимой информации; символы [;/ /,] – разделители. Здесь // означает пробел. Если в качестве разделителя используется символ «;» или пробел, то выводимый текст размещается один возле другого без пробелов, при выводе чисел одна позиция резервируется для знака числа. Если в качестве разделителя используется запятая, то каждое новое сообщение печатается в новой зоне. Оператор *Print* формирует выводную строку, которая разбита на 5 зон по 14 символов. Это позволяет размещать выводимую информацию по колонкам. Если текст не умещается в зоне, то он занимает соседнюю зону, а новая порция информации начинает печататься в соседней свободной зоне. В список выражений может помещаться любая информация, в том числе и алгебраические выражения. Выводимые выражения в списке разделяются запятыми. В конце строки могут быть расположены управляющие символы: точка с запятой или запятая. При наличии этих управляющих символов курсор остается в текущей позиции, и следующий оператор Print будет выводить информацию с этой позиции. Действуют эти управляющие символы так же, как и управляющие символы после комментария.

В качестве выражений могут использоваться константы, переменные, функции, выражения. Для преобразования числовых данных в строку символов используется функция *Str*.

При отладке программы для вывода данных можно использовать метод *Print* объекта *Debug*:

Debug.Print <список переменных>.

Переменные в списке разделяются запятыми

Пример 3.2. Использование оператора Print

```
Print “X =”; x ' выводится значение x
Print “X =”;x, “Y =”;y ' выводятся значения x и y в соседних зонах
Print “X =” x, “Y =”;Sin(x) ' выводится значение x, вычисляется и
                            ' выводится значение функции Sin(x)
Print “Автор”; S ' выводится переменная символьного типа
Debug.Print x, y ' данные выводятся в окно непосредственного наблюдения
```

Примеры линейных программ

Пример 3.3. Даны переменные A и B. Требуется поменять их значения.

Решение:

1. Выбор метода решения. Выполнить обмен данными с использованием вспомогательной переменной. Словесный алгоритм:

Объявить переменные A, B и T, где T – вспомогательная переменная. Сохранить значение переменной A в переменной T, присвоить переменной A значение B, присвоить переменной B значение вспомогательной переменной T.

2. Алгоритм решения будет включать пять операторов (рис. 3.6).

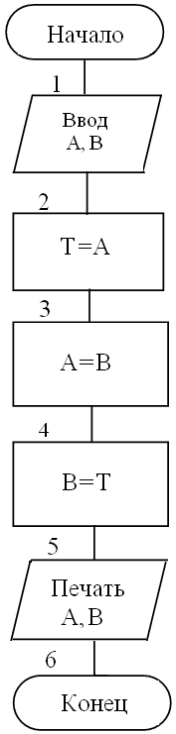


Рис. 3.6. Схема алгоритма примера 3.3

Запись алгоритма решения задачи на алгоритмическом языке

алг обмен_данными
дано A, B
надо поменять значения A и B
нач
T := A

A := B
B := T
рез A, B
кон

3. Запись алгоритма на языке программирования (Visual Basic 6.0):

```
Dim a As Single, b As Single, T As Single
a = Val(InputBox("Введите значение A"))
b = Val(InputBox("Введите значение B"))
T = a
a = b
b = T
Print "a =", a, "b =", b
```

Примечание: в первой строке переменные a, b, T объявлены как переменные вещественного типа.

4. Для отладки программы достаточно ввести значения a и b и проконтролировать результат.

Пример 3.4. Вычислить площадь треугольника, если известны длины сторон *a*, *b*, *c*.

Решение:

1. *Выбор метода решения.*

Задача может быть легко решена, если воспользоваться формулой Герона: площадь треугольника равна корню квадратному из произведения полупериметра на разности полупериметра и каждой из сторон треугольника.

2. *Математическая модель* задачи может быть представлена двумя формулами:

$$p = \frac{a+b+c}{2}; \qquad (3.2)$$

$$S = \sqrt{p(p-a)(p-b)(p-c)}\,. \qquad (3.3)$$

Для оформления отчетной документации необходимо описать все переменные, используемые в программе, по форме, представленной на рисунке 3.7. Это позволит также систематизировать исходные данные и результаты вычислений, избежать ошибок при программировании, например, повторного использования имен переменных. Схема алгоритма решения данной задачи аналогична схеме, представленной на рисунке 3.6.

STREUG (Имя программы)	Вычисление площади треугольника (Назначение программы)		Лист	1
			Листов	1
Переменная			Комментарий	
Обозначение в формуле	Имя переменной	Тип переменной		
a	длина стороны *a*	вещественная	длина стороны *a*	
b	длина стороны *b*	вещественная	длина стороны *b*	
c	длина стороны *c*	вещественная	длина стороны *c*	
p	длина полупериметра	вещественная	длина полупериметра	
S	площадь треугольника	вещественная	площадь треугольника	

Рис. 3.7. Форма для описания переменных задачи

3. *Запись алгоритма на языке программирования*:

```
Private Sub Form_Click()
    Rem Вычисление площади треугольника
    a = Val(InputBox("Введите значение A"))
    b = Val(InputBox("Введите значение B"))
    c = Val(InputBox("Введите значение C"))
    p = (a + b + c)/2
    S = Sqr(p * (p – a) * (p – b) * (p – c))
    Print "S =";S
End Sub
```

Чтобы сделать программу более удобной для чтения и понимания алгоритма, в нее рекомендуется вводить комментарии. Комментарии вводятся с помощью оператора ***Rem*** или символа апостроф (‘).

4. *Отладка программы*. Для отладки программы необходимо ввести данные, для которых результат известен или может быть рассчитан вручную.

Например, для прямоугольного равнобедренного треугольника длина гипотенузы равна $a\sqrt{2}$, а площадь S равна $\frac{a^2}{2}$.

При $a = 1$, $S = 0{,}5$.

Пример 3.5. Вычисление первой и второй производных.

Если функция задана в виде таблицы, то производная от функции в заданной точке может быть вычислена численными методами как предел отношения конечных разностей $\Delta y = y_1 - y_0$ и Δx:

$$y' = \frac{\Delta y}{\Delta x},$$

где Δx – шаг, разность между соседними значениями аргумента. Или, взяв приращения слева и справа от точки x, получают центральную разность:

$$f'(x) = \lim_{\Delta x \to \pm 0} \frac{(f(x+\Delta x) - f(x-\Delta x)}{2\Delta x} + 0(\Delta x). \qquad (3.4)$$

Отсюда вытекает способ численного дифференцирования. Если заменить предел Δx его конечным значением h, то получим приближенные формулы для вычисления первой и второй производных:

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}; \qquad (3.5)$$

$$f''(x) \approx \frac{f(x+h) + f(x-h) - 2f(x)}{h^2}. \qquad (3.6)$$

Эти выражения представляют собой усеченные интерполяционные многочлены (многочлен Стирлинга). Одной из серьезных проблем в данном случае является выбор величины шага h. *При уменьшении шага уменьшается ошибка усечения, но возрастает ошибка округления.* Поэтому стремятся выбрать оптимальную величину шага, при которой *ошибка усечения* – Δ (ошибка, вызванная отбрасыванием правых членов разложения) и *ошибка округления* – ε будут примерно равны. Для формулы (3.4) оптимальный шаг определяется из выражения:

$$h = \sqrt[3]{\frac{3\varepsilon}{M_3}}$$

или

$$\Delta = \frac{1}{6h}\left|\Delta^3 y\right| \approx \frac{1}{2}\frac{\varepsilon}{h}, \qquad (3.7)$$

где h – шаг;

$\Delta^3 y$ – конечная разность 3-го порядка;

ε – абсолютная погрешность вычисления функции;

M_3 – максимальное значение конечной разности 3-го порядка.

Таким образом, ошибка усечения равна примерно половине ошибки округления, деленной на шаг. Полная погрешность не превзойдет при этом $0{,}5\varepsilon/h$.

Простейшим способом выбора шага является обеспечение выполнения неравенства $|f(x + h) - f(x)| > \xi$, где ξ – некоторое малое число. При вычислении производной это исключает вычитание близких по значению чисел, которое обычно приводит к увеличению погрешности вычисления.

Пример 3.6. ПК выводит результат с 8 знаками после запятой, при этом семь знаков точные, то есть ошибка округления ε не превышает половины восьмого знака после запятой. Требуется определить значение шага при вычислении производной первого порядка, чтобы ошибка усечения не превышала 0,0001 (т. е. $\Delta \le 0{,}0001$).

Решение. Согласно условию задачи абсолютная погрешность вычисления функции равна $0{,}5 \cdot 10^{-7}$. Тогда из формулы (3.7) получаем:

$$h = \frac{\varepsilon}{2\Delta} = \frac{0{,}00000005}{2 \cdot 0{,}0001} = 0{,}00025.$$

При $\varepsilon = 0{,}001$ и $\Delta = 0{,}001$ величина шага будет равна 0,5.

Для выражения (3.6) величина шага определяется из следующего соотношения:

$$\Delta = \frac{1}{12h^2}\left|\Delta^4 y\right| \approx 4\frac{\varepsilon}{h^2}. \qquad (3.8)$$

В условиях примера 3.7 величина шага будет равна ~ 0,05.

Таким образом, для получения приемлемого результата не следует стремиться сильно уменьшать шаг приращения аргумента, а также следует учитывать точность, с которой вычисляются значения аргумента и функции. Напомним, *абсолютная погрешность суммы не превышает суммы погрешностей слагаемых, абсолютная погрешность произведения и частного от деления двух чисел не превышает наибольшей из абсолютных погрешностей сомножителей (делимого и делителя).*

Пример 3.7. Вычисление первой и второй производных в заданной точке.

```
REM Вычисление первой и второй производных по табличным данным
Private Sub Form_Click()
    Dim x As Single, y As Single, y1 As Single, y2 As Single, h As
    Single, p1 As Single, p2 As Single
    x = Val(InputBox("Введите значение X")) ' значение аргумента
    y = Val(InputBox("Введите значение Y")) ' значение функции
    в точке x
    y1 = Val(InputBox("Введите значение Y1")) ' значение функции
    в точке слева от x
    y2 = Val(InputBox("Введите значение Y2")) ' значение функции
    в точке справа от x
    h = Val(InputBox("Введите значение шага"))
    p1 =(y2 – y1)/(2 * h)
    p2 = (y2 + y1 – 2 * y)/h^2
    Print "Первая производная =";p1
    Print "Вторая производная =";p2
End Sub
```

3.3.2. Разветвляющиеся программы

Схемы разветвляющихся алгоритмов

С выбором вариантов действий человек сталкивается постоянно в повседневной жизни, часто не задумываясь об этом: пойти на лекцию или в кино, пойти с другом в театр или на дискотеку, купить нужную вещь или подождать. Выбор решения при этом зависит как от внутренних, так и от внешних факторов: есть настроение или его нет, есть на улице дождь или нет, есть деньги в бумажнике или нет и т. д. Выбор решения сопровождается, как правило, постановкой вопроса «Что будет, если ... ?» Выбор может быть простой или сложный. При простом выборе имеется всего два варианта решения (две альтернативы), при сложном выборе – несколько. При наличии двух альтернатив задача выбора формулируется следующим образом:

Если <условие> То <действие первое> [, иначе <действие второе>].

Квадратные скобки означают, что эта часть выражения не обязательная.

Если <*условие*> То <*действие первое*> – краткая форма записи условного выражения.

Если <*условие*> То <*действие первое*> иначе <*действие второе*> – полная форма записи условного выражения.

Если условие выполняется, *то* выполняется действие первое, *иначе* выполняется действие второе. Здесь *если*, *то* и *иначе* – ключевые слова. Если число альтернатив больше двух, то операция выбора повторяется многократно, выбирается последовательно одна из двух альтернатив. Схемы алгоритмов данных операторов приведены на рисунке 3.8.

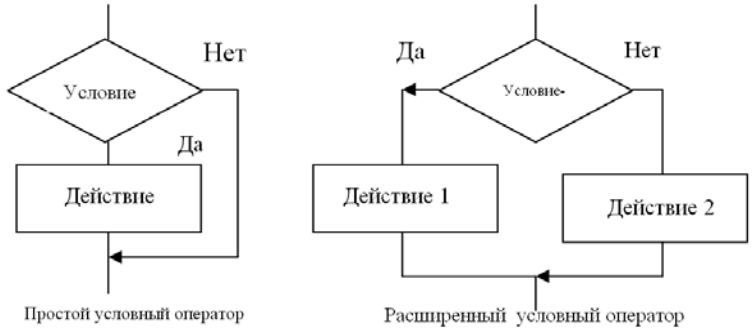


Рис. 3.8. Схемы условных выражений

Понятие о полной группе событий

События, связанные с выбором решения, образуют полную группу событий, если к данной группе событий нельзя добавить никакие другие события. Например, если в урне находятся белые и черные шары, то при последовательном вынимании шаров из урны возможно только два исхода: вынут белый шар или черный шар. При бросании монеты возможно тоже два исхода: монета легла «орлом» или «решкой» (вероятность того, что монета встанет на ребро, очень мала, и ей можно пренебречь).

Выберем на числовой оси *X* произвольную точку *A*. В этом случае случайная величина *x* может принять по отношению к точке *A* одно из трех значений: меньше *A*, больше *A* или равно *A*. Это полная группа событий. Графическое представление этих условий приведено на рисунке 3.9.

Условия записываются с помощью условных выражений. В условных выражениях в качестве операндов используются

переменные и выражения, операнды соединяются знаками отношений: =, >, <, >=, <=, <>, операторами AND, OR, NOT.

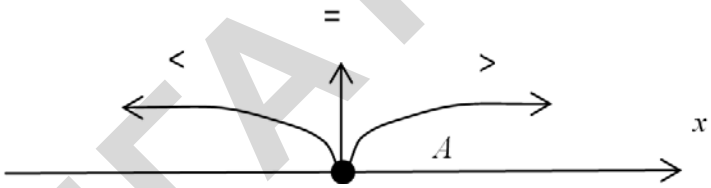


Рис. 3.9. Полная группа событий для точки на числовой оси

Примеры простых условных выражений:

x < a; x > a; Sin(x) <= 1; a + b <= a * b, при a <> b и a > 1, b > 1.

Примеры сложного выражения: a < x < b.

То же самое можно записать с использованием логического оператора AND:

x > a AND x < b.

Если группа событий полная, то для решения задачи число проверок должно быть на единицу меньше, чем число исходов.

Пример 3.8. Отобразить графически условия x < a; x >= a.

Решение приведено на рисунке 3.10.

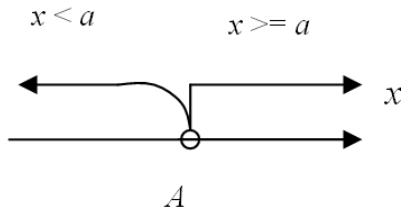


Рис. 3.10. К примеру 3.8

Пример 3.9. Отобразить графически условие *x <= a*, *a < x < b, x >= b*.

Решение приведено на рисунке 3.11.

Пример 3.10. Написать программу для проверки условий согласно рисунку 3.12.

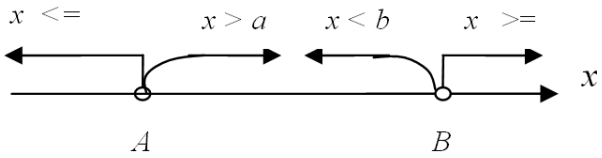


Рис. 3.11. К примеру 3.9

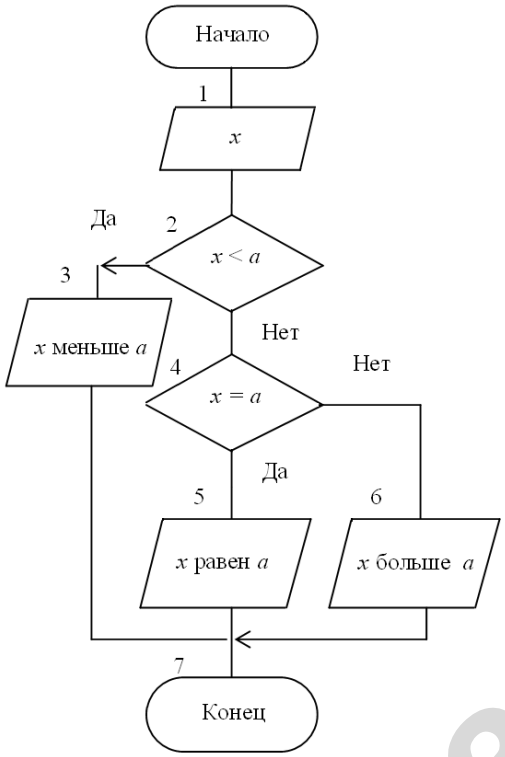


Рис. 3.12. Схема алгоритма к примеру 3.10

Решение.

```
алг Проверка_значения X
    дано x
    нач
        если x < a то
            печать "X меньше a"
```



```
        иначе если x = a то
            печать "X равен a"
        иначе
            печать "X больше a"
        конец если
    все
кон
```

Пример 3.11. Известная сказка о вещем камне. Налево пойдешь – жизнь потеряешь, прямо пойдешь – коня потеряешь, направо пойдешь – женату быть. Требуется разработать алгоритм для решения данной задачи.

Решение. Выполним формализацию задачи – введем обозначения: налево – 1, прямо – 2, направо – 3; выбор пользователя обозначим как *i*. Группа событий неполная, так как неизвестно, что будет, если вариант не выбран (вероятно, герой сказки в этом случае должен вернуться домой или, как его старшие братья, остаться гулять в чистом поле и ждать случая вернуться к отцу не с пустыми руками). При данных условиях программа должна проверить все варианты выбора. Опишем решение задачи на алгоритмическом языке.

```
алг Вещий_камень
    дано i ‘ вариант выбора
    нач
        если i = 1 то
            Текст1 = “Жизнь потеряешь”: Вывод Текст1
        иначе если i = 2 то
            Текст1 = “Коня потеряешь”: Вывод Текст1
        иначе если i = 3 то
            Текст1 = “Женату быть”: Вывод Текст1
        иначе
            Текст2 = “Нет выбора”: Вывод Текст2
        конец если
    все
кон
```

Схема алгоритма приведена на рисунке 3.13.

Пример 3.12. Вычислить значения функции:

$$Y = \begin{cases} x^2 - 5, \text{если } x < -2, \\ -x^2 + 5, \text{если } -2 \le x \le 4, \\ x^3 - 5x - 1, \text{если } x > 4. \end{cases}$$

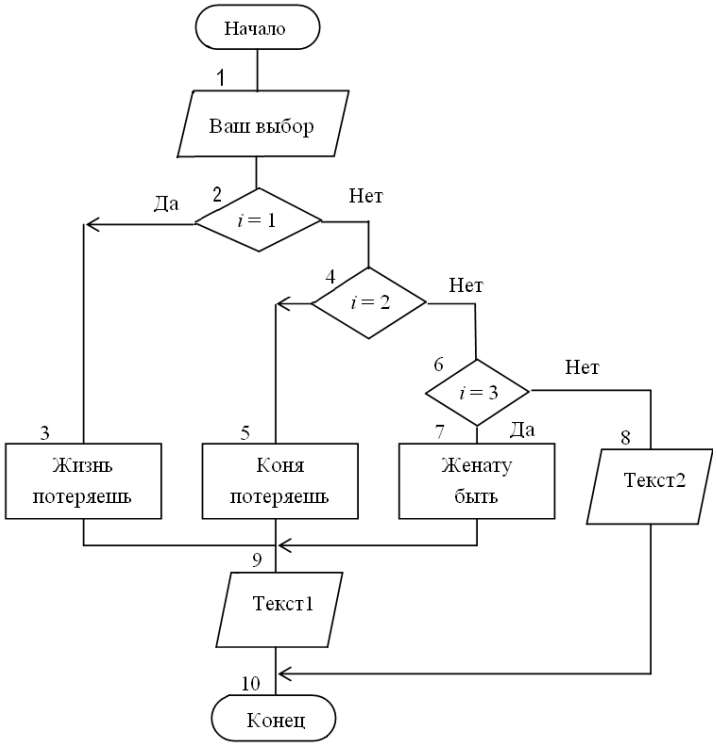


Рис. 3.13. Схема алгоритма к примеру 3.11

Решение. В данной задаче группа событий полная, поэтому для вычисления значения *Y* при любых значениях аргумента достаточно выполнить две проверки.

Описание решения задачи на алгоритмическом языке и алгоритм решения приведены на рисунке 3.14.

Выбор решения в разветвляющихся программах осуществляется с помощью условного оператора If.

Различают однострочные и многострочные условные операторы.

Форматы однострочного оператора IF:

а) простой однострочный оператор:

If <условие> Then <операторы>.

При выполнении оператора If проверяется условие и, если оно истинно, то выполняется действие, указанное после опции Then.

Если выражение ложно, то управление передается на оператор, следующий за оператором If. Управление может передаваться также на метку, например m1.

```
алг условный_оператор
  дано x
нач
  если x < –2 то
     y = x² – 5
  иначе если x > 4 то
y = x³ – 5x – 1
  иначе
     y = –x² + 5
  все
  рез y
кон
```

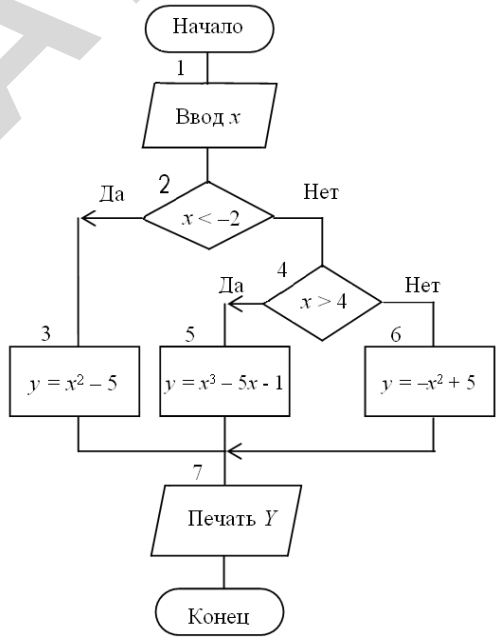


Рис. 3.14. Схема алгоритма к примеру 3.12

*If x < a Then y = Sin(x) * Exp(2^Log(x))*

If x < e Then Goto m1

б) простой расширенный оператор IF:

If <условие> Then <операторы 1 > Else <операторы 2>.

При выполнении оператора If, если условие истинно, то выполняются операторы, указанные после опции Then, в ином случае выполняются операторы, следующие за опцией Else. После выполнения соответствующей группы операторов управление передается на оператор, следующий за оператором If.

*If x < a Then y = Sin(x) * Exp(2^Log(x)) Else y = 3 * x^2 – 5 * x + 4*

После опций Then и Else может быть указано несколько операторов, разделенных двоеточием. Однако число операторов ограничено длиной строки. Этих недостатков лишен многострочный оператор IF.

При записи операторов следует обращать внимание на структуру записи. Структура должна соответствовать той, что указана в примере.

Достоинство многострочного оператора IF состоит в том, что число операторов в группах не ограничено.

Форматы многострочного оператора If:

а) простой	б) расширенный

If <условие> Then
  <первая группа операторов>
Else
  <вторая группа операторов>
End If
	If <условие> Then
	  <первая группа операторов>
	ElseIf <условие> Then
	  <вторая группа операторов>
	Else
	  <третья группа операторов>
	End If

Для условий примера 3.12 программа будет иметь следующий вид:

А. При использовании однострочного оператора If	Б. При использовании многострочного оператора If

x = Val(InputBox("Введите значение X"))
a = Val(InputBox("Введите значение A"))
b = Val(InputBox("Введите значение B"))
  If x < a Then y = x^2 – 5: Goto m1
  If x > b Then y = x^3 – 5 * x – 1: Goto m1
  y = –x^2 + 5
  m1:
  Print "x ="; x, "y ="; y
	x = Val(InputBox("Введите значение X"))
	a = Val(InputBox("Введите значение A"))
	b = Val(InputBox("Введите значение B"))
	  If x < a Then
	  ElseIf x > b Then
	  Elsey = x^2 – 5
	  y = x^3 – 5 * x – 1
	  y = –x^2 + 5
	  End If
	  Print "x ="; x, "y ="; y

Оператор выбора SELECT CASE

Оператор *Select Case* очень похож по структуре и выполняемым задачам на многострочный оператор *If/Then/Else*.

Оператор *Select Case* является оператором выбора. Он обеспечивает переход на ту или иную подпрограмму в зависимости от

результатов вычислений. Так же, как и многострочный If, оператор Select Case позволяет сформировать хорошо читаемую программу. В отличие от оператора If оператор Select Case имеет больший набор опций для управления выбором. Он очень удобен для обработки пунктов меню.

Синтаксис оператора Select Case

```
Select Case <переключающее выражение>
  Case <Значение 1> : REM Значение 1 – условие выборки
    <Блок операторов 1>
  Case <Значение 2>
    <Блок операторов 2>
    ...
  Case Else
    <Блок операторов n+1>
End Select
```

В качестве переключающего выражения может использоваться переменная числового или строкового типа или арифметическое выражение.

В качестве *Значения* для блока Case могут использоваться следующие формы:

число или список значений, например: *Case 1, 3, 5, 8*;

выражение [, выражение]. *Выражение* – любое числовое или строковое выражение, которое должно совпадать с типом переключающего выражения;

выражение TO выражение, например: *Case 1 TO 10*. Данное выражение означает, что значением может быть любое число из указанного диапазона;

IS выражение. Ключевое слово *IS* представляет собой условие, например *Case IS < 5*.

Блок *Case Else* выполняется тогда, когда не выполнено ни одно из предыдущих условий.

Пример 3.13. Использование оператора Select Case.

Организовать вычисление определенного интеграла тремя методами: методом средних прямоугольников, методом трапеций, методом Симпсона.

```
Private Sub Form_Click()
  Cls ' Очистка формы
  Print "Укажите метод вычисления интеграла: 1 – метод средних"
  Print "прямоугольников, 2 – метод трапеций, 3 – метод Симпсона"
  metodint = Val(InputBox("Укажите метод вычисления интеграла"))
```

```
Print
Select Case metodint ' metodint – переключающее выражение
   Case 1
      Print “Вычисляется интеграл методом средних прямоугольников”
      ' <программа вычисления интеграла методом средних прямоугольников>
   Case 2
      Print “Вычисляется интеграл методом трапеций”
      ' <программа вычисления интеграла методом трапеций>
   Case 3
      Print “Вычисляется интеграл методом Симпсона”
      ' <программа вычисления интеграла методом Симпсона>
   Case Else
      Print “Неправильный ввод. Введите правильно номер варианта”
      Beep ' Выдача звукового сигнала
   End Select
End Sub
```

3.3.3. Циклические программы

Циклом называется процедура, в которой вычислительные операции выполняются многократно заданное число раз или до достижения некоторой переменной, вычисляемой в теле цикла, определенного, наперед заданного значения.

Понятие о циклах и их классификация приведены в разделе 3.2.

Циклы могут быть организованы с использованием оператора If или с использованием специальных операторов: *For/Next*, *While/Wend*, *Do/Loop*. Примеры организации циклов приведены в таблице 3.3, на рисунках 3.15 и 3.16.

Оператор FOR/NEXT

Оператор For служит для организации циклов с заданным числом повторений. Цикл относится к циклам с постусловием. Синтаксис оператора:

For i = Інач To Ікон Step di
   <тело цикла>
Next i

Таблица 3.3

Операторы циклов

а) цикл с постусловием REM использование для организации цикла оператора IF i = інач: REM заголовок цикла M1: <тело цикла, вычисление функции F(i)> i = i + di IF i <= ікон THEN GOTO M1: REM конец цикла PRINT “Результат”; F	REM использование оператора FOR / NEXT FOR i = інач TO N STEP di <тело цикла, вычисление функции F(i)> NEXT i PRINT “Результат”; F Схема алгоритма приведена на рис. 3.15
б) цикл с предусловием REM Использование оператора IF F = <выражение> i = інач M1: IF F <= ε THEN GOTO M2 <тело цикла, вычисление F(i)> i = i + di GOTO M1 M2: PRINT “Результат”; F ---------- REM Использование оператора REM WHILE / WEND F = <выражение> i = інач WHILE F > ε <тело цикла, вычисление F(i)> i = i + di WEND PRINT “Результат”; F	Схема алгоритма цикла с предусловием приведена на рис. 3.16

В данном формате *Інач* – начальное значение переменной цикла, *Ікон* – конечное значение переменной цикла, а *di* – шаг приращения значения переменной цикла.

Операторы *For* и *Next* образуют операторные скобки, между которыми заключено тело цикла.

Циклы применяются для решения самых разнообразных задач:

табулирования функций;

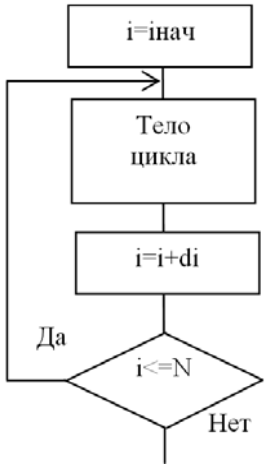


Рис. 3.15. Цикл с постусловием

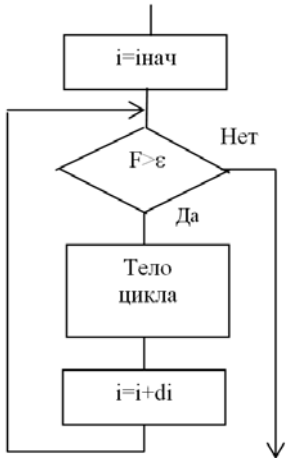


Рис. 3.16. Цикл с предусловием

определения экстремумов функций;

вычисления сумм конечных и бесконечных рядов;

вычисления определенных интегралов;

решения алгебраических и трансцендентных уравнений численными методами;

операций с матрицами;

автоматизированного ввода и вывода данных.

Редкая вычислительная процедура обходится без использования циклов.

Табулирование функции одной переменной

Под табулированием понимают конструирование или вычисление (составление) различных математических таблиц.

Наиболее широко этот метод использовался до появления вычислительной техники. Для вычисления значений тригонометрических функций, интегралов широко использовались таблицы, рассчитанные в научных центрах, например, *таблицы Брадиса*. Табличные методы задания функций могут применяться и в малых вычислительных машинах для сокращения времени вычисления с использованием методов интерполяции и экономии памяти, необходимой для хранения сложных программ вычисления по точным формулам. Табулирование может применяться также для построения графиков функций.

Суть табулирования функции состоит в том, что весь диапазон изменения независимой переменной разбивают на равные интервалы и для каждого значения аргумента в граничных точках интервалов (*узлах интерполяции*) вычисляется значение функции одним из известных методов с требуемой точностью. Результаты расчетов представляются в виде таблицы, в одной из колонок которой приводятся значения аргумента, а в другой – соответствующие им значение функции. Алгоритм расчета представляет собой цикл типа «До», так как число шагов обычно известно заранее, или сочетание циклов «До» и «Пока», если заданы требования к точности вычислений.

Пример 3.14. Составить таблицу значений функции:

$$y = x^2 + 2x - 4$$

для x, изменяющегося в диапазоне от 0 до 10 с шагом 0,5.

Решение. Введем обозначения: *Xn* – начальное значение аргумента, *Xk* – конечное значение аргумента, *Dx* – шаг приращения аргумента. Программа работает следующим образом (рис. 3.17):

Блок 1. Вводим исходные данные: *Xn*, *Xk*, *Dx*.

Блок 2. Присваиваем текущему значению аргумента *x* начальное значение *Xn*.

Блок 3. Вычисляем значение функции при текущем значении аргумента.

Блок 4. Выводим на экран или печать значения *x* и *y*.

Блок 5. Увеличиваем значение аргумента на величину шага.

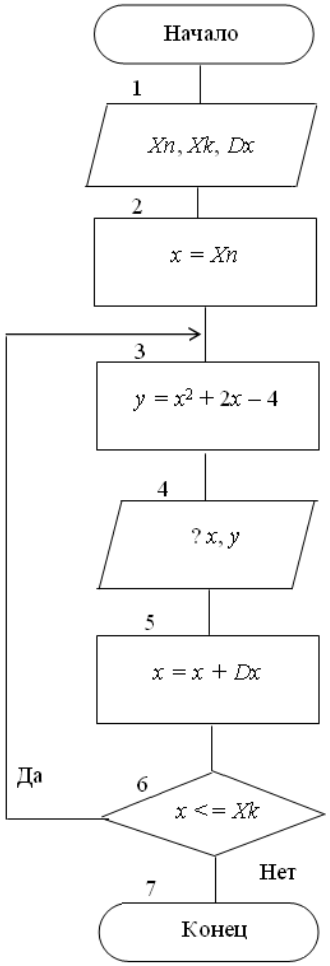


Рис. 3.17. Схема алгоритма к задаче 3.14

Блок 6. Проверяем условие окончания цикла. Если условие выполняется, то есть текущее значение аргумента меньше или равно конечному значению аргумента, то возвращаемся на блок 3 и повторяем вычислительную процедуру (блоки 3, 4, 5, 6). Если условие не выполняется, то выход из цикла. Конец программы.

В данном примере число шагов явно не задано. Переменной цикла является *x*. В начале цикла *x* присвоено начальное значение – 0. В конце

цикла значение *x* автоматически увеличивается на величину шага (это внутренняя операция цикла For/Next, в программе это не отражено, но показано на схеме алгоритма). Текущее значение переменной *x* сравнивается с конечным значением. Если *x* меньше или равно конечному значению, то цикл повторяется. Все названные параметры: *x*, *Xk*, *Dx* – указаны в заголовке цикла. Так как значение шага не вычисляется, а задается пользователем, то число шагов может оказаться не целым. Поэтому конечное значение *Xk* увеличено на половину шага, чтобы не потерять значение функции на конце отрезка.

К примеру 3.14:

Запись программы на алгоритмическом языке

алг Табулирование_функции
  дано *Xn, Xk, Dx*
  нц для x от Xn до Xk + Dx/2 с шагом Dx
    $y = x^2 + 2x - 4$
    рез x, y
  кц
кон

Запись программы на языке программирования

```
REM Табулирование функций
Xn = Val(InputBox("Введите значение Xn"))
Xk = Val(InputBox("Введите значение Xk"))
Dx = Val(InputBox("Введите значение Dx"))
For x = Xn To Xk + Dx/2 Step Dx
    y = x^2 + 2 * x – 4
    Print x, y
Next x
```

Пример 3.15. Напечатать таблицу перевода температуры из градусов по шкале Цельсия (C) в градусы по шкале Фаренгейта (F) для значений температуры от 10 °C до 30 °C с шагом 5 °C. Формула перевода: F = 1,8C + 32.

Решение. Т. к. известно начальное значение, конечное значение температуры и шаг, то число шагов можно вычислить до начала цикла. Следовательно, для решения задачи можно применить оператор For/Next:

```
Private Sub Form_Click()
    Tn = Val(InputBox("Введите начальное значение температуры °C"))
    Tk = Val(InputBox("Введите Конечное значение температуры °C"))
    Dt = Val(InputBox("Укажите шаг изменения температуры °C"))
```

T, °C	T, °F
10	50
15	59
20	68
25	77
30	86

```
    N = Int((Tk – Tn)/Dt) + 1
    Print Tab(10);“T °C“; Tab(20);“T °F“
    Tc = Tn
    For i = 1 To N
        F = 1.8 * Tc + 32
        Print Tab(10); Tc; Tab(20); F
        Tc = Tc + Dt
    Next i
End Sub
```

Здесь Tc – вспомогательная переменная – текущее значение температуры по Цельсию, N – число шагов. В отличие от примера 3.14, в задаче применен другой подход к организации цикла. Переменная цикла i – целочисленная переменная, изменяется от 1 до N с шагом 1. Это счетчик циклов. Значение текущей температуры в градусах Цельсия вычисляется в теле цикла: Tc = Tc + Dt.

Вычисление сумм и произведений ряда чисел

Пусть требуется вычислить сумму и произведение натурального ряда чисел от 1 до N, то есть вычислить выражения:

$$S = \sum_{i=1}^{N} i, \ \ i \in \overline{1,N}, \tag{3.9}$$

$$P = \prod_{i=1}^{N} i, \ \ i \in \overline{1,N}. \tag{3.10}$$

Для вычисления суммы чисел переменной S до входа в цикл присваивается начальное значение нуль – $S = 0$, а для вычисления произведения переменной P присваивается начальное значение единица $P = 1$. В теле цикла проводится вычисление S: к текущему значению суммы прибавляется текущее значение переменной цикла $S = S + i$. Для вычисления произведения текущее значение произведения P умножается на текущее значение переменной цикла – $P = P * i$.

Пример 3.16. Вычислить сумму и произведение чисел натурального ряда от 1 до N.

Решение. Введем обозначения: Xn – начальное значение аргумента, N – число членов ряда, Dx – шаг, S – сумма чисел, P – произведение чисел.

Схема алгоритма приведена на рисунке 3.18. Данный алгоритм не отвечает свойству массовости, так как не позволяет вычислять отдельно сумму четных или нечетных чисел. Чтобы обеспечить

такую возможность, необходимо формировать запрос, что надо подсчитать. Кроме того, необходимо обеспечить контроль значения Xn, так как при вычислении произведения чисел Xn не может быть равным нулю, при вычислении суммы или произведения четных чисел начальное значение Xn должно быть четным, а при вычислении суммы или произведения нечетных чисел – нечетным.

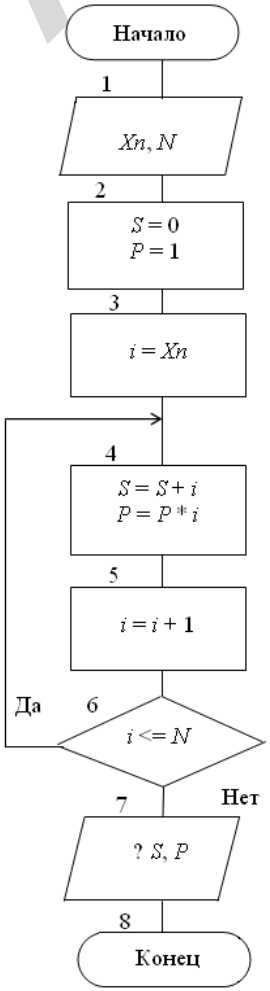


Рис. 3.18. Схема алгоритма к примеру 3.16

Проверить четность числа можно с помощью функции MOD. Функция MOD вычисляет отношение двух целых чисел, при этом возвращается значение 0, если остаток от деления равен нулю, то есть число четное, и возвращается значение один, если остаток от деления не равен нулю, то есть число нечетное. Например: 5 MOD 2 – результат 1, 6 MOD 2 – результат 0.

К примеру 3.16

```
алг Вычисление суммы
    дано Xn, N
    нач
        S = 0
        P = 1
        нц для i от Xn до N с шагом Dx
            S = S + i
            P = P * i
        кц
        рез S, P
кон
```

```
REM Вычисление суммы
REM и произведения чисел
Xn = Val(InputBox("Введите значение Xn"))
N = Val(InputBox("Введите значение N"))
Di = Val(InputBox("Введите значение Di"))
S = 0: P = 1
FOR i=Xn TO N STEP Di
    S = S + i
    P = P * i
NEXT i
PRINT "S =";S, "P =";P
```

Решение примера 3.16 с учетом изложенных замечаний дано в примере 3.17.

Пример 3.17. Вычисление суммы и произведения чисел.

```
Private Sub Form_Click()
Rem Вычисление суммы четных чисел
    Cls ' Очистка формы
    Print "Для вычисления суммы введите число 1"
    Print "Для вычисления произведения введите число 2"
    Print "Для вычисления суммы и произведения введите число 3"
    K = Val(InputBox("Выберите вариант вычисления"))
```



```
    Cls ' Очистка формы
    Print "Для вычисления выражения для четных чисел введите число 1"
    Print "Для вычисления выражения для нечетных чисел введите число 2"
    Print "Для вычисления выражения для любых чисел введите число 3"
    k1 = Val(InputBox("Укажите вариант вычисления выражения"))
    Cls
    Xn = Val(InputBox("Введите значение начального члена ряда"))
    N = Val(InputBox("Введите значение конечного члена ряда"))
    Di = Val(InputBox("Введите значение шага"))
    S = 0: P = 1
    For i = Xn To Xk Step Di
      Select Case k
        Case 1
        If i Mod 2 = 0 And k1 = 1 Then S = S + i
        If i Mod 2 > 0 And k1 = 2 Then S = S + i
        If k1 = 3 Then S = S + i
        'Print "S ="; S
        Case 2
          If i > 0 Then
          If i Mod 2 = 0 And k1 = 1 Then P = P * i
          If i Mod 2 > 0 And k1 = 2 Then P = P * i
          If k1 = 3 Then P = P * i
          'Print "P ="; P
          End If
        Case 3
          If i > 0 Then
          If i Mod 2 = 0 And k1 = 1 Then S = S + i: P = P * i
          If i Mod 2 > 0 And k1 = 2 Then S = S + i: P = P * i
          If k1 = 3 Then S = S + i: P = P * i
          'Print "S ="; S, "P ="; P
          End If
      End Select
    Next i
    If k = 1 Then Print "S ="; S
    If k = 2 Then Print "P ="; P
    If k = 3 Then Print "S ="; S, "P="; P
End Sub
```

Оператор WHILE/WEND

Оператор WHILE служит для организации циклов с параметром (бесконечных циклов или циклов типа «ПОКА»), относится к циклам с предусловием. Условием окончания цикла является достижение функцией, вычисляемой в теле цикла, некоторого наперед заданного значения. Синтаксис оператора:

WHILE <условие>
    <тело цикла>
WEND

Тело цикла выполняется в том случае, когда условие истинно. Если условие ложно, то программа выходит из цикла. Особенностью использования данного цикла является то, что значение вычисляемой функции должно быть известно перед входом в цикл. Если начальное значение функции меньше заданной величины ε, то программа не войдет в цикл. Схема алгоритма приведена в таблице 3.3 и на рисунке 3.16.

Оператор DO

Оператор DO – универсальный оператор, служит для организации циклов типа «ПОКА». Он может быть организован как цикл с предусловием или цикл с постусловием. Когда в теле цикла используется опция WHILE, то тело цикла выполняется в том случае, если условие истинно (положительная логика). Когда используется опция UNTIL, то тело цикла выполняется в том случае, если условие ложно (отрицательная логика).

Оператор DO:

а) цикл с предусловием

DO <WHILE|UNTIL>
    <тело цикла>
LOOP

б) цикл с постусловием

DO
    <тело цикла>
LOOP <WHILE|UNTIL>

Пример 3.18. Составить таблицу значений функции e^x для заданного значения x с точностью 0,0001.

Решение.

Функцию e^x можно представить в виде ряда:

$$e^x = 1 + \frac{x^1}{1} + \frac{x^2}{1 \cdot 2} + \frac{x^3}{1 \cdot 2 \cdot 3} + \ldots + \frac{x^i}{i!} + \ldots$$

или

$$e^x = 1 + \sum_{1}^{\infty} \frac{x^i}{i!}. \qquad (3.11)$$

Данный ряд сходящийся. Под точностью в данном случае понимается значение отбрасываемого члена ряда. Полная ошибка может быть в несколько раз больше значения отбрасываемого члена ряда. Для знакопеременных рядов ошибка вычисления суммы сходящегося ряда не превышает значения отбрасываемого члена ряда.

Алгоритм вычисления функции реализуется с помощью цикла типа «Пока» (рис. 3.19).

Первый блок вводит исходные данные: x – значение аргумента и E – точность вычисления значения текущего члена ряда. Второй и третий блоки осуществляют начальное присвоение переменной цикла, S – начальное значение суммы, P – вспомогательная переменная для вычисления факториала. Начальное значение суммы равно единице, так как в выражении (3.11) имеется соответствующее слагаемое. Начальное значение y также принимаем равным 1, чтобы на первом шаге программа вошла в цикл. Четвертый блок проверяет условие окончания цикла – достижение заданной точности вычисления текущего члена ряда. Если условие выполняется, то выполняется тело цикла, иначе осуществляется переход на блок 7 – печать результата и выход из программы – блок 8. Пятый блок обеспечивает вычисление значения функции: вычисляется текущее значение факториала – P, значение текущего члена ряда – Y и значение функции на текущем шаге – S.

С точки зрения вычислительной математики блок 5 построен нерационально, так как на каждом шаге необходимо вычислять выражение $x^i/(i!)$, кроме того, при больших значениях i наступает переполнение разрядной сетки. Скорректируем математическую модель:

$$e^x = 1 + \sum_{1}^{\infty} \frac{\prod_{1}^{i} x}{\prod_{1}^{i} i} = 1 + \sum_{1}^{\infty} \prod_{1}^{i} \frac{x}{i}. \qquad (3.12)$$

В данной модели степень переменной x заменена ее произведением, а в числителе факториал заменен произведением ряда натуральных чисел от единицы до i. При такой модели значение текущего элемента будет равно значению предыдущего элемента, умноженного на x и деленного на i.

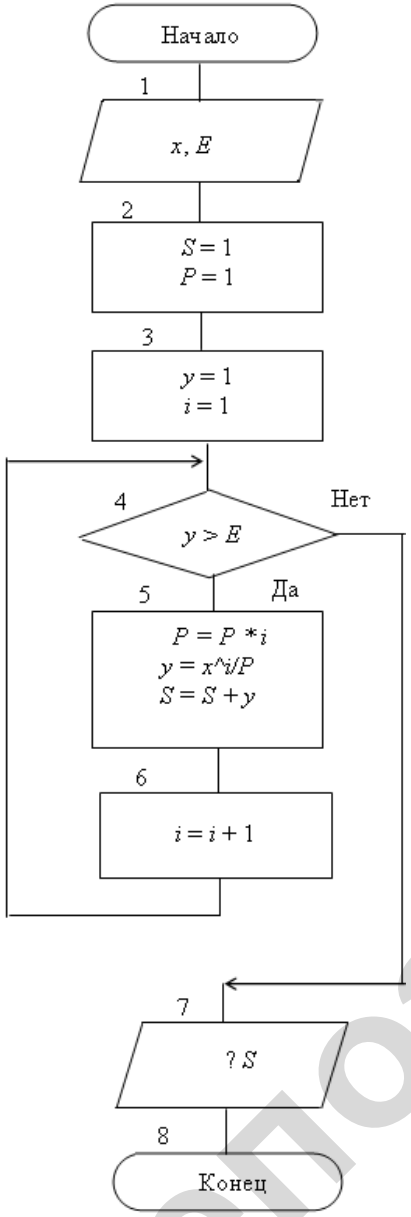
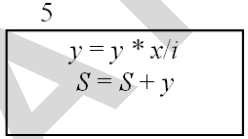


Рис. 3.19. Схема алгоритма к примеру 3.18

С учетом сделанных замечаний блок 5 алгоритма (рис. 3.19) будет выглядеть следующим образом:



Данный пример показывает, как от правильного построения математической модели зависит эффективность алгоритма.

Вычисление определенного интеграла

Пример 3.19. Пусть требуется вычислить площадь фигуры, ограниченной прямыми $Xn = a$, $Xk = b$, $y = 0$ и *графиком функции* $y = e^x + 5\sin(x)$, с точностью e.

Решение. Площадь фигуры может быть вычислена с помощью определенного интеграла. Для вычисления определенного интеграла численными методами необходимо организовать цикл типа «ДО», а чтобы вычислить интеграл с заданной точностью, применяется метод двойного прохода, и реализуется эта процедура с помощью цикла «Пока».

Пусть функция $y = f(x)$ неотрицательна и непрерывна на отрезке $[a; b]$. Тогда площадь $S(x)$ криволинейной трапеции представляет собой функцию от x (рис. 3.20).

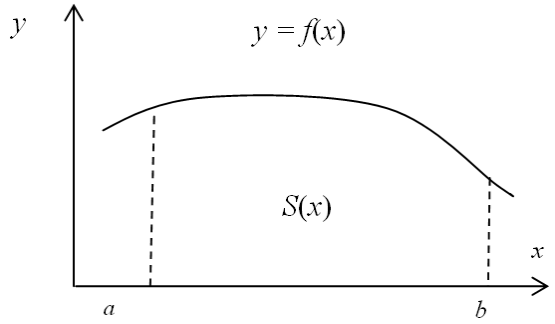


Рис. 3.20. К определению интеграла

Производная от $S(x)$ равна $f(x)$ всюду на отрезке $[a; b]$. Иными словами, $S(x)$ оказывается *первообразной* для $f(x)$.

Процедура нахождения первообразной $S(x)$ от функции $f(x)$ называется интегрированием.

Приращение любой первообразной для функции $f(x)$, получаемое при переходе от а к b, называется *определенным интегралом* функции $f(x)$, взятым в пределах от a к b, и обозначается символом $\int_a^b f(x)dx$. Здесь a и b называют соответственно нижним и верхним пределами интегрирования.

С геометрической точки зрения определенный интеграл представляет собой площадь криволинейной трапеции, ограниченной графиком функции $f(x)$, прямыми $x = a$, $x = b$ и осью x (рис. 3.21):

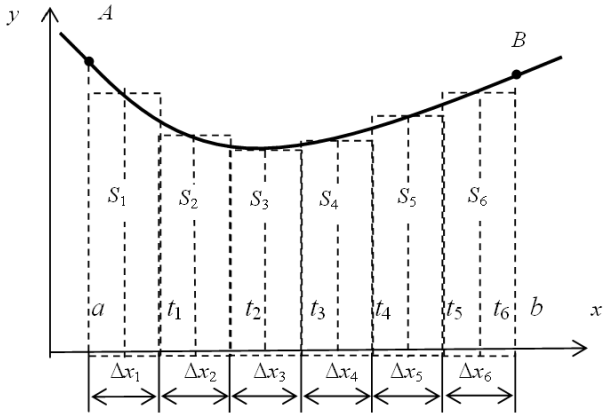


Рис. 3.21. Вычисление определенного интеграла методом средних прямоугольников

$$S = \int_a^b f(x)dx . \qquad (3.13)$$

По определению:

$$S = \int_a^b f(x)dx = F(b) - F(a),$$

где $F(b)$ и $F(a)$ – первообразные для функции $f(x)$ при $x = b$ и $x = a$ соответственно. Для большинства функций первообразную $f(x)$ не удается выразить через элементарные функции, поэтому чаще всего интеграл вычисляют *численными методами*.

Сущность метода численного интегрирования состоит в том, что отрезок интегрирования $[a; b]$ разбивают на n элементарных отрезков $[x_{i-1}, x_i]$ и заменяют интеграл функции $f(x)$ суммой прямоугольников $S_i = f(t_i) * \Delta x_i$ (рис. 3.21), где $f(t_i)$ – значение функции в некоторой точке (t_i) внутри i-го отрезка; Δx_i – длина отрезка, $\Delta x_i = x_i - x_{i-1}$, то есть:

$$S_n = S_1 + S_2 + \ldots + S_{n-1} + S_n + R = \sum_{i=1}^{n} f(t_i)\Delta x + R . \qquad (3.14)$$

Здесь S_n – интегральная сумма, R – погрешность усечения. Предел этой суммы при неограниченном увеличении числа отрезков разбиения, когда длина наибольшего из элементарных отрезков Δx_i стремится к нулю, есть интеграл от функции $f(x)$ на отрезке $[a; b]$:

$$\int_a^b f(x)dx = \lim \sum_{i=1}^{n} f(t_i)x_i , \quad \max \Delta x_i \to 0 .$$

Простейший метод численного интегрирования – *метод средних прямоугольников*. В нем используется замена интеграла интегральной функцией (3.14). В этом случае все отрезки Δx_i равны между собой, а в качестве $f(t_i)$ принимается значение функции в середине отрезка. Пусть h – длина элементарного отрезка, $h = (b - a)/n$; $x_{i-1/2}$ – значение аргумента в середине i-го отрезка, $x_{i-1/2} = a + (2i - 1)(b - a)/(2n)$, тогда можно записать:

$$\int_a^b f(x)dx = \sum_{i=1}^{n} h f(x_{i-1/2}) = \sum_{i=1}^{n} hf\left(a + hi - \frac{h}{2}\right) = \sum_{i=1}^{n} h_i f\left(a + (2i-1)\frac{h}{2}\right). \qquad (3.15)$$

Схема алгоритма приведена на рисунке 3.22.

В других методах вычисляют не площадь прямоугольника, а площадь криволинейной трапеции, заменяя функцию $f(x)$ на элементарном отрезке интерполирующей функцией. Если в качестве интерполирующей функции используется линейная функция, то получаем *метод трапеций*:

$$\int_a^b f(x)dx = \frac{1}{2}\sum_{i=1}^{n} h_i \left(f(x_i) + f(x_{i=1})\right) + R . \qquad (3.16)$$

В зависимости от используемой интерполирующей функции различают методы Ньютона, Симпсона, Ромберга и др. В методе Симпсона, например, в качестве интерполирующей функции ис-

пользуется многочлен второй степени $ax^2 + bx + c$. Эти методы обеспечивают получение более точных результатов при меньшем числе шагов.

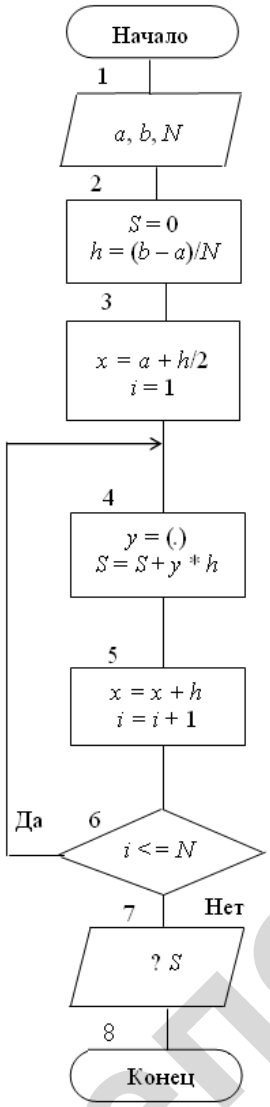


Рис. 3.22. Схема алгоритма к примеру 3.19

Формула трапеций (3.16) после преобразования принимает вид:

$$S = h\left(\frac{y_0 + y_n}{2} + \sum_i^{n-1} y_i\right), \qquad (3.17)$$

а формула Симпсона записывается следующим образом:

$$S = \frac{h}{3}\left(y_0 + y_{2m} + 2\left(y_2 + y_4 + \ldots + y_{2m-2}\right) + 4\left(y_1 + y_3 + \ldots + y_{2m-1}\right)\right). \qquad (3.18)$$

Программа вычисления определенного интеграла методом средних прямоугольников с заданной точностью

Программа содержит два вложенных цикла: цикл For/Next и цикл While/Wend. Первый цикл – внутренний, он обеспечивает вычисление площади криволинейной трапеции при заданном числе отрезков разбиения. Второй цикл – внешний, этот цикл обеспечивает достижение заданной точности. Для достижения заданной точности используется метод *двойного прохода*. Суть этого метода состоит в следующем. Вычисляется площадь криволинейной трапеции *S* при заданном числе отрезков разбиения *N*, и результат запоминается в переменной *S*1. Затем увеличивают число отрезков разбиения вдвое и снова вычисляют площадь криволинейной трапеции. После этого вычисляют ошибку интегрирования по приближенной формуле. Процесс вычисления заканчивается, когда ошибка интегрирования станет меньше или равной некоторой наперед заданной величине. Блок 5 на рисунке 3.23 – предопределенный процесс (см. рис. 3.22).

```
REM Вычисление определенного интеграла с заданной точностью
a = Val(InputBox("Введите нижнюю границу отрезка интегрирования"))
b = Val(InputBox("Введите верхнюю границу отрезка интегрирования"))
N = Val(InputBox("Укажите число отрезков разбиения"))
E = Val(InputBox("Укажите требуемую точность"))
s = 0: R = 1 + e
WHILE R > e
    S1 = S
    S = 0
    h = (b – a)/N
    x = a + h/2
        FOR i = 1 TO N
```

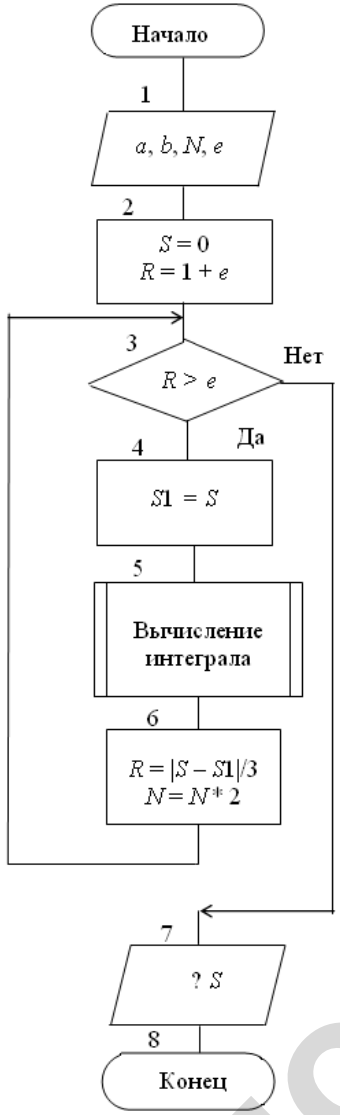


Рис. 3.23. Вычисление определенного интеграла

```
y = f(x)
s = s + y * h
```



```
    x = x + h
  NEXT i
  R = ABS(S – S1)/3
  N = N * 2
WEND
PRINT "S="; S
```

В программе обозначено: S – площадь фигуры на текущем шаге, $S1$ – площадь фигуры на предыдущем шаге, R – погрешность усечения.

Оценка погрешности усечения осуществляется по сложным формулам, однако для практических целей можно воспользоваться приближенными формулами (принцип Рунге):

$$\Delta \approx \frac{1}{3}\left|I_n - I_{2n}\right| \text{ – для формулы трапеции,} \qquad (3.19)$$

$$\Delta \approx \frac{1}{15}\left|I_n - I_{2n}\right| \text{ – для формулы Симпсона.} \qquad (3.20)$$

Здесь I_n – значение интеграла при разбиении отрезка интегрирования на n частей, а I_{2n} – значение интеграла при разбиении отрезка на $2n$ частей.

Решение алгебраических и трансцендентных уравнений

Нелинейные уравнения вида $F(x) = 0$ принято называть *алгебраическими*, если они содержат только алгебраические функции, и *трансцендентными*, если они содержат другие функции (тригонометрические, показательные, логарифмические и т. д.).

При решении нелинейных уравнений всегда возникают серьезные трудности. В аналитическом виде можно получить решение алгебраического уравнения не выше второй степени. Для решения уравнений большей степени, а также трансцендентных уравнений можно использовать графические и численные методы.

Под численным методом решения уравнений понимают метод нахождения численных значений корней уравнения с заданной точностью.

Пусть дано уравнение $f(x)= 0$, где $f(x)$ – функция, непрерывная на отрезке $[a; b]$ и дифференцируемая на интервале $]a; b[$, т. е. включая и граничные значения.

Корнем уравнения $f(x) = 0$, где $f(x)$ – функция, непрерывная и дифференцируемая на отрезке $[a; b]$ и в граничных точках, называется всякое число c, принадлежащее отрезку $[a; b]$, такое, что $f(c) = 0$.

Процесс нахождения корней уравнения состоит из двух этапов:
1) отделения корней;
2) уточнения значения корней с заданной точностью *h*.

Отделением корней уравнения называется процесс выделения из области определения функции f(x) отрезков [a ; b_i], в каждом из которых содержится один, и только один, корень уравнения f(x) = 0.

Под уточнением значения корня с заданной точностью понимают сужение границ отрезка отделения корня [a_i; b_i] до длины, не превосходящей заданной точности h.

Отделение корней численными методами

Процедура отделения корней уравнения численными методами сводится к табулированию функции на заданном отрезке с некоторым шагом *h* и проверке на каждом шаге условия знакопостоянства функции. Если на границах элементарного отрезка *h* значения функции $y1 = f(x)$ и $y2 = f(x + h)$ имеют одинаковый знак, то корня на этом отрезке нет, если $y1$ и $y2$ имеют разные знаки, то на отрезке $[x; x + h]$ имеется корень. Для проверки знакопостоянства функции на отрезке $[x; x + h]$ достаточно проверить условие $y1y2 < 0$. Если условие выполняется, то это означает, что на данном отрезке есть корень, в противном случае делается вывод об отсутствии корня на отрезке $[x; x + h]$. На рисунке 3.24 отделено два корня. Корни имеются на третьем и шестом отрезках.

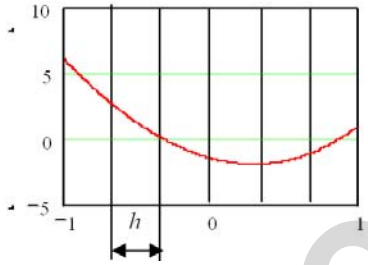


Рис. 3.24. К методу отделения корня на отрезке

Уточнение значения корня

Уточнение значений корней на отрезках отделения осуществляется различными методами одномерной поисковой оптимизации: деления отрезка пополам, простых итераций, касательных и др.

При выборе метода следует исходить из:
а) сложности подготовки задачи к решению;
б) быстродействия алгоритма;
в) времени и точности решения задачи.
Рассмотрим некоторые методы.

Метод дихотомии (деления отрезка пополам)

Достоинством метода является его простота, а также то, что этот метод не накладывает никаких ограничений на исследуемую функцию. Достаточно, чтобы на данном отрезке был корень, и притом только один.

Суть метода состоит в следующем. Отрезок отделения корня $[a; b]$ делят пополам и точку деления *c* принимают за значение корня (рис. 3.25). Сразу же проверяют, если значение функции в точке *c* станет меньше или равно требуемой точность вычисления значения функции, т. е. $f(c) \le \varepsilon$, то точка *c* принимается за значение корня и процесс поиска корня заканчивается. В противном случае определяют, на каком из отрезков – $[a; c]$ или $[c; b]$ – функция меняет знак, и выбирают его для последующего анализа. Если $F(a)F(c) \le 0$, то корень находится на отрезке $[a; c]$, в ином случае корень находится справа от точки *c*. Пусть $F(a)F(c) \le 0$, то есть корень лежит на отрезке $[a; c]$. Тогда точку *b* переносят в точку *c*, то есть $b = c$, и этот отрезок снова делят пополам и так далее. Каждый раз корень уравнения оказывается локализованным на все меньшем и меньшем отрезке. Процесс поиска корня заканчивается, когда длина отрезка $[a; b]$ станет меньше или равна заданной точности уточнения корня ε или значение функции в точке *c* станет меньше или равно требуемой точности вычисления значения функции в этой точке. Метод всегда обеспечивает сходимость процесса, то есть последовательность приближенных значений корня c_1, c_2, c_3, …, c_n будет сходиться к самому значению корня *c*. Алгоритм уточнения значения корня на отрезке отделения методом деления отрезка пополам приведен на рисунке 3.26.

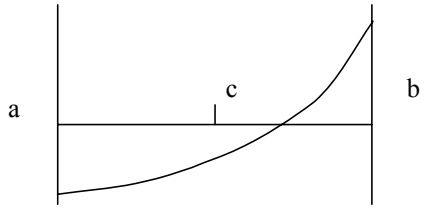


Рис. 3.25. К методу деления отрезка пополам

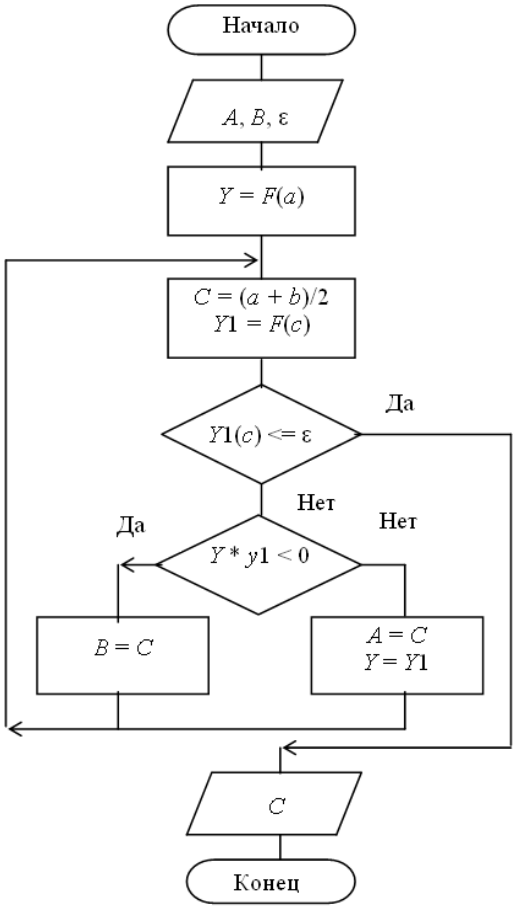


Рис. 3.26. Алгоритм решения уравнения методом деления отрезка пополам

Метод итераций

Метод простых итераций более эффективен по сравнению с методом дихотомии, но имеет ограничения. Функция должна быть непрерывной и дифференцируемой на отрезке отделения. Метод обеспечивает сходимость алгоритма, если $f'(x) < 0$.

Для реализации метода функцию $f(x) = 0$ необходимо преобразовать к рекуррентному виду путем эквивалентных преобразований:

$$x = \varphi(x). \qquad (3.21)$$

В итерационных алгоритмах рекуррентная формула имеет вид

$$X_{i+1} = \varphi(x_i). \qquad (3.22)$$

Для решения уравнения необходимо задать начальное приближение x_0 на отрезке отделения корня $[a; b]$, выбрать шаг изменения переменной Δx и последовательно решать уравнение вида 3.22. Табулировать необходимо правую часть этого выражения. Начальное приближение выбирается произвольно. Однако рекомендуется в качестве начального приближения выбирать одну из границ отрезка отделения корня, для последующих формул используется значение корня на предыдущем шаге, т. е. $f(x_i)$.

В результате решения этого уравнения получаем ряд приближений значения корня:

$$X_{i+1} = \varphi(x_i),\ i = 0, 1, 2, \ldots \qquad (3.23)$$

Процесс уточнения корня прекращается, когда выполняется условие:

$$||x_{i+1} - x_i| < \varepsilon \text{ или } f(x_{i+1}) < \varepsilon. \qquad (3.24)$$

Время поиска зависит от выбранного начального приближения x_0 и шага Δx.

Пример 3.20. Пусть дано уравнение:

$$x\cos(x) - \lg(x + 1{,}1) = 0\,.$$

Требуется уточнить корни уравнения на отрезке $[a; b]$ с точностью $\varepsilon = 0{,}001$.

Решение. Преобразуем уравнение к виду (3.21):

$$x = \frac{\lg(x + 1{,}1)}{\cos(x)}\,. \qquad (3.25)$$

Представим выражение (3.25) в виде, удобном для использования в итерационном цикле:

$$x_i = \frac{\lg(x_{i-1} + 1{,}1)}{\cos(x_{i-1})}\,. \qquad (3.26)$$

Алгоритм итерационного цикла представляет собой обычный цикл типа «Пока» и может быть реализован как цикл с предусловием или как цикл с постусловием. Но в данном случае его удобнее организовать как сочетание цикла For/Next и условного оператора IF:

```
Private Sub Iterazij()
    a = Val(InputBox("Введите начальное значение"))
    b = Val(InputBox("Введите конечное значение"))
    N = Val(InputBox("Укажите число итераций"))
    delta = Val(InputBox("Укажите требуемую точность поиска корня"))
    ' для примера можно взять значения a = 0: b = 2: N = 10: delta = 0.001
    x = a
    For i = 1 To N
        x1 = Log(x + 1.1)/Log(10)/Cos(x)
        Print x, a, b, x1
        If Abs(x1 – x) <= delta Then Print "Корень ="; x1: Exit Sub
        x = x1
        If x >= b + delta/2 Then Print "Нет корня на отрезке": Exit Sub
    Next i
End Sub
```

Метод касательных (метод Ньютона)

Пусть на отрезке $[a; b]$ отделен корень c уравнения $f(x) = 0$ и $f(x)$ – функция, непрерывная на отрезке $[a; b]$, и на интервале $(a; b)$ существуют отличные от нуля производные f' и f''.

Выберем начальное приближение x_0.

Если на отрезке $[a; b]$ $f'(x)f''(x) > 0$, то нулевое приближение корня выбираем в точке b: $x_0 = b$, если же $f'(x)f''(x) < 0$, то нулевое приближение выбираем в точке a: $x_0 = a$.

Построим график функции $y = f(x)$. Так как функция вогнутая и возрастающая (рис. 3.27), то $f'(x) > 0$ и $f''(x) > 0$. Следовательно, начальное приближение корня принимаем в точке b. Проведем касательную к графику функции в точке $B(b, f(b))$. Ее уравнение будет иметь вид:

$$y = f(b) + f'(b)(x - b). \quad (3.27)$$

Найдем точку пересечения касательной с осью x (точка b_1). Так как в этой точке $y = 0$, то, решая уравнение относительно x, получим:

$$x = b - \frac{f(b)}{f'(b)}. \quad (3.28)$$

Это и есть первое приближение решения уравнения – x_1.

Проведем касательную к графику функции в точке b_1 $(x_1; f(x_1))$. Найдем точку пересечения касательной с осью x – x_2:

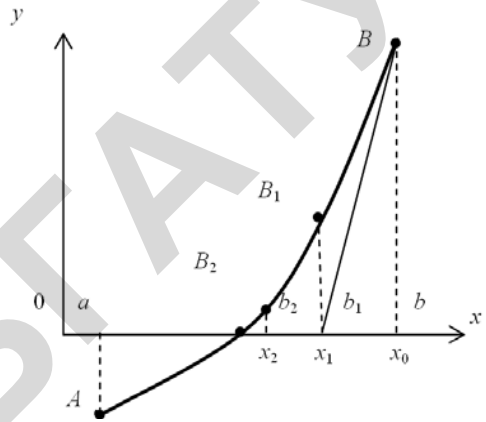


Рис. 3.27. Уточнение корня на отрезке методом касательных

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}. \quad (3.29)$$

Это второе приближения уравнения.

Для произвольной точки i получим:

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}. \quad (3.30)$$

Отношение $f(x_i)/f'(x_i)$ есть не что иное, как приращение значения аргумента – Δx_i. Действительно, так как $f'(x_i)$ численно равно тангенсу угла наклона касательной к графику функции:

$$f'(x_i) = \mathrm{tg}(\alpha) = Bb/bb_1, \text{ то } x_0 - x_1 = bb_1 = Bb/\mathrm{tg}(\alpha). \quad (3.31)$$

На каждом шаге значение Δx_i меняется.

Таким образом, формула (3.30) дает последовательность приближений x_i корня. Процесс вычислений заканчивается, когда $f(x_{i+1})$ становится меньше некоторой наперед заданной величины ε, точности поиска корня.

Недостатком метода касательной является необходимость нахождения производных.

Приближенный поиск глобальных экстремумов

Во многих задачах математического анализа требуется найти максимальное или минимальное значение функции (экстремум) на

заданном отрезке [*a*; *b*]. Экстремум может находиться на концах отрезка или внутри него. Функция может быть выпукла (выпукла вверх) или вогнута (выпукла вниз). Имеются простые способы анализа функций на отрезках, основанные на определении знака первой и второй производной.

Если функция непрерывна на отрезке, то:

если первая производная $f'(x)$ не меняет знак на концах отрезка, то функция является монотонно убывающей, если $f'(x) < 0$, и возрастающей, если $f'(x) > 0$. Монотонные функции достигают экстремума на концах отрезка;

если на концах отрезка производная функции меняет знак, то экстремум находится внутри отрезка. Причем если вторая производная больше нуля, то в точке экстремума функция достигнет минимума. Если вторая производная меньше нуля, то в точке экстремума функция достигает максимума.

Приближенный поиск экстремума функции внутри отрезка может быть выполнен с помощью табулирования функции. На каждом шаге табулирования выполняется сравнение вычисленного значения функции с некоторой переменной, принятой за максимум, и если значение функции больше этой переменной, то переменной присваивается значение функции. Аналогично выполняется поиск минимума. В качестве начального значения экстремумов функции можно принять любое значение функции на отрезке табулирования.

Пример 3.21. Найти приближенное значение глобальных экстремумов функции $y = F(x)$ на отрезке [A; B].

Решение. Введем две переменные – YMax и YMin. В качестве начального значения для переменной YMax можно взять большое отрицательное число, например, –1E38, а для переменной YMin – большое положительное число +1E38. В качестве начальных значений переменным YMax и YMin можно присвоить также значение функции на одной из границ отрезка табулирования функции, например, Ymax = F(Xa), Ymin = F(Xa).

Алгоритм поиска глобального экстремума функции:

```
алг Поиск_мин_мах                  | REM Приближенный поиск
    дано A, B, Dx                  | REM экстремума функции
нач                                | A = Val(InputBox("Укажите нача-
Ymax = F(A): Ymin = F(A)           | ло отрезка"))
    нц для x от A до B + Dx/2      | B = Val(InputBox("Укажите конец
    с шагом Dx                     | отрезка"))
```



```
        y = F(x)                   | Dx = Val(InputBox("Укажите шаг
      если y > Ymax то             | табулирования")): x = A
        Ymax = y: Xmax = x         | Ymax = (<выражение>): Ymin =
      все                          | Ymax
      если y < Ymin то             | FOR x = A TO B + Dx/2 STEP Dx
        Ymin = y: Xmin = x         |   y = (<выражение>)
      все                          |   IF y > Ymax THEN Ymax = y:
    кц                             |   Xmax = x
    рез Ymax, Ymin                 |   IF y < Ymin THEN Ymin = y:
кон                                |   Xmin = x
                                   | NEXT x
                                   | PRINT "Xmax =";Xmax, "Ymax =";
                                   | Ymax
                                   | PRINT "Xmin =";Xmin, "Ymin =";
                                   | Ymin
```

В данном примере $F(x)$ – любая заданная функция.

Так как шаг табулирования выбирается произвольно, то можно потерять значение функции на конце отрезка. Чтобы этого не произошло, условием окончания цикла принимается не B, а $B + Dx/2$. Схема алгоритма во многом будет похожа на схему, представленную на рисунке 3.17, только в тело цикла включаются дополнительно проверки значений функции на минимум и максимум.

Точность поиска зависит от выбранного шага. Чем меньше шаг, тем точнее получаемое решение, но увеличивается время решения.

Если на отрезке один экстремум, то наиболее эффективным методом поиска экстремума будет один из методов одномерной поисковой оптимизации. Для этой цели могут быть использованы те же методы, что и для уточнения корней: метод дихотомии, «золотого сечения», простых итераций.

Пример 3.22. Найти экстремумы функции $y = 5x^2 - 8x + 1$ на отрезке [–1; +1].

Решение. Находим первую производную функции y:

$$y' = 10x - 8;\ y'(a) = -18;\ y'(b) = 2.$$

Функция имеет экстремум на заданном отрезке, так как первая производная на концах отрезка меняет знак. Вторая производная функции $y'' = 10$, то есть больше нуля. Следовательно, функция имеет на отрезке [-1; 1] точку минимума.

Метод «золотого сечения»

В отличие от метода итераций, метод «золотого сечения» не требует задания начального значения корня. Достаточно указать требуемую точность поиска. Анализируют не саму функцию, а ее производную. В этом методе отрезок отделения [*a*; *b*] делится на неравные части. Одна точка (*D*) задается на расстоянии 0,618, а другая (*C*) – на расстоянии 0,382 длины отрезка [*a*; *b*] от точки *a* (рис. 3.28), и определяют, в какой из точек – *C* или *D* – производная функции меняет знак по отношению к знаку производной от этой функции в точке *a*. Если в точке *D* производная функции знак не меняет, то сразу переходят к анализу отрезка [*d*; *b*].

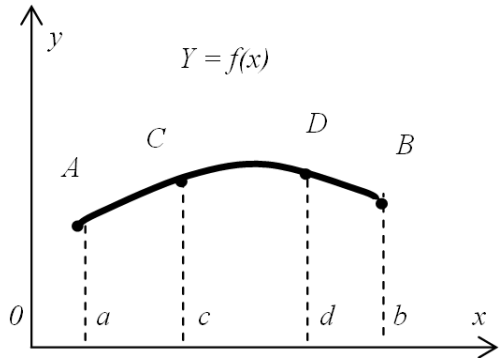


Рис. 3.28. Поиск экстремума функции методом «золотого сечения»

Если в точке *d* производная функции меняет знак, то проверяют значение производной функции в точке *c*. Если в точке *c* производная функции знак не меняет, то переходят к анализу отрезка [*c*; *d*], в противном случае анализируют отрезок [*a*; *c*]. Данный метод обеспечивает более быструю сходимость, чем метод дихотомии и метод итераций.

Пример 3.23. Программа для поиска экстремума функции $\sin(x^2) + 0{,}5$ методом «золотого сечения».

```
Option Explicit
‘ Объявление типов переменных осуществляется в разделе Общие окна Программа
Dim x As Single, y As Single, a As Single, b As Single, e As Single
Dim k As Single, c As Single, d As Single, y1 As Single
----------------------------------------------------------------------------------
```



```
Private Function fnf(x) ‘ функция пользователя
    fnf = Sin(x ^ 2) + .5
End Function
----------------------------------------------------------------------------------
Private Function fnf1(x) ‘ производная от функции пользователя
    fnf1 = 2*x*Cos(x ^ 2)
End Function
----------------------------------------------------------------------------------
Private Sub Form_Click() ‘ Обработчик события Click формы
    REM Определение экстремума функции методом золотого сечения
    Cls
    a = Val(InputBox(“Укажите начало отрезка”))
    b = Val(InputBox(“Укажите конец отрезка”))
    e = Val(InputBox(“Укажите требуемую точность”))
    k = b ‘ сохранение значения b
    m1:
    c = a + .382 * (b – a): d = a + .618 * (b – a)
    y = fnf1(a) ' fnf1 – производная от функция пользователя в точке a
    IF b – a <= e Then x = (b + a) / 2: Print “экстремум:”; x, “y =”;
    fnf(x): Exit Sub ‘ печать результата и выход из программы
    b = c: y1 = fnf1(b)
    IF y * y1 < 0 Then Goto m1 ‘ проверяется отрезок [A; C]
    a = c: b = d: y = y1: y1 = fnf1(b)
    IF y * y1 < 0 Then Goto m1 ‘ проверяется отрезок [C; D]
    a = d: b = k: y = y1: y1 = fnf1(b)
    IF y * y1 < 0 Then Goto m1 ‘ проверяется отрезок [D; B]
End Sub
```

3.3.4. Массивы

Понятие об индексированных переменных

Массивом называется совокупность элементов, имеющих одно имя и отличающихся индексом.

Например, двумерный массив:

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1j} & \dots & a_{1n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{i1} & a_{i2} & \dots & a_{ij} & \dots & a_{in} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mj} & \dots & a_{mn} \end{bmatrix}. \qquad (3.32)$$

Здесь a_{ij} – элемент массива. Индекс ***i*** означает номер строки, ***j*** – номер столбца.

В данном разделе изложены лишь общие сведения о массивах, необходимые для понимания сущности данного понятия, и способы их использования.

Массивы можно классифицировать по числу измерений и способу распределения памяти. По числу измерений массивы делятся на одномерные и многомерные, а по способу распределения памяти на статические и динамические.

Число индексов в многомерных массивах ограничено, например, в системе Visual Basic оно не превышает шестидесяти. Верхнее значение индекса может быть равно 32 767, а нижнее – 0. Индексы массивов целочисленные.

По умолчанию нумерация элементов массива начинается с нуля. Для изменения индексации с нуля на единицу используется оператор ***Option Base N***, где N может принимать значения 0 и 1. Однако при объявлении массивов можно задавать произвольные значения верхних и нижних границ массива.

Объявление массивов осуществляется оператором Dim. Размерность массива может задаваться константами или переменными. При объявлении массивов можно указывать и тип массива:

```
Dim A(10)
Dim A1(m), B(m,n)
Dim C(нижняя_граница TO верхняя_граница)
Dim A2(1 TO m), B2(5 TO 10, 1 TO 20)
```

Dim A(10) – одномерный массив, содержит 11 элементов.

Dim B(5 TO 10, 1 TO 20) – двухмерный массив, имеет 6 строк и 20 столбцов. Нумерация строк начинается с 5, а нумерация столбцов с единицы. При объявлении массивов можно указать и их тип.

В зависимости от способа распределения памяти массивы делятся на массивы со статическим распределением памяти и массивы с динамическим распределением памяти.

Массивы со статическим распределением памяти (для краткости – статические массивы) объявляются операторами ***Dim*** с атрибутом ***Static***. Если размерность массива задана константами, то массив считается статическим:

```
Dim [Static] ABC(1 TO 5, 1 TO 2).
```

Размерность статического массива не может быть изменена при выполнении программы.

Массивы с динамическим распределением памяти (для краткости – динамические массивы) объявляются оператором ***Dim*** с атрибутом ***Dinamic***. Размерности динамического массива задаются переменными. Динамический массив объявляется дважды: первый раз в разделе общие формы без указания размерности, а второй раз в процедуре с указанием размерности оператором ReDim:

```
Dim [Dinamic] A () As Variant ‘ объявление без указания размерности
---------------------------------------------------------------------------------
ReDim A(m,n) As Variant ‘ повторное объявление в процедуре с указанием размерности.
```

Ввод данных в массив и вывод данных из массива

Так как число элементов массива представляет собой всегда счетное множество, то для работы с массивами используется оператор цикла For/Next.

Пример 3.24. Ввод и вывод одномерного массива.

```
10    Option Base 1 ‘ запись в разделе общие
11    Dim A() As Single
---------------------------------------------------------------------------------
20    Private Sub Command_Click() ‘ Обработчик события кнопки
30    Rem Ввод одномерного массива
40    N = Val(InputBox(“Укажите размерность массива”))
50    ReDim A(N) As Single
60    For i = 1 TO N
70       A(i) = Val(InputBox(“Введите” & Str( i) & “элемент массива”))
80    Next i
```

Нумерация строк введена для удобства объяснения. По щелчку мыши на экран выводится окно диалога с запросом указания размерности массива, строка 40. Переобъявляется одномерный массив А размерностью N (строка 50). Затем организуется цикл с заданным числом повторений (строки 60–80). В строке 70 на экран выводится запрос на ввод значения элемента массива.

```
100   Rem Вывод одномерного массива
110   For i = 1 TO N
120       Print A(i),
130   Next i
140   End Sub
```

Алгоритм вывода массива на форму или печать подобен алгоритму ввода массива. В данном алгоритме необходимо обратить внимание на символ «запятая» в конце оператора Print (строка 120).

При наличии данного символа данные будут выводиться в пять колонок. Если этот символ удалить, то данные будут выведены в один столбец (см. описание оператора Print). При использовании в качестве управляющего символа не запятой, а точки с запятой данные также будут выводиться в одну строку (см. раздел 3.3.1).

При вводе (выводе) данных в многомерный массив необходимо организовать столько вложенных циклов, какова размерность массива.

Пример 3.25. Ввод и вывод двухмерного массива.

```
Option Base 1 ‘ запись в разделе общие
Dim A() As Single
---------------------------------------------------------------------------------
Private Sub Command_Click() ‘ Обработчик события кнопки
N = Val(InputBox(“Укажите число строк”))
M = Val(InputBox(“Укажите число столбцов”))
    ReDim A(n, m) As Single
    Rem Ввод данных в двухмерный массив
    For i = 1 To n
        For j = 1 To m
        A(i) = Val(InputBox(“Введите a(” & Str( i) & “,” & Str(j) & “) элемент массива”))
        Next j
    Next i
    Rem Вывод данных из двухмерного массива
    For i = 1 To n
        For j = 1 TO m
            Print A(i, j);
        Next j
        Print
    Next i
End Sub
```

В данном примере вывод элементов массива осуществляется в виде матрицы. Это обеспечивается за счет использования разделителя «;» в конце оператора Print. Символ «;» в конце оператора Print оставляет курсор в текущей строке. Когда ввод данных в первую строку закончится, то оператор Print без параметров возвращает курсор в начало строки.

Процедуры ввода данных в массив и вывода данных из массива типичные, поэтому их можно оформить в виде подпрограмм.

3.3.5. Операции с массивами

Массивы широко используются для хранения и обработки данных. При работе с файлами данных, вводе и выводе информации удобнее всего записывать данные предварительно в массив.

Умножение массива на число, вектор, массив

В последующих примерах операции объявления массивов, ввода и вывода данных из массивов опущены.

Пример 3.26. Умножить все элементы двухмерного массива на число *C*.

Решение. Для умножения массива на число необходимо умножить на это число каждый элемент массива.

```
For i = 1 To n
    For j = 1 To m
        A(i,j) = A(i, j)*C;
    Next j
Next i
```

Пример 3.27. Найти скалярное произведение векторов $\vec{a}$ и $\vec{b}$.

Решение. При скалярном умножении вектора на вектор получается переменная, значение которой равно сумме произведений каждого элемента второго вектора на соответствующий элемент первого вектора: $c = \sum a_i b_i$.

```
C = 0
For i = 1 To n
    C = C + A(i) * B(i)
Next i
```

Пример 3.28. Умножить двумерный массив *A* на вектор $\vec{b}$, результаты поместить в вектор $\vec{c}$.

Решение. При умножении массива на вектор необходимо, чтобы число столбцов в массиве было равно числу элементов в векторе. При умножении массива на вектор получается новый вектор, в котором число элементов равно числу строк в массиве, а каждый элемент этого вектора равен сумме произведений каждого элемента строки массива на соответствующий элемент вектора, т. е. $c_i = \sum a_{ij} b_j$. Примером может быть решение системы уравнений матричным способом: $A \cdot \vec{x} = \vec{b}$.

```
FOR i = 1 TO n
```

```
    FOR j = 1 TO m
        C(i) = C(i) + A(i, j) * B(j);
    NEXT j
NEXT i
```

Пример 3.29. Умножить двумерный массив *A* на массив *B*, результаты поместить в массив *C*.

Решение. При умножении массива *A* на массив *B* необходимо, чтобы число столбцов в массиве *A* было равно числу строк в массиве *B*. При умножении массива *A* на массив *B* получается новый массив *C*, в котором число строк равно числу строк первого массива, а число столбцов – числу столбцов второго массива, а каждый элемент этого массива равен сумме произведений каждого элемента строки массива *A* на соответствующий элемент столбца *B*, т. е. $c_{ik} = \sum a_{ij} b_{jk}$.

```
REM Умножение массива на массив
    N = Val(InputBox("Укажите число строк в массиве A"))
    M = Val(InputBox("Укажите число столбцов в массиве A"))
    P = Val(InputBox("Укажите число столбцов в массиве B"))
    ReDIM A(n, m) As Variant
    ReDim B(m,p) As Variant
    ReDim C(n,p) As Variant

REM ввод данных в массивы A и B опущен

FOR k = 1 TO p
    FOR i = 1 TO n
        FOR j = 1 TO m
            C(i,k) = C(i,k) + A(i, j) * B(j,k)
        NEXT j
    NEXT i
NEXT k
```

Вычисление сумм элементов массивов

Пример 3.30. Вычислить сумму элементов одномерного массива.

Решение.

```
S = 0
FOR i = 1 TO n
    S = S + A(i)
NEXT i
PRINT S
```


Пример 3.31. Вычислить сумму нечетных элементов двухмерного массива.

Решение. Элемент массива считается четным, если сумма его индексов четная, и наоборот – элемент массива считается нечетным, если сумма индексов элемента массива нечетная. Например, A(0,0), A(1,5) – четные элементы массива, а элементы A(0,1), A(4,7) – нечетные. Поэтому для определения суммы нечетных элементов массива необходимо в цикле подсчитывать сумму индексов элемента массива и проверять, является эта сумма четной или нечетной. Для проверки четности суммы индексов можно использовать арифметический оператор MOD (см. раздел 2).

```
S = 0
FOR i = 1 TO n
    FOR j = 1 TO m
        k = i + j
        IF k MOD 2 <> 0 THEN S = S + A(i,j)
    NEXT j
NEXT i
PRINT S
```

Поиск минимального и максимального элементов массива

Пример 3.32. Найти максимальный и минимальный элементы одномерного массива.

Решение. При поиске минимального и максимального значений элементов массива необходимо ввести вспомогательные переменные, например, Amax и Amin, и присвоить им в качестве начального значения значение любого элемента массива, например, нулевого. Для запоминания индекса максимального или минимального элемента массива также необходимо ввести вспомогательные переменные:

```
Amax = A(0): Amin = A(0)
For i = 1 To n
    If A(i) > Amax Then Amax = A(i): Imax = i
    If A(i) <Amin Then Amin = A(i): Imin = i
Next i
Print "Imax =";Imax, "Amax =";Amax
Print "Imin =";Imin, "Amin =";Amin
```

Сортировка массивов

Одной из операций с массивами является сортировка данных.

Известно много различных способов сортировки массивов. Однако, независимо от используемого способа, в программе осущест-

вляется сравнение текущего элемента массива с другим элементом этого же массива и, в зависимости от результатов сравнения, осуществляется обмен данными между этими элементами. Алгоритм обмена данными рассмотрен в примере 3.3.

Самая простая процедура сортировки – перебор.

```
FOR i = 1 TO n – 1
    FOR j = i + 1 TO n
        IF A(j)<A(i) THEN T=A(j): A(j)=A(i) : A(i)=T
    NEXT j
NEXT i
```

Приведенная процедура обеспечивает сортировку по возрастанию значения элементов массива. Для сортировки элементов массива по убыванию их значений достаточно поменять знак отношения в третьей строке.

Вычисление многочленов. Схема Горнера

Пример 3.33. Вычислить многочлен $P(x) = 5x^3 + 2x^2 – 15x – 6$ при $x = 2{,}5$.

Решение. Задачу можно решить, что называется, «в лоб» – записать выражение по правилам языка VB 6.0. Однако для повышения скорости вычисления целесообразно заменить операции умножения операциями сложения и вычитания. Особенно это актуально в управляющих ПК, работающих в режиме реального времени.

Для вычисления многочлена используется *схема Горнера* (рис. 3.29).

Здесь $A(n)$ – вектор коэффициентов многочлена, n – степень многочлена, c – значение аргумента, BO – коэффициент b_i, BN – текущее значение многочлена.

Пусть задан многочлен:

$$P(x) = a_0x^n + a_1x^{n-1} + ... + a_{n-1}x + a_n. \qquad (3.33)$$

Требуется вычислить значение многочлена при $x = c$, т. е.:

$$P(c) = a_0c^n + a_1c^{n-1} + ... + a_{n-1}c + a_n.$$

Запишем эту формулу в виде:

$$P(c) = (...((a_0c + a_1)c + a_2)c + a_3)c + ... + a_{n-1})c + a_n.$$

Мы заменили операцию возведения в степень операциями умножения и сложения, которые выполняются значительно быстрее, чем операция возведения в степень.

Введем вспомогательный вектор b размерностью n:

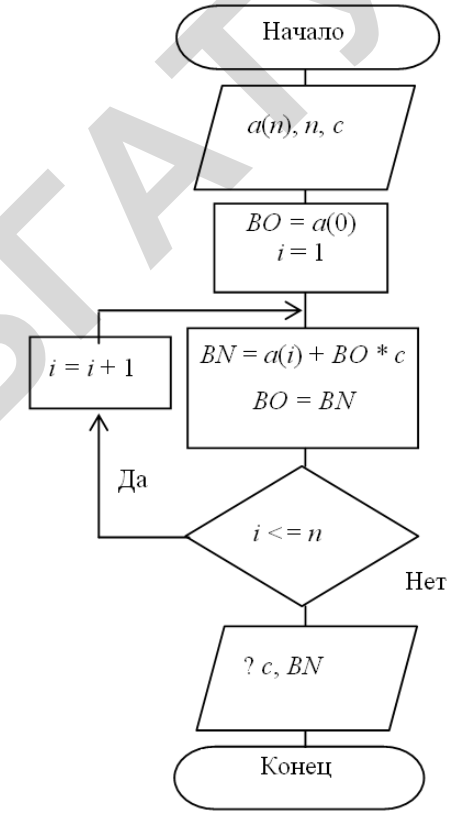


Рис. 3.29. Алгоритм вычисления многочлена

$$b_0 = a_0$$
$$b_1 = b_0c + a_1$$
$$b_2 = b_1c + a_2$$
$$...$$
$$b_n = b_{n-1}c + a_n.$$

Отсюда $b_n = P(c)$.

Таким образом, вычисление значения многочлена $P(x)$ при $x = c$ сводится к повторению элементарных операций:

$$b_i = a_i + b_{i-1}c,$$

где $i = 1, 2, ..., n$, $b_0 = a_0$.

Алгоритм вычисления многочлена $P(x)$ при $x = c$ по схеме Горнера приведен на рисунке 3.29. При вычислениях мы обошлись без использования вектора b благодаря использованию рекуррентной формулы.

Пример вычисления многочлена по схеме Горнера:

В общем виде		В числовом выражении:	
c	$a_0\ a_1\ a_2\ \ldots\ a_n$ + $b_0c\ b_1c\ \ldots\ b_{n-1}c$	2,5	5 2 –15 –6 + 12,5 36,25 53,125
$b_0\ b_1\ b_2\ \ldots\ b_n = P(c)$		5 14,5 21,25 47,125	

То есть $P(2.5) = 47{,}125$.

3.3.6. Интерполирование функций

При решении многих задач значения функции задаются в виде двумерной таблицы, в одной строке которой задаются значения аргумента, а в другой соответствующие значения функции. Точки, в которых определены значения функции, называются *узлами интерполяции*. Но этого бывает недостаточно, и требуется найти значения функций в некоторых точках, отличных от узлов интерполяции. Вычисление значения функции $f(x)$ в точках x, отличных от узлов интерполяции, называется *интерполяцией*.

Методы интерполяции используются в управляющих ЭВМ для сокращения времени вычислений. Например, требуемое состояние технологического процесса можно задать в виде таблицы значений функций в контрольных точках. Значение текущих параметров процесса в точках, отличных от узлов интерполяции, рассчитывается ЭВМ с использованием указанных методов.

Для поиска значений функции в точках, отличных от узлов интерполяции, необходимо построить функцию F, принимающую в узлах интерполяции те же значения, что и функция $f(x)$, то есть $F(x_0) = f(x_0)$, $F(x_1) = f(x_1)$, ..., $F(x_n) = f(x_n)$. Таких функций может быть построено бесконечное множество.

Способ построения функции F, принимающей в узлах интерполяции те же значения, что и функция f, называется *интерполированием*, а функцию F называют *интерполирующей функцией*. Геометрически интерполированию соответствует нахождение плавной кривой, проходящей через заданные точки $(x_i; f(x_i))$ (рис. 3.30). Так как таких кривых может быть бесчисленное множество, то задача

отыскания интерполирующей функции в общей постановке не решается однозначно. В качестве интерполирующей функции обычно используются полином Лагранжа, многочлен Ньютона, непрерывные дроби, многочлен вида $Pn(x) = a_0 x^n + a_1 x^{n-1} + \ldots + a_{n-1}x + a_n$. Наибольшей простотой с точки зрения вычисления обладают интерполяционный полином Лагранжа и интерполяционный многочлен Ньютона.

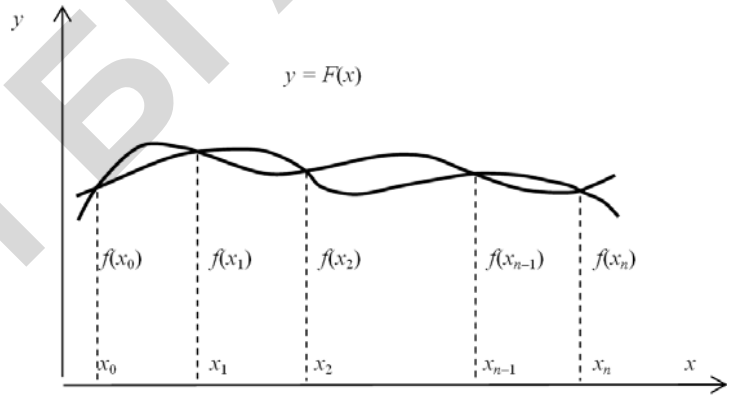


Рис. 3.30. Интерполирование

Интерполяция функций методом Лагранжа

Интерполяционный многочлен Лагранжа имеет следующий вид:

$$L_n(x) = \sum_{i=0}^{n} \frac{(x-x_0)(x-x_1)\ldots(x-x_{i-1})(x-x_{i+1})\ldots(x-x_n)}{(x_i-x_0)(x_i-x_1)\ldots(x_i-x_{i-1})(x_i-x_{i+1})\ldots(x_i-x_n)}. \qquad (3.34)$$

При $n = 1$ получаем формулу для линейной интерполяции:

$$L_1 = y_0 \frac{x-x_1}{x_0-x_1} + y_1 \frac{x-x_0}{x_1-x_0}. \qquad (3.36)$$

Для ускорения вычислений многочлена при различных значениях n применяют схему Эйткена, в которой не требуется явного построения многочлена:

$$y(x, x_0, x_1) = \frac{1}{x_1 - x_0} \begin{vmatrix} y_0 & x_0 - x \\ y_1 & x_1 - x \end{vmatrix},$$

$$y(x,x_0,x_2)=\frac{1}{x_2-x_0}\begin{vmatrix} y_0 & x_0-x \\ y_2 & x_2-x \end{vmatrix}, \qquad (3.36)$$

$$y(x,x_0,x_1,x_2)=\frac{1}{x_2-x_1}\begin{vmatrix} y(x,x_0,x_1) & x_1-x \\ y(x,x_0,x_2) & x_2-x \end{vmatrix}.$$

Теперь для вычисления значения функции в произвольной точке числовой оси, не совпадающей с узлами интерполяции, следует вычислить $y(x) = L_n(x)$.

Программа интерполяции функций методом Лагранжа

Входные данные:

N – количество интервалов интерполяции;

A – значение аргумента;

X(*N*), *F*(*N*) – массивы значений аргумента и функции;

Выходные данные:

F1 – значение функции.

Программа

```
Rem интерполяция полиномом Лагранжа по Эйткену
Dim X() As Single, F() As Single
---------------------------------------------------------------------------
Private Sub Form_Click()
N = Val(InputBox("Укажите число интервалов интерполяции"))
ReDim X(N): ReDim F(N)
For i = 0 TO N
    X(i) = Val(InputBox("Укажите значение X в" & Str(i) & "узле"))
    F(i) = Val(InputBox("Укажите значение Y в" & Str(i) & "узле"))
Next i
A = Val(InputBox("Укажите значение аргумента x"))
For j = 0 To N – 1
    For i = j + 1 To N
        F(i) = ((A – X(j)) * F(i) – (A – X(i)) * F(j))/(X(i) – X(j))
    Next i
Next j
F1 = F(N)
Print "Значение функции"; F1
End Sub
```

Интерполяционная формула Ньютона

Интерполяционная формула Ньютона имеет следующий вид:

$$P_{n-1}(x) = y_1 + (x-x_1)f(x_1;x_2) + (x-x_1)(x-x_2)f(x_1;x_2;x_3) + \\ + (x-x_1)(x-x_2)\ldots(x-x_{n-1});$$

$$f(x_1;x_2;\ldots;x_n)=\sum_{i=0}^{n-1}A_k\varphi_i\,, \qquad (3.39)$$

где $A_0 = y_1$, $A_k = f(x_1;x_2;\ldots;x_{k+1})$ – разделенные разности k-го порядка.

Программа интерполяции по Ньютону

Программа позволяет найти значение функции по значению переменной в промежуточной точке между узлами интерполяции.

Входные данные:

N – степень полинома;

X(*N*), *Y*(*N*) – массивы *x*, *y*;

A – значение аргумента.

Выходные данные:

L – значение полинома в точке *A*.

Программа:

```
Dim X(), Y()
Rem подпрограмма интерполяции по Ньютону
    N = Val(InputBox("Укажите степень полинома"))
    ReDim X(N): ReDim Y(N)
    For i = 0 TO N
        X(i) = Val(InputBox("Укажите значение X в" & Str(i) & "узле"))
        Y(i) = Val(InputBox("Укажите значение Y в" & Str(i) & "узле"))
    Next i
    A = Val(InputBox("Укажите значение аргумента x"))
    L = Y(0): S = 1
    For i = N To 1 Step –1
        i1 = N – i
        For k = 0 To I – 1
        Y(k) =(Y(k + 1) – Y(k))/(X(k + 1 + i1) – X(k))
    Next k
    S = S * (A – X(i1)): L1 = L
    L = L + Y(0) * S
Next i
Print "Значение функции";L
```

3.3.7. Обработка строковых переменных

Понятие о строковых переменных

Строковые переменные представляют собой цепочки любых символов, заключенных в кавычки. Максимальная длина строки – 32 767 символов, минимальная – ноль, пустая строка.

Для объявления символьных переменных используется суффикс – знак доллара $. Тип символьных переменных может быть объявлен также с помощью оператора DEFSTR:

A$, C1$ – строковые переменные;

DEFSTR C, S – все переменные, имена которых начинаются с символов C и S, считаются переменными символьного типа.

Тип переменной, объявленный оператором DEFSTR, можно изменить с помощью суффикса $. Строковые переменные можно объединять, используя операцию конкатенации & или +:

C2$ = C$ & C1$ или C2$ = C$ + C1$.

Ввод символьных переменных

Для ввода символьных переменных могут использоваться операторы Let и InputBox.

При использовании оператора *Let* символьная переменная заключается в кавычки:

Let A$ = “Брестская крепость – герой”

При вводе символьных переменных с помощью оператора InputBox строковые переменные записываются без кавычек. Данные вводятся в строку ввода окна диалога:

S = InputBox(“Введите фамилию первого летчика-космонавта”)

Переменная S должна быть переменной символьного типа.

Вывод символьных переменных

Для вывода символьных переменных на экран могут использоваться операторы Print и метод Print объекта Debug.

Синтаксис оператора Print при выводе символьных переменных ничем не отличается от его синтаксиса при выводе числовых данных. При выводе символьных констант они заключаются в кавычки:

Print “День победы! Как он был от нас далек…”

Функции для обработки символьных переменных

Visual Basic имеет значительное число функций для работы с символьными переменными. Рассмотрим некоторые из них.

Примечание 1. В данном разделе все строковые переменные будут обозначены символом C, а числовые переменные – символом N.

Примечание 2. Ряд функций могут использоваться для обработки как строк символов, так и числовых данных. Поэтому они имеют два синтаксиса, например:

Ltrim(Строка любого типа)

Ltrim$(Строка символьного типа) As String

В первом случае функция может обрабатывать строки любого типа: числовые или символьные. Во втором случае в качестве аргумента

должна быть строка символьного типа, и возвращает функция также строку символьного типа.

По назначению функции для работы с символьными переменными можно разделить на ряд групп.

Перевод символов таблицы ASCII в число и чисел в код ASCII: Asc, Chr$.

Синтаксис функции Asc: Asc(C).

Функция Asc выделяет из строки символов первый символ и возвращает ASCII код числа. Например, Asc(Азбука) – результат число 128 – код русской буквы А.

*Синтаксис функции Chr$*: Chr$(N).

Функция Chr$ возвращает символ, соответствующий коду числа в таблице ASCII. Например, Chr$(129) – результат символ Б.

Перевод чисел из машинного представления в строку символов и из символьного представления в числовое: Str и Val.

Синтаксис функции Str: Str(N).

Функция Str переводит число в строку символов.

Синтаксис функции Val: Val(C).

Функция Val переводит строку символов в число.

Функции для обработки строк: Len, Left$, Right$, Mid$, Ltrim$, Rtrim$, Instr; Lcase$, Ucase$.

Функция *Len(C)* возвращает длину строки.

Функции *Left$(C,N)*, *Right$(C,N)* – выделяют из строки символов C N символов слева и справа соответственно.

Функция *Mid$(C,N1,N2)* – выделяет из строки символов C N2 символов начиная с позиции N1. Например:

C1 = Mid$(“Шумит, гудит зеленый лес”,8,5)

Print C1

Результат: гудит

Функция Mid$ выделяет из строки «Шумит, гудит зеленый лес» 5 символов с восьмой позиции – слово «гудит».

Функция Mid$ может использоваться также и для замены части текста. Синтаксис функции при замене текста:

Mid$(C,N1,N2) = C1

В данном случае в строке C заменяется N2 символов строкой C1, начиная с позиции N1. Если длина строки C1 больше N2, то лишние символы отбрасываются. Если длина строки C1 меньше N2, то недостающие символы заполняются пробелами. Длина строки C при этом не изменяется. Результат замены сохраняется в строке C. Например:

C$ = “Отражается месяц в пруду”

Mid$(C,12,5) = “башня”
Print C

Результат: Отражается башня в пруду

Функции *Ltrim$(C) и Rtrim$(C)* отбрасывают лидирующие и хвостовые пробелы соответственно.

Функция *Lcase$(C)* переводит прописные буквы в строчные.

Функция *Ucase$(C)* переводит строчные буквы в прописные.

Функция *Instr([i],C1,C2)* осуществляет поиск первого вхождения строки символов С2 в строку символов С1 и возвращает номер позиции, с которой строка символов С2 входит в строку символов С1. Если вхождение не найдено, то возвращается значение 0. Поиск может осуществляться с начала строки или с позиции i; если i отсутствует, то поиск начинается с первой позиции.

C = “Светит месяц, светит ясный”
C1 = “месяц”
n = Instr(1,C,C1)
Print n

Результат: 8 – то есть начальная позиция слова «месяц» – восемь.

C = “светит месяц, светит ясный”
C1 = “светит”
n = Instr(C$,C1$)
Print n

Результат: 1.

Найдено первое вхождение слова «*светит*» в строку С. Для поиска других вхождений необходимо организовать цикл.

Функции генерирования строк символов: Space$, String$.

Функция *Space$(n)* генерирует строку из *n* пробелов.

Функция *String$* имеет два формата.

В первом формате: *String$(n,C)* – функция выделяет из строки С первый символ и генерирует строку, содержащую *n* этих символов.

Во втором формате: *String$(n,Cod)* – функция генерирует строку из n символов, определяемых кодом *Cod*. Здесь *Cod* – код кодовой таблицы ASCII. *Cod* может принимать значения от 31 до 255.

Типовые процедуры обработки символьных переменных

Строка символов рассматривается программой как одномерный массив. Символы можно сортировать и сравнивать так же, как и числа. Операции отношения над текстовыми операндами базируются на посимвольном сравнении их числовых кодов. В кодовой

таблице ASCII, например, цифры имеют меньшие номера, чем буквы. Латинские буквы имеют меньшие номера, чем символы русского алфавита. Прописные буквы имеют меньшие номера, чем строчные буквы соответствующего алфавита.

При сравнении двух строк символов программа сравнивает их посимвольно. Если строки разной длины, то строка с большим числом символов считается большей, если остальные символы совпадают. Если символы в строках разные, то при сравнении учитываются результаты сравнения по первым n символам, где *n* – число символов в короткой строке.

Рассмотрим некоторые типовые алгоритмы обработки символьных переменных.

Пример 3.34.. Выделить *i*-й символ из текста.

Задача решается с помощью функции Mid$:

C1 = Mid$(C,i,1)

Пример 3.35. Найти позицию, в которой располагаются заданный символ или цепочка символов.

Задача решается с помощью функции Instr:

n = Instr(C1,C2)

Пример 3.36. Удалить символ или цепочку символов.

Решение:

- Найти позицию, с которой заданный символ или цепочка символов входят в строку (функция Instr(C,C1)).
- Выделить часть строки слева от символа (функция Left$(C,N)).
- Выделить часть строки справа от символа (функция Right$(C,N)).
- Объединить полученные строки.

Пример 3.37. Вставить текст между *i*-м и (*i* + 1)-м символами.

- Разделить строку на две подстроки в месте нахождения заданного символа.
- Прибавить к левой части строки вставляемый текст и пробел.
- Прибавить к полученной строке правую часть строки.

Пример 3.38. Разделить текст на строки, если ограничитель – спецсимвол (рис. 3.31).

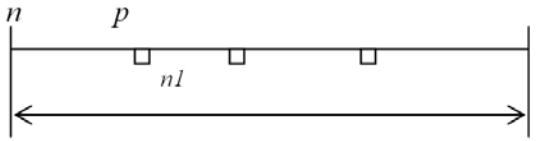


Рис. 3.31. К примеру 3.38

Решение:

1. Определить длину текста.
2. Найти первый спецсимвол (*p*).
3. Если *p* = 0, то напечатать текст от *n* до *n*1 и завершить программу; иначе напечатать текст от *n* до *p*.
4. Запомнить позицию пробела: *n* = *p* + 1 – и повторить операции по пп. 2–4.

```
Private Sub Form_Click()
    Cls
    c$ = “во поле береза стояла”
    c1$ = “ ”
    n = 1 ' текущее значение переменной цикла
    n1 = Len(c$)
    m:
    IF n < n1 Then
        p = Instr(n,c$,c1$) ' p – позиция спецсимвола (пробела)
        IF p = 0 Then Print Mid$(c$,n,n1 – (n – 1)): Exit Sub
        Print Mid$(c$,n,p – n)
        n = p + 1
        Goto m
    End IF
End sub
```

Пример 3.39. Сравнить символы или цепочки символов:
: “a” < “b”; “ab” = “ab”; “ac” > “ab”; “abc” < “abd”, но “abf” > “abd”; “abc” < “abcd”, но “fbc” > “abcd”

Пример 3.40. Выделить слово из текста.

Решение:

- Вычислить длину слова (функция LEN).
- Найти позицию вхождения слова в строку символов (функция INSTR).
- Выделить слово (функция MID$).

Пример 3.41. Определить, является буква гласной или согласной:

- Ввести строку текста.
- Ввести строку символов, содержащую только гласные, Cg$.
- Выделить один символ из текста в позиции i (функция Mid$(C,i,1)).
- Проверить, принадлежит ли символ строке гласных Cg$ (функция Instr(C,C1)). Если принадлежит, то это значит, что символ – гласная.

Контрольные вопросы

1. Что такое табулирование функций? Как оно осуществляется?
2. Поясните порядок вычисления многочлена по схеме Горнера.
3. В чем различие между аналитическими и численными методами решения уравнений?
4. Что значит отделить корни уравнения?
5. Что понимается под уточнением корней уравнения?
6. Отделите корни уравнения $2x^3 + 5x^2 – 3 = 0$.
7. Отделите корни уравнения $3x^3 – 2x^2 + 5 = 0$.
8. Назовите методы уточнения корней уравнения на отрезке отделения.
9. В чем заключается сущность метода дихотомии?
10. В чем отличие метода «золотого сечения» от метода дихотомии?
11. В чем заключается сущность метода простых итераций?
12. Назовите условия наличия экстремума внутри отрезка.
13. Сформулируйте условия выпуклости и вогнутости функции.
14. Что такое интеграл? Дайте геометрическую интерпретацию определенного интеграла.
15. Поясните способы вычисления определенного интеграла.
16. Что такое интерполирование функций?
17. Что такое символьная переменная?
18. Приведите классификацию функций для обработки символьных переменных.
19. Что лежит в основе сравнения символьных переменных?
20. На чем основан принцип обработки символьных переменных?

Заключение

В настоящем разделе мы ознакомились с основами программирования. Основными элементами языка программирования являются команды, операторы, функции, выражения. Программа представляет собой последовательность операторов. Для реализации программы используется несколько базовых структур: линейные, выбор, вычисляемый переход, циклы.

Для ввода данных используются операторы Let, InputBox.

Для вывода данных используются операторы Print, метод Print объекта Debug.

Процедуры выбора реализуются с помощью операторов If/Then/Else и Select Case. Оператор If имеет две структуры: однострочный If и многострочный If. Многострочный If позволяет соз-

давать хорошо структурированные, легко читаемые и поверяемые программы. Оператор Select Case также позволяет создавать хорошо структурированные программы и имеет больше возможностей, чем оператор If.

Циклы реализуются с помощью операторов For/Next, While/Wend, Do/Loop. Оператор For/Next создает циклы с постусловием, используется для организации циклов с заданным числом повторений. Оператор While/Wend служит для организации циклов с параметром (бесконечных циклов). Данный оператор позволяет создавать циклы с предусловием и реализует положительную логику, то есть тело цикла выполняется в том случае, когда условие истино. Оператор Do/Loop – универсальный оператор. Он позволяет создавать как циклы с предусловием, так и циклы с постусловием. При этом могут создаваться циклы как с положительной логикой, так и с отрицательной логикой.

Названные операторы позволяют создавать программы для решения самых разных задач: вычисления арифметических и алгебраических выражений, табулирования функций, исследования функций, решения задач математического анализа численными методами, работы с массивами.

Основная трудность в усвоении вопросов программирования лежит не в изучении операторов, а в знании методов математического анализа и умении составлять алгоритмы программ.

4. ТЕКСТОВЫЙ ПРОЦЕССОР MICROSOFT WORD 2010

Ключевые слова: Microsoft Office, Microsoft Word, диаграмма, лента, макросы, слияние документов, таблица, текстовый процессор, шаблон.

Офис 2010 имеет шесть модификаций в зависимости от состава приложений, включенных в пакет (табл. 4.1).

Таблица 4.1

Состав пакетов программ Microsoft Office 2010

Приложение	Стартовый	Для дома и учебы	Для дома и бизнеса	Стандартный	Профессиональный	Профессиональный плюс
Excel	Да	Да	Да	Да	Да	Да
Word	Да	Да	Да	Да	Да	Да
OneNote	Нет	Да	Да	Да	Да	Да
PowerPoint	Нет	Да	Да	Да	Да	Да
Outlook	Нет	Нет	Да	Да	Да	Да
Publisher	Нет	Нет	Нет	Да	Да	Да
Access	Нет	Нет	Нет	Нет	Да	Да
Communicator	Нет	Нет	Нет	Нет	Нет	Да
InfoPath	Нет	Нет	Нет	Нет	Нет	Да
SharePoint Workspace (Groove)	Нет	Нет	Нет	Нет	Нет	Да

Офис 2010 существенно отличается от своих предшественников. Первая его особенность – вместо стандартного главного меню с выпадающими меню второго и последующих уровней появились *ленты*. Нельзя сказать, что опытные пользователи с восторгом вос-

приняли это новшество. Поиск нужных команд на ленте значительно затруднен, их не сразу можно найти на длинной ленте. Однако при более глубоком знакомстве с лентой можно отметить и существенные достоинства, и, прежде всего, следует сказать о том, что все команды снабжены подробными всплывающими подсказками, помогающими понять назначение команды и область ее применения. Это, несомненно, важный фактор при изучении возможностей текстового процессора и удобства пользования им. Большим недостатком, по мнению автора, является обилие шаблонов, встроенных в команды, заготовок разного рода на все случаи жизни, выбрать из которых нужный сложнее, чем создать собственный шаблон. Вторая примечательная особенность – это возможность создавать веб-страницы и совместно работать над документом. Эта возможность обеспечивается с помощью программы Office Live Workspace. Этот сервис позволяет хранить файлы непосредственно в сети на виртуальном диске объемом до 5 Гбайт и вести совместную работу с коллегами над одним документом в режиме реального времени.

4.1. MICROSOFT WORD 2010. НАЧАЛЬНЫЕ СВЕДЕНИЯ

Назначение

Текстовый процессор Microsoft Word 2010 представляет собой интегрированную среду для создания и редактирования документов сложной структуры. Он обеспечивает ввод, редактирование и форматирование текста, вставку диаграмм, таблиц и рисунков, обмен данными с другими приложениями Windows, например, Excel, работу с гипертекстовыми документами, просмотр веб-страниц и размещение документов на веб-страницах, подготовку писем и их рассылку по электронной почте. Наличие средств форматирования текста, вставки фотографий и рисунков позволяет использовать Word 2010 как малую издательскую систему. Текстовый процессор содержит большой набор шаблонов, облегчающих создание стандартных документов, позволяет создавать записи в *блоге*[11].

[11] Блог (англ. blog, от web log – Интернет-журнал событий, Интернет-дневник) – веб-сайт, основное содержимое которого – регулярно добавляемые записи, содержащие текст, изображения или мультимедиа. Для блогов характерны недлинные записи временной значимости, отсортированные в обратном хронологическом порядке (последняя запись сверху). Отличия блога от традиционного дневника обусловливаются средой: блоги обычно публичны и предполагают сторонних читателей, которые могут вступить в публичную полемику с автором (в комментарии к блогозаписи или своих блогах). [Википедия]

Word 2010 создает документы в формате, отличном от формата предыдущих версий текстовых процессоров. Расширение имени файла – .docx. Поэтому файлы документов, подготовленных в Word 2010, нельзя прочитать в предыдущих версиях текстового процессора. Для обеспечения *совместимости* с предыдущими версиями Word при сохранении файла необходимо установить тип файла «Документ Word 97–2003».

Окно программы

Запуск программы осуществляется через команду ***Программы*** главного меню, щелчком мыши по кнопке *W* на рабочем столе или в строке состояния.

Выход из программы осуществляется командой ***Файл, Выход*** главного меню или комбинацией клавиш *Alt + F4*.

Окно программы Word 2010 приведено на рисунке 4.1. В верхней части расположена строка заголовка *1*, в которой выводится имя редактируемого документа (по умолчанию предлагается имя *Документ X*) и название программы. В левой части строки заголовка размещается кнопка системного меню (W), а в правой – кнопки свертывания, развертывания/восстановления и закрытия окна *6*.

Ниже строки заголовка расположены меню Файл и Лента *2*, панель быстрого доступа *3*, рабочее окно и строка состояния *11*.

В рабочем окне расположены:

окно документа *5*;

горизонтальная и вертикальная линейки прокрутки *9*. Горизонтальная линейка прокрутки появляется в том случае, когда ширина окна программы становится уже окна документа. Выше вертикальной линейки прокрутки расположены кнопка управления делением окна документа разделительной линией на две части по горизонтали *7* и кнопка управления линейками *8*, позволяющая показать или убрать линейки, не обращаясь к командам ленты Вид. На вертикальной линейке прокрутки расположена Кнопка перехода к объекту *10*. В окне документа расположен курсор ввода (точка вставки), который отмечает место вставки текста или других объектов в документ *13*;

горизонтальная и вертикальная линейки *4*, *12*. Над вертикальной линейкой расположена кнопка вставки табуляторов *14*.

Меню Файл

Кнопка меню Файл (кнопка Office) расположена в верхнем левом углу экрана. Это меню содержит команды работы с файлами: создания, открытия, закрытия, сохранения, печати и др.

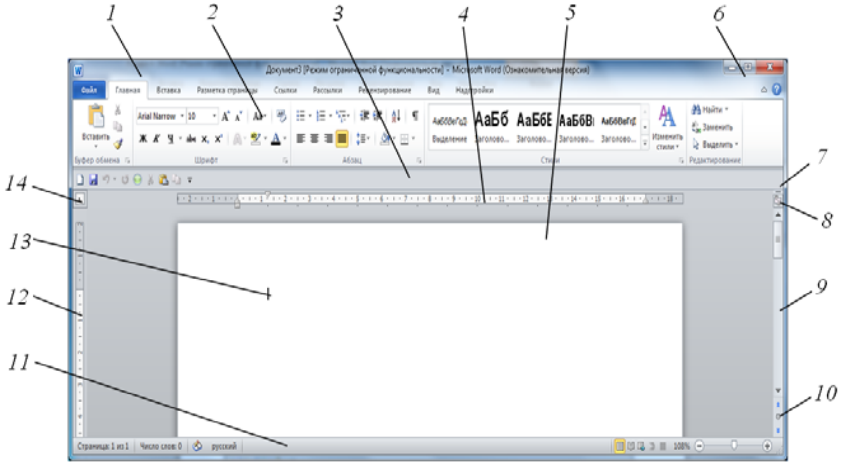


Рис. 4.1. Окно программы Word 2010:
1 – строка заголовка; *2* - лента; *3* – панель быстрого доступа; *4* – горизонтальная линейка; *5* – окно документа; *6* – кнопки свертывания, развертывания и закрытия окна; *7* – кнопка деления окна документа; *8* – кнопка управления линейками; *9* – вертикальная линейка прокрутки; *10* – кнопка перехода к объекту; *11* – строка состояния; *12* – вертикальная линейка; *13* – курсор (точка вставки); *14* – установка табуляторов

При выборе команды Файл открывается окно диалога (рис. 4.2), в котором отображаются функции, доступные в выделенной опции (команде) меню. По умолчанию открывается окно диалога Сведения, в котором приведены сведения о редактируемом документе. Программа имеет дружественный интерфейс, что позволяет достаточно легко освоиться с ее возможностями. Для выхода из меню щелкните по ярлычку ленты Главная или ярлычку другой ленты.

Лента

Лента – это новшество, предложенное фирмой Microsoft. Ленты обеспечивают доступ к командам в зависимости от выполняемой операции. Впервые ленты появились в офисе 2007 и были встречены старыми пользователями в штыки, но со временем они прижились и даже стали копироваться другими производителями программ. И тем не менее, ленты содержат множество заготовок, найти среди которых нужную не так-то просто, некоторые настройки стали довольно сложны и неудобны, например, вставка номеров страниц.

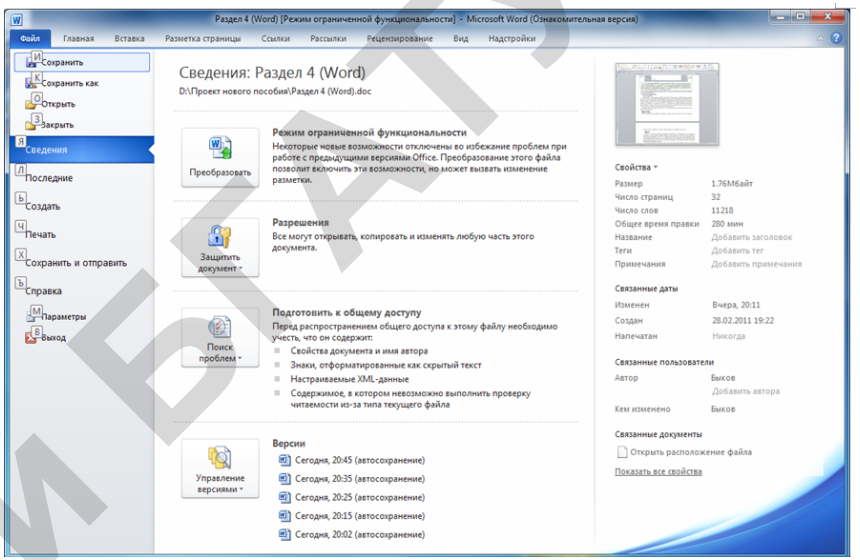


Рис. 4.2. Меню Файл

Лента состоит из трех элементов: *вкладок*, *групп* и *команд*.

Вкладки. Лента содержит тринадцать ***основных вкладок*** и множество ***вкладок инструментов***. На вкладках собраны команды по их функциональному назначению. Доступ к вкладкам осуществляется с помощью ярлычков (которые можно принять за команды главного меню). В Word 2010 имеется возможность добавлять на ленту новые кнопки (в Word 2007 такой возможности нет), а также добавлять новую вкладку, новую группу на существующую вкладку с любым набором команд. Для простоты изложения в тексте документа вкладки ленты будем называть иногда просто лентами.

Команды – это кнопка, поле для ввода информации или меню. Все кнопки снабжены всплывающими подсказками.

Группы. Каждая вкладка содержит несколько групп команд, объединенных по функциональному назначению. Например, на вкладке Главная имеется пять групп: **Буфер обмена**, **Шрифт**, **Абзац**, **Стили**, **Редактирование**. Вид группы Шрифт приведен на рисунке 4.3. Группа содержит набор команд для редактирования шрифтов: выбор типа шрифта, размера, начертания (полужирный, курсив, подчеркнутый) и другие команды. В правом нижнем углу группы имеется стрелка, щелчок по которой мышкой откроет окно

диалога для настройки параметров шрифтов, знакомое пользователям по прежним версиям текстового процессора (рис. 4.4).

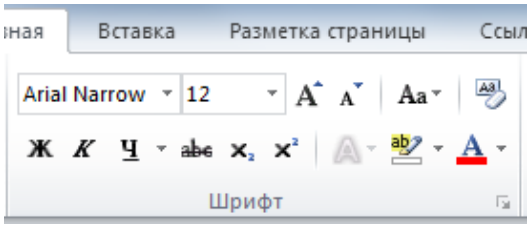


Рис. 4.3. Группа Шрифт вкладки Главная

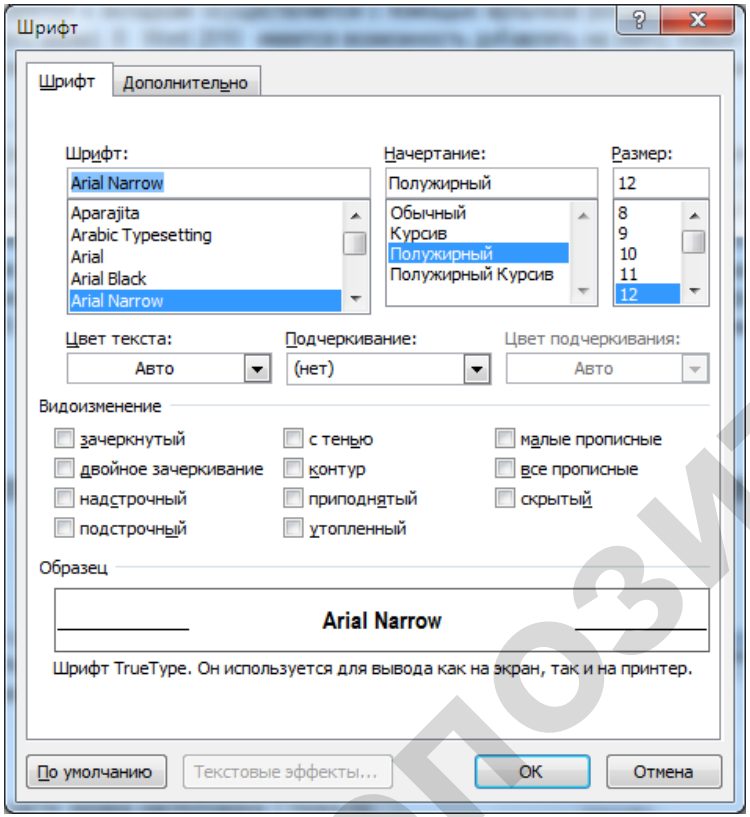


Рис. 4.4. Окно диалога настройки параметров шрифтов

Настройка ленты

Для настройки ленты необходимо выполнить следующие операции:

- Вызовите контекстное меню ленты щелчком мыши по свободному участку ленты и выберите пункт меню ***Настройка ленты*** или выберите команду ***Параметры, Настройка ленты*** в ***меню Файл***. Откроется окно диалога (рис. 4.5), в правой части которого будет расположена структура всех вкладок ленты, а в левой – список всех доступных кнопок и команд.

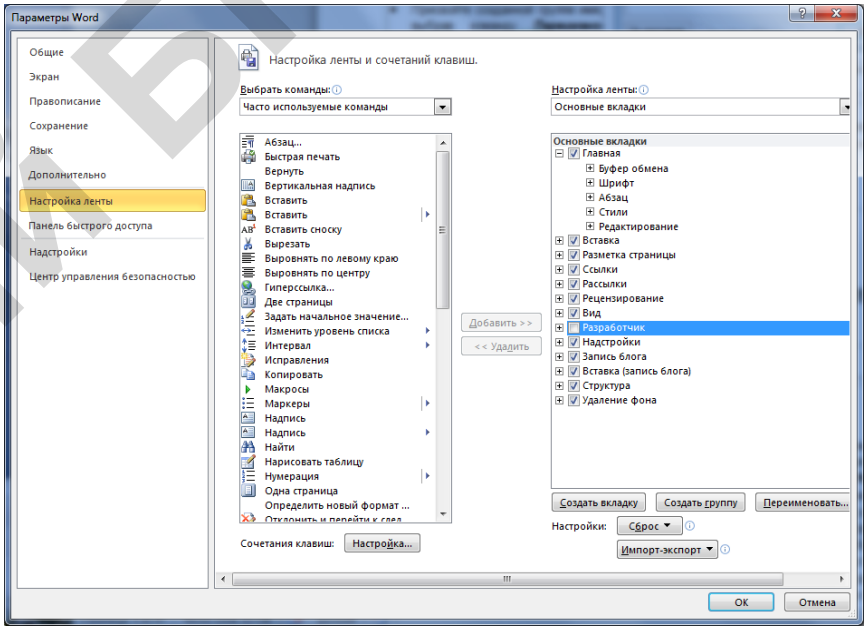


Рис. 4.5. Настройка ленты

- Разверните структуру нужной вкладки щелчком мыши по значку + и создайте в ней новую группу кнопкой ***Создать группу***, расположенной в нижней части окна.
- Присвойте созданной группе имя, выбрав команду ***Переименовать***.
- Выделите в левом списке нужную кнопку и переместите ее в созданную группу командой ***Добавить***.

Ленту можно скрыть. Для этого можно воспользоваться контекстным меню или кнопкой в виде стрелки, расположенной над верх-

ним правым краем ленты. Здесь же находится символ «?» – вызов справки (рис. 4.1).

Панель быстрого доступа

Панель быстрого доступа позволяет разместить на ней те кнопки, которые должны быть всегда под рукой и которых нет на вкладке Главная, например, кнопки Сохранить, Откат, Создать новый документ. Справа от установленных кнопок расположена кнопка раскрывающегося списка ***Настройка панели быстрого доступа***. Эта кнопка открывает всплывающее меню, в котором содержится набор команд. Команду можно поместить на панель, щелкнув по ней мышкой. У активных кнопок появляются метки. Повторный щелчок по команде удаляет команду с Панели быстрого доступа. Если среди команд списка нет нужной команды, щелкните по кнопке ***Другие команды***. Откроется окно диалога, подобное окну диалога на рисунке 4.5 для добавления команд на панель быстрого доступа. В окне диалога справа расположена Панель быстрого доступа, а слева – список команд. Выделите нужную команду и щелкните по кнопке ***Добавить***, а для удаления команды с Панели быстрого доступа выделите команду в правом списке и щелкните по кнопке ***Удалить***.

Строка состояния

Строка состояния расположена в нижней части окна программы (рис. 4.6). В левой части строки состояния выводится информация о редактируемом документе: номер текущей страницы, общее число страниц в документе, число слов в документе, используемый язык ввода. В правой части строки состояния расположены кнопки управления режимами просмотра документа, а также управление масштабом его представления. Аналогичные команды размещены в группах ***Режимы просмотра документа*** и ***Масштаб*** вкладки ***Вид***.

Рис. 4.6. Строка состояния

В Microsoft Word 2010 предусмотрено пять режимов просмотра: Разметка страницы, Режим чтения, Веб-документ, Структура и Черновик.

Режим Разметка страницы – основной режим. В этом режиме документ представляется со всеми элементами форматирования в том виде, как он будет выглядеть при печати.

Режим чтения. В этом режиме окно документа развертывается на весь экран. В нем нельзя вносить изменения в документ, но можно просматривать текст, вставлять примечания, выделять фрагменты текста цветом. Можно воспользоваться при необходимости мини-переводчиком.

Режим Веб-документа представляет документ в виде веб-страницы.

Режим Структура удобен для просмотра структуры документа. При переходе в этот режим открывается вкладка Структура (рис. 4.7). Раскрывающийся список ***Показать уровень*** позволяет управлять представлением документа, что показать: заголовки первого уровня, двух уровней и т. д. Раскрывающийся список слева позволяет выбрать уровень текста, одинарные стрелки слева и справа от списка позволяют повысить или понизить уровень заголовка на один уровень, двойная стрелка слева от списка повышает заголовок до первого уровня, а двойная стрелка справа от списка понижает уровень заголовка до обычного текста. Стрелки «вверх» и «вниз» позволяют перемещаться по структуре документа, кнопка «+» развертывает структуру раздела, а кнопка «–» сворачивает структуру раздела.

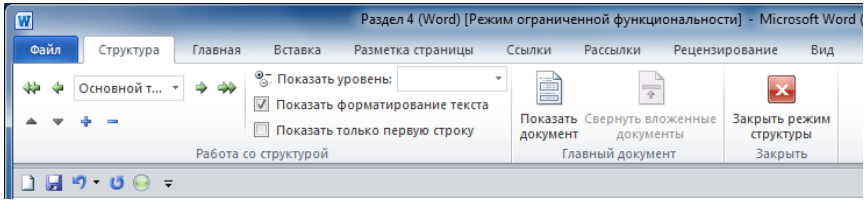


Рис. 4.7. Вкладка Структура

Режим Черновик позволяет просматривать текст документа. В этом режиме многие элементы форматирования не отображаются, в том числе рисунки, диаграммы. Отсутствует разделение документа на страницы. Границы страниц отмечаются линией из точек, а границы разделов отмечаются двойной линией из точек. Границы разделов можно удалить, выделив линию и нажав клавишу ***Delete***.

Линейки прокрутки

Линейки прокрутки служат для просмотра документа. На вертикальной линейке прокрутки имеется несколько кнопок. Кнопки с одиночной стрелкой служат для перемещения текста на одну

строку в соответствующем направлении, двойные стрелки перемещают текст на страницу. Быстрое перемещение по тексту удобно с помощью ползунка: зацепите ползунок мышью и перемещайте в требуемом направлении. При этом слева от ползунка появляется всплывающее окно сообщения, в котором отображаются номер страницы и наименование раздела.

На вертикальной линейке прокрутки имеется кнопка – ***Выбор объекта перехода***. При щелчке мышью по этой кнопке открывается меню (рис. 4.8), в котором можно выбрать объект для быстрого перехода: страница, разделы, примечания, сноски, концевые сноски, поля, таблицы, рисунки, заголовки, исправления, а также команды Найти, Переход и Отмена.

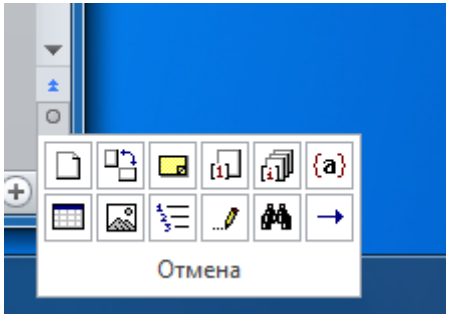


Рис. 4.8. Меню кнопки Выбор объекта перехода

Создание и открытие документов

Начало работы с документом начинается либо с создания нового документа, либо с открытия уже существующего документа. И в том, и в другом случае начинать работу необходимо с открытия меню ***Файл***.

Для создания нового документа введите в окне программы Microsoft Word команду ***Файл, Создать***[12]. Открывается окно диалога (рис. 4.9). Выберите ***Новый документ*** и щелкните по кнопке ***Создать***. Все документы создаются на основе шаблонов. Шаблон

[12] Имеется несколько способов создания документов, также как и выполнения других команд. В данном пособии будем рассматривать, один, классический способ, присущий текстовому процессору. С другими способами пользователь познакомится по мере приобретения опыта и навыков работы в операционной среде Windows.

содержит набор стилей форматирования текстов, оформления документа, определенный состав объектов, полей ввода и т. п. Документ, который создается командой ***Новый документ, Создать***, тоже основан на шаблоне *Нормальный*. Microsoft Word 2010 предлагает пользователю большое число шаблонов разного назначения. Если имеющихся шаблонов недостаточно, то можно поискать нужный шаблон в Интернете на сайте ***Office.com***. Имеется возможность создать свой *пользовательский шаблон*, который будет сохранен в папке Мои шаблоны. Для создания документа на основе шаблона необходимо выбрать в окне диалога нужный шаблон и щелкнуть по кнопке Создать.

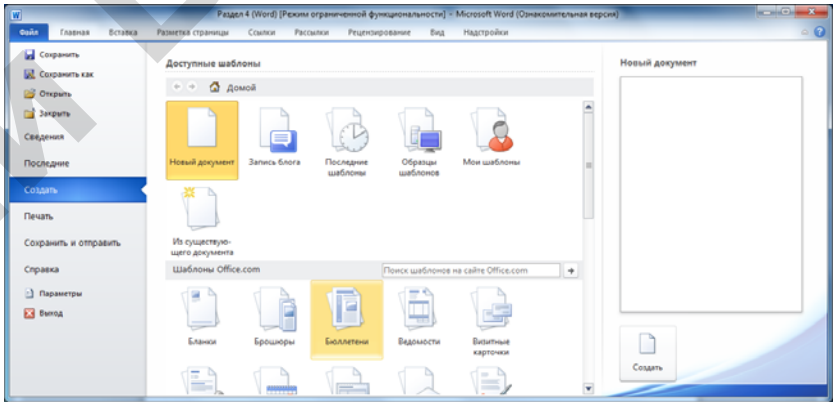


Рис. 4.9. Создание нового документа

Для открытия существующего документа введите команду ***Файл, Открыть*** и выберите в окне диалога нужный файл (рис. 4.10). Окно ввода *Имя файла* имеет раскрывающийся список последних файлов, с которыми работал пользователь. Список *Сервис* открывает доступ к сетевому диску, если он создан пользователем. Открывающийся список справа от окна ввода Имя файла позволяет установить фильтр для ограничения числа доступных файлов. Можно установить фильтр Документы Word, тогда будут доступны только документы, созданные в данной версии текстового процессора. При установке фильтра Документы Word 97–2003 будут доступны только документы, созданные в предыдущих версиях текстового процессора и т. д., по умолчанию фильтр отсутствует, и поэтому в окне диалога присутствуют все папки и все файлы, имеющиеся на выбранном диске. Для

завершения операции открытия файла следует щелкнуть по кнопке ***Открыть***. Кнопка Открыть имеет открывающийся список, который позволяет управлять способом открытия документа, с которым полезно познакомиться. Одной из опций этого списка является опция ***Открыть и восстановить***. Этой опцией следует воспользоваться, если возникли проблемы с открытием файла документа.

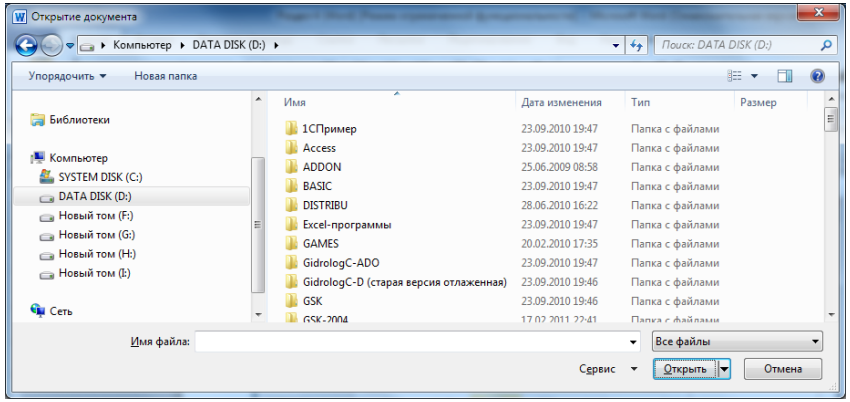


Рис. 4.10. Окно диалога Открытие документа

Другой способ найти файлы, с которыми работал недавно пользователь, – воспользоваться командой ***Последние*** меню ***Файл*** (рис. 4.9). В этом случае откроется внушительный список файлов и папок, к которым пользователь обращался недавно.

Сохранение документа

Первое, о чем должен позаботиться пользователь, работая в редакторе Word, это о сохранении созданного документа. В случае аварийного выключения питания или каких-либо неисправностей имеется опасность потерять результаты всей работы или ее части. Поэтому сразу же после загрузки программы и открытия нового документа сохраните его на диске командой ***Файл, Сохранить*** или ***Файл, Сохранить как***. При этом открывается окно диалога (рис. 4.11).

Окно сохранения документов стандартное для всех приложений Windows. В верхней части окна диалога, ниже заголовка расположен открывающийся список *Предыдущие расположения 4*, в котором отображается спецификация (полное имя) текущей папки. Слева от списка расположены две кнопки навигации *1*, а справа –

строка для поиска файлов *7*. Ниже списка *Предыдущие расположения* расположена строка меню *5*. В центральной части окна диалога расположены два списка: левый список *3* содержит список папок, а правый список *6* – содержание активной папки. Выберите нужный диск и папку для сохранения файла. В нижней части окна диалога расположены два списка: Имя файла и Тип файла. В строке ввода Имя файла вводится имя файла или выбирается из раскрывающегося списка. Тип файла также выбирается из списка. По умолчанию предлагается тип файла «Документ Word», расширение имени файла .docx. Для обеспечения совместимости с предыдущими версиями текстового процессора выберите тип файла *Документ Word 97–2003*.

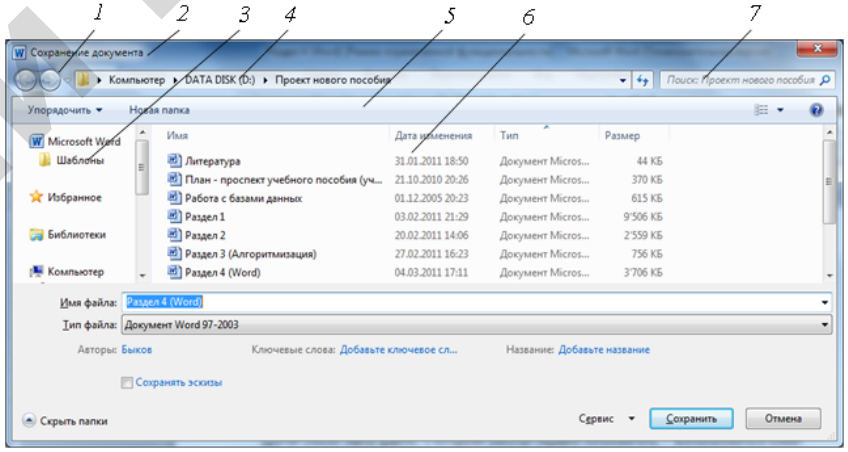


Рис. 4.11. Окно диалога Сохранение документа:
1 – кнопки навигации; *2* – строка заголовка; *3* – список папок список;
4 – предыдущие расположения; *5* – строка меню; *6* – список файлов и папок активной папки; *7* – строка поиска

В процессе работы целесообразно периодически сохранять информацию на диске, чтобы минимизировать возможные потери информации. Кроме того, для этого можно установить режим автосохранения: выберите в меню ***Файл*** команду ***Параметры, Сохранение*** и установите флажок ***Автосохранение каждые…***, а также установите интервал времени, через который будет выполняться автосохранение документа. По умолчанию установлен

интервал времени автосохранения 10 минут. Желательно также установить флажок *Сохранять последнюю автосохраненную версию при закрытии без предупреждения*.

Печать документов

Для вывода документа на печать необходимо воспользоваться командой ***Печать*** меню ***Файл*** или соответствующей кнопкой панели быстрого доступа.

В Word 2010 нет традиционной команды Предварительный просмотр. Очевидно, разработчики решили, что многостраничный просмотр документа путем изменения его масштаба вполне компенсирует эту потерю.

При вводе команды ***Печать*** меню Файл на экран выводится диалоговое окно настройки параметров печати (рис. 4.12). В этом окне можно выбрать, что печатать: весь документ, текущую страницу, указанные номера страниц или выделенный фрагмент текста. Дополнительные возможности управления выводом на печать предоставляют раскрывающиеся списки *Напечатать все страницы* и *Односторонняя печать*. Режим Односторонняя печать позволяет печатать документ на обеих сторонах листа с ручной перекладкой листов: вначале печатаются все нечетные страницы, затем программа предлагает переложить листы. После перекладки листов достаточно нажать кнопку Ок, чтобы продолжить печать.

Контрольные вопросы

1. Назовите основные элементы окна программы Microsoft Word?
2. Назовите основные вкладки ленты.
3. Как выполнить настройку ленты?
4. Как добавить кнопки на Панель быстрого доступа?
5. Поясните назначение кнопок в строке состояния.
6. Как сохранить документ на диске?
7. Как вывести документ на печать?

4.2. ВВОД И РЕДАКТИРОВАНИЕ ТЕКСТА

Ввод текста

Текст документа вводится после точки ввода, которая отображается мигающей вертикальной чертой (курсор). Конец абзаца или текста обозначается специальным символом ¶. Чтобы показать символы конца абзацев, следует активизировать кнопку *Непечатаемые символы*

в группе *Абзац* вкладки *Главная*. Текст можно вводить в любой точке рабочей страницы, для этого щелкните дважды мышью в нужном месте страницы.

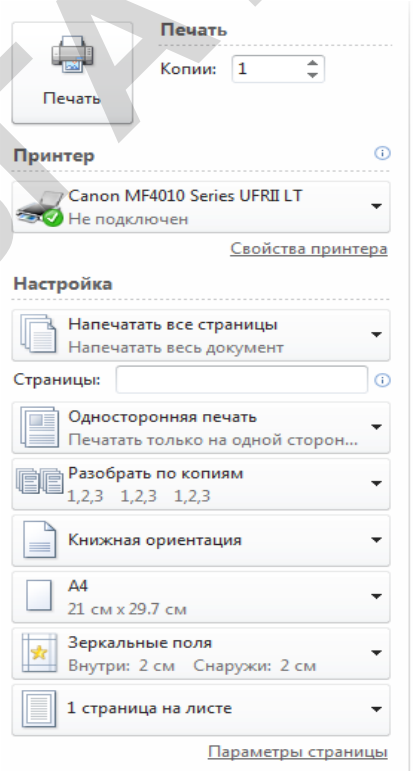


Рис. 4.12. Окно настройки параметров печати

При вводе длинного текста он автоматически переносится на другую строку. При нажатии клавиши Enter курсор ввода переходит на другую строку в положение абзацного отступа. Для вставки пустой строки нажмите клавишу Enter.

Чтобы разбить строку на две строки, установите курсор в точку раздела и нажмите клавишу Enter. Для объединения строк установите курсор в конец первой строки и нажмите клавишу Delete (Del) или установите курсор в начало второй строки и нажмите клавишу BackSpace (Возврат на шаг).

Выделение текста

Все операции в редакторе: копирование, перемещение, удаление – выполняются над выделенным текстом. Для выделения текста можно использовать несколько способов:

- Отдельное слово выделяется двойным щелчком мыши.
- Абзац выделяется тройным щелчком мыши.
- Другой способ выделения абзаца – установите указатель мыши на поле выделения и протяните мышью по этому полю до последней строки выделяемого текста. *Поле выделения* – левое поле документа. Когда указатель мыши находится на поле выделения, он приобретает форму стрелки ⇗.
- Для выделения произвольного фрагмента текста установите указатель мыши в начало выделяемого текста и протяните мышью до конца выделяемого текста. То же можно сделать с помощью клавиш управления перемещением курсора при нажатой клавише Shift: установите курсор в начало строки выделения, нажмите клавишу Shift и, не отпуская ее, нажмите и удерживайте клавишу «Вправо» или «Вниз» до тех пор, пока курсор ввода не достигнет конца выделяемого текста.

Перемещение или копирование текста

Для копирования текста можно пользоваться командами ***Копирование*** и ***Вставка*** группы *Буфер обмена* вкладки *Главная* или использовать прием перетаскивания выделенного текста мышью при нажатой клавише Ctrl. Эти операции удобно выполнять также с помощью контекстного меню.

При копировании текста, таблиц, рисунков и других объектов с помощью команд меню или панели инструментов следует придерживаться следующего алгоритма:

- Выделите копируемый текст или объект.
- Введите команду ***Копировать***. Выделенный текст помещается в буфер обмена. Из буфера обмена текст может быть вставлен в любое место текущего документа или другого документа.
- Переместите курсор в место вставки.
- Введите команду ***Вставить***. При вставке объекта предусмотрено три параметра вставки, которым соответствуют кнопки *Сохранить исходное форматирование*, *Объединить форматирование* и *Сохранить только текст*. Первый режим предусматривает, что фрагмент текста вставляется с тем форматированием, которое он имел в источнике. В третьем режиме сохраняется только текст без элементов форматирования, при этом тип шрифта приводится

в соответствие с типом шрифта текущего документа, а вот второй режим позволяет сохранить форматирование исходного текста, но тип шрифта устанавливается в соответствии с типом шрифта текущего документа.

Перемещение отличается от копирования тем, что исходный текст удаляется. В этом случае для помещения текста (объекта) в буфер целесообразно использовать команду ***Вырезать*** группы *Буфер обмена*. При перемещении объекта мышью клавиша Ctrl не нажимается.

Команда ***Вставить*** группы *Буфер обмена* имеет подменю, позволяющее выбрать режим вставки объекта.

Полезно также запомнить комбинации клавиш: [Ctrl + C] – забрать выделенный текст в буфер обмена и [Ctrl + V] – вставить текст из буфера обмена.

Поля

Документ, подготовленный пользователем, содержит ряд элементов: поля, заголовки, форматированный текст, абзацы, колонтитулы, таблицы, рисунки, графики, формулы и т. п. Основные элементы документа приведены на рисунке 4.13.

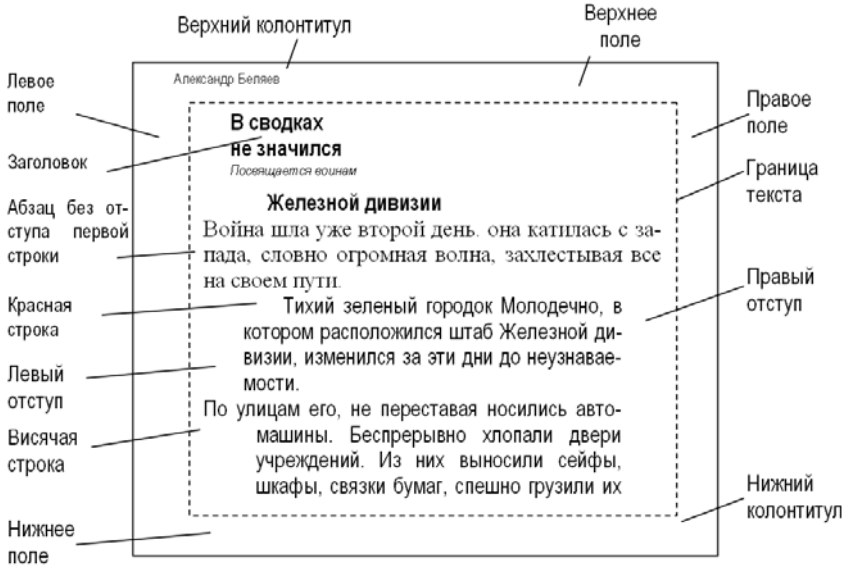


Рис. 4.13. Фрагмент оформления страницы документа

Поля ограничивают текст документа. Левое поле используется обычно для подшивки и принимается равным 20–25 мм, правое поле – не менее 10 мм. Верхнее и нижнее поля используются для размещения колонтитулов. Ширина верхнего поля принимается равной 20 мм, а нижнего поля – 20–25 мм. Размеры полей можно настраивать. Настройка параметров страницы осуществляется с помощью команд группы ***Параметры страницы*** на вкладке ***Разметка страницы*** или линейками. Там же можно вызвать окно диалога Параметры страницы (рис. 4.14) щелчком по кнопке в правом нижнем углу группы. Окно диалога имеет три вкладки.

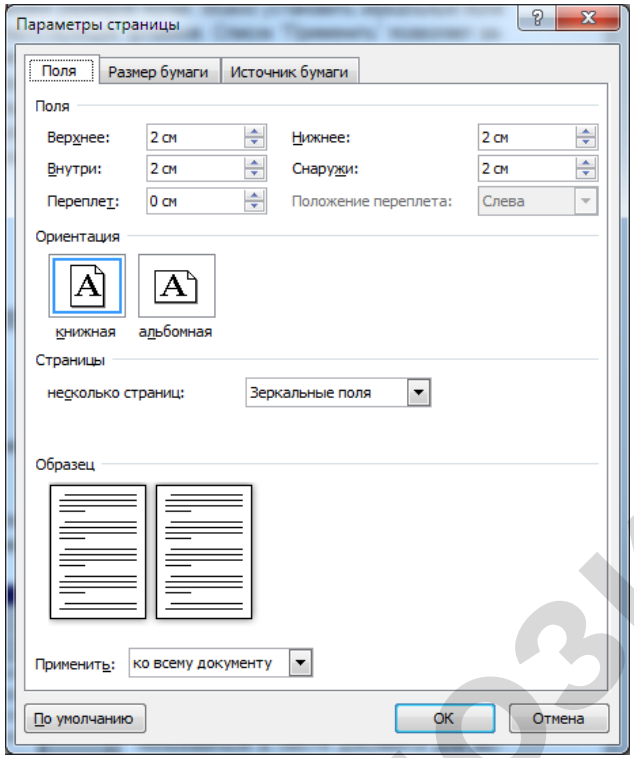


Рис. 4.14. Окно диалога Параметры страницы

Вкладка *Поля* предназначена для установки размеров полей. Можно установить ориентацию страниц (книжную или альбомную), зеркальные поля или несколько страниц на листе с помощью

одноименного открывающегося списка. Список «Применить» позволяет задать область действия установленных параметров: ко всему документу или от текущего листа до конца документа.

Вкладка *Размер бумаги* позволяет выбрать размер бумаги из заданных стандартных форматов или установить свой пользовательский формат, а также произвести настройку параметров печати.

Вкладка *Источник бумаги* дает возможность выполнять настройку разделов, колонтитулов, выравнивания текста на странице, установки границ и нумерации строк.

Для удобства размещения таблиц, рисунков и других объектов, вставляемых в документ, полезно установить на страницы границы текста (рис. 4.13). Граница текста отображается пунктирной линией только в электронном документе. Для ее установки введите команду ***Файл, Параметры, Дополнительно*** и в группе ***Показывать содержимое документа*** установите флажок ***Показывать границы текста***.

Заголовки

Заголовки оформляются с помощью стилей заголовков из списка ***Стили***, расположенного в группе ***Стили*** на вкладке ***Главная*** ленты (рис. 4.15). Здесь же имеются два списка для изменения стилей документа. Выделенные стили можно настраивать с помощью контекстного меню.

Стили, используемые в заголовках, не должны использоваться в тексте документа для выделения других фрагментов текста. При оформлении заголовков с помощью стилей открывается возможность автоматизировать процедуру составления оглавления (содержания) документа.

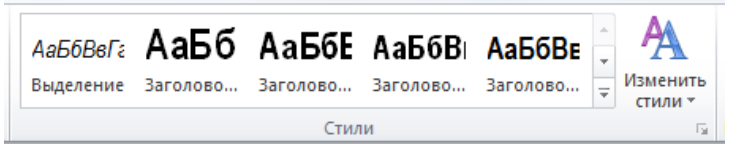


Рис. 4.15. Группа Стили вкладки Главная

Оформление шрифтов

Чтобы сделать текст более читабельным, удобным для чтения и восприятия пользователем, применяют различное оформление шрифта. К параметрам шрифта, подлежащим настройке, относятся: тип шрифта, размер, начертание (полужирный, курсив, подчеркну-

тый), цвет, межсимвольный интервал и ряд дополнительных эффектов.

Настройку параметров шрифтов осуществляют с помощью команд группы ***Шрифт*** (рис. 4.16) вкладки ***Главная*** ленты или окна диалога ***Шрифт*** (рис. 4.4), которое вызывается кнопкой, расположенной в правом нижнем углу группы *Шрифт*. Окно диалога имеет две вкладки: *Шрифт* и *Дополнительно*. На вкладке *Шрифт* настраиваются практически все параметры шрифтов. Все изменения немедленно отображаются в окне *Образец*. Для печатного текста наиболее подходящими являются шрифты Times New Roman – шрифт с засечками и Arial – прямоугольный шрифт. Не все шрифты имеют в своем составе русские символы. На вкладке *Дополнительно* нас интересует один параметр – межзнаковый интервал. Он может быть обычным, разреженным или уплотненным. Для разреженного и уплотненного шрифта предусмотрена возможность регулировки степени разрядки или уплотнения. При этом используется типографская единица измерения – пункт. Другие команды этой вкладки касаются настройки типографских параметров шрифтов.

Рис. 4.16. Группы Шрифт и Абзац вкладки Главная ленты

Оформление абзацев

Основной текст состоит из абзацев (рис. 4.13). В литературе абзац – это фрагмент текста, который содержит, как правило, законченную мысль. Для текстового процессора абзац – это текст, заключенный между двумя нажатиями клавиши Enter. При нажатии клавиши Enter автоматически вставляется символ возврата каретки. Точка вставки (курсор) переводится на новую строку в положение отступа первой строки. При оформлении абзацев могут настраиваться следующие параметры: уровень текста, выравнивание, отступы от границ текста, первая строка, межстрочный интервал и расстояние между абзацами, положение на странице. Параметры абзацев настраиваются с помощью команд группы Абзац (рис. 4.16) вкладки *Главная* ленты или окна диалога *Абзац* (рис. 4.17), вызываемого кнопкой, расположенной в правом нижнем углу группы *Абзац*.

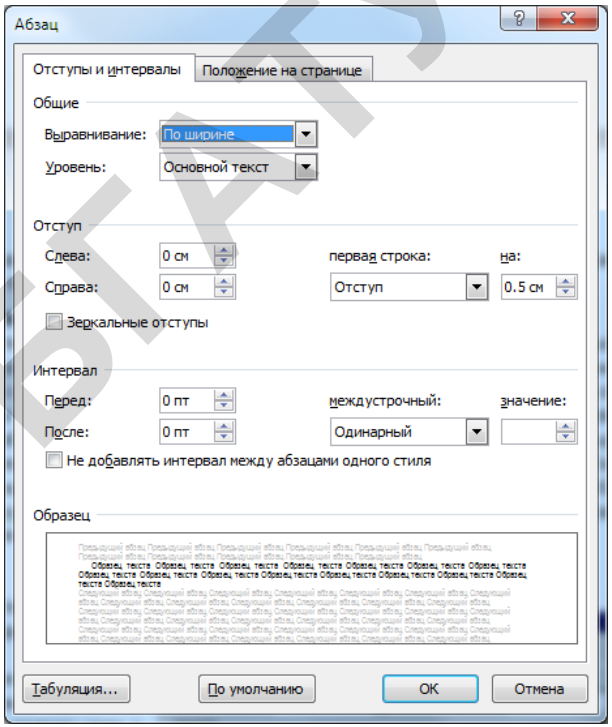


Рис. 4.17. Окно диалога для настройки параметров абзаца

Уровень текста выбирается из списка, он определяет высоту шрифта.

Выравнивание может принимать четыре значения: по левому краю, по центру, по правому краю и по ширине. Выравнивание по ширине предпочтительнее при оформлении документов, так как в этом случае оба края текста будут ровными.

Отступы слева и справа определяют расстояние текста от границы текста, но не от края страницы.

Первая строка может иметь три значения: без выступа, выступ вправо – красная строка или выступ влево – висячая строка (рис. 4.13).

Интервал между абзацами и *междустрочный интервал* измеряются в пунктах.

Вкладка *Положение на странице* определяет возможность использования висячей строки, управления переносом абзацев при

переходе текста на следующую страницу, переносом слов, нумерацией строк, параметрами надписей.

Часть параметров абзацев можно регулировать с помощью горизонтальной линейки (рис. 4.18).

Рис. 4.18. Горизонтальная линейка

На горизонтальной линейке имеется три маркера: два в левой части и один в правой части линейки. Эти маркеры позволяют устанавливать отступы и выступ первой строки. Если маркеры «Отступ слева» и «Отступ первой строки» совмещены, то выступ у первой строки отсутствует, если маркер «Отступ первой строки» находится справа от маркера «Отступ слева», то формируется красная строка, а если маркер «Отступ первой строки» находится слева от маркера «Отступ слева», то формируется выступ – висячая строка (рис. 4.19).

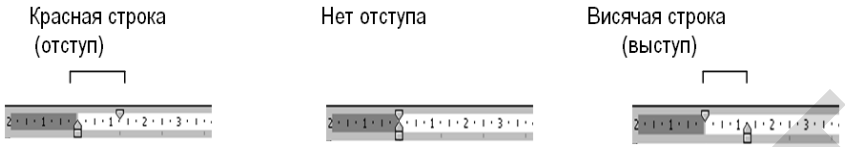


Рис. 4.19. Форматирование абзацев с помощью горизонтальной линейки

На горизонтальной линейке в левой и правой частях имеются серые полоски – это установленные значения полей. Размеры полей можно изменить с помощью мыши. Установите указатель мыши на границу раздела поля таким образом, чтобы он принял вид двунаправленной стрелки, и перетащите границу раздела в нужном направлении.

Табуляторы

Выше вертикальной линейки (рис. 4.1) находится кнопка для установки табуляторов. Табуляторы позволяют управлять размещением текста. По умолчанию текст выравнивается по левому краю, этому состоянию соответствует маркер табуляции ∟, маркер

⊥ означает выравнивание по центру, а маркер ⌋ – выравнивание по правому краю, четвертый тип маркера – выравнивание по десятичному разделителю и пятый тип маркера – с чертой |. Маркер применяется к выделенному абзацу или абзацам. Установить позиции табуляции можно также с помощью команды ***Табуляция*** окна диалога ***Абзац*** (рис. 4.17).

Границы и заливка

Границы и заливка применяются для выделения текста, оформления страниц и абзацев. Для этой цели можно воспользоваться кнопкой ***Границы*** в группе ***Абзац*** вкладки ***Главная*** ленты или командой ***Границы страницы*** в группе ***Фон*** вкладки ***Разметка страницы*** ленты.

Выделите текст, который требуется заключить в рамку, и введите команду ***Границы страницы*** из группы ***Разметка страницы*** – открывается окно диалога Границы и Заливка (рис. 4.20). Окно диалога имеет три вкладки. Вкладка *Граница* позволяет установить границы на выделенный фрагмент текста, а вкладка *Страница* позволяет установить границы страницы для всего документа или раздела, первой страницы или всех страниц, кроме первой. При установке границ имеется возможность выбрать тип рамки, тип линии, а также цвет и толщину линии. В правой части окна диалога имеется окно, в котором отображается образец рамки, созданной пользователем. Рядом с образцом расположены четыре кнопки. С помощью этих кнопок можно управлять отображением границ: установить границы с одной, двух, трех или со всех сторон. Вкладка *Заливка* позволяет установить цвет фона для абзаца или выделенного фрагмента текста. Для отмены заливки или границ выделите абзац или фрагмент текста, для которого необходимо выполнить удаление границ или заливки, войдите в окно диалога и выберите объект со словом «нет»: ***Граница, Тип – нет*** или ***Заливка, Нет заливки***.

Списки

Редактор Word позволяет автоматизировать процесс создания списков. Списки могут быть нумерованные, маркированные и многоуровневые (иерархические). Примеры списков приведены на рисунке 4.21.

Для оформления списков необходимо воспользоваться кнопками в группе Абзац (рис. 4.16) вкладки Главная Ленты. Списки можно оформлять двумя способами:

1. Написать текст, выделить его, выбрать в раскрывающемся списке вида списка нужный образец маркера (номера) и щелкнуть

по нему мышью. При необходимости можно изменить значок или стиль оформления номера.

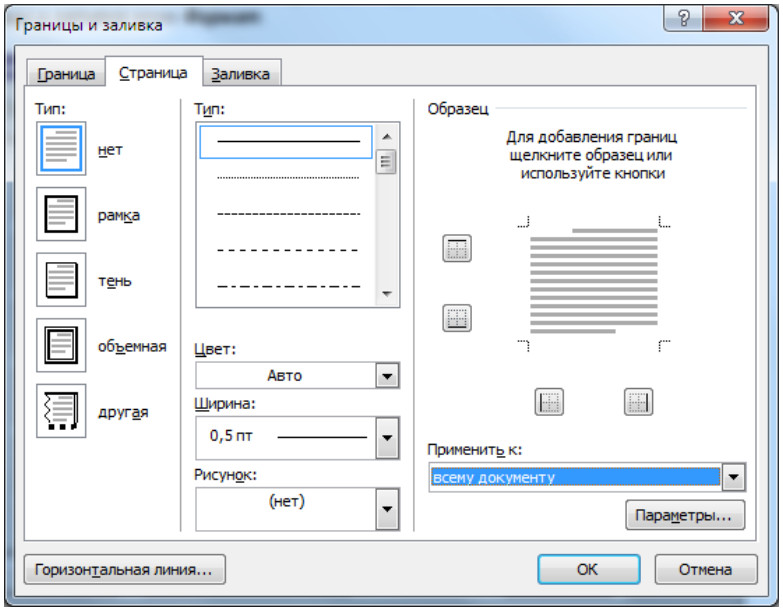


Рис. 4.20. Окно диалога Границы и заливка

Список		
нумерованный	маркированный	многоуровневый
1. Петров П. П. 2. Иванов Г. К. 3. Цветков П. С. 4. Полежаев Н. Р.	• Клубника • Черешня • Слива • Виноград	1. Глава 1 1.1. Раздел 1 1.1.1. Подраздел 1 1.1.2. Подраздел 2

Рис. 4.21. Примеры видов списков

2. Выбрать нужный вид списка. В точке вставки появится первый номер списка или маркер. Ввести текст первого элемента списка. Нажать клавишу Enter. При переходе на новую строку автоматически появляется следующий номер.

Необходимо отметить, что данная функция в Word 2010 реализована хуже, чем в предыдущих версиях.

Группа Редактирование

Следует отдельно остановиться на группе *Редактирование* вкладки Главная. Здесь расположены кнопки Найти, Заменить и Выделить. Первые две кнопки предназначены для поиска и замены текста в документе. А кнопка Выделить содержит подменю: *Выделить все*, *Выбор объектов*, *Выделить текст, имеющий такой же формат*, *Область выделения*.

Команда ***Выделить все*** позволяет выделить весь текст документа, к которому можно применить общую команду, например, копирование, удаление, перемещение, изменение типа шрифта, параметров абзаца и т. п.

Команда ***Выбор объектов*** применяется для выделения разрозненных графических объектов. При активизации этой команды можно выделять группу графических объектов обводкой рамкой (установить курсор в верхнем левом углу выделяемой области, нажать левую кнопку мыши и обвести пунктирной рамкой выделяемые объекты; объекты, которые полностью попадут в рамку, будут выделены) или выделять объекты щелчком мыши при нажатой клавише Ctrl.

Команда ***Выделять текст, имеющий такой же формат*** позволяет выделять фрагменты текста, размещенные в разных частях документа и имеющие одинаковый формат, для применения к ним общей команды форматирования.

Если доступна команда ***Область выделения***, то при ее активизации справа открывается список всех графических элементов. Теперь выделить эти элементы можно путем выбора из списка. Достоинство этой команды состоит также в том, что имеется возможность редактировать элементы даже сгруппированного рисунка. Данная команда, как и многие другие команды, *недоступна в режиме ограниченной функциональности*, то есть когда редактируется документ, например, с расширением .doc, подготовленный в предыдущих версиях текстового процессора.

Контрольные вопросы

1. Перечислите основные элементы форматирования текста.
2. Поясните назначение кнопок в группе Шрифт ленты.
3. Какие параметры шрифтов Вы знаете, как их установить?
4. Поясните назначение кнопок в группе Абзац ленты.
5. Какие параметры абзацев Вы знаете, как их установить?
6. Как создать маркированный (нумерованный, иерархический список)?

7. Как изменить параметры списка?
8. Как установить границы абзаца, страницы?
9. Как сохранить документ на диске?
10. Как вывести документ на печать?

4.3. ОФОРМЛЕНИЕ ДОКУМЕНТА

Деловой документ должен быть определенным образом оформлен. В него могут вставляться различные объекты: номера страниц, сноски, рисунки, формулы, таблицы, диаграммы, оглавление и т. п. Рассмотрим некоторые из возможностей текстового процессора по оформлению документов.

Вставка различных объектов в документ осуществляется с помощью вкладок ***Вставка***, ***Ссылки***, ***Рецензирование*** ленты.

Вставка объектов групп Колонтитулы, Текст и Символы вкладки Вставка

Вставка номера страниц

Вставка номеров страниц осуществляется командой ***Номер страницы*** группы ***Колонтитулы*** вкладки ***Вставка*** (рис. 4.22). Команда имеет раскрывающийся список, который предлагает выбрать место размещения номера страницы и установить формат страницы. При выборе места размещения номера страницы, например, внизу страницы, открывается список вариантов размещения номера: слева, справа, по центру и другие варианты.

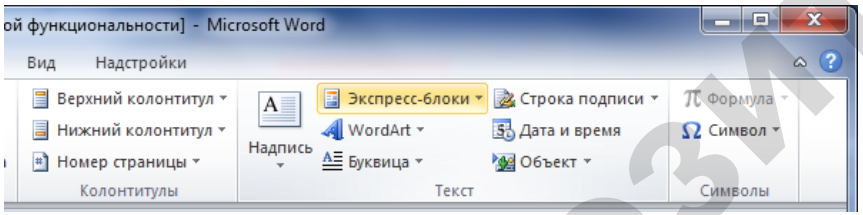


Рис. 4.22. Вставка объектов групп Колонтитулы, Текст и Символы

При выборе команды ***Формат номера страницы*** открывается окно диалога (рис. 4.23), которое предоставляет возможность выполнить дополнительные настройки: установить формат номера, включить номер главы, указать начальный номер страницы.

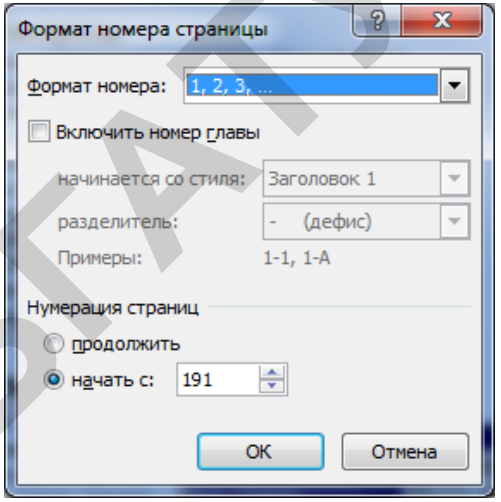


Рис. 4.23. Настройка параметров номеров страниц

При установке номера страницы необходимо также указать, с какой страницы следует начать нумерацию страниц: с четной страницы, с нечетной страницы или выбрать другой вариант из списка ***Начать раздел*** в окне диалога ***Параметры страницы*** на вкладке ***Источник бумаги*** (рис. 4.24).

Колонтитулы

Колонтитул – это короткий текст, расположенный в верхнем или нижнем поле страницы. Это может быть краткое сообщение, например, фамилия автора документа, наименование раздела, номер страницы, дата и время разработки документа, рисунки и т. п. По умолчанию все колонтитулы одинаковые, однако, можно изменить эту настройку. При выборе команды *Вставить колонтитул* открывается вкладка ленты *Работа с колонтитулами* (рис. 4.25), которая имеет ряд групп: колонтитулы, вставка, переходы, параметры, положение и закрыть.

Группа *Параметры* позволяет установить три разных вида колонтитулов: колонтитул первой страницы, колонтитулы четной и нечетной страниц. На первой странице колонтитулы обычно не пишутся. Текстовый процессор Word – многослойный документ. Колонтитулы пишутся во втором слое. Текст документа в этом режиме недоступен для редактирования, но может отображаться на странице при установке флажка *Показать текст документа*

в группе *Параметры*. В группе *Переходы* размещены кнопки для перехода к нижнему и верхнему колонтитулам, а также кнопки перехода к следующему и к предыдущему колонтитулам, поэтому для перемещения по колонтитулам не надо листать страницы. Кнопка *Как в предыдущем разделе* позволяет оформить колонтитулы в текущем разделе так же, как и в предыдущем разделе. Колонтитулы могут устанавливаться одинаковые во всем документе или в пределах раздела. Данные параметры могут настраиваться в окне диалога ***Параметры страницы*** (рис. 4.24) – список ***Применить*** (ко всему документу, к разделу, до конца документа).

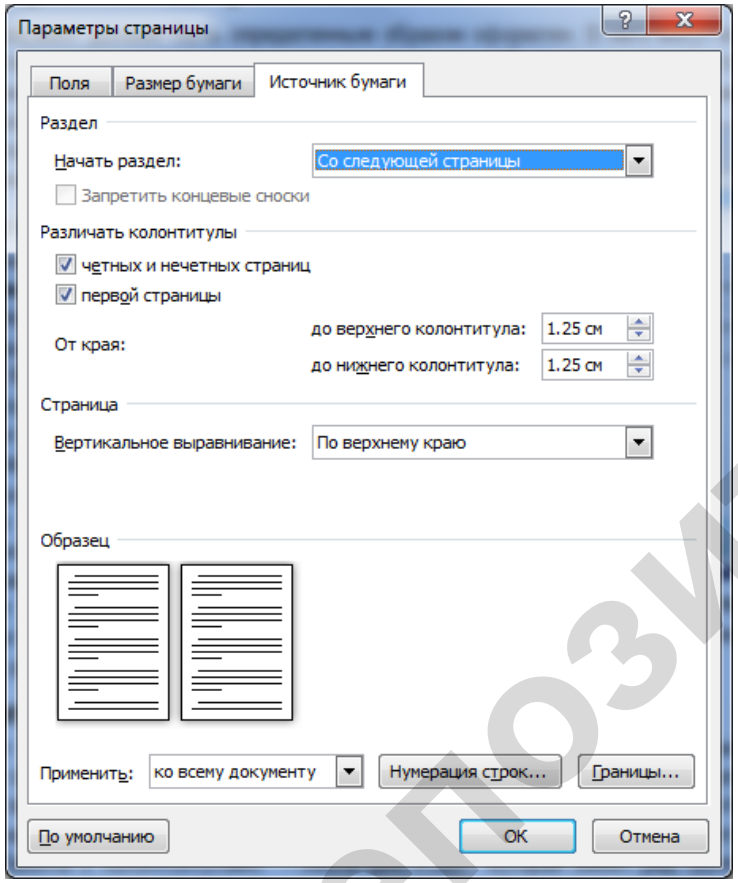


Рис. 4.24. Настройка параметров колонтитулов

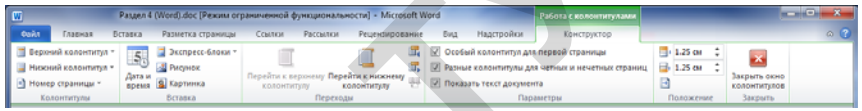


Рис. 4.25. Настройка параметров колонтитулов

В группе ***Текст*** ленты ***Вставка*** собран ряд команд для вставки текста и объектов (рис. 4.22): Буквица, Надпись, Строка подписи, Экспресс-блоки, WordArt, а также команды вставки даты и времени и объектов.

Надпись

Данный объект позволяет вставлять предварительно отформатированный текст. Это по сути дела текстовый документ, заключенный в рамку, к которому можно применять все элементы форматирования текста, а также вставлять другие объекты: рисунки, картинки, формулы и т. п. Рамку можно убрать с помощью команды *Формат объекта*. При вводе команды открывается меню заготовок размещения надписей. Внизу списка разместилась команда ***Нарисовать надпись***, которую пользователь может использовать для создания собственных форматов надписей. Созданную надпись можно сохранить в коллекции надписей. Надпись можно вставить также из меню ***Вставка***, группа ***Иллюстрации***, команда ***Фигуры***, объект ***Надпись*** (А).

Буквица

Позволяет создать большую заглавную букву в начале слова (как в детских книгах). Для создания буквицы выделите заглавную букву в первом слове предложения и введите команду ***Вставка, Текст, Буквица***. В окне диалога выберите способ размещения буквицы: в тексте или на поле. При необходимости выберите другой тип шрифта, начертание и другие параметры буквицы.

Дата и время

Дата и время вставляются ***командой Дата и Время*** из группы ***Текст***. После ввода команды открывается одноименное окно диалога, которое позволяет вставлять в документ поле даты и времени. Имеется возможность выбрать формат представления даты и времени. Флажок *Обновлять автоматически* позволяет установить режим автоматического обновления даты. Если флажок установлен, то при каждой загрузке документа будет устанавливаться текущая дата.

Строка подписи

Строка подписи – новинка Word 2010. Позволяет вставить цифровую подпись, заверяющую подлинность документа. Для создания

цифровой подписи законодательством Республики Беларусь установлены специальные процедуры.

WordArt

Данная команда позволяет вставлять броские, красочные заголовки, выбираемые из коллекции шрифтов. При выборе команды на экране появляется шаблон, в который необходимо ввести свой текст. К данному тексту можно добавлять некоторые элементы форматирования, например, курсив.

Экспресс-блоки

Позволяет вставлять предварительно созданные, отформатированные фрагменты текста с возможностью его повторного использования, автотекст, поля Word, свойства документа.

Команда Объект

Данная команда позволяет вставлять текст из файлов, а также загружать в документ объекты, подготовленные другими приложениями: Word, Excel, Power Point, Mathcad, графический редактор Bitmap Image, редактор формул Microsoft Equation 3.0 и др.

Вставка спецсимволов

Вставка символов, отсутствующих на клавиатуре, осуществляется с помощью команды ***Символы*** из группы Символы (рис. 4.22). Для ввода символа откройте список ***Символ*** и выберите команду ***Другие символы***. Откроется окно диалога *Символ* (рис. 4.26). Окно имеет две вкладки: Символы и Специальные знаки. В верхней части окна расположены два списка. Меняя тип шрифта в левом списке и набор символов в правом списке, можно быстро найти нужные символы. Для вставки символа выделите его мышью – символ выделяется на синем фоне, щелкните мышью по кнопке ***Вставить***.

Вставка формул

Часто при оформлении научных работ, отчетов требуется вписать в текст документа формулу. Раньше это делалось квалифицированными специалистами с помощью перьевой ручки и туши. Редактор Word предоставляет в распоряжение пользователя инструменты для выполнения этой работы самостоятельно пользователем. Нельзя сказать, что вписывание формул с помощью программ редактора быстрее, чем ручкой, скорее наоборот. Но повышение качества формул несомненно. Для вписывания формулы

в текст документа имеется специальная программа ***Microsoft Equation 3.0***, которая вызывается командой ***Объект, Объект***, вкладка ***Создание*** из группы ***Текст*** вкладки ***Вставка***. При загрузке программы на экране появляется панель инструментов (рис. 4.27), которая содержит набор элементов для вставки в формулы. При выборе любого значка открывается всплывающая панель с набором значков, соответствующих выбранной теме. Каждый значок имеет знакоместа, куда вписываются символы. Выбор знакоместа осуществляется мышью.

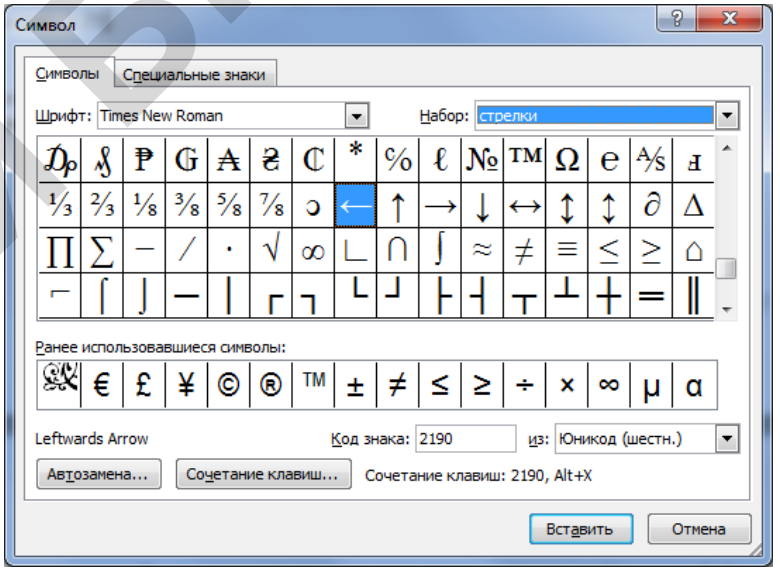


Рис. 4.26. Вставка специальных символов

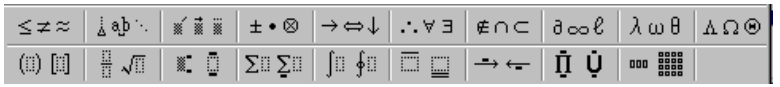


Рис. 4.27. Панель инструментов программы Microsoft Equation 3.0

В Word 2010 появился дополнительный инструмент для вставки формул: библиотека математических символов и стандартных математических формул, например, Бином Ньютона: $(x+a)^n = \sum_{k=0}^{n} \binom{n}{k} x^k a^{n-k}$.

Эта библиотека вызывается командой ***Формула*** в группе ***Символы*** вкладки ***Вставка.***

Пример вставки формулы:

$$\cos^3(x^2)\sqrt[3]{\frac{\lg(x)}{e^2}}+\sum \Rightarrow \cos^3(x^2)\sqrt[3]{\frac{\lg(x)}{e^2}}+\sum_{i=1}^{n} i^2 .$$

Вставка рисунков

Редактор Word позволяет вставлять в текст документа картинки из набора рисунков или из файла, рисованные объекты, подготовленные средствами редактора или другими графическими редакторами, например, Image Bitmap, CorelDraw и др. Вставка рисунков осуществляется с помощью команд из группы ***Иллюстрации*** вкладки ***Вставка*** (рис. 4.28). Технология вставки картинки следующая: щелкните мышкой по значку ***Картинка*** – откроется окно диалога Картинка (рис. 4.29). Откройте список ***Искать объекты*** и установите флажки у той группы объектов, которые хотите найти, щелкните по кнопке ***Начать***.

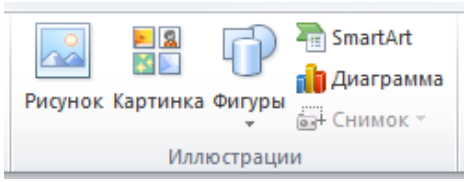


Рис. 4.28. Вставка картинок и рисованных объектов

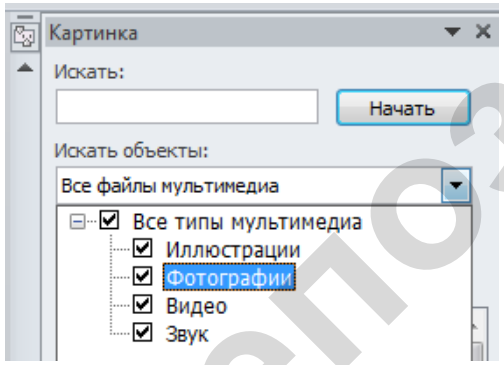


Рис. 4.29. Окно диалога вставки картинок

Открывается библиотека рисунков. Выберите нужный рисунок и щелкните по нему мышкой – рисунок вставляется в документ в точке вставки.

Команда ***Рисунок*** позволяет вставить рисунок из файла. После щелчка мышью по кнопке открывается окно диалога ***Вставка рисунка***. По умолчанию открывается папка ***Изображения***. Если в ней нет нужных рисунков, поищите их в других папках. Удобно воспользоваться окном поиска файлов по их расширению (.emf, .wmf, .bmp, .jpg).

Для вставки объекта из графического редактора ***Image Bitmap*** выполните следующее: введите команду ***Вставка, Текст, Объект*** и выберите на вкладке ***Создание*** в списке ***Тип объекта*** объект ***Image Bitmap*** – откроется окно графического редактора. Создайте рисунок. Для вставки созданного рисунка в документ щелкните мышью по видимой части окна редактора Word. Чтобы отредактировать рисунок, щелкните по нему дважды мышью – откроется графический редактор.

Рисунки можно создавать и с помощью встроенных средств редактора. Откройте список ***Фигуры*** в группе ***Иллюстрации*** – открывается панель инструментов (рис. 4.30), на ней расположены инструменты для рисования и фигуры.

Рассмотрим назначение некоторых элементов.

Инструменты для рисования: *Линия*, *Стрелка*, *Прямоугольник* и *Овал* – служат для изображения соответствующих объектов. Для установки этих элементов на страницу документа выделите соответствующий объект мышью, переместите указатель мыши в нужную точку рабочего листа, нажмите левую клавишу мыши и протащите указатель мыши в нужном направлении. Примеры построенных объектов приведены на рисунке 4.31. Чтобы выделить объект, щелкните по нему мышью. Выделенный объект отмечается маркерами в виде маленьких прямоугольников. Для изменения размеров объекта зацепите мышью за маркер и протащите в нужном направлении. Все операции выполняются над выделенным объектом. Для выделенного объекта можно изменить цвет, тип линии, тип штриха с помощью соответствующих инструментов. Выделенный объект можно удалить, нажав клавишу *Del*.

Как только будет нарисован первый объект, открывается лента ***Средства редактирования, Формат***, которая предоставляет в распоряжение пользователя дополнительные элементы управления рисованными объектами (рис. 4.32). Содержание вкладки зависит от типа выделенного объекта.

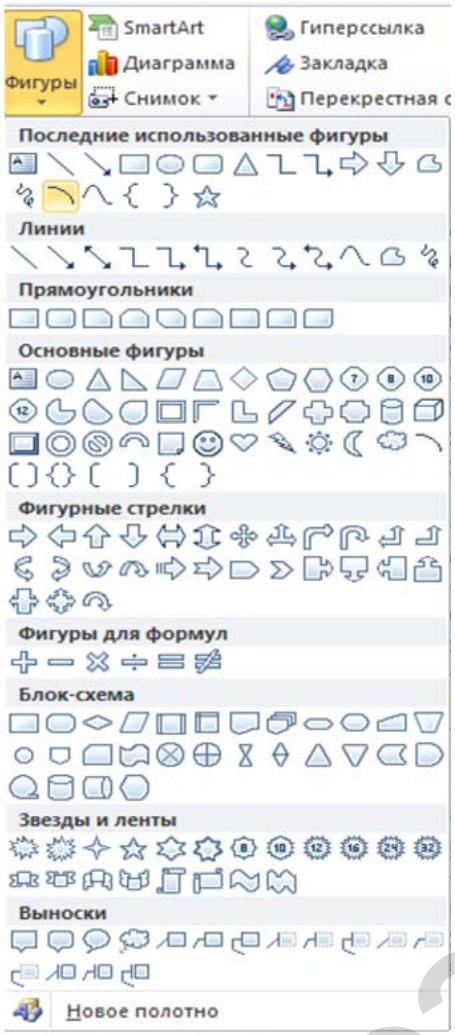


Рис. 4.30. Инструменты и фигуры для рисования

Рис. 4.31. Примеры объектов рисования

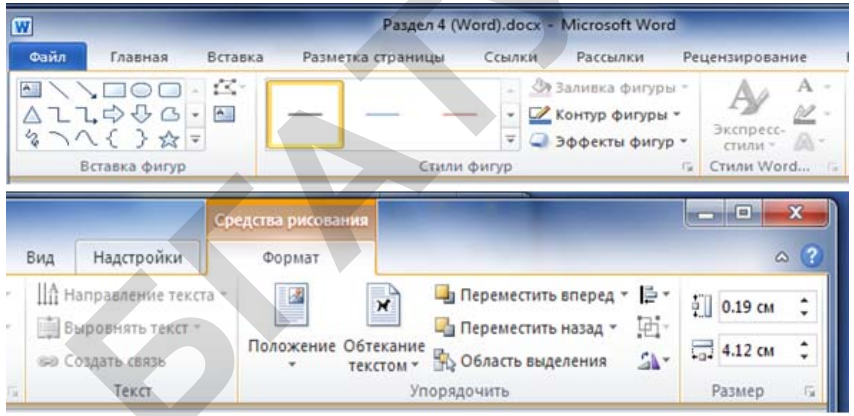


Рис. 4.32. Дополнительные элементы управления

Используя инструменты рисования, можно изобразить любую фигуру, например, пирамиду или конус (рис. 4.33).

Рис. 4.33. Примеры построения фигур

Построить пирамиду легко: нарисуйте прямую линию, сделайте пять копий этой линии и соедините их концы, перетаскивая линии за маркеры выделения, невидимые линии сделайте пунктирными. Чтобы фигура не рассыпалась при перемещении, ее надо сгруппировать: введите команду ***Выделить, Выбор объектов*** в группе ***Редактирование*** вкладки ***Главная***, установите указатель мыши выше левого верхнего края рисунка, нажмите клавишу мыши и обведите всю фигуру. Все объекты, полностью попавшие в контур выделения, выделяются. Данный прием работает в режим ограниченной функциональности. В режиме Word 2010 для выполнения данной процедуры выберите команду ***Область выделения*** на вкладке ***Средства рисования***, группа ***Упорядочить*** (рис. 4.32) – откроется список рисованных объектов. Объекты теперь можно выделять, выбирая их мышью из списка. Выбор группы объектов осуществля-

ется при нажатой клавише ***Ctrl***. После выделения нужных объектов вызовите контекстное меню и выполните команду ***Группировать***.

Построить конус будет несколько сложнее. В группе ***Основные фигуры*** (рис. 4.30) есть инструмент ***Дуга***. Дуга при выделении кроме восьми маркеров выделения имеет еще два маркера в виде ромба желтого цвета. Маркеры желтого цвета позволяют изменять форму дуги, а другие маркеры – размеры. Для построения основания конуса нам нужно построить две дуги эллипса, одну выпуклую вверх пунктирную, вторую – вогнутую вниз сплошную. Постройте первую дугу, она по умолчанию будет выпуклая вверх. Сделайте копию этой дуги и разверните ее на 180 градусов. Чтобы развернуть дугу, воспользуйтесь инструментом ***Повернуть*** в группе ***Упорядочить*** (рис. 4.32) или выделите объект, зацепите мышью за круглый маркер зеленого цвета – вокруг него появится черная стрелка – и выполните поворот объекта на нужный угол. Соединив две дуги, получим основание конуса. Стиль линии и ее толщину можно установить с помощью списка ***Контур фигуры*** в группе ***Стили фигур***. Теперь остается добавить две образующие, сгруппировать рисунок – и конус готов.

Команды ***Переместить вперед*** и ***Переместить назад*** позволяют управлять положением объектов друг относительно друга.

Команды ***Положение*** и ***Обтекание текстом*** позволяют установить, как объект будет взаимодействовать с окружающим его текстом. Чаще всего применяют обтекание вокруг рамки и обтекание сверху и снизу. После установки режима обтекания объект можно легко переместить в нужное положение с помощью мыши: установите указатель мыши на объект – указатель принимает вид двунаправленных стрелок, нажмите клавишу мыши и перемещайте рисунок в нужное положение.

Группа ***Упорядочить*** вкладки ***Средства рисования*** аналогична группе ***Упорядочить*** вкладки ***Разметка страницы***.

Сетка

Для удобства рисования можно установить на лист сетку командой ***Сетка*** из группы ***Показать*** вкладки ***Вид***. Границы сетки действуют как магнит, притягивая рисуемые объекты к узлам сетки. К сожалению, параметры сетки не регулируются. В офисе 2003 параметры сетки можно было настраивать.

Вставка объектов вкладки Ссылки

Вкладка Ссылки содержит шесть групп команд: Оглавление, Сноски, Ссылки и списки литературы, Названия, Предметный указатель, Таблица ссылок. Указанные группы команд позволяют подготовить к изданию брошюру, дипломный проект, диссертацию или учебное пособие в соответствии с требованиями, предъявляемыми к печатной продукции.

Оглавление

Оглавление является обязательным атрибутом любого более или менее сложного документа. Оглавление (Содержание) – это список заголовков разделов, подразделов с указанием номеров страниц. В редакторе Word вставка оглавления предельно упрощена. Для вставки оглавления выполните следующее:

- Напишите заголовки и подзаголовки к тексту документа и оформите их с помощью стилей заголовков вкладки Главная ленты. Заголовок первого уровня должен быть самым крупным. Заголовки следующих уровней имеют меньшую высоту шрифта. Редактор позволяет создавать до 9 уровней заголовков (на практике редко используют больше трех-четырех уровней);
- Укажите место вставки списка.
- Введите команду ***Оглавление*** на вкладке ***Ссылки***. Откроется список стандартных вариантов оформления оглавления, содержащего три уровня заголовков. Если эти варианты Вас не устраивают, тогда щелкните по кнопке Оглавление в конце списка. В этом случае открывается окно диалога (рис. 4.34). Выберите формат, заполнитель, число уровней оглавления и щелкните по кнопке ОК.

Таким образом, единственное, что необходимо сделать, чтобы Word смог автоматически сформировать оглавление документа, – оформить заголовки с помощью стилей заголовков.

При необходимости стиль заголовка можно изменить. Для изменения стилей заголовков можно воспользоваться командой ***Изменить стили*** вкладки ***Главная*** ленты. Данная команда позволяет выбрать стили заголовков из шаблонов или создать собственный стиль, используя наборы шрифтов, цветов и стилей абзацев.

Сноски

Сноска – это краткий комментарий к тексту, пояснение какого-либо термина, описание событий, связанных с какой-нибудь датой, и тому подобное. Различают обычную и концевую сноски. *Обычная сноска* помещается в конце текущей страницы, *концевая сноска* – в конце документа. Для вставки сноски установите курсор после текста, к которому будет относиться сноска, введите команду ***Вставить сноску*** из группы ***Сноска***. Для создания собственного стиля сноски вызовите окно диалога (рис. 4.35) кнопкой в правом

нижнем углу группы Сноски, выберите тип сноски, способ нумерации и щелкните по кнопке ОК. В месте размещения курсора появится номер сноски. Редактируется сноска, как обычный текст. Для удаления сноски выделите и удалите ее номер.

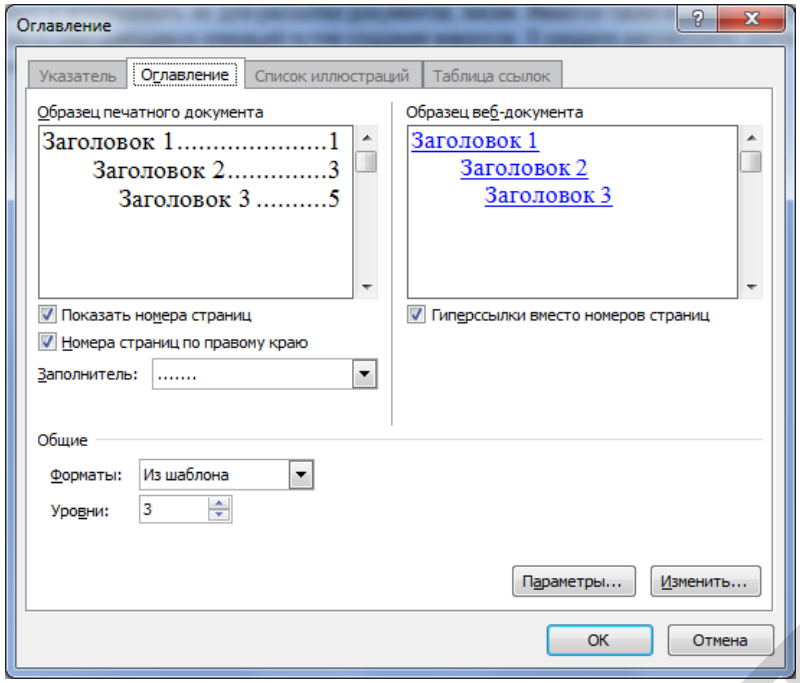


Рис. 4.34. Окно диалога для вставки Оглавления

Ссылки и списки литературы

Эта новинка Word 2010 облегчает создание списка литературных источников, использованных при разработке документа. В группе ***Ссылки и списки литературы*** размещены кнопка *Управление источниками* и списки *Стиль*, *Вставить ссылку* и *Список литературы*. Список ***Стиль*** позволяет выбрать стиль оформления списка литературы. Список ***Вставить ссылку*** добавляет в текст документа ссылку на источник фрагмента текста, вставленного в документ, в соответствии с выбранным стилем. Ссылка выбирается из списка. Если списка нет, то предлагается форма для ввода данных об источнике. Кнопка ***Управление источниками*** позволяет

создать список источников, использованных в данном документе. Кнопка Список литературы автоматически формирует список литературных источников, использованных при разработке документов, или цитируемых трудов.

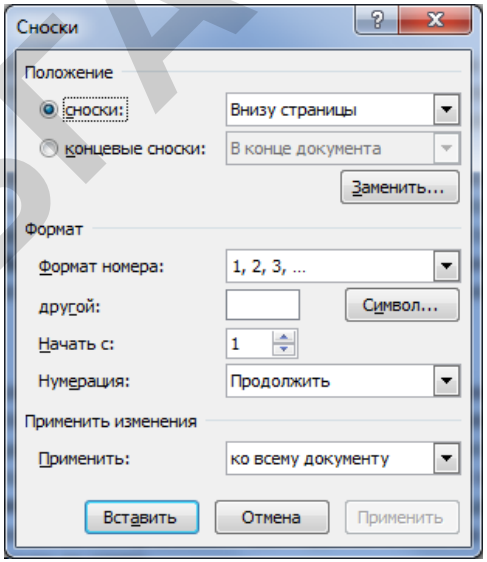


Рис. 4.35. Вставка сноски

Названия

Команда ***Вставить название*** из группы ***Названия*** позволяет автоматизировать нумерацию рисунков, таблиц, формул и облегчает составление списка иллюстраций документа. При вводе команды открывается окно диалога *Название* (рис. 4.36). В строке ввода после номера рисунка следует ввести название этого рисунка. Номера рисунков программа формирует автоматически. В группе *Параметры* расположены два списка: «подпись» и «положение». Список «подпись» содержит список постоянной части названия. Эта часть может изменяться пользователем. Для этого щелкните по кнопке Создать и в открывшемся окне диалога введите название, например, Рис. Это название попадает в список и может использоваться в дальнейшем в качестве подписи номера рисунка. Список «положение» позволяет указать место размещения надписи: над выделенным объектом или под выделенным объектом.

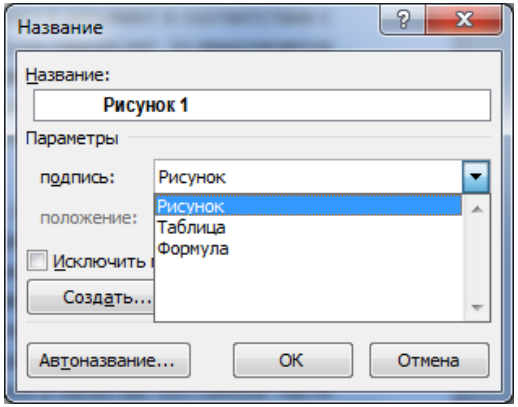


Рис. 4.36. Вставка названий

Кнопка ***Список иллюстраций*** позволяет автоматически создать список рисунков, таблиц или формул, если они были созданы с помощью команды ***Вставить название***.

Перекрестная ссылка

Перекрестная ссылка связывает текст в документе или разрешает пользователям переходить к нужному тексту или объекту. Имеется два вида ссылок: первый тип ссылки связывает два объекта *в одном документе*. Второй тип ссылок позволяет переходить от *одного документа к другому* – это *гиперссылки*.

Первый тип ссылки дает пользователю только информацию, например: «*Подробнее об этом можно узнать в разделе Сноски*». При изменении местоположения раздела пользователь все равно найдет его по названию.

Второй тип ссылки применяется в электронных документах для автоматического перехода к требуемому разделу. Если на поле ссылки навести указатель мыши, то он превращается в руку, а при щелчке мышью по этому полю пользователь перейдет к требуемому разделу. При этом открывается панель инструментов Web. Чтобы вернуться назад, надо щелкнуть мышью по кнопке Назад на панели инструментов Web.

При создании ссылки в документ вставляется поле с выбранным типом ссылки, выделенное серым цветом. Перекрестная ссылка может ссылаться на нумерованные абзацы, заголовки, закладки, сноски, рисунки, таблицы и другие *объекты*, созданные средствами Word. Например, нельзя сделать сноску на заголовок, если он не

оформлен с помощью стиля заголовков, нельзя сослаться на слово Рисунок, если он не создан с помощью команды Название меню Вставка.

Для создания перекрестной ссылки выполните следующее:

- В месте перекрестной ссылки напишите нужный текст, например: «Подробнее об этом можно узнать в разделе Сноски».
- Выделите слово Сноски (на этом месте может быть любой текст, так как он будет заменен на выбранный) и введите команду ***Вставка, Перекрестная ссылка*** – открывается окно диалога Перекрестная ссылка (рис. 4.37).

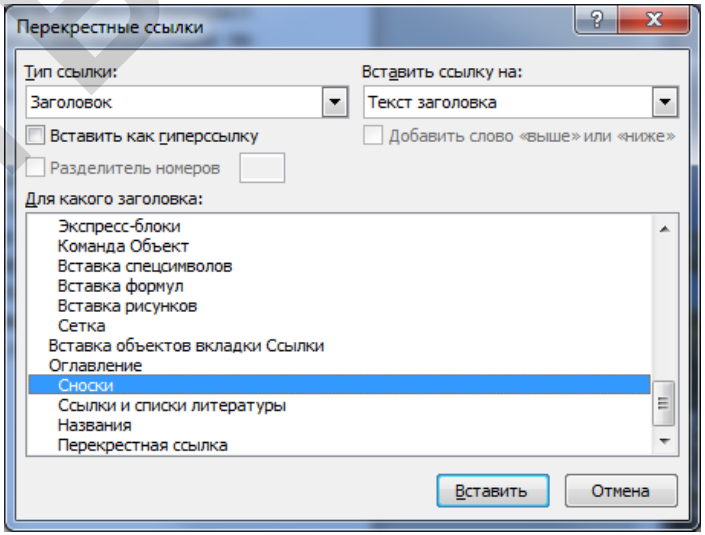


Рис. 4.37. Создание перекрестной ссылки

- Выберите в списке «*Тип ссылки*» требуемый тип, например, Заголовок.
- В списке «*Вставить ссылку на*:» выберите объект, на который будете ссылаться: Текст заголовка, Номер страницы и др.
- В окне «Для какого заголовка:» выделите заголовок Сноски.
- Нажмите кнопку ***Вставить***.

Если предполагается создавать гиперссылку, то установите флажок «Вставить как гиперссылку» в окне диалога. Для завершения работы по созданию перекрестной ссылки щелкните по кнопке Вставить. С помощью команды Перекрестная ссылка можно

создавать гиперссылки для перехода к нужному разделу в текущем документе.

Вставку Перекрестной ссылки и Гиперссылки можно выполнить также с помощью команд из группы ***Ссылки*** вкладки ***Вставка***. Команда ***Гиперссылка*** позволяет делать ссылку на любой документ, в том числе и ссылку на ресурсы Интернета (рис. 4.38). Для создания гиперссылки с помощью данного окна диалога необходимо:

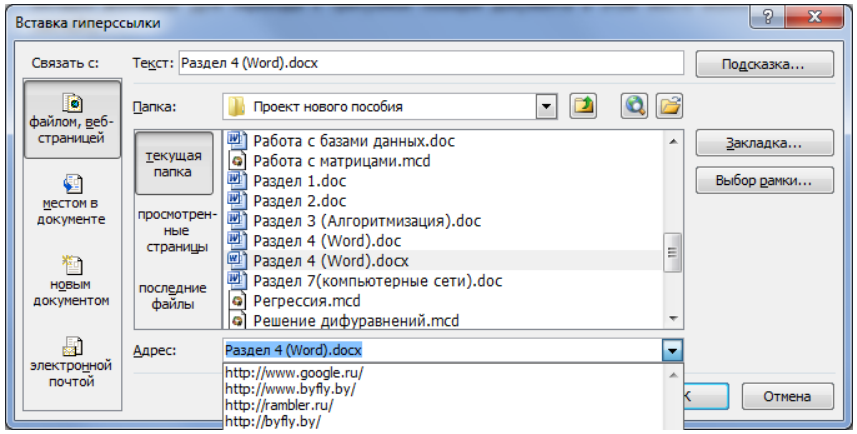


Рис. 4.38. Вставка гиперссылки

- Написать и выделить текст гиперссылки (ключевое слово).
- Вызвать окно диалога.
- Выбрать файл на своем компьютере или в Интернете.
- Выбрать закладку в выбранном файле. В качестве закладки программа выбирает стандартные элементы текстового процессора: Заголовки, Текст, Текстовые поля, Поля со списком, Флажки.
- Завершить работу, щелкнув по кнопке ОК.

Для перехода к требуемой позиции документа можно поставить в нужном месте *закладку* командой ***Закладка*** из группы ***Ссылка*** вкладки ***Вставка***.

Для удаления гиперссылки вызовите контекстное меню поля гиперссылки и введите команду Удалить гиперссылку.

Закладка

Закладка предназначена для присвоения имени определенной позиции в документе. На закладки можно ссылаться при создании

гиперссылок и ряда других полей Word. Для вставки закладки выделите объект (текст, формулу, заголовок, таблицу и др.) и введите команду ***Вставка, Закладка***. Укажите имя закладки. Чтобы просмотреть закладки, введите команду ***Файл, Параметры, Дополнительно*** и в группе ***Показывать содержимое документа*** установить/снять флажок ***Показывать закладки***. Закладки выделяются квадратными скобками. Для удаления закладки введите команду ***Вставка, Закладка***, выделите нужное имя закладки и щелкните по кнопке ***Удалить*** в окне диалога.

Предметный указатель

В больших документах для удобства поиска требуемой информации иногда вставляют указатели – алфавитные списки ключевых слов. Для вставки ключевого слова или фразы в указатели выделите этот текст и щелкните по кнопке ***Пометить элемент*** в группе ***Предметный указатель*** или нажмите комбинацию клавиш [***Alt + Shift + X***] – открывается окно диалога ***Определение элемента указателя*** (рис. 4.39). В строке ввода Элемент указателя основной будет записано выделенное слово. К основному элементу указателя можно добавить еще два дополнительных элемента. Введите в строке ввода «***Дополнительный***» первый дополнительный элемент, а затем через двоеточие – второй дополнительный элемент. Чтобы запомнить выделенные элементы, щелкните по кнопке ***Пометить*** или ***Пометить все***. Во втором случае будут помечены все вхождения ключевого слова в документ.

При необходимости вставить перекрестную ссылку активизируйте соответствующий переключатель и добавьте текст после «См…» в строке ввода. Для включения в указатель диапазона страниц выберите сначала этот диапазон и отметьте его закладкой. Затем установите переключатель Диапазон страниц и выберите в списке требуемую закладку.

Для завершения операции по формированию Указателя перейдите на последнюю страницу документа. Вставьте разрыв страниц, введите заголовок для предметного указателя и введите команду ***Предметный указатель*** на вкладке ***Ссылки***. Откроется одноименное окно диалога (рис. 4.34). Выделите вкладку ***Указатель***, выберите стиль оформления Предметного указателя и щелкните по кнопке ОК для завершения работы.

Контрольные вопросы

1. Изобразите лист формата А4 и покажите основные элементы оформления документа.

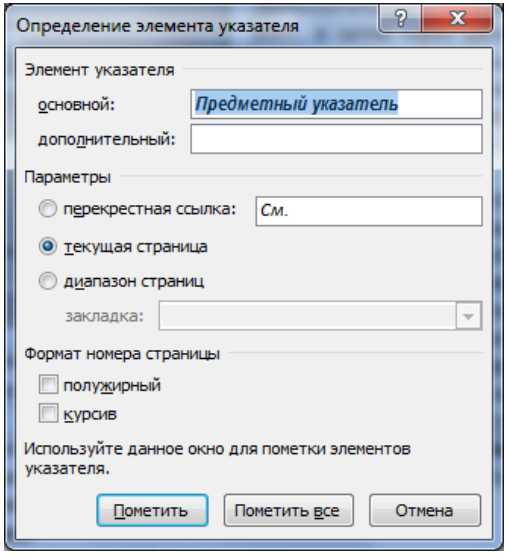


Рис. 4.39. Определение элементов указателя

2. Как вставить в документ картинку, дату, сноску, номера страниц?

3 Как вставить в документ формулу? Как используется мастер формул?

4. Какие виды колонтитулов можно установить в документе? Как это выполнить?

5. Расскажите алгоритм вставки оглавления.

6. Расскажите назначение основных инструментов в группе Иллюстрации вкладки Вставка.

7. Как нарисовать схему алгоритма в документе Word?

8. Как вставить в документ перекрестную ссылку, гиперссылку?

4.4. ТАБЛИЦЫ

Таблицы удобно применять для представления упорядоченного набора данных, списков, управления размещением текста, оформления логотипов и т. п. В таблицах можно проводить также несложные расчеты. Если требуется выполнить сложные расчеты, то следует вставить таблицу Excel.

Текстовый процессор Word 2010 имеет средства для работы с таблицами и обработки данных. Все команды для работы с таблицами сосредоточены в меню команды ***Таблица*** вкладки ***Вставка*** и на вкладке ***Работа с таблицами*** ленты. Вкладка ***Работа с таблицами*** открывается при создании или выделении таблицы. Она имеет две вкладки: ***Конструктор*** и ***Макет***.

Создание таблиц

Создание таблицы осуществляется командой ***Таблица*** на вкладке ***Вставка*** (рис. 4.40). После ввода команды открывается меню команды Таблица. Оно позволяет создать новую таблицу несколькими способами:

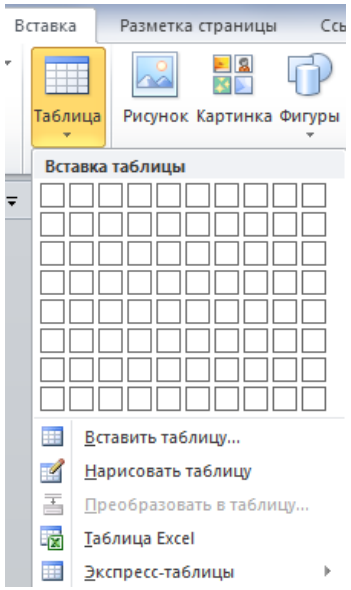


Рис. 4.40. Меню команды Таблица вкладки Вставка

- Выделить мышкой нужное число строк и столбцов в макете таблицы.
- Вставить таблицу командой ***Вставить таблицу***. После ввода команды откроется окно диалога, в котором необходимо указать требуемое число строк и столбцов.
- Нарисовать таблицу.
- Вставить таблицу Excel.

• Вставить таблицу из набора шаблонов таблиц – Экспресс-таблицу.

Таблица состоит из строк и столбцов. По умолчанию все столбцы нумеруются латинскими буквами от A до Z, а строки цифрами. Поэтому при написании формул к ячейке следует обращаться по ее адресу, например, A1.

Каждая ячейка таблицы представляет собой самостоятельный документ, в который можно помещать числа, текст, рисунки, даты, таблицы. Редактируется текст в ячейке по тем же правилам, что и в основном документе. Ширина столбцов может изменяться путем перетаскивания границ ячеек с помощью мыши непосредственно на таблице или на горизонтальной линейке. Высота строки автоматически увеличивается при переносе текста, но может также изменяться путем перетаскивания линий разметки строк мышью.

Таблица становится активной, если щелкнуть мышью по любой ячейке. Для выделения строки подведите указатель мыши к левой ячейке снаружи таблицы так, чтобы он превратился в стрелку ⇗ и щелкните мышью. Другой способ выделения строки – щелкните мышью по левой ячейке строки, нажмите клавишу мыши и протащите указатель мыши по строке. Для выделения столбца подведите указатель мыши к верхней строке снаружи так, чтобы он превратился в черную стрелку ⬇ и щелкните мышью. Выделить всю таблицу можно протаскиванием мыши или щелчком мыши по специальному маркеру, который появляется в верхнем левом углу таблицы при ее активизации (рис. 4.41). Подведите указатель мыши к границе строки или столбца внутри таблицы – появляется маркер выделения таблицы, подведите к нему указатель и щелкните мышью. Зацепив таблицу мышью за этот маркер, можно перемещать ее по листу.

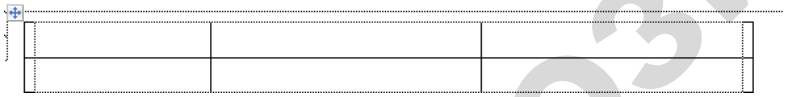


Рис. 4.41. Таблица с маркерами выделения

Команды вкладки Конструктор

Все команды управления относятся только к выделенной таблице. Команды вкладки *Конструктор* обеспечивают оформление таблицы различными стилями, настройку стиля, толщины и цвета линий сетки

(рис. 4.42). Команды ***Строка заголовка***, ***Строка итогов***, ***Первый столбец***, ***Последний столбец*** позволяют установить особый вид форматирования этих элементов таблицы. Команды ***Чередующиеся строки*** и ***Чередующиеся столбцы*** позволяют применить разные стили форматирования к четным и нечетным строкам/столбцам.

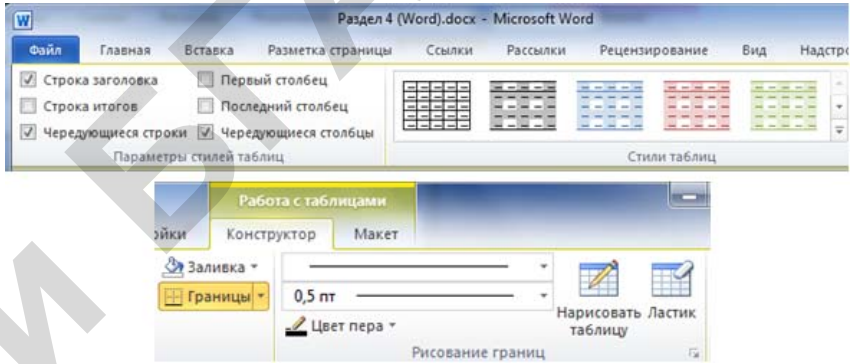


Рис. 4.42. Вкладка Конструктор

Команды вкладки Макет

Вид вкладки Макет приведен на рис. 4.43.

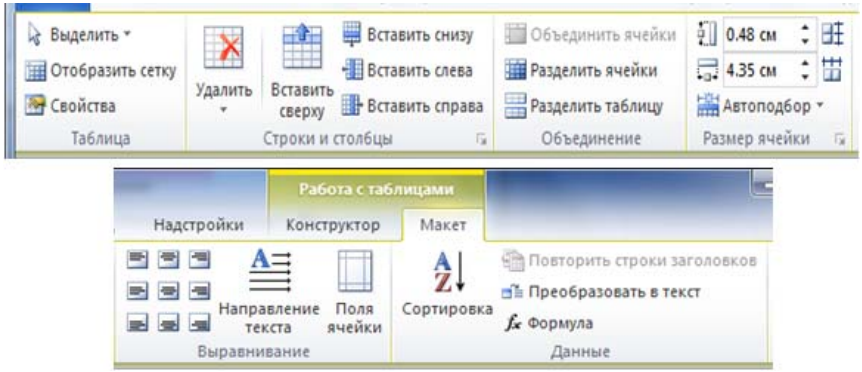


Рис. 4.43. Вкладка Макет

Команды группы ***Таблица*** позволяют управлять выделением строк и столбцов, отображением сетки и открытием окна диалога *Свойства* сетки.

Команда ***Свойства*** позволяет управлять размещением таблицы на листе, настраивать параметры таблицы, строк, столбцов и ячеек. Эту команду можно вызвать также через контекстное меню таблицы.

Команды группы ***Строки и столбцы*** позволяют добавлять столбец слева, столбец справа, строку сверху, строку снизу. Вставка строки (столбца) осуществляется относительно выделенной строки (столбца). Дополнительные команды управления вставкой строк и столбцов содержатся во всплывающем меню, которое вызывается щелчком мыши по кнопке в правом нижнем углу группы.

Команда ***Удалить*** позволяет удалить таблицу, строку, столбец или ячейку.

Группа ***Объединение*** позволяет управлять объединением или разбиением выделенных ячеек, что дает возможность создавать таблицы произвольной структуры (рис. 4.44).

<table>
<tr><td>1</td><td colspan="3">2</td><td>3</td></tr>
<tr><td></td><td>5</td><td colspan="2">6</td><td></td></tr>
<tr><td></td><td></td><td>7</td><td>8</td><td></td></tr>
<tr><td rowspan="2">4</td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
</table>

Рис. 4.44. Пример объединения и разбиения ячеек

Таблица на рисунке 4.44 имеет три столбца (1, 2, 3). Область 4 получена путем объединения двух ячеек первого столбца. Столбцы 5 и 6 получены путем деления четырех ячеек столбца 2 на два столбца, а столбцы 7 и 8 получены путем деления трех ячеек столбца 6 на два столбца.

Команда ***Разделить таблицу*** разбивает таблицу на две части по горизонтали по текущей строке.

Команды группы ***Размер ячеек*** позволяют устанавливать требуемые размеры ячеек. А команда ***Автоподбор*** позволяет автоматически настраивать ширину столбцов и высоту строк по содержимому или ширине окна.

Команды группы ***Выравнивание*** управляют выравниванием текста в ячейках по ширине и высоте, изменением ориентации текста в ячейках, настройкой полей в ячейках и интервалов между ячейками.

Группа ***Данные*** позволяет выполнять сортировку данных в таблице, преобразовать таблицу в текст, вставлять формулы и управлять переносом первой строки на другую страницу.

Команда ***Повторить строки заголовков*** обеспечивает автоматическое формирование верхней шапки таблицы при переходе таблицы на другую страницу.

Команда ***Преобразовать в текст*** позволяет преобразовывать таблицу в обычный текст. При преобразовании таблицы в текст программа просит указать разделитель, которым будут отделяться данные: символ табуляции, точка с запятой, запятая или иной символ.

Команда ***Сортировка*** позволяет отсортировать таблицу по возрастанию или убыванию значения. В качестве значения может быть текст, дата, число. Выделите таблицу, введите команду ***Сортировка*** – откроется одноименное окно диалога (рис. 4.45). Окно диалога позволяет задать одновременно три условия сортировки. Укажите, какие столбцы будут использоваться первым, вторым и третьим, тип данных в каждом столбце и направление сортировки (по возрастанию или убыванию значения). При задании последовательности сортировки необходимо исходить из следующего принципа: первый параметр должен быть наиболее общим, то есть он должен разбивать всю совокупность данных на наименьшее число групп.

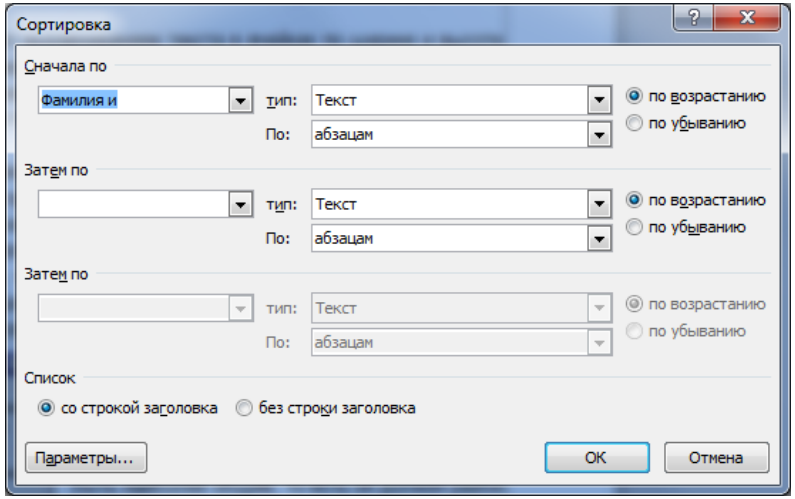


Рис. 4.45. Окно диалога Сортировка

Пусть имеется список рабочих (рис. 4.46). Определить, в какой последовательности следует выбирать поля таблицы для сортировки, если:

Фамилия и инициалы	Номер цеха	Табельный номер	Трудовой стаж
Петров И. С.	1	А342	5
...			

Рис. 4.46. Пример таблицы для определения порядка выбора полей для сортировки

а) требуется отсортировать таблицу по трем полям: Фамилия и инициалы, Номер цеха, Трудовой стаж;

б) требуется составить список рабочих по цехам.

Для пункта а) первым параметром для сортировки следует выбрать Номер цеха, вторым параметром можно выбрать трудовой стаж, а третьим фамилию. В этом случае программа распределит рабочих сначала по цехам, потом в переделах цехов распределит рабочих по трудовому стажу и в последнюю очередь в пределах этих групп распределит их в алфавитном порядке.

Для пункта б) в первую очередь необходимо отсортировать таблицу по номеру цеха, а затем по фамилиям и инициалам.

Формула – таблица позволяет проводить несложные вычисления. Выделите ячейку, в которую будет вводиться формула, и введите команду ***Формула***. Откроется окно диалога (рис. 4.47). Строка Формула служит для ввода формулы. Формула должна начинаться со знака «равно» и может содержать ссылки на ячейки таблицы и функции. Например:
=A1*B1+C1*D1, =PRODUCT(A1:C1),
=AVERAGE(C1:C10) и т. д.

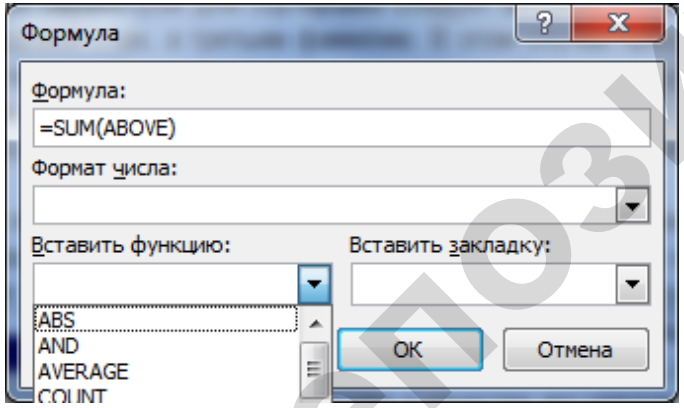


Рис. 4.47. Ввод формул

В функциях в качестве аргументов могут использоваться адреса ячеек, списки адресов ячеек, указанных через символ «;», или диапазоны. Диапазон – непрерывная область ячеек, обозначенная номером левой верхней и правой нижней ячеек группы, разделенных двоеточием. Например:

SUM(A1;C1;E1) – вычисляется сумма значений ячеек A1, C1 и E1;

SUM(A1:D8) – вычисляется сумма значений ячеек, расположенных в блоке (диапазоне) ячеек A1:D8.

Список *Вставить функцию* содержит ряд встроенных арифметических и логических функций. Например:

ABS() – абсолютная величина числа;

AVERAGE() – среднее значение списка или диапазона чисел;

COUNT() – количество значений списка или диапазона чисел;

MAX() – максимальное значение в списке или диапазоне чисел;

MIN() – минимальное значение в списке или диапазоне чисел;

PRODUCT() – произведение значений в списке или диапазоне.

В качестве аргумента у встроенных функций могут использоваться также служебные слова LEFT – список всех ячеек слева, ABOVE – список всех значений выше, например, SUM(ABOVE).

Пример 4.1. Выполнить вычисления, используя встроенные функции.

	Среднее	Минимум	Максимум	Произведение	Сумма	Количество
	3	41	25	2	5	123
	13	7	213	3	6	15
	34	12	29	4	2	46
	25	76	15	5	9	73
Результат	=AVERAGE (B2:B5)	=MIN (C2:C5)	=MAX (D2:D5)	=PRODUCT (E2:E5)	=SUM (ABOVE)	=COUNT (G2:G5)

Пример 4.2. Выполнить вычисления, используя встроенные функции.

Округление		Модуль		Целое		Условное выражение
исходное	формула	исходное	формула	исходное	формула	
12,344	=ROUND(A3;2)	–75	=MOD(C3;2)	1,234	=INT(E3)	–0,165

1) округлить числа до двух знаков;
2) найти модуль от деления числа на два;
3) найти целое от числа, большего единицы;
4) вычислить условное выражение по данным таблицы:

если A3 > C3, тогда A3/C3, иначе A3 + C3; =IF(A3 > C3; A3/C3; A3 + C3)

Так как условие выполняется, то возвращается результат согласно первому выражению – A3/C3.

Контрольные вопросы

1. Для чего предназначены таблицы в редакторе Word?
2. Как создать таблицу?
3. Как добавить строку, столбец, ячейку?
4. Как настроить свойства таблицы, строки, столбца?
5. Как осуществляются вычисления в таблице?

4.5. ШАБЛОНЫ

Основой каждого документа является шаблон. Шаблон – это набор параметров форматирования текста, абзацев, списков, элементов автотекста, макросов.

Редактор Word имеет стандартный шаблон Normal, который загружается при открытии нового документа. Кроме того, он имеет большое количество других шаблонов для создания записок, факсов, писем, публикаций и т. п. Чтобы найти требуемый шаблон, введите команду ***Создать***, открывается окно диалога (рис. 4.48). Выберите подходящий по содержанию шаблон и щелкните по кнопке ***Создать***. Шаблон может содержать поля ввода с текстом, который нужно заменить на свой, например, название организации, домашний адрес и т. п.

В документ можно вставлять поля различного вида с помощью команды Вставка, Экспресс-блоки, Поля.

Word 2010 позволяет создавать и собственные шаблоны – пользовательские.

Для создания пользовательского шаблона введите команду ***Файл, Создать*** и выберите в окне диалога папку ***Мои шаблоны*** (рис. 4.48) – откроется окно диалога Создать (рис. 4.49). Активизируйте значок ***Новый документ***, установите в группе Создать переключатель ***Шаблон*** и щелкните по кнопке ОК.

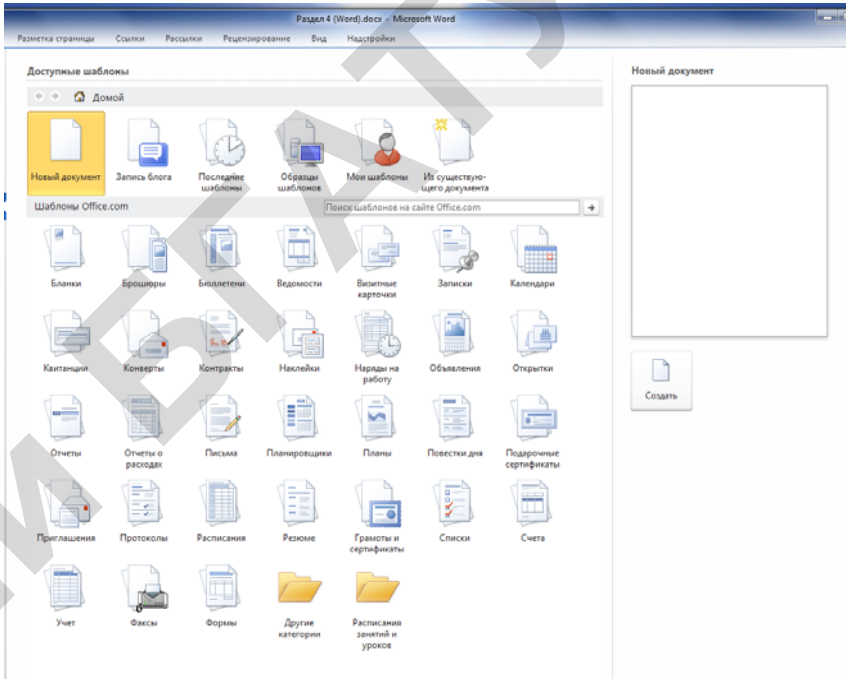


Рис. 4.48. Окно диалога Создание документа

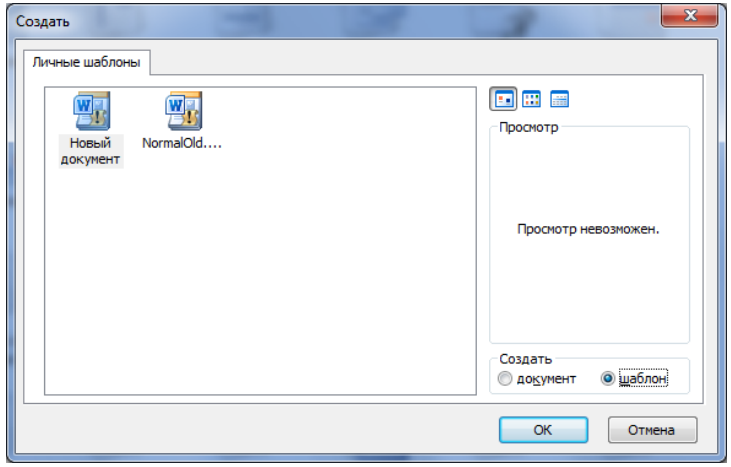


Рис. 4.49. Создание шаблона пользователя

В шаблон документа введите постоянный текст и вставьте необходимые поля с помощью команды ***Вставка, Экспресс-блоки*** (Автотекст, Поле и Организатор стандартных блоков). В шаблон можно вставлять такие объекты, как текст (A), поле ввода (ab|), флажки, списки. Элементы управления для вставки в шаблоны размещены в группе ***Элементы управления*** вкладки ***Разработчик*** ленты. В режиме ограниченной функциональности эти кнопки недоступны. Для совместимости с предыдущими версиями текстового процессора имеются формы предыдущих версий, список которых вызывается кнопкой ***Список предыдущих версий*** из группы ***Элементы управления*** (рис. 4.50).

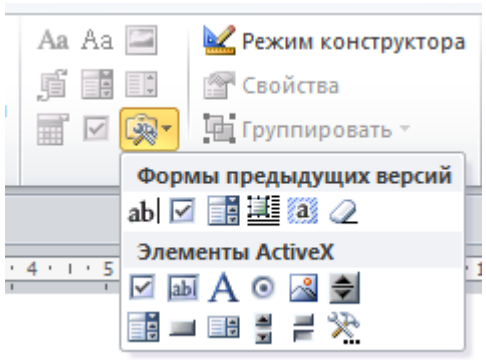


Рис. 4.50. Поля для создания шаблонов

При сохранении документа шаблон сохраняется с расширением .dotx (.dot) в папке Мои шаблоны.

Создание шаблона пользователя

В группе Элементы управления (рис. 4.50) имеются следующие элементы для создания полей (слева направо и сверху вниз):

форматированный текст (Aa) – ввод форматированного текста;

обычный текст (Aa) – ввод обычного текста;

рисунок – вставка элемента управления содержимым «рисунок»;

коллекция стандартных блоков – служит для вставки стандартных блоков;

поле со списком – содержит текстовое поле с раскрывающимся списком, позволяет вносить данные во время работы;

раскрывающийся список – содержит список данных;

календарь – служит для вставки даты;

флажок может иметь два состояния: установлен или снят.

Установка элементов управления осуществляется в режиме конструктора (режим установлен по умолчанию). Настройка параметров элементов управления также осуществляется в режиме конструктора, вызов свойств объектов осуществляется кнопкой Свойства в группе Элементы управления или через контекстное меню.

Окно свойств элемента управления содержимым «Форматированный текст» приведено на рисунке 4.51.

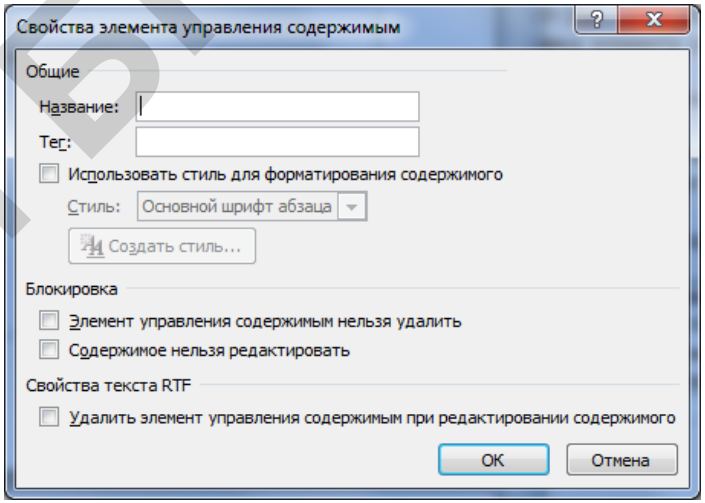


Рис. 4.51. Элемент управления содержимым «Форматированный текст»

Элемент управления «Обычный текст», в отличие от элемента управления «Форматированный текст», позволяет размещать текст в несколько абзацев.

На рисунке 4.52 приведено окно свойств элемента управления содержимым «Поле со списком». В окне свойств раскрывающегося списка по умолчанию записана информация «Выберите элемент», которая выводится на экран в качестве подсказки во время работы. Окно позволяет дополнять, сортировать, изменять или удалять данные.

Строка ввода Название позволяет ввести название объекта, например, «*Ввод текста*», а строка Тег – выводит «скобки» вокруг подсказки «Выберите элемент». В примере это слова «*мой текст*». Название появляется при выделении объекта, а Тег появляется при выходе из режима Конструктора.

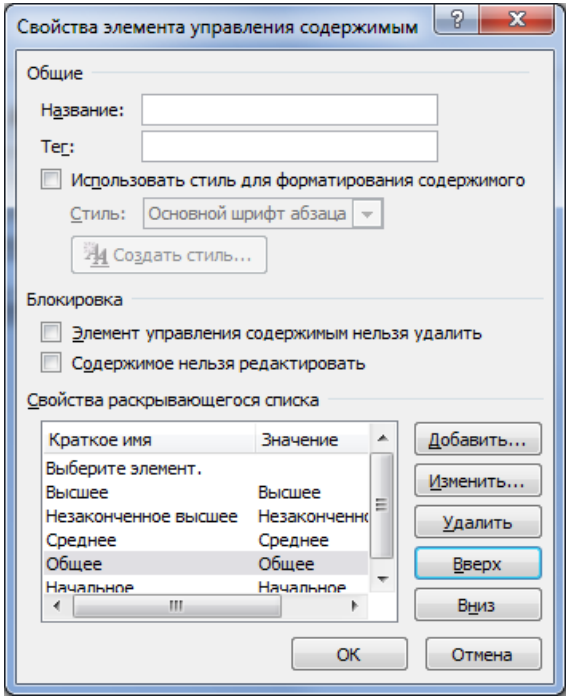


Рис. 4.52. Элемент управления содержимым «Поле со списком»

Вид поля ввода во время работы приведен на рис. 4.53.

Рис. 4.53. Поле ввода во время работы

На рисунке 4.54 приведен пример создания шаблона.

Порядок создания шаблона:

а) введите команду ***Файл, Создать, Мои шаблоны, Новый документ***, в группе *Создать* установите переключатель «шаблон»;

б) создайте таблицу и впишите постоянный текст в первую колонку;

в) вставьте во вторую колонку элементы управления:

Анкета	
Фамилия, инициалы	Место для ввода текста
Дата рождения	14.05.2011
Социальное положение	Служащий
Образование	Среднее специальное
Член профсоюза	☐
Домашний адрес	Место для ввода текста

Рис. 4.54. Пример создания шаблона

текстовые поля в строки Фамилия, инициалы и Домашний адрес;

элемент управления *Раскрывающийся список* в строке Социальное положение;

элемент управления *Поле со списком* в строке Образование;

элемент управления *Флажок* в строке Член профсоюза;

г) установите защиту на все, кроме полей ввода: введите команду ***Ограничить редактирование*** в группе ***Защита*** вкладки ***Разработчик***. Установите флажок в пункте 2 «***Разрешить только указанный способ редактирования документа***», выберите в списке «***Ввод данных в поля форм***» и далее ***Включить защиту***;

д) сохраните документ как шаблон. Папка Шаблоны открывается по умолчанию.

Использование шаблона

Введите команду ***Файл, Создать, Мои шаблоны***, выберите в окне диалога ***Личные шаблоны*** файл ***Анкета*** и щелкните по кнопке ***Ок***.

Использование полей ввода предыдущих версий

Формы предыдущих версий содержат следующие элементы управления:

текстовое поле служит для вставки текста, чисел, дат и времени;

флажок – предназначен для вставки элементов управления Флажок. Флажок может иметь два состояния: установлен или снят;

раскрывающийся список – служит для ввода данных из списка;

затенение – предназначено для выделения полей формы серым цветом;

защита – обеспечивает защиту постоянной информации от изменения. После установки защиты курсор будет перемещаться только по полям формы.

Для установки поля формы в документ установите курсор в точку ввода и щелкните мышью по требуемой кнопке на панели инструментов Формы. Для удаления поля формы выделите его мышью и нажмите клавишу Del.

Параметры Текстового поля

После установки текстового поля в документ нужно, при необходимости, настроить его параметры. Окно настройки параметров вызывается щелчком мыши по соответствующей кнопке на панели инструментов, двойным щелчком мыши по полю формы или через контекстное меню. Окно настройки параметров текстового поля приведено на рисунке 4.55. Оно имеет несколько списков и строку ввода. Список «Тип» позволяет установить тип поля: обычный текст, число, дата, время или вычисления. Счетчик «Максимальная длина» позволяет установить ограничения на длину поля формы. Список «Формат» позволяет устанавливать формат текста, даты, времени или числа. Строка ввода «Текст по умолчанию» позволяет вводить, в зависимости от установленного типа, текст, число, дату или формулу. Для каждого поля формы автоматически формируется закладка.

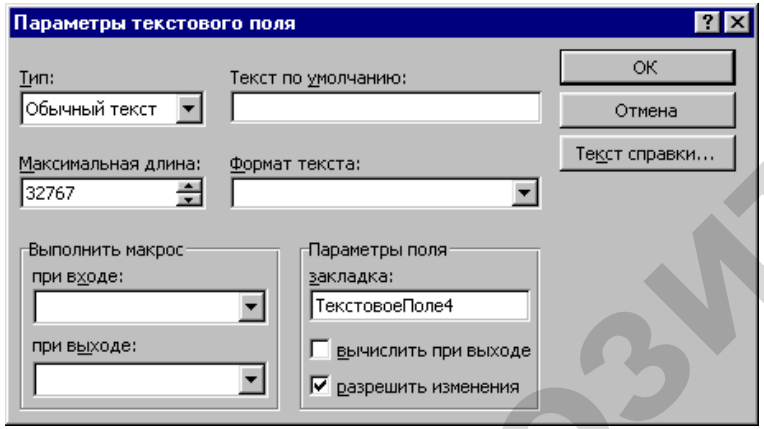


Рис. 4.55. Окно настройки параметров текстового поля

Окно настройки параметров поля со списком позволяет вводить данные в список (рис. 4.56), а также изменять порядок следования элементов списка с помощью кнопок Порядок.

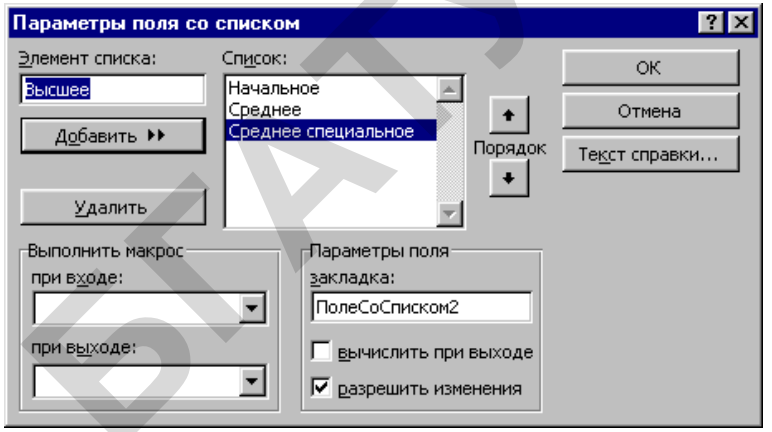


Рис. 4.56. Окно настройки параметров поля со списком

Пример создания шаблона

Рассмотрим порядок создания собственного шаблона на примере Анкеты участника соревнований (условный пример).

Предположим, что нам нужно вести учет спортсменов, прибывающих на соревнование по легкой атлетике. При этом требуется отразить следующие сведения: фамилия, имя, отчество, домашний адрес, номер телефона, дата рождения, возрастная группа, вид соревнования, семейное положение, дистанция, личные достижения, является ли обладателем рекордов: мирового, европейского, республиканского.

Разработаем форму анкеты (рис. 4.57).

Порядок работы:

1. Введите команду Файл, Создать, Шаблоны на моем компьютере. В окне диалога Шаблоны щелкните по кнопке Новый документ, установите переключатель Шаблон и щелкните по кнопке ОК.

2. Введите неизменяемую часть информации (Заголовки, таблицу, текст, рисунки и др.).

3. Вставьте поля формы и настройте их свойства:

Место проведения соревнований – по умолчанию Стадион «Динамо»;

Дата – поле типа Дата, длина 15, с шаблоном и примером заполнения;

Время – текстовое поле, длина 10, заполнено по умолчанию;

Анкета участника соревнований		
Место проведения соревнований:		
Дата: Время:		
Возрастная группа →		
Вид соревнования	Дистанция	Личное достижение
Фамилия, имя, отчество		
Дата рождения		
Адрес		
Город		Индекс →
Страна		Телефон →
Дополнительные сведения: Являетесь ли обладателем рекордов: мирового ☐ европейского ☐ республиканского ☐		

Рис. 4.57. Пример шаблона до настройки свойств полей

Возрастная группа – поле со списком;

Вид соревнований – поле со списком;

Дистанция – поле со списком;

Личное достижение – простой текст, длина 10;

Фамилия, имя, отчество – простой текст, длина 40;

Место проведения соревнований – текстовое поле, длина поля 20, текст;

Дата рождения – поле типа Дата с шаблоном ДД/ММ/ГГ, длина 8, пример заполнения;

Адрес, Город – простой текст, длина поля – 50;

Страна – простой текст, длина 30, текст по умолчанию;

Почтовый индекс – простой текст с примером заполнения;

Телефон – число, длина 8, пример заполнения;

Сведения об обладании рекордом – флажки, не установлены.

4. Установите защиту.

5. Сохраните шаблон командой Файл, Сохранить, введите имя файла.

Файл сохраняется по умолчанию с расширением .dot в папке Шаблоны.

Пример шаблона после настройки полей и установки защиты приведен на рисунке 4.58.

Анкета участника соревнований		
Место проведения соревнований: Стадион «Динамо»		
Дата: 08/06/04 Время: 10 часов		
Возрастная группа → общая (до 21)		
Вид соревнования Бег	Дистанция 10 км	Личное достижение 27,35 мин
Фамилия, имя, отчество	Иванов Петр Михайлович	
Дата рождения	12/05/1984	
Адрес	ул. Запрудная, д. 40, кв. 1	
Город	Бобруйск	Индекс → 224000
Страна	Беларусь	Телефон → 23-15-67
Дополнительные сведения: Являетесь ли обладателем рекордов: мирового ☐ европейского ☐ республиканского ☒		

Рис. 4.58. Пример шаблона после настройки полей и установки защиты

Для редактирования шаблона введите команду ***Файл, Создать, Шаблоны на моем компьютере***, выделите в окне диалога ***Шаблоны*** шаблон, установите переключатель ***Шаблон*** и щелкните по кнопке ОК. Вызовите панель инструментов Формы и снимите защиту. Теперь можно приступать к редактированию шаблона.

Контрольные вопросы

1. Что такое шаблон, для какой цели он создается?
2. Расскажите алгоритм создания шаблона?
3. Какие поля можно использовать в шаблоне и как они вставляются?
4. Как настроить свойства (параметры) полей формы?

4.6. СЛИЯНИЕ ДОКУМЕНТОВ

Подготовка документов к рассылке

Часто приходится рассылать один и тот же документ по разным адресам, например, приглашение на презентацию книги, юбилей учебного заведения, симпозиум и др. Программа Word позволяет автоматизировать эту работу. Для подготовки к рассылке документа, оформления конвертов и наклеек имеется *Мастер слияния документов*.

Для выполнения операции слияния необходимо подготовить шаблон документа с полями слияния и источник данных для заполнения этих полей.

Шаблон документа может быть подготовлен заранее и сохранен на диске как документ Word или создан в процессе выполнения работы.

Источник данных представляет собой обычную таблицу. Таблица может быть подготовлена средствами Word или другими приложениями, например, с помощью электронной таблицы, системы управления базой данных и др. Пример шапки таблицы приведен на рисунке 4.59. Каждая строка таблицы является записью, относящейся к одному объекту, а каждая ячейка – элементом данных.

Обращение	Имя	Фамилия	Должность	Организация	Адрес1	Адрес2	Город	Область	Индекс

Рис. 4.59. Пример источника данных для слияния

Выполнение операций по подготовке документов к рассылке осуществляется с помощью вкладки ***Рассылка*** (рис. 4.60). При этом выполняется ряд последовательных шагов:

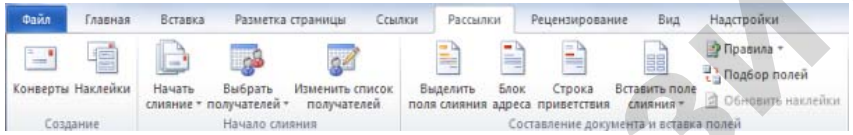


Рис. 4.60. Фрагмент вкладки Рассылка

Настройка основного документа. Основной документ может содержать текст, графику, рисунки и другие сведения, которые необходимо сообщить адресатам. В тексте документа необходимо предусмотреть обращение к адресату, при необходимости предусмотреть место для адреса и другие вставки, имеющие индивидуальное значение, например, номера счетов, суммы кредитов и др.

Подключение документа к источнику данных, если он существует. Если источника данных нет, то его необходимо создать. Источником данных может быть текстовый файл, содержащий сведения об адресате. Файл может быть подготовлен в Word, в одном из приложений Windows или другом источнике данных, совместимом с текстовым процессором.

Уточнение списка получателей или элементов. Word после загрузки файла данных создает копии документа для каждой записи. Однако рассылка может осуществляться не всем адресатам списка, а только некоторым, поэтому ненужные записи надо исключить из списка.

Вставка в документ полей слияния – текстовых заполнителей. При слиянии эти поля заполняются данными из списка.

Предварительный просмотр и завершение слияния. Прежде чем печатать документ, каждую запись можно просмотреть.

Процедура создания документа рассылки в Word 2010 выполняется следующим образом:

1. Откройте новый документ и напишите основной текст документа, зарезервировав место для вставки полей, или загрузите в шаблон документ, если шаблон письма уже имеется, например, как показано на рисунке 4.61. Созданный документ сохраните на диске.

«Обращение» «Имя» «Отчество»

Сердечно поздравляем Вас с 66 годовщиной ПОБЕДЫ Советского народа в Великой Отечественной войне над фашистской Германией!

Желаем Вам крепкого здоровья и плодотворной работы по воспитанию подрастающего поколения в духе беззаветной любви и преданности своему Отечеству.

Совет Ветеранов Брестской области

08.05.2011

Рис. 4.61. Шаблон основного документа с полями слияния

2. Откройте вкладку ***Рассылки***, в группе ***Начало слияния*** откройте список ***Начать слияние*** и выберите в всплывающем списке команду ***Пошаговый мастер слияния*** (рис. 4.62).

Откроется первое окно диалога Слияние (рис. 4.63), которое подведет нас к желанной цели за шесть шагов. Выберем тип документа *Письмо* из предложенного меню и перейдем ко второму этапу: выбору документа. Мастер предлагает три варианта: *Текущий*

документ, *Шаблон* или *Существующий документ*. Так как документ нами создан, выберем *Текущий документ* и перейдем к третьему окну диалога Выбор получателей. В качестве источника данных программа предлагает использование существующего списка, контакты в электронной почте, а при отсутствии таковых предлагает создать новый список. Если источник данных существует, активизируйте соответствующий переключатель и щелкните по кнопке ***Обзор***. Открывается окно программы Проводник, с помощью которого необходимо найти и загрузить существующий файл данных. Для сокращения области поиска можно указать требуемый тип файлов: документы Word, книги Excel, базы данных Access и др. При этом необходимо позаботиться о совместимости текстового процессора с источником данных. Для этого введите команду ***Файл, Параметры, Дополнительно*** и в группе ***Общие*** установите флажок ***Подтверждать преобразование формата файла при открытии***.

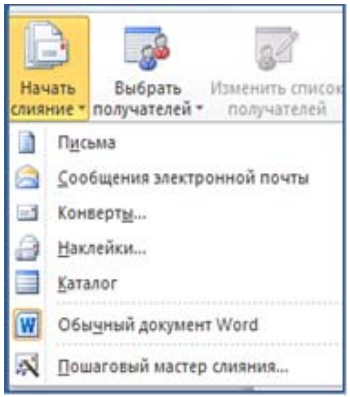


Рис. 4.62. Всплывающее меню команды Начать слияние

При отсутствии источника данных выберите режим Создание списка и щелкните по кнопке ***Создать***. При этом откроется окно диалога (рис. 4.64), предлагающее набор полей для заполнения. Команда *Создать запись* добавляет новую запись в список, команда Настройка столбцов обеспечивает добавление или удаление новых полей, а также упорядочивает их следование путем перемещения вверх или вниз по списку. После окончания ввода данных щелкните по кнопке ОК – программа предложит сохранить базу

данных на диске. Сохраните данные и переходите к следующему этапу – *Создание письма* (рис. 4.65).

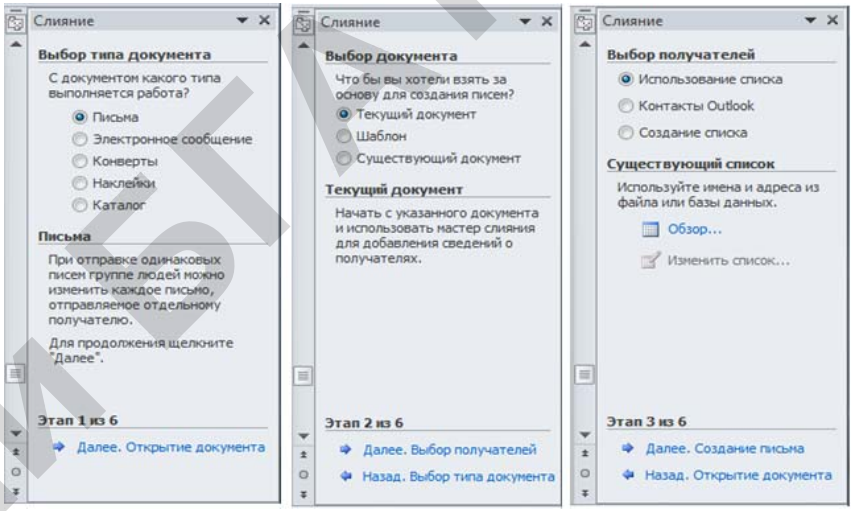


Рис. 4.63. Мастер Слияния. Шаги 1–3

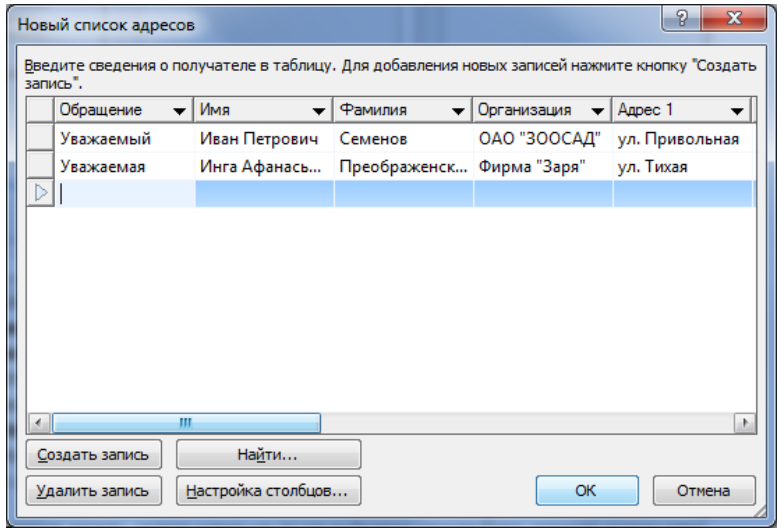


Рис. 4.64. Создание списка рассылки

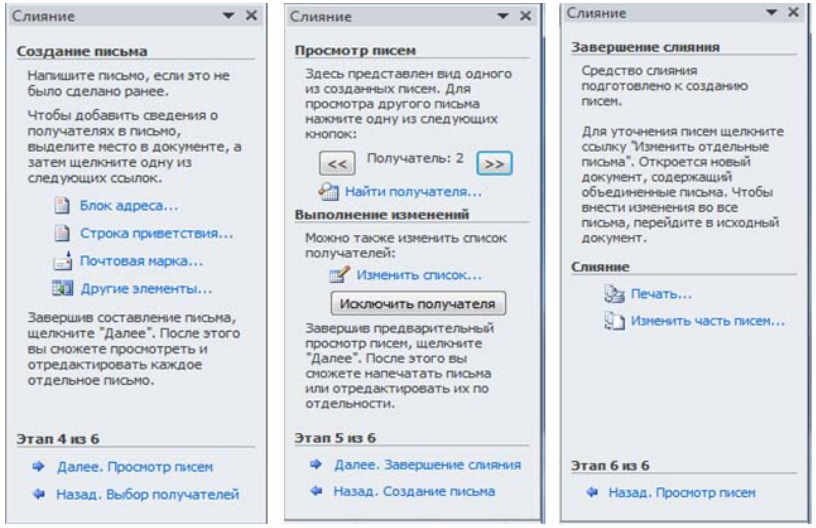


Рис. 4.65. Мастер слияния. Шаги 4–6

Команда ***Блок адреса*** предложит вставить поля для адреса получателя, команда ***Строка приветствия*** поможет выбрать обращение и форму приветствия. Очевидно, что это следует делать в том случае, когда в базе данных нет поля *Обращение*. Команда ***Почтовая марка*** используется при работе с конвертами. Команда ***Другие элементы*** позволит нам выполнить работу по вставке полей в отведенные им места. При вводе команды Другие элементы открывается окно диалога со списком полей. Перетащите мышкой нужные поля в отведенные для них места и переходите к следующему этапу – ***Просмотр писем***. В этом окне можно посмотреть все отобранные письма. Кнопка ***Исключить получателя*** позволяет исключить текущий элемент списка. Кнопка ***Изменить список*** откроет базу данных, в которой можно вручную снять флажки напротив тех записей, которые не следует отправлять. В этом окне можно также выполнить сортировку записей командой ***Сортировка*** и установить фильтр для автоматического отбора записей по выбранным критериям командой ***Фильтр***. Окно диалога Фильтр (рис. 4.66) позволяет формировать критерии отбора с использованием логических условий И и ИЛИ. В приведенном примере в выходной список будут помещены записи, в которых содержится обращение Уважаемый и адресаты находятся в городе Минске или Бресте.

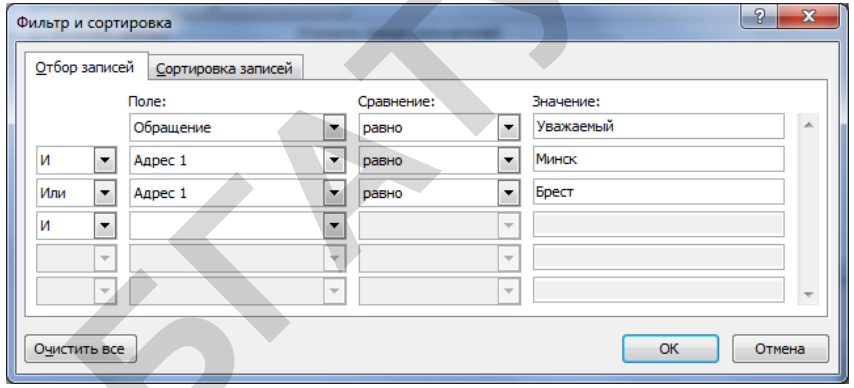


Рис. 4.66. Окно диалога Фильтр

На шестом, последнем шаге можно отправить письма на печать или в файл. Окно диалога предлагает две команды: Печать и Изменить часть писем. При выборе команды ***Печать*** слияние данных с полями документа происходит в процессе печати, документа, содержащего все отправленные письма, не создается. При выборе команды ***Изменить часть писем*** программа предложит создать новый документ, в который будут помещены все отправленные письма. И в первом, и во втором случае открывается еще одно окно диалога, которое предложит указать программе, что печатать: все, текущую запись или указанные записи.

По окончании слияния получим готовый документ. При этом программа попытается улучшить оформление документа (рис. 4.67).

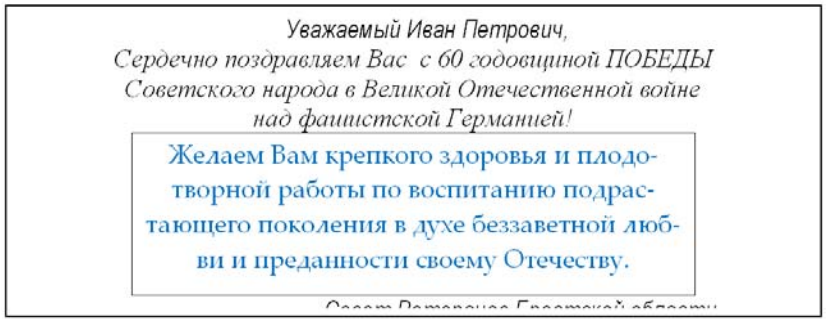


Рис. 4.67. Шаблон основного документа после слияния

Возобновление слияния

Если по какой-либо причине нельзя завершить работу по слиянию документов, сохраните основной документ. Word сохраняет основной документ, источник данных и данные поля. При повторной загрузке основного документа Word запросит подтверждение на открытие документа. Если ответить Да, то программа вернется к тому состоянию документа, когда была прекращена работа.

Создание конвертов

После создания документов их необходимо разложить по конвертам, надписать адреса и разослать по почте. Понятно, что это тоже трудоемкая работа. Поэтому целесообразно поручить ее мастеру слияния документов.

При подготовке конвертов может использоваться та же информационная база, которая использовалась при создании документов. Откройте список ***Начать слияние*** на вкладке ***Рассылки*** и выберите команду ***Конверты***. В окне диалога (рис. 4.68) выберите формат конверта, настройте параметры печати. Закройте окно диалога, щелкнув по кнопке ОК. Запишите в верхнем поле адрес отправителя. Перейдите в нижнее поле для вставки адреса получателя. Откройте список ***Выбрать получателей*** и выберите источник данных. Если база данных существует, выберите команду ***Использовать существующий список***. Щелкните по кнопке ***Блок адреса*** в группе ***Составление документа и вставка полей*** – в документ вставляется поле слияния Блок адреса. Щелкните по кнопке ***Просмотреть результаты*** в одноименной группе – в блок адреса вставляется адрес получателя из базы данных. Завершите слияние, щелкнув мышью по кнопке ***Найти и объединить*** и выберите команду ***Печать***. После этого программа предложит настроить список для печати конвертов.

При использовании команды ***Конверты*** из группы ***Создание*** вкладки ***Рассылки*** принцип подготовки и печати конвертов аналогичен описанному ранее, но ввод адресов получателей осуществляется вручную. Адрес отправителя можно сохранить кнопкой Добавить/Изменить для использования в следующих сеансах. Имеется возможность выбора адресов отправителей и получателей из списка электронной почты Outlook Express.

Контрольные вопросы

1. Для чего используется мастер слияния документов?
2. Что такое основной документ, как его создать?
3. Что такое источник (база) данных, как он создается?

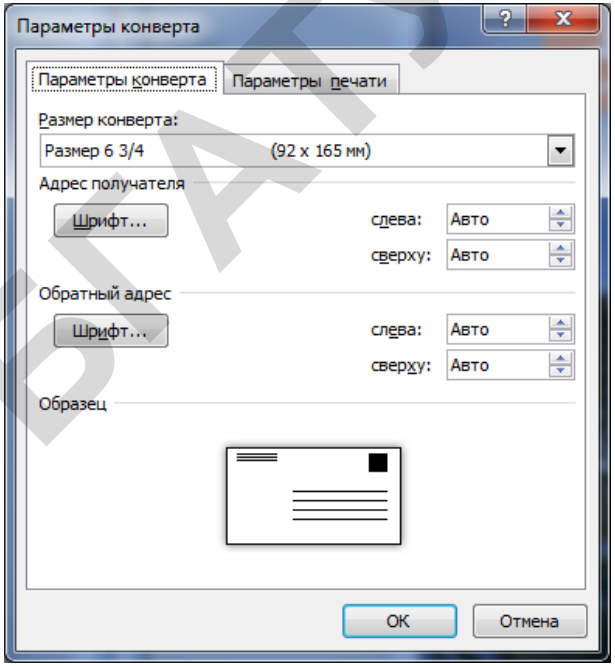


Рис. 4.68. Печать конвертов

4. Расскажите порядок слияния документов.
5. Как осуществляется выбор адресатов при рассылке писем?
6. Как заадресовать конверты для рассылки писем?

4.7. МАКРОКОМАНДЫ

Макрокоманды, или макросы, представляют собой набор команд, с помощью которых можно автоматизировать выполнение повторяющейся задачи. Макросы позволяют легко и быстро выполнять часто повторяющиеся действия и тем самым повышать производительность труда пользователя.

Макрокоманды хранятся в шаблонах. Выбором шаблона для сохранения ограничивается круг документов, в которых можно использовать ту или иную макрокоманду. Место хранения макроса указывается при его создании.

Макрокоманда может быть написана пользователем с использованием встроенного языка программирования Visual Basic или записана с помощью средств записи программы Word. Макрокоманда имеет имя. Имя макрокоманды может содержать до 36 символов, в имени не допускается использование пробелов. Макрокоманде можно назначить комбинацию клавиш или кнопку на панели инструментов.

Запись макроса средствами записи программы Word

Проще всего создать макрос с помощью встроенных средств записи. Продумайте последовательность действий, необходимых для выполнения нужной операции (сценарий), а затем повторите их в режиме записи макроса. Это необходимо делать всегда, так как в макрос будут записаны все команды, в том числе и ошибочные. Создадим для примера макрос для удаления текста справа от точки вставки.

Требуемая последовательность действий: установить курсор в строку текста слева от удаляемого текста; выделить текст, нажав комбинацию клавиш Shift + End; нажать клавишу Delete.

Начать запись макроса можно командой ***Макросы, Запись макроса*** из группы ***Макросы*** вкладки ***Вид*** или одноименной командой из группы ***Код*** вкладки ***Разработчик***.

Алгоритм записи макроса:

- Установите курсор в любое место в тексте.
- Введите команду Макросы, Запись макроса.
- Введите имя макроса в строке ввода Имя макроса (рис. 4.69).

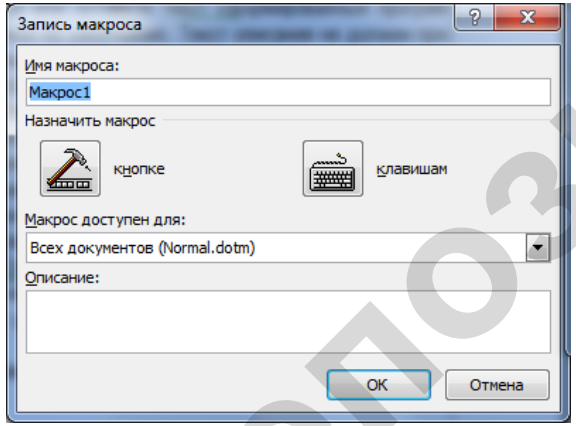


Рис. 4.69. Окно диалога записи макроса

- Выберите в списке «Макрос доступен для» документ, где будет храниться макрос.
- В окне Описание дайте, при необходимости, описание макроса или оставьте текст, сформированный программой по умолчанию. Текст описания не должен превышать 255 символов. При вводе длинного текста для перехода на новую строку нажмите Shift + Enter.
- Макросу можно назначить кнопку на панели инструментов или комбинацию клавиш с помощью кнопок в группе Назначить макрос.

После выполнения всех необходимых операций щелкните по кнопке ОК. С этого момента все ваши действия будут записаны в макрос:

- Выполните последовательность действий для удаления текста справа согласно разработанному ранее сценарию.
- После выполнения всех необходимых команд остановите запись макроса командой ***Макрос, Остановить запись***.

Выполнение макроса

Для запуска макроса установите курсор слева от текста, который надо удалить, и введите команду ***Вид, Макрос, Макросы***, выберите в списке нужный макрос и щелкните по кнопке ***Выполнить***.

Процесс применения макроса значительно упрощается, если назначить макросу комбинацию клавиш или назначить кнопку на панели инструментов.

Безопасность при работе с макросами

При работе с макросами следует соблюдать осторожность, так как макросы могут содержать вирусы и компьютер или локальная сеть могут быть заражены. Поэтому при открытии документа, содержащего макросы, система безопасности операционной системы выдает предупреждение: появляется желтая панель сообщений со значком щита и кнопкой ***Включить содержимое***. Если известно, что макрос поступил из надежного источника, то щелкните по кнопке ***Включить содержимое***: файл откроется как надежный документ.

Предупреждение системы безопасности Запуск макроса отключен Включить содержимое

Для изменения режима запуска макросов выполните следующее: введите команду ***Файл, Параметры, Центр управления безопасностью, Параметры центра управления безопасности***, Выберите один из приведенных режимов (рис. 4.70):

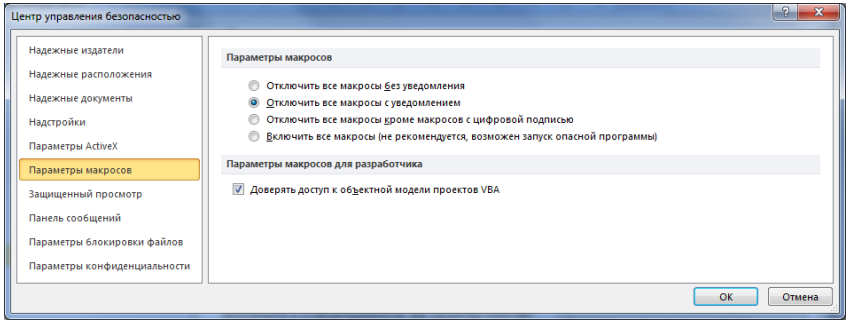


Рис. 4.70. Настройка параметров запуска макросов

отключить все макросы без уведомления – запуск макросов будет невозможен;

отключить все макросы с уведомлением – этот режим установлен по умолчанию;

отключить все макросы, кроме макросов с цифровой подписью. В этом случае предполагается, что макрос поступил из надежного источника и не содержит вирусов;

включить все макросы – этот режим не рекомендуется, так как возможно заражение компьютера макровирусами.

Назначение макроса кнопке на панели инструментов

Назначить макрос кнопке панели инструментов или назначить макросу комбинацию клавиш можно сразу же при создании макроса в окне диалога Запись макроса (рис. 4.69). Выберите команду ***Назначить Макрос кнопке***. Открывается окно диалога ***Параметры Word*** (рис. 4.71). Выделите кнопку в левом окне и щелкните по кнопке ***Добавить***. Кнопка закрепится на панели быстрого доступа.

Если вид кнопки с длинной надписью не устраивает Вас, то ее можно настроить. Выделите новую кнопку и щелкните по кнопке Изменить. Открывается окно ***Изменение кнопки*** (рис. 4.72), в котором можно выбрать подходящую кнопку и изменить либо совсем удалить отображаемое имя кнопки.

Назначение макросу комбинации клавиш

При выборе в окне диалога ***Запись макроса*** (рис. 4.69) команды ***Назначить макрос клавишам*** откроется окно диалога Настройка клавиатуры (рис. 4.73). Введите в окне *Новое сочетание клавиш* сочетание клавиш, например, [Ctrl + Shift + L] и щелкните по кнопке ***Назначить***. Введенная комбинация клавиш переместится из

окна *Новое сочетание клавиш* в окно *Текущее сочетание*. Закройте окно диалога и продолжите запись макроса.

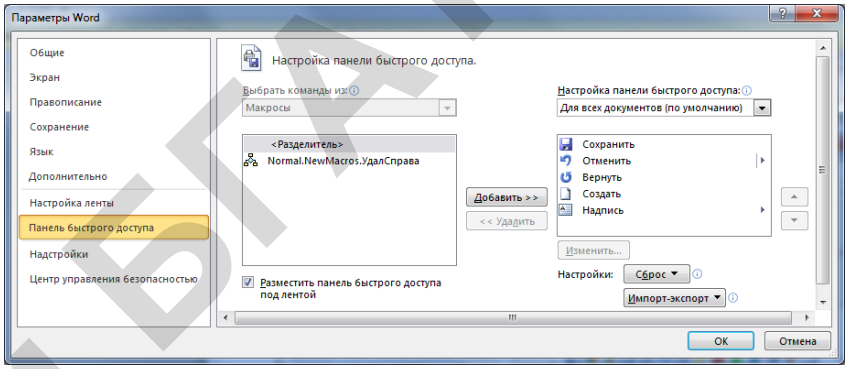


Рис. 4.71. Назначение макроса кнопке панели инструментов

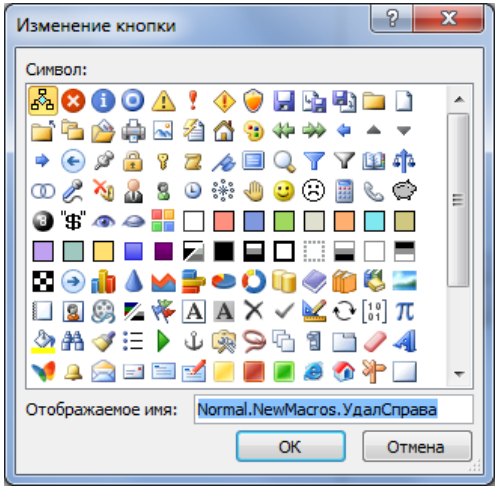


Рис. 4.72. Окно диалога настройка макроса

Запись макроса с помощью встроенного языка программирования Visual Basic

Для записи макроса с помощью встроенного языка программирования Visual Basic введите команду ***Visual Basic*** в группе ***Код*** вкладки ***Разработчик*** – открывается окно разработки проекта (рис. 4.74).

Если нет вкладки ***Разработчик*** (по умолчанию она не активирована), то выполните следующие действия: введите команду ***Файл, Параметры, Настройка ленты*** и в списке ***Основные вкладки*** установите флажок ***Разработчик***.

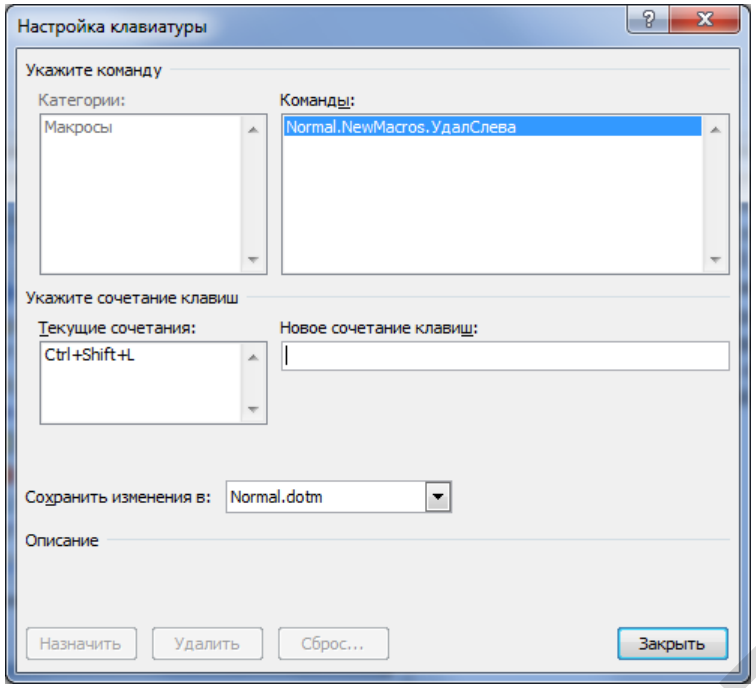


Рис. 4.73. Назначение комбинации клавиш макросу

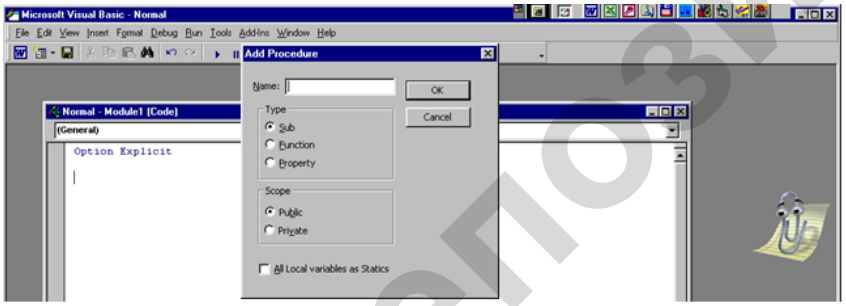


Рис. 4.74. Окно разработки программы Visual Basic

Введите команду ***Insert, Module*** – открывается окно разработки программы ***Module1***. Введите команду ***Insert, Procedure*** – открывается окно добавления процедуры ***Add Procedure***. Введите имя процедуры, например, ***Codirovka***, установите переключатели ***Sub*** – ключевое слово заголовка процедуры и ***Public*** – общая (процедура будет доступна всем формам проекта) и щелкните по кнопке ***OK*** – программа возвращается в окно проекта Module1. В этом окне появится шаблон процедуры, две строки команд, между которыми необходимо записать текст программы:

```
Public Sub Codirovka()
    ‘Текст программы
End Sub
```

Если Вы умеете программировать или позаимствовали текст понравившейся процедуры у товарища, напишите здесь или скопируйте сюда текст программы.

Часто бывает, что мы забываем переключить клавиатуру с русского на английский текст или наоборот и тогда досадуем на себя за такую оплошность. Проблему легко устранить с помощью программы, текст которой приведен ниже. Строки программы, начинающиеся с апострофа и оператора Rem, – комментарии.

```
Rem Программа перекодировки текста с русского языка на английский и наоборот
Public Sub Codirovka() ‘ заголовок процедуры
    On Error Resume Next ‘ обработчик ошибок
    ‘ Объявление переменных:
    Dim n As Integer, strSetRus As String, strSetEng As String
    Dim strMisStr As String, strCurrChar As String
    Dim strNewStr As String, numChrPos As Integer
    ‘ strMisStr – переменная для хранения ошибочного текста
    strMisStr = Selection.Text ‘ присвоение выделенной строки переменной
    ‘ strSetEng – строка, содержащая символы английского языка
    strSetEng = "QWERTYUIOP{}ASDFGHJKL:ZXCVBNM<>qwerty
    uiop[]asdfghjkl; zxcvbnm,."
    ‘ strSetRus – строка, содержащая символы русского алфавита
    ‘ русские и английские символы записаны не в алфавитном порядке,
    ‘ а в соответствии с раскладкой клавиатуры
```

```
    strSetRus = "ЙЦУКЕНГШЩЗХЪФЫВАПРОЛДЖЯЧСМИТЬБЮ
    йцукенгшщзхъфывапролджячсмитьбю"
    ‘ цикл: извлекается символ из строки с ошибочным текстом
    и сравнивается со
    ‘ строками, содержащими английский и русский текст. Если
    в ошибочной
    ‘ строке английский текст, то он будет заменен на русский и на-
    оборот
    ‘ исправленный текст помещается в переменную strNewStr
        For n = 1 To Len(strMisStr)
            strCurrChar = Mid(strMisStr, n, 1)
            numChrPos = InStr(strSetEng, strCurrChar)
        If numChrPos <> 0 Then
            strCurrChar = Mid(strSetRus, numChrPos, 1)
        Else
            numChrPos = InStr(strSetRus, strCurrChar)
            strCurrChar = Mid(strSetEng, numChrPos, 1)
        End If
        strNewStr = strNewStr & strCurrChar
    Next n
    ‘ выделенному тексту присваивается новая строка
    Selection.Text = strNewStr
End Sub
```

Когда программа будет написана, сохраните ее на диске командой ***File, Save Normal***, а затем вернитесь в документ командой ***File, Close and Return to Microsoft Word***. Наша программа попала в список макросов.

Чтобы применить ее, выделите ошибочный текст и введите команду ***Сервис, Макрос, Макросы***. Выделите макрос ***Codirovka*** и щелкните по кнопке ***Выполнить***. Для удобства использования назначьте макрос комбинации клавиш или кнопке панели инструментов, как описано выше в данном разделе.

Контрольные вопросы

1. Что такое макрос? Как его создать?
2. Как используются макросы?
3. Как назначить макрос кнопке панели инструментов?
4. Как назначить макросу комбинацию клавиш?
5. Как написать макрос с помощью встроенного языка программирования?

Заключение

В настоящем разделе мы познакомились с текстовым процессором Microsoft Word. Он обладает большими возможностями по редактированию текста, размещению текста в виде колонок, созданию списков, бюллетеней, вставки рисунков и рисованных объектов, формул. Имеется возможность создавать таблицы и управлять ими, проводить несложные вычисления. Редактор позволяет создавать базы данных и использовать их для рассылки документов, писем. Имеется также возможность автоматизации часто повторяющихся операций путем создания макросов. В разделе рассмотрены далеко не все возможности процессора, что ограничено рамками учебного пособия.

5. ПРЕЗЕНТАЦИЯ

Ключевые слова: выдача (выдач), конспект доклада, презентация, представление презентации, раздаточный материал, слайд.

5.1. ВИДЫ ПРЕЗЕНТАЦИЙ

Презентации предназначены для наглядного представления результатов выполненной работы, разъяснения целей, задач и планов, обучения и т. п.

Презентации представляют собой один из видов мультимедийного документа в виде слайдов, то есть документа, объединяющего в себе различные способы представления информации: текст, таблицы, диаграммы и графики, звук, видеоизображения, анимацию, спецэффекты.

Классификация презентаций

Все презентации можно классифицировать по некоторым признакам, определяющим требования к их оформлению и представлению: по способу представления, по способу управления представлением, по области применения.

По способу представления информации презентации делятся на линейные и со сценарием.

В *линейных презентациях* материал расположен по порядку: начало, продолжение, завершение. Такой вид презентации используется в торговле для представления сведений о товаре, рекламных роликах, в обучающих презентациях.

Презентации *со сценарием* предполагают показ слайдов, снабженных анимированными объектами, видеоматериалом, звуковым сопровождением, а также спецэффектами.

По способу управления представлением презентации можно разделить на *интерактивные* и *непрерывные* (рис. 5.1).

Интерактивные презентации выполняются под управлением пользователя. Они построены на основе диалога программы

с пользователем и должны предоставлять ему возможность самостоятельно выбирать направление просмотра, переходить от одного раздела к другому, используя элементы управления.

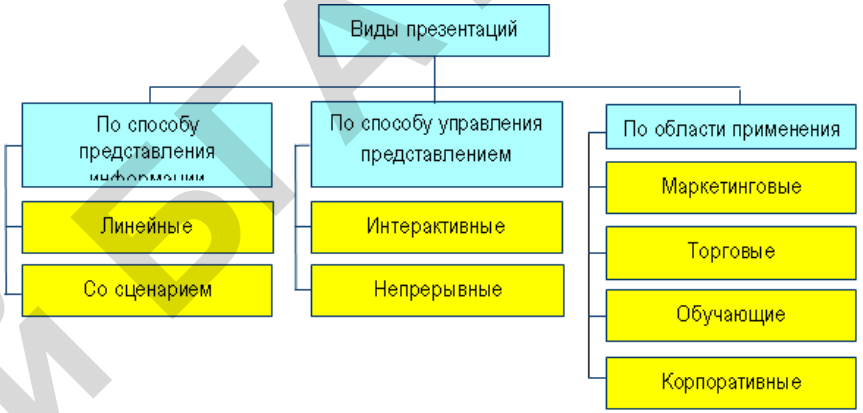


Рис. 5.1. Классификация презентаций

Непрерывные презентации позволяют предоставлять информацию в автоматическом режиме. Это завершенные программные продукты. Такие презентации используются в рекламе, различного рода экспозициях.

По области применения презентации можно разделить на следующие виды.

Маркетинговые – отражают направления деятельности компании, виды услуг.

Торговые – используются дилерами или торговыми агентами при заключении сделок, продаже товаров и услуг.

Обучающие презентации – используются при обучении персонала, в учебных заведениях при чтении лекций, выполнении лабораторных работ и практических заданий.

Корпоративные презентации – ориентированы на потенциальных инвесторов или освещают финансовую деятельность компании и т. д.

Средства создания презентаций

Фирма Microsoft предоставляет пользователям обширный набор приложений ОС Windows для создания презентаций: Power Point, Action, Animation Works Interactive, Multimedia ToolBook,

MS Publisher. Однако наиболее популярной является программа Power Point.

Основные понятия, используемые при работе в среде Power Point

Презентация – это набор слайдов и слайд-фильмов, раздаточные материалы, а также конспект и планы докладов, хранящиеся в одном файле.

Файлы Power Point имеют расширение ***.pptx***. Старые версии Power Point сохраняли файлы с расширением ppt. Файлы этих версий не совместимы. Для обеспечения возможности использования в предыдущих версиях программы презентаций, подготовленных в Power Point 2010, при сохранении файлов необходимо использовать тип файлов *Презентации PowerPoint 97–2003 (*.ppt)*.

Слайд – отдельная страница презентации, включающая разные объекты Power Point: заголовок, текст, графику, диаграммы, таблицы, рисунки, рисованные объекты, фотографии, формулы, видеоклипы, видеофильмы. Слайды можно распечатывать на бумаге или на прозрачной пленке.

Раздаточный материал – это распечатанные в компактном виде несколько слайдов на одной странице с целью закрепления восприятия слушателями темы доклада и возможности самостоятельно вернуться к теме доклада.

Конспект доклада – текст доклада, при печати которого на каждой странице будут выведены уменьшенное изображение слайда и текст, составляющий его содержание.

5.2. СРЕДА РАЗРАБОТКИ ПРЕЗЕНТАЦИЙ

Рабочее окно программы Power Point представляет собой стандартное окно Windows (рис. 5.2) и включает строку заголовка, ленту, панель быстрого доступа, рабочее окно, строку состояния. В рабочем окне размещены три объекта: шаблон слайда, окно структуры документа и Заметки к слайду. *Шаблон слайда* занимает большую часть экрана, на нем, собственно, и ведется разработка слайда. *Окно структуры* документа предназначено для быстрого просмотра и перемещения по слайдам и имеет две вкладки: Слайды и Структура. В режиме Слайды информация выводится в виде последовательности мини-слайдов. В режиме Структура на экран выводится только текстовая информация,

разделенная на кадры, рисунки не выводятся. В данном окне можно копировать, вставлять, удалять и перемещать слайды. Например, для перемещения слайда выделите его, нажмите и удерживайте левую клавишу мыши, Начните перемещение слайда – к указателю мыши прикрепляется маленький прямоугольник; при достижении перемещаемым слайдом места вставки между слайдами появляется горизонтальная черная полоса, отпустите клавишу мыши.

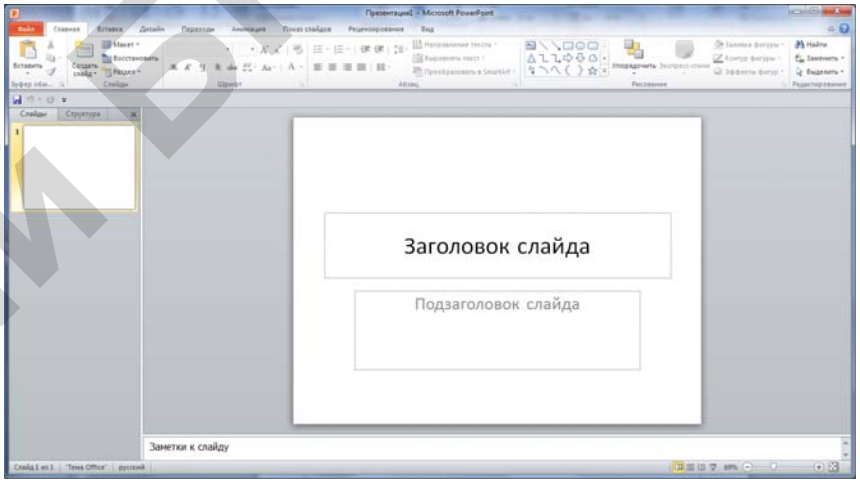


Рис. 5.2. Интерфейс программы Power Point

Заметки к слайду могут содержать дополнительную текстовую информацию, поясняющую содержание текущего слайда.

Структура ленты и строки состояния аналогична структуре ленты и строки состояния этих объектов в текстовом процессоре Word (см. раздел 4.1).

В левой части строки состояния слева указаны номер текущего слайда и общее число слайдов в презентации, тема оформления слайдов, используемый язык редактирования документа. В правой части строки состояния расположены кнопки режимов просмотра, настройки масштаба и кнопка ***Вписать слайд в окно***. Кнопка *Вписать слайд в окно* подгоняет размеры слайда к размеру рабочего окна, сохраняя соотношение сторон слайда.

5.3. СОЗДАНИЕ ПРЕЗЕНТАЦИИ

Можно указать несколько способов создания презентации:

начать с чистого листа и все операции выполнять в Power Point;

использовать для презентации заранее подготовленный текст документа;

использовать шаблоны Power Point;

использовать существующую презентацию.

Начать с чистого листа. Этот способ можно использовать, когда мало текстовой информации и Вы собираетесь читать доклад с демонстрацией кадров, давая пояснения. При больших объемах текста этот путь не эффективен.

Использование готового текстового документа. Можно ускорить создание презентации, для этого надо руководствоваться следующим алгоритмом:

1. Выполнить некоторые подготовительные операции над исходным текстом:

создать копию текстового документа и использовать ее для дальнейшей работы;

выделить в документе фрагменты текста приемлемого объема, которые будут помещены в один слайд, и оформить их стилем заголовка № 2 (размер шрифта заголовков можно уменьшить);

расставить заголовки к каждому фрагменту текста и оформить их стилем заголовка № 1;

определить текст, который будет помещен в качестве заметок к слайдам;

сохранить полученный документ на диске и закрыть документ в текстовом процессоре.

2. Загрузить документ в программу Power Point:

ввести команду ***Файл, Открыть***, установить в окне диалога ***Открытие документа*** тип файлов ***Все***;

загрузить подготовленный файл презентации. Программа автоматически формирует слайды, внося заголовки, оформленные стилем Заголовок 1, в заголовок слайда, а текст, оформленный стилем Заголовок 2, в слайд. Таблицы, рисунки, формулы автоматически не переносятся.

3. Открыть в текстовом процессоре Word подготовленный ранее файл презентации:

вставить пустые слайды в местах размещения таблиц, формул, рисунков. Добавление новых слайдов в Power Point осуществляется командой ***Создать слайд*** вкладки ***Главная*** ленты;

скопировать и вставить в слайды необходимые рисунки;

скопировать и вставить в область Заметки к слайду текст заметок.

Рекомендация

При вставке рисунков используйте команду ***Вставить, Специальная вставка, Рисунок (расширенный метафайл)***. При вставке таблиц и формул удобно воспользоваться ***Контекстным меню*** и выбрать режим вставки ***Сохранить исходное форматирование***.

4. Приступить к оформлению документа.

Использование шаблона. Power Point предоставляет пользователю возможность создать презентацию на определенную тему, используя готовые шаблоны, например, Обучение, Отчет о состоянии проекта, Рекламный буклет и т. п. Введите команду ***Файл, Создать***: перед Вами откроется окно диалога, в котором можно выбрать понравившуюся тему. В том числе представлена презентация Power Point 2010, которая позволит пользователю ознакомиться с возможностями программы, загрузив шаблон и открыв его в режиме показа. При недостатке шаблонов можно поискать что-нибудь подходящее на сайте Office.com.

Здесь же можно подобрать и тему оформления.

Использование существующей презентации. В этом случае важно сохранить стиль оформления, а не содержание. Сохраните слайды, которые будут вам необходимы, и замените в них текст. Остальные слайды удалите. В принципе, достаточно оставить заголовок и один из слайдов, чтобы сохранить стиль оформления документа.

Управление внешним видом рабочей среды

Лента содержит несколько постоянно открытых вкладок: Главная, Вид, Вставка, Дизайн, Переходы, Анимация, Показ, Рецензирование (рис. 5.2). Каждая из вкладок содержит набор команд, сгруппированных по функциональному назначению.

Управление внешним видом рабочего окна программы осуществляется с помощью вкладки ***Вид*** и группы команд в правой части строки состояния, о которых упоминали ранее. Вкладка Вид (рис. 5.3) содержит несколько групп команд, позволяющих управлять режимами просмотра, настраивать параметры слайдов, выдач, заметок, внешний вид окна редактора, управлять масштабом изображения, открытыми окнами.

Режимы просмотра презентации и образцы документов

Power Point предоставляет пользователю четыре режима просмотра презентации: Обычный, Сортировщик слайдов, Чтение и Показ слайдов.

Рис. 5.3. Лента, вкладка Вид

Обычный режим просмотра слайдов предназначен для разработки презентации. Вид документа в этом режиме приведен на рисунке 5.2.

Сортировщик слайдов позволяет быстро просматривать презентацию и осуществлять перемещение отдельных слайдов или групп слайдов в требуемое положение. Этот режим соответствует многостраничному режиму просмотра документов в текстовом процессоре Word.

Режим чтения – это полноэкранный режим просмотра документа. В этом режиме отображаются и спецэффекты, примененные к кадрам.

Показ слайдов – это представление презентации целевой аудитории. Управление показом в ручном режиме осуществляется с помощью клавиш управления курсором «вверх» и «вниз», PgUp, PgDn, а также команд контекстного меню.

Команда ***Страница заметок*** из группы ***Режимы просмотра презентаций*** позволяет просмотреть страницу заметок вместе со слайдом.

Группа команд ***Режимы образцов*** позволяет просматривать и настраивать образцы слайдов, выдач и заметок для разработки и печати. Здесь ***выдачи*** – это слайд, выводимый на печать.

На рисунке 5.4 приведен фрагмент окна при настройке ***Образцов слайдов***. В левой части экрана выведены образцы оформления слайдов. В этом режиме можно установить типы шрифтов, их размеры, образец заголовка, а также размеры и шрифты даты, колонтитула и номера слайда (в нижнем правом углу образца слайда). Установленные параметры будут автоматически применяться ко всем вставляемым слайдам. Однако если пользователь редактирует шрифт слайда в процессе его разработки, то связь с настройками прерывается. Команда ***Образец выдач*** позволяет настроить режим печати слайдов: ориентацию, число слайдов на странице, колонтитулы и др. Команда ***Образец заметок*** позволяет настраивать для печати конспект доклада.

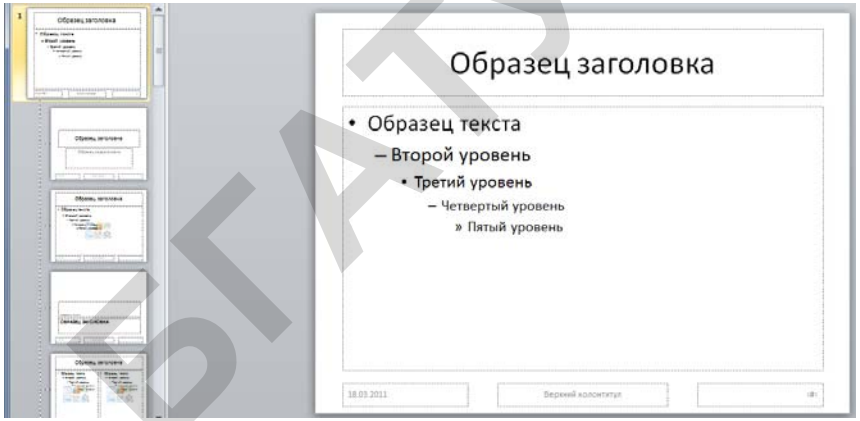


Рис. 5.4. Фрагмент окна при настройке образцов слайдов

Создание слайдов

Power Point 2010 имеет мощные средства для разработки слайдов: ввода и редактирования текста, вставки различных объектов – и в этом смысле мало чем уступает текстовому процессору Word 2010. Все необходимые элементы для создания слайда сосредоточены на вкладке ***Главная***. На ней размещены:

группа ***Буфер обмена*** – содержит команды вырезания, копирования, вставки с использованием буфера обмена, форматирования по образцу – достаточно щелкнуть дважды мышкой по кнопке Форматирование по образцу, чтобы применить этот формат к другим частям документа;

группа ***Слайды*** позволяет создать новый слайд. При этом команды ***Создать слайд*** и ***Макет*** имеют одно и то же назначение и предоставляют пользователю шаблоны размещения информации на слайде (рис. 5.5). Команда ***Восстановить*** – восстанавливает форматирование слайда, заданное по умолчанию. Команда ***Раздел*** предназначена для упорядочения слайдов в больших презентациях. Разделам можно присваивать наименования. Разделы можно просматривать в режиме сортировщика слайдов и в обычном режиме, однако режим сортировщика более подходит для определения структуры презентации и распределения слайдов по разделам. Для создания раздела щелкните правой кнопкой мыши между слайдами, где предполагается вставить раздел. В выбранном месте появится метка *Раздел без заголовка*. Раздел можно переименовать с помо-

щью контекстного меню. Вызовите контекстное меню метки *Раздел без заголовка*, выберите команду ***Переименовать раздел***, введите имя раздела и щелкните по кнопке ***Переименовать***, аналогично раздел можно удалить;

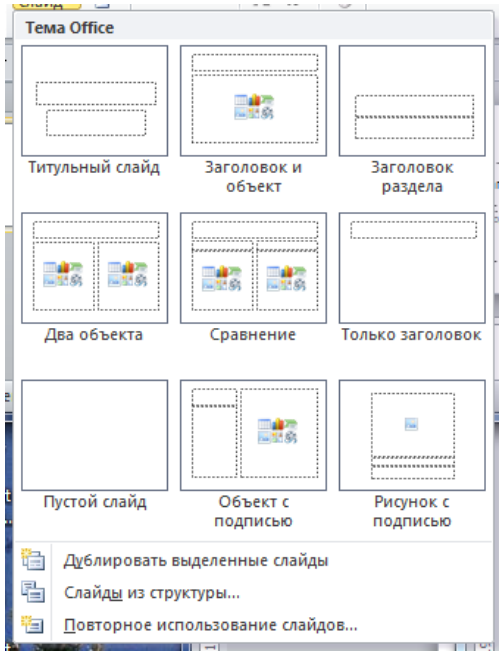


Рис. 5.5. Образцы слайдов темы Office

группы ***Шрифт, Абзац, Рисование, Редактирование*** аналогичны соответствующим командам Word 2010.

Вкладка ***Рецензирование*** позволяет проверять орфографию, добавлять примечания к тексту заметок, пользоваться справочниками, переводчиком текста с других языков, выбирать язык редактирования справок, примечаний, интерфейса.

Шаблоны слайдов содержат местозаполнители. Пустые местозаполнители служат для ввода текста, местозаполнители с картинками служат для вставки объектов: таблиц, графиков и диаграмм, рисунков SmartArt, рисунков из файлов, картинок, клипов мультимедиа и даже фильмов (рис. 5.6). Двойной щелчок по соответствующему объекту вызывает окно диалога вставки этого объекта. В пустые

слайды можно вставлять разные объекты, кроме текста. Для вставки текста в пустой слайд необходимо вставить ***Надпись*** (вкладка Главная, группа Рисование, Фигуры или вкладка Вставка, группа Иллюстрации, Фигуры).

Рис. 5.6. Образцы слайдов темы Office

Вставка объектов

Вкладка ***Вставка*** Power Point 2010 мало чем отличается от одноименной вкладки ленты Word 2010. Здесь наше внимание может привлечь группа Мультимедиа. Она предоставляет возможность вставки видеоматериалов из файла, видеосайта, организатора клипов. Список типов видеофайлов, которые можно использовать в Power Point, практически не ограничен. Звук можно воспроизводить из звукового файла, из организатора клипов, а также имеется возможность записать звук с микрофона.

Для воспроизведения файла звука или видео требуется универсальный проигрыватель, который воспроизводит мультимедийные файлы и управляет такими устройствами воспроизведения, как приводы компакт- и видеодисков.

Группа ***Ссылки*** позволяет вставить гиперссылки на веб-страницу, рисунок, адрес электронной почты или программу. Имеется возможность для каждого объекта установить действие, выполняемое по щелчку мыши или наведении на объект указателя мыши. К таким действиям относятся, например, переход на гиперссылку, запуск программы, запуск макроса.

Объект SmartArt служит для визуального представления информации в виде списков, последовательности переходов, иерархии, отражения связи между объектами (рис. 5.7). После выбора элемента SmartArt открывается вкладка ленты для работы с данным объектом, содержащая две вкладки: Конструктор и Формат, позволяющие вести настройку параметров объектов.

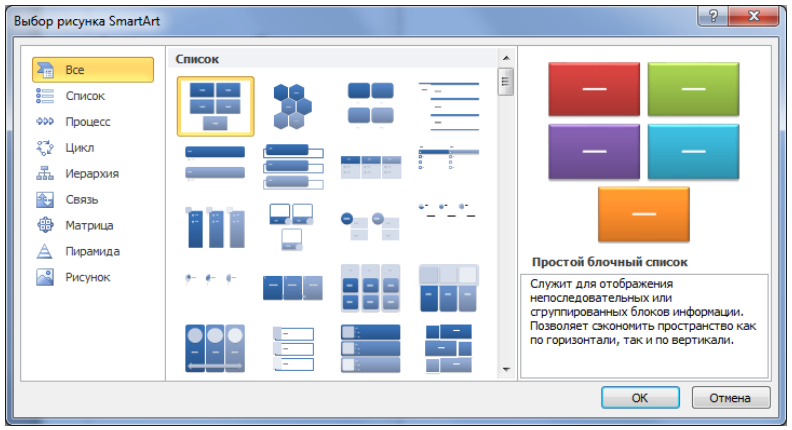


Рис. 5.7. Вставка объектов SmartArt

Для создания рисунка SmartArt выполните следующее:

на вкладке Вставка ленты в группе Иллюстрации щелкните по кнопке SmartArt;

в диалоговом окне Выбор рисунка SmartArt выберите необходимый тип: Список, Процесс, Цикл и др. и соответствующий макет;

введите текст.

Если на слайде уже есть текст, его можно преобразовать в рисунок SmartArt:

щелкните рамку с текстом на слайде, который нужно преобразовать;

на вкладке ***Главная*** в группе ***Абзац*** щелкните по кнопке ***Преобразовать в рисунок SmartArt***;

в коллекции выберите нужный макет рисунка SmartArt.

В Power Point 2010 добавлен новый тип рисунков SmartArt, который позволяет использовать фотографии. Чтобы создать такой графический объект, необходимо выполнить следующее:

вставьте макет рисунков SmartArt;

введите поясняющий текст.

5.4. ОФОРМЛЕНИЕ СЛАЙДОВ

После создания слайдов можно подумать и об улучшении внешнего вида Вашего творения. Это легко выполнить с помощью вкладок ***Дизайн***, ***Переходы*** и ***Анимация***.

Вкладка ***Дизайн*** (рис. 5.8) предоставляет пользователю возможность выбрать тему из предлагаемого списка. Выбранная тема может быть применена ко всем слайдам или к выделенным слайдам. Обычно выбирается одинаковый тип для всех слайдов. Исключение можно сделать для титульного листа. Хотя титульный лист темы и так всегда оформляется иначе, чем остальные слады этой темы.

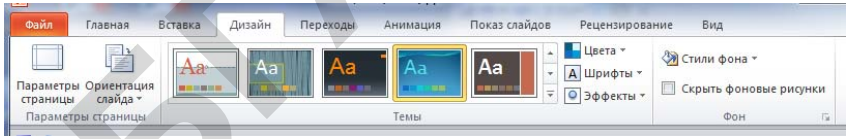


Рис. 5.8. Лента, вкладка Дизайн

Применение темы документа придает документу профессиональный вид. Тема документа представляет собой набор вариантов форматирования, включающих набор цветов, шрифтов (в том числе шрифты заголовков и основного текста) и эффектов (в том числе варианты линий и способы заливки). Применяемые темы влияют на стили, которые можно использовать в документе.

В дополнение к выбранной теме можно изменить цвет, шрифты и эффекты и сохранить созданную тему как пользовательскую, присвоив ей оригинальное имя.

Дополнительных эффектов можно добиться, изменив ***стиль фона***. Стиль фона выбирается из шаблонов: ровная заливка, градиентная заливка, узор заполнения и цветовое решение. Дополнительные возможности предоставляет команда ***Фон***, а также команда ***Формат фона*** в окне диалога ***Стили фона***.

Вкладка ***Переходы*** (рис. 5.9) позволяет создать эффекты, происходящие при смене кадра. После выбора варианта перехода можно изменить его параметры. Здесь же можно подобрать звук, сопровождающий смену кадра и временные параметры перехода.

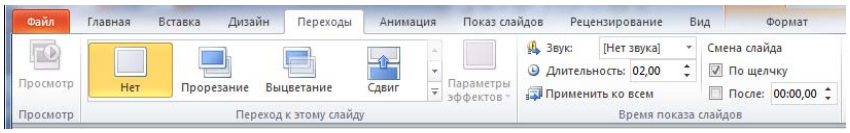


Рис. 5.9. Лента, вкладка Переходы

Вкладка ***Анимация*** (рис. 5.10) создает дополнительные эффекты при смене кадра. Анимация – это добавление к тексту или объекту специального видео- или звукового эффекта. Можно применить анимацию к тексту, рисункам, фигурам, таблицам, графическим элементам SmartArt и другим объектам в презентации. К одному и тому же объекту анимационные эффекты можно применять последовательно один за другим. Все анимационные эффекты, примененные к слайду и его элементам, нумеруются и отображаются на слайде. Во время показа слайдов они не отображаются. Естественно, во всем надо знать меру. Поэтому лишние или неудачные варианты анимации можно удалить. Для этого выделите номер эффекта, который хотите удалить, и в окне диалога ***Стили анимации*** группы ***Анимация*** одноименной вкладки выберите команду ***Нет***.

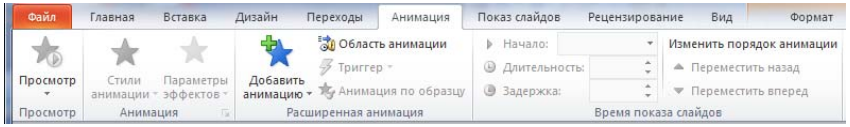


Рис. 5.10. Лента, вкладка Анимация

5.5. ПРЕДСТАВЛЕНИЕ ПРЕЗЕНТАЦИИ

Подготовка презентации к представлению целевой аудитории – это завершающий этап по разработке презентации. Настройка параметров представления презентации осуществляется с помощью команд вкладки ***Показ слайдов*** (рис. 5.11).

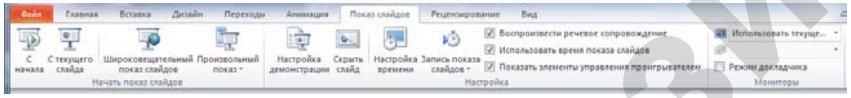


Рис. 5.11. Лента, вкладка Показ слайдов

Начнем с настройки монитора. Прежде всего можно изменить разрешение экрана монитора. При большем разрешении можно показать больше деталей изображения, но при меньшем разрешении экрана показ слайдов осуществляется быстрее. Этот параметр необходимо согласовать с разрешением, поддерживаемым проектором. Большинство проекторов поддерживает разрешение 1024×768

точек на экране, это вполне достаточное разрешение для просмотра большинства презентаций.

Power Point позволяет использовать несколько режимов показа слайдов:

непрерывный показ с начала;

показ с текущего кадра;

произвольный показ только отдельных кадров;

широковещательный показ удаленным зрителям, использующим средства веб-браузера.

Настройка презентации осуществляется с помощью одноименного окна диалога (рис. 5.12). Программой предусмотрено три режима показа слайдов:

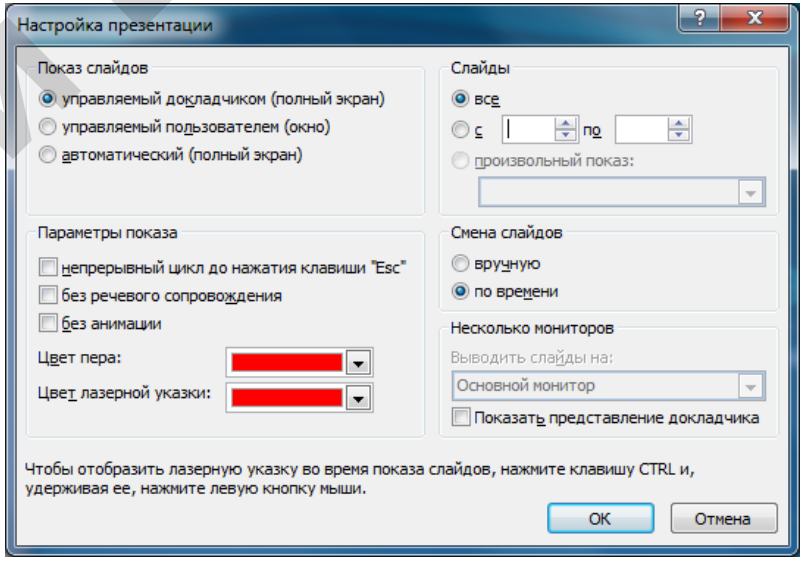


Рис. 5.12. Настройка презентаций

управляемый докладчиком. Это полноэкранный режим, то есть слайд разворачивается на весь экран. Навигация осуществляется с помощью клавиш управления курсором, клавиш PgUp, PgDn, а также контекстного меню и кнопок управления, установленных на слайдах. В этом режиме управление показом осуществляет докладчик в соответствии с разработанным ранее сценарием. Докладчик не связан никакими ограничениями при представлении презен-

тации: можно пропускать слайды, изменять время представления слайда на экране, сопровождая каждый слайд необходимыми пояснениями, возвращаться к ранее просмотренным слайдам, записывать во время доклада речевое сопровождение и т. д.;

управляемый пользователем. Это режим просмотра слайдов в окне документа. В этом режиме можно просматривать, копировать и печатать слайды. Управление просмотром осуществляется с помощью клавиш PgUp и PgDn и кнопок навигации в строке состояния;

автоматический. Это полноэкранный режим. Смена слайдов осуществляется без участия докладчика по заранее разработанному сценарию.

Power Point поддерживает многоэкранный режим докладчика, то есть при возможности подключения второго монитора или второго ноутбука можно вести показ сразу в двух режимах: полноэкранном и в «режиме докладчика» с использованием времени доклада и заметок.

Другие команды окна диалога ***Настройка презентации*** позволяют управлять речевым сопровождением, анимацией, показом слайдов.

Команда ***Настройка времени*** группы ***Настройка*** запускает презентацию в полноэкранном режиме и позволяет провести репетицию представления презентации с фиксацией времени представления каждого слайда. Эти данные можно сохранить для использования в дальнейшем в автоматическом режиме. Во время настройки программа запускается в режиме окна и появляется панель инструментов ***Запись*** (рис. 5.13). Для перехода к следующему кадру нажмите кнопку ***Далее*** (слева), для приостановки записи нажмите кнопку ***Пауза***. Кнопка ***Повторить*** (правая кнопка) позволяет повторить запись времени для текущего слайда. По окончании записи времени показа последнего слайда появится окно диалога, в котором будет указано общее время показа презентации и предложение сохранить данные настройки. Чтобы сохранить последние записанные временные интервалы, ответьте Да, чтобы удалить текущие временные интервалы, ответьте Нет. После этого программа открывается в режиме сортировщика слайдов, где у каждого слайда будет указано время его показа. Чтобы слайды автоматически сменялись через указанное время, необходимо установить флажок ***Использовать время показа слайдов*** в группе ***Настройка***.

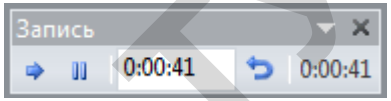


Рис. 5.13. Настройка времени показа слайдов

Кнопка ***Запись показа слайдов*** позволяет выбрать слайд, для которого ведется запись, и очистить речевое сопровождение и время записи выбранного слайда.

Произвольные показы

Произвольные показы позволяют создавать презентацию внутри другой презентации, группировать презентации, группировать отличающиеся кадры, присвоить им имя и переходить к ним во время показа слайдов. Для создания произвольного показа введите команду ***Произвольный показ***, в одноименном окне диалога выберите команду ***Создать*** – откроется окно диалога ***Задание произвольного показа*** (рис. 5.14), которое позволяет осуществлять отбор слайдов для произвольного показа из слайдов презентации. Кнопки Вверх, Вниз позволяют изменять последовательность слайдов.

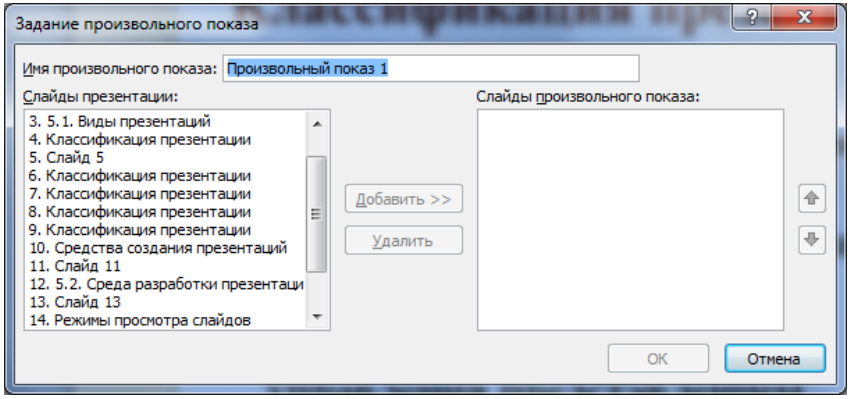


Рис. 5.14. Создание произвольного показа

Работа с отдельными файлами презентаций в разных окнах

Power Point 2010 позволяет запустить на одном мониторе одновременно несколько презентаций. При этом полностью поддерживаются эффекты анимации и мультимедиа. Используя вкладку ***Вид*** группы ***Окно***, можно установить режим одновременного показа на

экране всех открытых документов - команда ***Упорядочить все***. Все окна управляются независимо друг от друга, что позволяет теперь запустить и просматривать одновременно несколько презентаций, используя режим ***Чтение***.

Создание страниц заметок

Страницы заметок можно создавать непосредственно во время разработки каждого слайда. По окончании создания презентации страницы заметок можно оформить с помощью команды ***Вид, Образец заметок***. При печати слайдов в верхней части страницы выводится миниатюра слайда.

Раздаточные материалы

Раздаточные материалы – это не что иное, как распечатанные слайды. Подготовка раздаточного материала к печати осуществляется командой ***Вид, Образец выдач***. После ввода команды появляется вкладка ***Образец выдач*** (рис. 5.15). На этой вкладке можно установить параметры страницы, ориентацию выдач (книжная или альбомная), ориентацию слайда (также книжная или альбомная), число слайдов на странице, указать необходимость вывода верхнего и нижнего колонтитулов, даты и времени, изменить тему и фон.

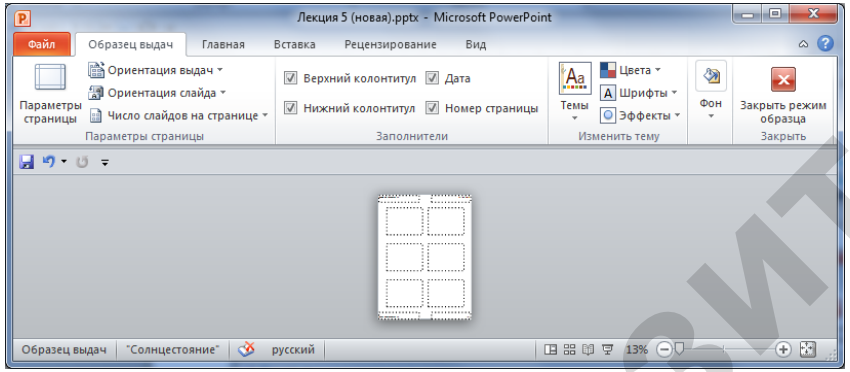


Рис. 5.15. Настройка образца выдач

Запись и синхронизация речевого сопровождения и движений указки

Речевое сопровождение можно записать до проведения презентации или непосредственно во время презентации. В последнем случае можно включить в запись комментарии и вопросы слушателей. Для того чтобы записать речевое сопровождение и синхронизировать его

с лазерной указкой, необходимо выполнить следующее. Введите команду ***Показ слайдов, Запись показа слайдов*** и установите флажок ***Речевое сопровождение и лазерная указка***, щелкните кнопку ***Начать запись***. Теперь можно записать речевое сопровождение. В лазерную указку превращается курсор мыши при нажатой левой клавише мыши и нажатой клавише Ctrl на клавиатуре.

Сохранение и использование презентаций

Сохранение презентации осуществляется командами ***Сохранить***, ***Сохранить как*** и ***Сохранить и отправить*** меню ***Файл***. Первые две команды сохраняют файлы на нашем компьютере. Команда ***Сохранить и отправить*** позволяет выполнить при сохранении некоторые преобразования и отправить файл по назначению.

В частности, при сохранении презентацию можно преобразовать в видеофайл – команда ***Создать видео*** (рис. 5.16). По умолчанию файл сохраняется в формате Windows Media Video (WMV). При этом могут быть сохранены и речевое сопровождение, и время записи. Если речевое сопровождение и движения лазерной указки не были синхронизированы и записаны, задайте при сохранении параметр *Не использовать записанное речевое сопровождение и время показа файлов*.

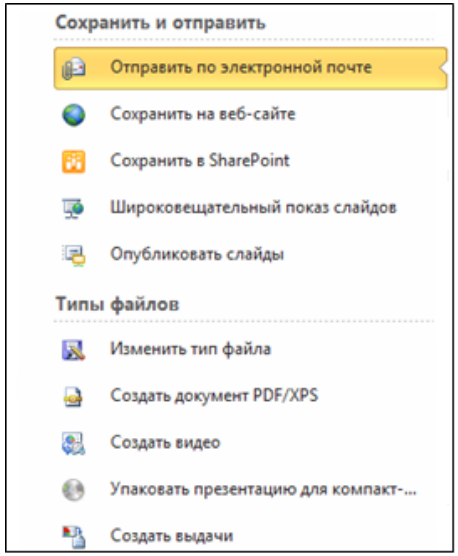


Рис. 5.16. Меню команды Сохранить и отправить

При отправке презентации по электронной почте рекомендуется использовать форматы .pdf и .xps. Достоинством этих форматов является то, что документы этих форматов одинаково выглядят на большинстве компьютеров, сохраняются шрифты, исходное форматирование и изображения, содержимое нельзя легко изменить.

Для совместного использования презентации ее необходимо сохранить на общедоступных сетевых ресурсах SharePoint.

Команда ***Широковещательный показ файлов*** позволяет транслировать презентацию в Интернете. Этой информацией смогут воспользоваться ваши партнеры, зарегистрированные в службе *Windows Live*.

Презентацию можно отправить как факс через Интернет без использования факсимильного аппарата. Однако для этого требуется поставщик службы факсов.

Дополнительные рекомендации по подготовке файлов презентаций к отправке содержатся в меню ***Файл, Сведения***. Это так называемое представление Backstage.

Контрольные вопросы

1. Что такое презентация, слайд?
2. Что такое раздаточный материал, конспект доклада?
3. Расскажите алгоритм создания презентации с использованием текстового документа.
4. Какие существуют режимы просмотра презентации?
5. Какие объекты могут использоваться для оформления презентации?
6. Какие эффекты можно использовать для улучшения восприятия материала?
7. Как подготовить раздаточный материал?
8. Как подготовить конспект доклада?
9. Как производится настройка параметров показа презентации?
10. Какие способы представления презентации вам известны, каковы их особенности?

6. ЭЛЕКТРОННАЯ ТАБЛИЦА EXCEL

Ключевые слова: Microsoft Excel, диаграмма, диапазон, книга, лист, списки, функция, электронная таблица, ячейка.

6.1. ОСНОВНЫЕ СВЕДЕНИЯ

Электронная таблица Excel – интегрированная система. Она предназначена для создания и обработки электронных таблиц, ведения экономических и инженерных расчетов, математического моделирования, создания и ведения однотабличных баз данных, представления результатов обработки таблиц и списков в виде диаграмм и графиков функций, подготовки выходных форм документов, сохранения их на дисках и вывода на печать, подготовки вебдокументов.

Для загрузки программы щелкните мышкой по значку приложения Microsoft Excel на рабочем столе или панели задач операционной системы Microsoft Windows 7. Для выхода из программы введите команду ***Файл, Выход***.

6.1.1. Описание рабочего окна Excel

Рабочее окно Microsoft Excel представляет собой стандартное окно Microsoft Office со специфическими элементами. В верхней части окна расположена строка заголовка, в которой располагаются кнопка системного меню, название приложения и имя файла, загруженного в окно (в начале работы по умолчанию выводится имя файла Книга1), кнопки свертывания и развертывания окна программы. Ниже расположены: кнопка меню Файл, лента, панель быстрого доступа, строка ввода данных, рабочее окно, строка состояния (рис. 6.1).

Лента имеет ту же структуру, что и лента Word 2010. Она имеет восемь постоянно открытых вкладок, ярлычки которых

видны на экране: Главная, Вставка, Разметка страницы, Формулы, Данные, Рецензирование, Вид и Надстройки. Каждая вкладка состоит из групп команд, в которых расположены кнопки управления. У большинства групп в правом нижнем углу имеется кнопка, которая вызывает окно диалога настройки параметров, известное пользователям по предыдущим версиям программы Microsoft Excel. Для увеличения рабочей области ленту можно свернуть соответствующей командой контекстного меню или комбинацией клавиш ***Ctrl + F1***.

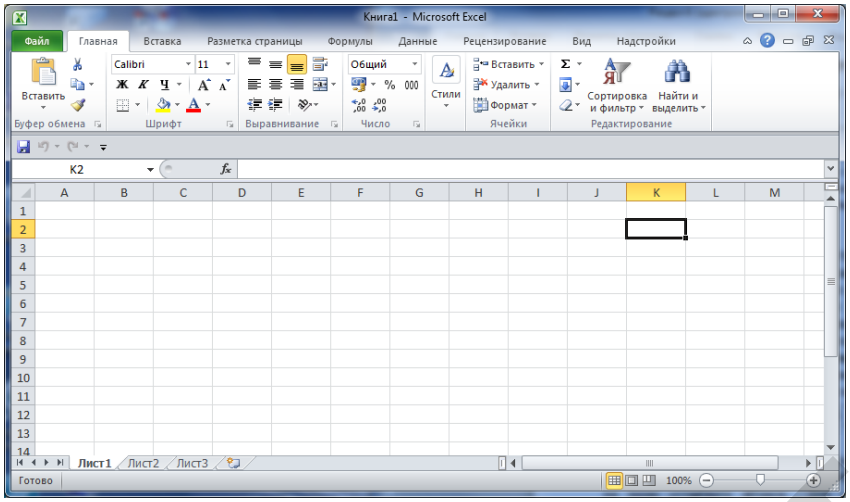


Рис. 6.1. Рабочее окно программы Excel

По умолчанию на экран не выводится одна важная вкладка ленты – Разработчик. Чтобы добавить ее на ленту, выполните следующее: введите команду ***Файл, Параметры, Настройка ленты***, в окне ***Основные вкладки*** установите флажок ***Разработчик*** и щелкните по кнопке ***ОК***.

Панель быстрого доступа содержит команды, используемые наиболее часто, состав этих команд может настраиваться пользователем. Порядок настройки панели быстрого доступа описан в разделе 4.1. Положение Панели быстрого доступа также может определить пользователь: над лентой или под лентой. Для изменения положения панели быстрого доступа воспользуйтесь контекстным меню.

Строка формул имеет три поля: *поле адреса ячейки (Имя)*, *поле управляющих клавиш* и *поле ввода данных (Строка формул)*. Поле *Имя* представляет собой раскрывающийся список, в нем указан адрес текущей ячейки или ее имя. Если щелкнуть мышкой по этому полю, ввести адрес ячейки и нажать клавишу Enter, то курсор электронной таблицы перейдет в указанную ячейку. Этот прием перехода к нужной ячейке называется *непосредственной адресацией*. В *поле управляющих кнопок* выводятся три кнопки: ☒ – отмена редактирования строки ввода, ☑ – окончание редактирования, *fx* – ввод функций. Названные кнопки появляются при активизации строки ввода. Поле *Ввод данных* предназначено для отображения вводимой информации или содержания выделенной ячейки. Для активизации строки ввода необходимо щелкнуть по ней мышью.

Строка состояния расположена в нижней части рабочего окна. В левой части строки состояния выводится информация о текущем состоянии электронной таблицы. В правой части расположены кнопки управления режимами просмотра таблицы: обычный, разметка страницы и страничный, а также элементы управления масштабом представления информации. Строка состояния имеет *контекстное меню*, которое позволяет управлять представлением информации в строке состояния.

Над строкой состояния размещены кнопки навигации, ярлычки листов, специальная кнопка ***Вставить лист*** и горизонтальная линейка прокрутки.

Рабочая книга, рабочий лист

Информация в электронной таблице сохраняется в виде *рабочих книг*. Имя книги выводится в строке заголовка. Рабочая книга состоит из листов различного типа. Максимально возможное число листов в рабочей книге – 256.

Рабочий лист состоит из пронумерованных строк и столбцов. Столбцы рабочих листов озаглавлены латинскими буквами от A до Z и их комбинациями по два и три символа, например, AA, AB, IU, …, XFD. Строки пронумерованы цифрами. Рабочий лист содержит 16 384 столбца и 1 048 576 строк. На пересечении строк и столбцов образованы ячейки. В одной из ячеек расположен *контур выделения* – ***курсор*** электронной таблицы. Рабочий лист имеет номер, который указан на ярлыке (рис. 6.1). Если щелкнуть правой кнопкой мыши по ярлыку, то откроется контекстное меню с перечнем команд для управления рабочим листом. Рабочие листы можно добавлять, удалять, копировать, переименовывать, перемещать, группировать, разгруппировывать.

Если требуется внести одинаковую информацию на несколько листов, то их можно сгруппировать. Для группировки листов нажмите клавишу Shift и щелкните мышью по ярлычкам группируемых листов. Чтобы разгруппировать рабочие листы, выделите их и выберите в контекстном меню команду Разгруппировать.

Ячейка и ее свойства

Ячейка является основным элементом таблицы.

Ячейка – область, образованная пересечением строки и столбца. Она обозначается номером столбца и строки, на пересечении которых находится. Например, A1, AV9999.

Диапазон (группа, блок) – непрерывная область ячеек, обозначенная номерами начальной и конечной ячеек, разделенными двоеточием или точкой, например, A1:C10, D8.H12. Ячейке или диапазону может быть присвоено уникальное имя.

Ячейка характеризуется следующими параметрами: адрес, содержание, значение, формат, статус.

Адрес ячейки

Адрес ячейки может быть абсолютным, относительным и смешанным.

Относительный адрес: A1, E7. Относительный адрес в операциях копирования автоматически настраивается.

Абсолютный адрес: $A$1, $E$7. Абсолютный адрес ячейки не меняется в операциях копирования, вставки или удаления ячеек, строк и столбцов.

Смешанный адрес: $A1, A$1. Если ячейке присвоен смешанный адрес, то при копировании будет меняться только тот параметр, перед которым не стоит знак $. Например: $D6 – при копировании ячейки будет меняться только номер строки; D$6 – при копировании будет меняться только адрес столбца.

Присвоение имени ячейке

Ячейке или диапазону ячеек может быть присвоено имя. В Excel 2010 для работы с именами создана отдельная группа – ***Определенные имена*** на вкладке ***Формулы***.

Присвоение имени осуществляется командой ***Присвоить Имя***. Для присвоения имени ячейке или диапазону ячеек необходимо:

1. Выделить ячейку (диапазон ячеек).
2. Ввести команду ***Формула, Присвоить имя***.
3. Выбрать область действия имен из списка Область: книга или лист. Ввести в строке ввода *Имя* диалогового окна *Создание имени* имя ячейки и щелкнуть по кнопке **ОК** (рис. 6.2). Область,

которой присваивается имя, отображается в строке *Диапазон*. В имени ячейки не допускаются пробелы. Чтобы сохранить смысловое содержание ячейки, рекомендуются следующие записи: *УдельноеСопротивление*, *МоментИнерции*, *СуммаПрибыли*, *НормаПрибыли* и т. д.

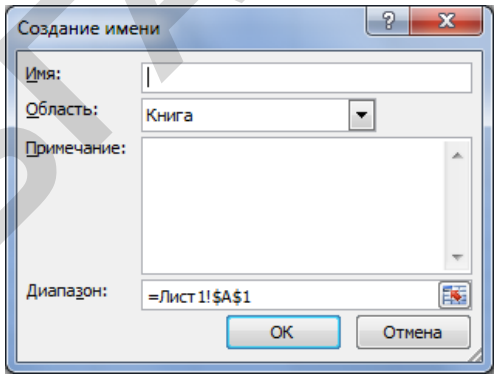


Рис. 6.2. Присваивание имен

Команда ***Диспетчер имен*** позволяет создавать имена, просматривать, удалять или изменять их.

Ячейке можно присвоить имя надписи, поясняющей содержание ячейки. Для использования надписей имеется команда ***Создание имен из выделенного диапазона***. Эта команда вызывает окно диалога, где можно указать место размещения поясняющей надписи по отношению к ячейке, которой присваивается имя: в строке выше, в столбце слева, в строке ниже или в строке справа.

В формулах можно ссылаться или на адрес ячейки, или на ее имя. Имя ячейки используется как *абсолютный адрес*. Для вставки имени ячейки в формулу служит команда ***Использовать в формуле***. Эта команда открывает список доступных имен.

Содержание ячейки

Содержание ячейки – это то, что вводится в нее через строку ввода. Поэтому ячейка может либо быть пустой, либо содержать данные: текст, текстовую константу, формулу, дату, время.

Значение ячейки

Значением ячейки могут быть число, текстовая константа, дата, время, сообщения об ошибках. Значением пустой ячейки и ячейки, содержащей текст, является ноль.

Число может быть представлено в виде целого числа (123), вещественного числа с десятичной точкой (0,0001785) или в экспоненциальной форме (1,785E–4). Дробная часть числа отделяется от целой части числа запятой. Этот параметр является настраиваемым, то есть в качестве разделителя целой и дробной частей числа может использоваться либо точка, либо запятая. Настройка этого параметра осуществляется в операционной системе Windows: ***Панель управления, Языки и региональные стандарты, Дополнительные параметры, Разделитель целой и дробной части***, выбрать из списка требуемый символ.

Текстовая константа – это строка символов, используемая в выражениях как операнд, при вводе текстовой константы она заключается в скобки и в кавычки, например, ("ноябрь").

Дата – значение функции дата.

Время – значение функции время.

***Сообщения об ошибках*:**

#ДЕЛ/0! – деление на ноль;

#ИМЯ? – не определено имя переменной в формуле;

#Н/Д! – нет допустимых значений, аргумент функции не может быть определен;

#ПУСТО! – итога не существует;

#ЧИСЛО! – избыточное число либо неверное использование числа, например, КОРЕНЬ(–1);

#ССЫЛКА! – неверная ссылка; ячейка, на которую она сделана, в рабочем листе не существует;

#ЗНАЧ! – неправильный тип аргумента; например, использование текста там, где необходимо число.

Если в формуле использовано одно из этих ошибочных значений, результат формулы также будет ошибочным. Ошибочные значения распространяются по всему рабочему листу, помечая все значения, зависящие от них, как некорректные. В этом случае достаточно найти и исправить ошибку, чтобы все остальные ячейки, связанные с ячейкой, содержащей ошибку, восстановили свое значение.

Формат ячейки

К формату ячейки относятся такие параметры ячейки, как ширина, режим отображения формул, формат отображения числовых величин, размещение содержимого ячейки, шрифт, цвет, границы, статус ячейки. Все основные параметры настройки свойств ячейки в Excel 2010 вынесены на вкладку Главная ленты (рис. 6.1).

Ширина ячейки может быть от 1 до 255 символов, по умолчанию – 9 символов, а высота – 409 точек. Длина записи для содержимого ячейки – до 32 767 символов, в том числе 1024 символа отображаются в ячейке, все символы отображаются в строке формул.

Режим отображения формул: формулы могут отображаться в виде формул или в виде значения. По умолчанию формулы отображаются в виде значения. Для перехода к режиму отображения формул необходимо ввести команду ***Показать формулы*** в группе ***Зависимости формул*** на вкладке ***Формулы***.

Формат отображения числовых величин. Числовые величины могут отображаться в виде целого (16, 154), вещественного числа (1,1755, 5,439), в показательной форме (1,45E–4), в денежном формате (345,32) или ($345,32), в процентном формате (35%). При представлении числа в процентном формате введенное число делится на сто. Изменение способа представления чисел, даты и времени осуществляется командами группы ***Число*** вкладки ***Главная*** (рис. 6.3), а также окна диалога ***Формат ячеек, Число***, вызываемого кнопками в правом нижнем углу групп Число, Шрифт или Выравнивание (рис. 6.4).

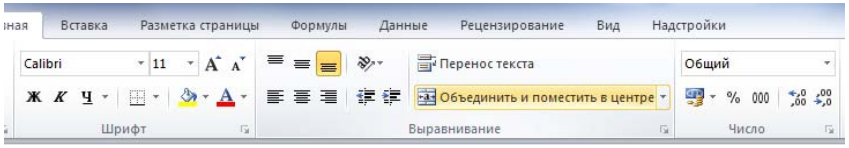


Рис. 6.3. Вкладка Главная ленты. Форматирование ячеек

Размещение содержимого ячейки. Содержимое ячейки может быть размещено справа, слева, по центру. По умолчанию текст прижимается к левому краю ячейки, числа – к правому краю ячейки. Текст может быть размещен горизонтально, вертикально или под определенным углом. Управление размещением содержимого ячейки осуществляется командами группы ***Выравнивание*** вкладки ***Главная*** и командами окна диалога ***Формат ячеек, Выравнивание*** (рис. 6.5). Важное значение имеет флажок ***Переносить по словам*** (то же команда ***Перенос текста*** группы ***Выравнивание***). При установке этого флажка текст будет автоматически переноситься в пределах установленной ширины столбца. Если флажок *Переносить по словам* сброшен, то вводимый текст располагается в одну

строку, и если соседняя ячейка заполнена, то на экране будет видна только часть текста, умещающаяся в ячейке.

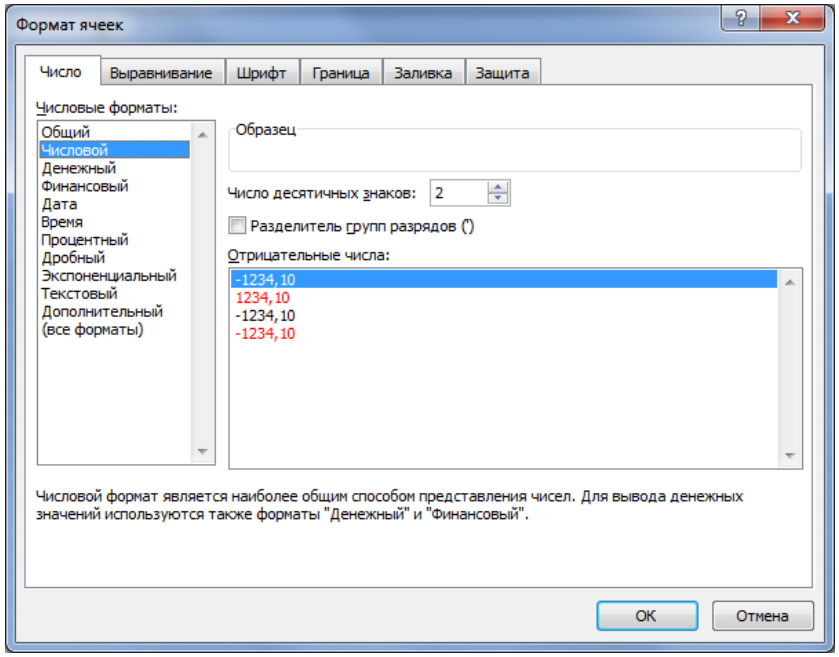


Рис. 6.4. Окно диалога Формат ячеек. Форматирование чисел

Шрифт. Параметры шрифта: начертание, стиль, цвет, интервал между символами, высота шрифта и другие эффекты, относящиеся к форматированию шрифта, устанавливаются с помощью команд группы Шрифт и окна диалога ***Формат ячеек, Шрифт***.

Границы ячейки и заливка. Стили обрамления и заполнения ячеек устанавливаются командами списков Границы и Заливка из группы Шрифт и командами вкладок ***Границы*** и ***Заливка*** окна диалога ***Формат ячеек*** соответственно.

Статус ячейки. Ячейка может иметь два статуса: защищена или не защищена. В защищенную ячейку нельзя внести информацию или изменить ее содержание. Установка режима защиты осуществляется командами вкладки ***Защита*** окна диалога ***Формат ячеек***. Эта команда позволяет установить защиту на ячейку или скрыть формулу. По умолчанию для всех ячеек установлен режим

защиты. Режим защиты ячейки вступает в силу только после защиты листа командами ***Защитить лист*** из группы ***Изменения*** вкладки ***Рецензирование***. При этом можно выборочно установить, какие элементы будут доступны пользователю. По умолчанию всем пользователям разрешено выделять заблокированные и незаблокированные ячейки. Поэтому, если необходимо обеспечить доступ к некоторым ячейкам, для этих ячеек надо снять защиту ячейки, а потом защитить лист.

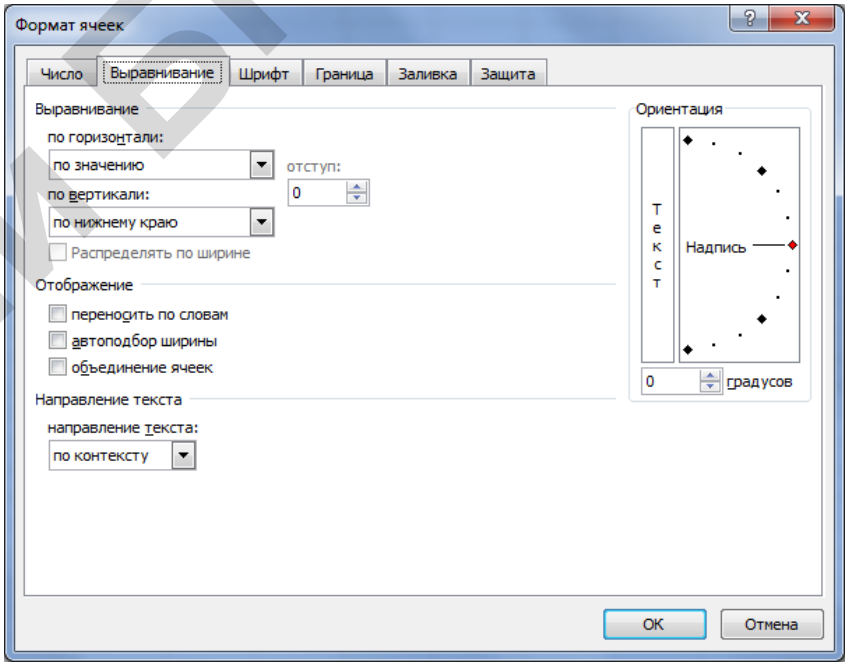


Рис. 6.5. Окно диалога Формат ячеек. Форматирование текста

Для отмены защиты ячейки достаточно отменить защиту листа командой ***Снять защиту листа***.

Можно защитить также структуру книги. В этом случае будет запрещено вносить изменения в структуру книги: перемещение, удаление или добавление листов.

Команда ***Доступ к книге*** позволяет вести одновременную работу с книгой нескольким пользователям через Интернет. Команда ***Защитить книгу и дать общий доступ*** обеспечивает доступ

к книге пользователей с одновременной защитой книги паролем. Команда ***Разрешить изменение диапазонов*** разрешает изменение некоторых диапазонов защищенной книги некоторым лицам.

6.1.2. Курсор таблицы

Курсор таблицы, или контур выделения, представляет собой рамку, окаймляющую всю ячейку (рис. 6.1). В правом нижнем углу рамки на пересечении сторон располагается маленький квадрат – ***маркер заполнения***. Этот маркер используется для заполнения ячеек рядом данных с постоянным шагом, а также для копирования формул. Для перемещения курсора используются клавиши управления перемещением курсора, а также клавиши Home – перейти в первую ячейку строки, [Ctrl + Home] – перейти в ячейку A1, [End + клавиши управления перемещением курсора] – последняя занятая ячейка в соответствующем направлении, используются также клавиши прокрутки и ряд других комбинаций клавиш. Непосредственная адресация осуществляется вводом адреса ячейки в поле «Адрес ячейки» Строки ввода данных.

6.1.3. Ввод данных

Данные вводятся в Строку ввода данных или непосредственно в ячейку. В первом случае выделите ячейку, в которую вводятся данные, и щелкните по Строке ввода данных. Введите нужную информацию. Для окончания ввода нажмите клавишу Enter или щелкните кнопку ✓ Строки ввода данных. Во втором случае выделите ячейку и вводите данные прямо в ячейку. По окончании ввода данных нажмите клавишу Enter.

При редактировании строки ввода используются клавиши:

1. ←, → – сдвиг курсора строки ввода на один символ в соответствующем направлении.
2. *Insert* – включение режима вставки символов.
3. *Delete* – удаление символа в позиции курсора.
4. *Home, End, Tab* – переход в начало или конец текста.
5. *SpaceBar* (ПРОБЕЛ) – сдвиг вправо с удалением символов или без удаления символов, в зависимости от режима Вставка/Замена.
6. *BackSpace* (ВОЗВРАТ НА ШАГ) – удаление символа слева от курсора.
7. *Esc* – удаление вводимого текста.

Для очистки ячейки выделите ее и нажмите клавишу Delete или Пробел и Enter. Очистить ячейку можно командой ***Очистить*** из группы ***Редактирование*** вкладки ***Главная***. После ввода этой команды откроется дополнительное меню с запросом, что очищать: *Все*, *Форматы*, *Содержимое*, *Примечания* или *Гиперссылки*.

Ввод текста. Признаком текста при вводе данных является апостроф ('), например: 'Сводная ведомость, 'Электронная таблица. По умолчанию вводимые данные воспринимаются как текст.

Ввод даты. Дата вводится в формате ДД.ММ.ГГ или ДД.ММ.ГГГГ: день, месяц, год (17.05.99). В качестве разделителя используется точка. Электронная таблица позволяет выводить дату на экран в различных форматах.

Ввод текстовых констант. Для ввода текстовых констант необходимо ввести символ = и текст в кавычках: =”Текст”. Для преобразования чисел или числовых значений выражений в текстовые константы следует воспользоваться функцией ТЕКСТ().

Тип данных в ячейке определяется при первом вводе. Для изменения формата ячейки следует воспользоваться окном диалога ***Формат ячеек*** (рис. 6.4, 6.5). Например, если в ячейку введена дата, то для ввода числа необходимо изменить формат ячейки:

вызовите окно диалога ***Формат ячеек*** кнопкой в правом нижнем углу группы ***Число***;

выберите вкладку ***Число***;

выберите в окне «***Числовые форматы***» тип «***Числовой***».

Ввод формул. Признаком формулы является знак =. Если при вводе формулы допущена ошибка, то программа выдает сообщение об ошибке. При вводе формулы без знака «равно» программа воспринимает вводимые данные как текст. Адреса ячеек вводятся только латинскими символами.

> ***Совет****: чтобы избежать ошибок при записи адресов ячеек, выбирайте их мышью. Установите курсор в точку ввода Строки ввода данных и щелкните мышью по ячейке, адрес которой надо вставить в формулу.*

При вводе вещественных чисел, в отличие от других языков программирования высокого уровня, используется десятичная запятая, а не точка.

Для ввода формул или ознакомления с функциями Excel можно использовать Мастера функций. Для этого необходимо щелкнуть кнопку f_x в стандартной панели инструментов или воспользоваться вкладкой ***Формулы*** ленты.

Примеры записи формул:
=A2+2 - сложение;
=24–12 – вычитание;
=F35/B7*$A$2 – делит значение ячейки F35 на значение ячейки B7 и умножает на значение ячейки A2. У ячейки A2 указан абсолютный адрес;
=СТАВКА*МЕСЯЦ – перемножаются значения, содержащиеся в ячейках с именами СТАВКА и МЕСЯЦ;
=ЕСЛИ(A2<B2;C3;D2*E17) – условное выражение. Если значение в ячейке A2 меньше значения в ячейке B2, то результат будет равен значению ячейки C3, иначе – произведению значений ячеек D2 и E17.

При записи формул, для указания адреса ячеек, значения которых не должны изменяться при копировании формул, обязательно используйте абсолютный адрес.

Для ускорения ввода признака абсолютного адреса – символа $ – можно воспользоваться функциональной клавишей **F4**: установите курсор строки ввода в любом месте адреса ячейки и нажмите клавишу F4.

Редактирование содержимого ячейки

Для редактирования содержимого ячейки необходимо выделить ее, при этом содержимое ячейки отображается в Строке ввода данных. Щелкните мышкой по Строке ввода данных и вносите необходимые изменения. Для окончания редактирования данных нажмите клавишу Enter или щелкните по кнопке ✓ строки ввода данных.

6.1.4. Приемы работы с электронной таблицей

- Выбор ячейки. Установите курсор на ячейку или щелкните по ней мышью.
- Выбор группы ячеек. Установите указатель мыши на первую ячейку группы (области), нажмите левую клавишу и протащите указатель по всем ячейкам группы. В конце области выделения отпустите клавишу мыши. Эту же операцию можно выполнить с помощью клавиш управления перемещением курсора при нажатой клавише Shift.
- Для выбора нескольких несвязанных областей необходимо нажать и удерживать клавишу Ctrl, а затем выделить требуемые области.

- Для отмены выделения ячеек щелкните мышью по чистому полю в любом месте экрана.
- Выбор строк и столбцов. Для выбора одной строки (столбца) щелкните мышью по номеру строки. Для выбора группы строк (столбцов) установите указатель мыши на номер первой строки, нажмите клавишу мыши и протащите ее указатель по всем строкам выделяемой группы. Отпустите клавишу мыши.
- Изменение размеров строк и столбцов. Для изменения высоты (ширины) строки (столбца) зацепите границу строки, т. е. подведите указатель мыши к границе строки так, чтобы он изменил свою форму на двунаправленную стрелку, нажмите левую клавишу мыши и протащите указатель мыши в нужном направлении до установки требуемых размеров строки. При уменьшении размеров до нуля строки будут спрятаны.
- Управление параметрами ячеек: вставка/удаление строк, столбцов, скрытие, восстановление строк, столбцов, установка ширины столбцов и высоты строк, управление листами осуществляется с помощью команд ***Вставить***, ***Удалить***, ***Формат*** группы ***Ячейки*** вкладки ***Главная***. Например, для восстановления спрятанных столбцов выделите столбцы слева и справа от спрятанных и введите команду ***Формат, Скрыть или отобразить, Отобразить столбцы***. Чтобы восстановить спрятанные строки, выделите строки сверху и снизу от спрятанных строк и введите команду ***Формат, Скрыть или отобразить, Отобразить строки***. Для быстрой установки ширины столбцов или высоты строк служат команды ***Автоподбор ширины*** и ***Автоподбор высоты*** меню ***Формат***.

Для вставки требуемого количества строк (столбцов) выделите столько строк, сколько нужно вставить, и выберите команду ***Вставить, Вставить строки на лист***. Для удаления строк (столбцов) выделите нужные строки и выберите команду ***Удалить, Удалить строки с листа***. При вставке (удалении) ячеек необходимо дополнительно указать способ освобождения (заполнения) места для ячейки: со сдвигом ячеек вниз (вверх) или вправо (влево).

- Форматирование ячеек. Для изменения формата ячейки (группы ячеек) выделите ячейку, выберите команду ***Формат, Формат ячейки***, в окне диалога Формат ячеек щелкните нужную вкладку и установите требуемые параметры. Для окончания работы щелкните кнопку ***ОК***. Форматировать ячейки можно как до, так и после ввода данных. С целью экономии времени применяйте формат сразу к группе ячеек после ввода данных.

Копирование ячеек

При копировании ячеек можно использовать команды ***Копировать*** и ***Вставить*** из группы ***Буфер обмена*** вкладки ***Главная*** ленты, команды контекстного меню.

Копирование с использованием маркера автозаполнения

Если копирование осуществляется в соседние ячейки, то его удобно выполнить с использованием маркера автозаполнения:

выделите копируемую ячейку;

зацепите мышью за маркер автозаполнения (подведите курсор к черному квадратику в правом нижнем углу курсора таблицы так, чтобы указатель мыши превратился в черный крестик) и протащите указатель мыши по всем ячейкам назначения.

Копирование с помощью команд группы Буфер обмена или контекстного меню

Для копирования с помощью команд группы Буфер обмена необходимо:

выделить ячейку или блок ячеек, подлежащих копированию;

выбрать команду Копировать. Вокруг выделенной ячейки появляется мерцающий контур;

выделить начальную ячейку, куда будут копироваться данные;

выбрать команду Вставить;

выбрать подходящую команду из меню (рис. 6.6). Содержание меню зависит от того, что копируется: текст, рисунок или формулы;

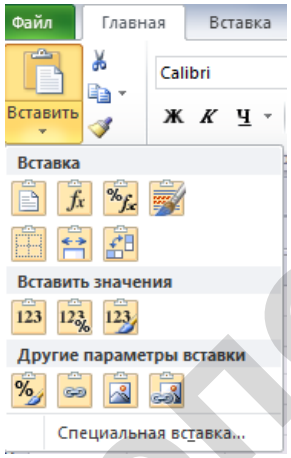


Рис. 6.6. Меню команды Вставить

отменить выделение блока. Чтобы удалить мерцающий контур у копируемых ячеек, щелкните мышью по Строке ввода данных или нажмите клавишу Esc.

Копирование с помощью мыши

Для копирования содержимого ячеек, содержащих текстовую информацию или формулы с абсолютными адресами данных, выделите с помощью мыши ячейку или блок ячеек, подлежащих копированию, нажмите и удерживайте клавишу Ctrl, перетащите выделенные ячейки на новое место, отпустите кнопку мыши, а затем отпустите кнопку Ctrl.

Для копирования формул, содержащих ссылки на ячейки с относительными или смешанными адресами, следует воспользоваться командой ***Вставить, Специальная вставка***:

выделите копируемые ячейки;

укажите место вставки;

введите команду ***Вставить, Специальная вставка*** из группы ***Буфер обмена*** и щелкните по кнопке ***Вставить связь***.

В этом случае в формуле сохраняются ссылки на ячейки, содержащие данные. При изменении данных в ячейках автоматически будут меняться значения в ячейках источника данных и в ячейках назначения. При использовании команды Специальная вставка можно выполнять арифметические операции сложения, вычитания, умножения и деления с данными источника данных и данными ячейки назначения, можно копировать только значения или формулы (рис. 6.7).

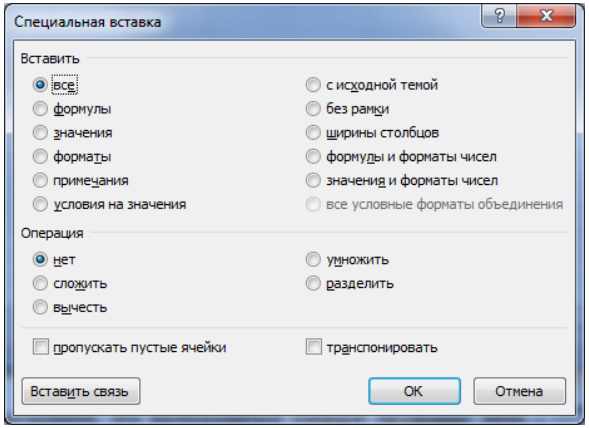


Рис. 6.7. Окно диалога Специальная вставка

6.1.5. Оформление таблицы

Для оформления таблицы: обрамления, заливки цветом – можно воспользоваться вкладками ***Граница*** и ***Заливка*** окна диалога ***Формат ячеек*** или одноименными кнопками в группе ***Шрифт*** вкладки ***Главная***. Можно воспользоваться также командами ***Форматировать как таблицу*** и ***Стили ячеек*** из группы ***Стили*** вкладки ***Главная***.

Для оформления таблицы с помощью команды ***Форматировать как таблицу*** выполните следующее:

выделите таблицу и введите команду ***Форматировать как таблицу***;

в окне ***Стили*** выберите нужный стиль оформления;

в окне диалога ***Форматирование таблицы*** (рис. 6.8) укажите область, подлежащую форматированию. Если таблица содержит заголовки, установите флажок Таблица с заголовками.

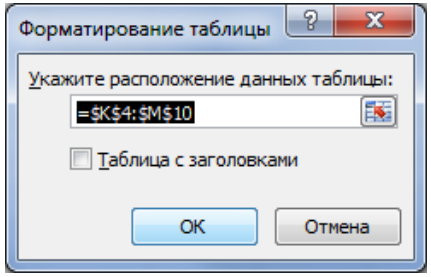


Рис. 6.8. Форматирование таблицы

Если на вкладке ***Работа с таблицами*** (Конструктор) в группе ***Параметры стилей таблиц*** установлен флажок ***Строка заголовка***, то над таблицей (или в строке заголовков, когда установлен флажок Таблица с заголовками в окне диалога Форматирование таблицы) появляется список (рис. 6.9). Эти списки содержат команды сортировки и фильтрации. В группе Параметры стилей таблиц содержатся и другие флажки, определяющие стиль оформления таблицы. Чтобы удалить эти списки, снимите флажок ***Строка заголовка*** в группе ***Параметры стилей таблиц***.

Условное форматирование

Часто бывает необходимо выделить какие-то результаты, чтобы обратить на них внимание: минимальные запасы сырья и материалов, предельно допустимые значения параметров среды обитания и др. В этом случае на помощь приходит ***Условное форматирование***. Команда Условное форматирование предоставляет пользователю большой набор вариантов выделения параметров, необходимых для контроля: отношение предпочтения больше, меньше и др., правила отбора первых и последних, гистограммы, цветовые шкалы, значки. Кроме того, пользователь может создать собственное правило выделения (рис. 6.10).

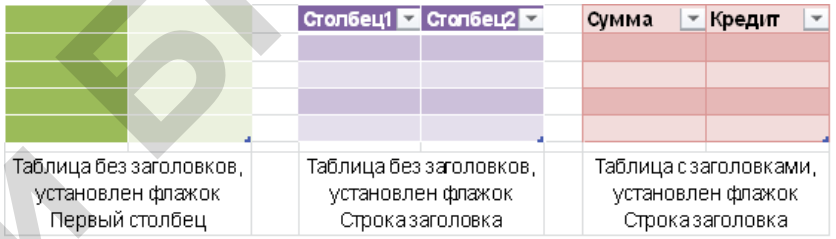


Рис. 6.9. Примеры оформления таблиц

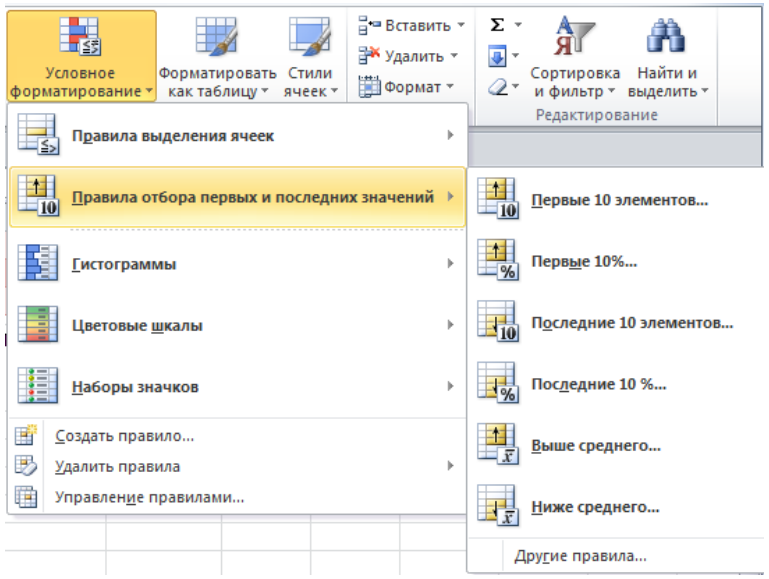


Рис. 6.10. Условное форматирование

Рассмотрим пример. Пусть в таблице учета материалов (рис. 6.11) нужно контролировать норму запаса материальных средств, чтобы вовремя сделать заказ на поставку материалов. Выделим ячейки, значения в которых необходимо контролировать, и введем команду ***Условное форматирование***, ***Создать правило***. Открывается окно диалога ***Создание правил форматирования*** (рис. 6.12). Выберем один из вариантов оформления, например, ***Форматировать только ячейки, которые содержат***. Окно диалога меняет свою форму (рис. 6.13). В первом списке окна *Измените описание правил* выберем контролируемый параметр, например, Значение ячейки. Во втором списке выберем знак отношения – меньше (при выборе «между» необходимо будет указать два параметра), в третьем окне укажем число 10 (или ссылку на ячейку, содержащую значение для сравнения). Для выбранной области установим форматирование: щелкнем по кнопке ***Формат*** – откроется окно диалога ***Формат ячеек***. Выберем в этом окне стиль форматирования контролируемого параметра. Если результаты будут просматриваться на экране, можно закрасить контролируемые ячейки цветом. Если результаты предполагается выводить на печать, можно применить форматирование шрифта: выделение курсивом, полужирным шрифтом, тип шрифта, перечеркивание, подчеркивание и т. д. Выберем выделение цветом. Результат условного форматирования приведен на рисунке 6.11. Условное форматирование можно применять к области данных или к одной ячейке. К ячейкам можно применить несколько форматов при различных значениях ячеек, применяя форматирование последовательно к выбранным ячейкам. Например, ячейки можно закрашивать желтым цветом, если запасы близки к критическому уровню, и красным цветом, когда запасы снизятся ниже критического значения.

Наименование	Наличие запасных частей
Втулка	10
Поршень	15
Карбюратор	5
Тормозные накладки	10
Коробка передач в сборе	2

Рис. 6.11. Пример условного форматирования

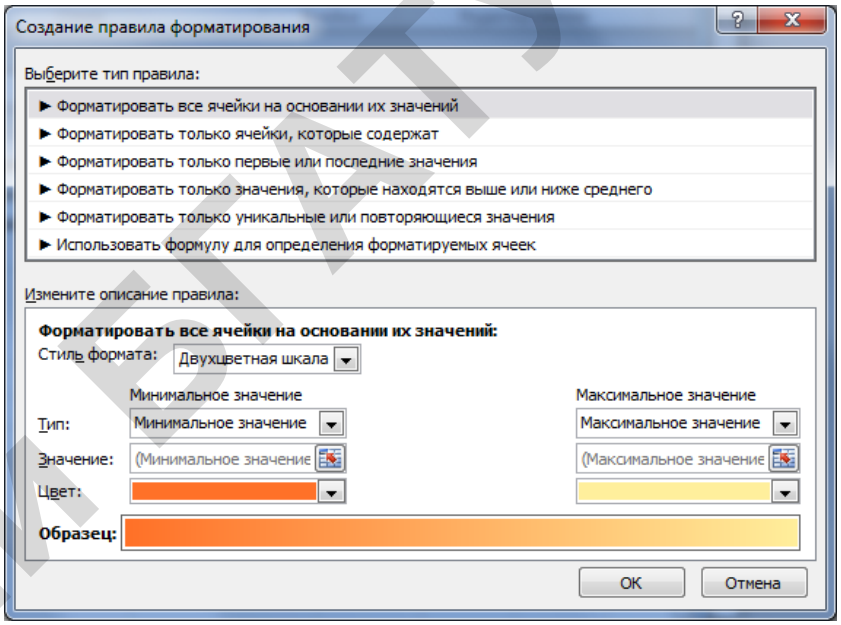


Рис. 6.12. Окно диалога Создание правил форматирования

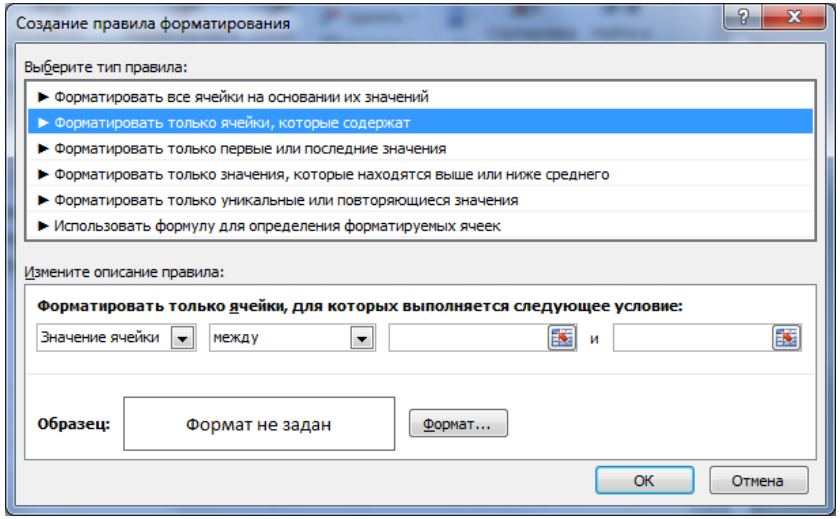


Рис. 6.13. Создание правила форматирования ячеек

При необходимости правила можно удалить для выделенных ячеек либо для всего листа командой ***Удалить правила***. Правилами можно управлять с помощью команды ***Управление правилами***. При выборе этой команды откроется окно ***Диспетчера правил условного форматирования*** (рис. 6.14).

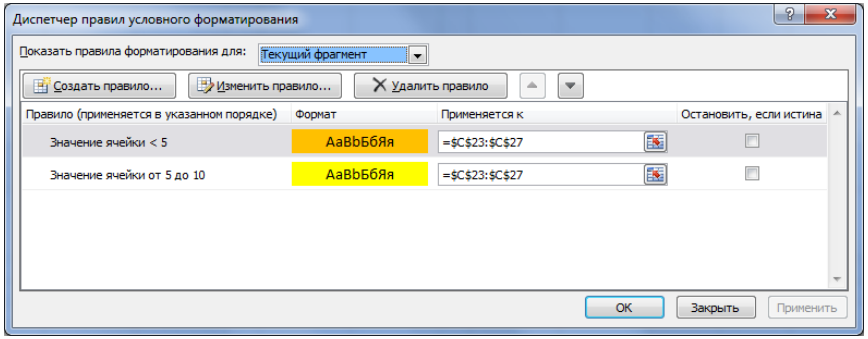


Рис. 6.14. Управление правилами

6.1.6. Управление окнами

Электронная таблица имеет многооконный интерфейс, следовательно, можно открыть несколько книг и работать с ними одновременно. Для управления окнами имеется группа ***Окна*** вкладки ***Вид*** ленты (рис. 6.15). Группа имеет ряд интересных и полезных команд. Команда ***Новое окно*** позволяет открыть текущий документ в новом окне. А команда ***Упорядочить все*** позволяет размещать окна удобным образом (рядом, сверху вниз, слева направо, каскадом, только окна текущей книги). Команда ***Разделить*** позволяет разделить текущее окно на две или четыре части и просматривать в каждом из них определенную часть текущего документа.

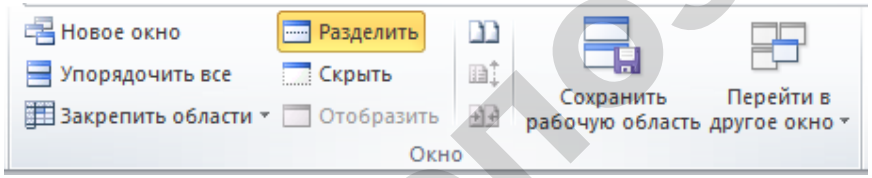


Рис. 6.15. Группа Окно вкладки Вид

Команда ***Закрепить области*** позволяет зафиксировать шапку таблицы, левую колонку или и то, и другое вместе. При работе с таблицами, содержащими большое число строк и столбцов, возникают затруднения, так как исчезает либо шапка таблицы, либо левая колонка. Для фиксации шапки таблицы подведите ее к верхнему краю таблицы и введите команду ***Закрепить области, Закрепить верхнюю строку***. Для фиксации левой колонки подведите левую колонку таблицы к левому краю экрана и введите команду ***Закрепить области, Закрепить первый столбец***. Для одновременной фиксации и шапки таблицы, и левой колонки выделите ячейку снизу и справа от фиксируемых строк и колонок и введите команду ***Закрепить области, Закрепить области***. Для отмены фиксации подокон выберите команду ***Закрепить области, Снять закрепление областей***.

Кнопка ***Скрыть*** скрывает текущее окно, а кнопка ***Отобразить*** восстанавливает закрытое окно.

В центре группы Окно расположены три кнопки для сравнения документов. Кнопка ***Рядом*** располагает рядом два листа для сравнения их содержимого. Кнопка ***Синхронная прокрутка*** обеспечивает одновременную прокрутку сравниваемых документов. Кнопка ***Восстановить расположение окна*** обеспечивает выделение каждому из сравниваемых документов половину рабочего окна.

Назначение других кнопок понятно из их названий.

6.1.7. Предварительный просмотр. Настройка параметров страниц

Перед печатью необходимо убедиться, что данные размещены на листе аккуратно, не выходят за границы листа. Возможно, потребуется каким-то образом изменить размещение материала на странице, уплотнить текст, чтобы сократить число страниц. Для этой цели в меню Файл имеется команда Предварительный просмотр, одноименная кнопка имеется и на панели инструментов Стандартная. Для просмотра таблицы введите команду ***Разметка страницы*** на вкладке ***Режимы просмотра книги*** вкладки ***Вид***. В этом режиме лист представляется в том виде, как будет выглядеть при печати, можно просмотреть начало и конец страниц, верхний и нижний колонтитулы, на экран выводятся горизонтальная и вертикальная линейки.

Для изменения параметров страницы откройте вкладку ***Разметка страницы*** и обратитесь к группе ***Параметры страницы*** (рис. 6.16).

Команды этой вкладки позволяют установить размеры страницы, выбрать ориентацию страницы (книжная или альбомная), задать/убрать область печати (диапазон ячеек, выводимый на печать); установить размеры полей, вставить разрывы страниц, добавить фоновый рисунок – команда Подложка, установить режимы печати заголовков. При выборе команды ***Печать заголовков*** открывается окно диалога ***Параметры страницы*** (рис. 6.17). Это же окно вызывается кнопкой в правом нижнем углу группы Параметры страницы. Окно диалога имеет несколько вкладок.

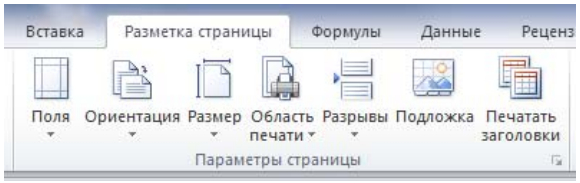


Рис. 6.16. Группа параметры страницы вкладки Разметка страницы

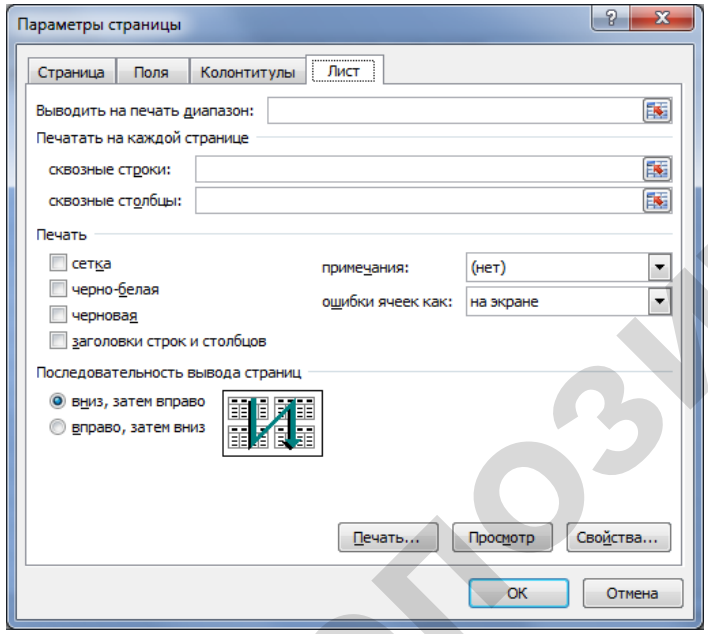


Рис. 6.17. Окно диалога Параметры страницы

Вкладка ***Лист***, представленная на рисунке 6.17, позволяет указать диапазон таблицы, выводимый на печать, строки и столбцы, которые следует печатать на каждой странице; управлять выводом на печать заголовков строк и столбцов таблицы, сетки, примечаний, ошибок; устанавливать порядок вывода страниц при печати. Не выходя из этого режима, можно просмотреть, как будет выглядеть страница при печати и, наконец, напечатать документ.

Вкладка ***Страница*** позволяет выбрать размер бумаги, качество печати, изменить ориентацию страницы при печати, масштаб, формат листа и др. Страница может иметь два способа ориентации: книжный формат и альбом. При книжном формате печать страницы идет поперек листа, при ориентации альбом – вдоль листа. В обоих случаях лист загружается в принтер одинаково – узкой стороной. Переключатель «Разместить не более чем на» позволяет автоматически выполнить масштабирование документа на указанное число страниц по ширине и высоте.

Вкладка ***Поля*** позволяет установить ширину левого и правого полей, высоту верхнего и нижнего полей и высоту поля для колонтитулов, а также центрирование текста документа на странице.

Вкладка ***Колонтитулы*** позволяет создать верхний и нижний колонтитулы или выбрать их из списков, а также управлять ими.

После выхода из режима просмотра в таблице пунктирными линиями будут отмечены границы страниц.

6.1.8. Сохранение, открытие и печать таблицы

Перед печатью целесообразно сохранить документ на диске, для этого необходимо выполнить следующее: введите команду ***Сохранить*** или ***Сохранить как***... из меню ***Файл***, укажите в строке ввода «Имя Файла» имя файла, при необходимости измените тип файла .xlsx на .xls для совместимости с предыдущими версиями программы Excel. Выберите диск и папку, щелкните по кнопке ОК.

Открытие сохраненного ранее документа осуществляется командой ***Файл, Открыть***. Выберите диск, папку. Если в списке не видите файл, который, по Вашему мнению, должен быть в этой папке, обратите внимание на тип файла, при необходимости измените тип файла. Программа хранит по умолчанию список последних 25 файлов, с которыми Вы работали. Поэтому для открытия файлов можно воспользоваться командой ***Файл, Последние*** и поискать нужные файлы в списке файлов или в списке папок.

Для вывода таблицы на печать выберите команду ***Печать*** из меню ***Файл***, укажите, что печатать (активные листы, всю книгу, выделенный фрагмент), ***Число копий*** и щелкните кнопку ***Печать***.

6.1.9. Настройка параметров таблицы

Приступая к работе с электронной таблицей, полезно ознакомиться с некоторыми настройками. Настройка параметров электронной таблицы осуществляется командой ***Файл, Параметры***. После ввода команды открывается окно диалога ***Параметры Excel*** (рис. 6.18). Команды собраны в группы.

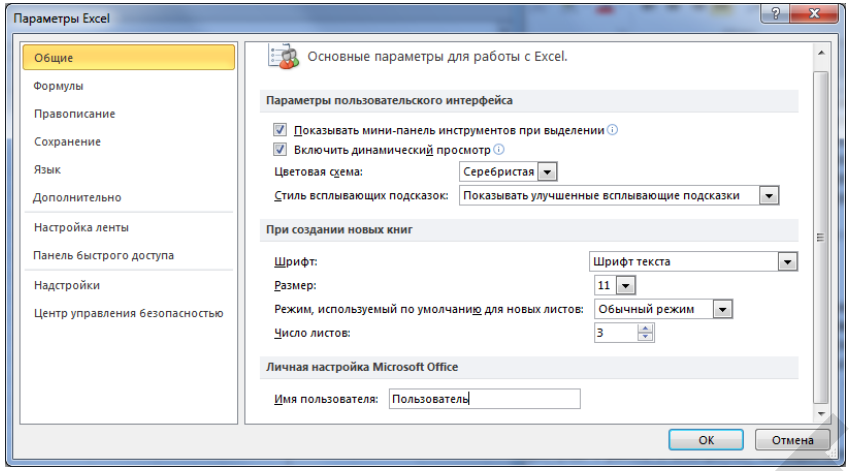


Рис. 6.18. Рабочее окно программы Excel

В группе ***Язык*** выберите язык, используемый по умолчанию при редактировании, в интерфейсе, при выводе справок.

В группе ***Сохранение*** установите интервал времени, через который рабочая книга будет сохраняться. По умолчанию установлен интервал 10 минут. Здесь же установите тип расширения файла при сохранении. Для обеспечения совместимости с предыдущими версиями электронной таблицы установите тип файла Книга Excel 97–2003 (*.xls). По умолчанию установлен тип файла Книга Excel (*.xlsx). Можно также изменить месторасположение файлов при сохранении, указав рабочий диск и каталог.

Группа ***Формулы*** содержит команды настройки параметров вычислений. По умолчанию установлен режим автоматических

вычислений. Это значит, что вычисления и пересчет значений ячеек производятся автоматически сразу же после ввода данных и нажатия клавиши Enter. В разделе Работа с формулами можно изменить стиль ссылок на ячейки R1C1 (R – Номер_Строки, C – Номер_Столбца). Этот стиль ссылок на номера ячеек применяется иногда при программировании. По умолчанию установлен стиль ссылок A1, то есть в адресе ячейки (ссылке) указывается номер столбца и номер строки.

Настройка ленты. Нет надобности без нужды настраивать ленту. Тем не менее, иногда в этом может возникнуть необходимость. Например, Вы не можете найти команду загрузки редактора VBA. Добавлять команды можно только в группы, созданные пользователем. Для добавления команды загрузки редактора VBA выполните следующие операции:

- Введите команду ***Настройка ленты***: откроется одноименное окно диалога (рис. 6.19).

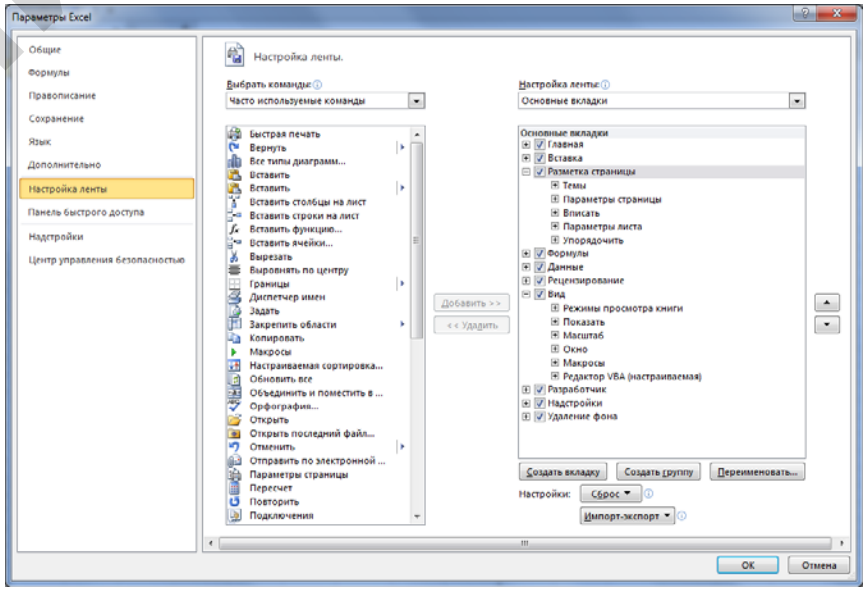


Рис. 6.19. Окно диалога Настройка ленты

- В правом списке ***Основные вкладки*** найдите Вкладку ***Разработчик*** и установите возле нее флажок.

При необходимости можно создать новую группу и установить на нее нужную Вам команду. Например:

- В правом списке ***Основные вкладки*** найдите вкладку ***Вид*** и разверните ее щелчком по кнопке +.
- Создайте новую группу кнопкой ***Создать группу***. Группа добавляется в активную вкладку. Переименуйте ее, присвоив имя, например, «Редактор VBA».
- В левом списке ***Выбрать команды*** выберите ***Все команды***.
- Найдите в списке команд команду ***Visual Basic*** и щелкните по кнопке ***Добавить***.
- Закройте окно, щелкнув по кнопке OK.

Команда ***Надстройки*** тоже может понадобиться нам, когда потребуются команды Пакета анализа. Для добавления команд на вкладку ***Данные, Анализ*** выполните следующие действия:

- Введите команду ***Надстройки*** и в окне диалога ***Управление надстройками Microsoft Office*** откройте список ***Управление***.
- Выберите в списке ***Надстройки Excel*** и выберите команду ***Перейти***.
- В окне диалога ***Надстройки*** установите флажки у тех надстроек, которые Вам нужны.

Центр управления безопасностью. К нему нам придется обратиться, если возникнут проблемы с выполнением макросов. Введите команду ***Центр управления безопасностью, Параметры центра управления безопасностью, Параметры макросов***. По умолчанию установлен режим ***Отключить все макросы с уведомлением***. Поэтому при запуске программы, содержащей макросы, на экран будет выводиться предупреждение. Если вы доверяете этим макросам, щелкните по кнопке ***Включить содержимое*** в строке предупреждения системы безопасности.

Контрольные вопросы

1. Как изменить ширину строки (столбца)?
2. Как вставить строку, группу строк (столбцов)?
3. Каким образом устанавливается формат ячейки?
4. Что такое Строка формул, какие поля она имеет?
5. Где можно увидеть адрес текущей ячейки?
6. Как перейти к новой ячейке?
7. Что такое абсолютный, смешанный и относительный адреса ячейки, как они записываются?
8. Как выполняется копирование ячеек?
9. Как ввести в ячейку текст, формулу?

10. Как сохранить таблицу на диске?
11. Как загрузить таблицу с диска?
12. Как вывести таблицу на печать?
13. Каким образом осуществляется фиксация шапки таблицы?

Заключение

В настоящем разделе мы ознакомились с возможностями электронной таблицы и основными приемами работы в ее среде. Единицей учета в Excel является Книга. Книга сохраняется в файле с расширением .xlsx. Для совместимости с более ранними версиями электронной таблицы необходимо изменить тип файлов, используемых по умолчанию, на Книга Excel 97–2003, расширение имени файла .xls.

Книга состоит из листов. Основным элементом листа является ячейка. Ввод информации в ячейку осуществляется через строку Формул. Каждая ячейка рассматривается программой как самостоятельный документ, к которому применяются все элементы форматирования. Все команды управления и форматирования сосредоточены в меню Файл и на ленте. Лента имеет большое число вкладок. Каждая вкладка имеет ярлычок. Основных вкладок десять, ярлычки этих вкладок образуют своеобразное главное меню. Другие вкладки появляются автоматически при вводе определенных команд или работе с определенными объектами.

Для успешной работы необходимо выполнить некоторые настройки электронной таблицы, выполняемые с помощью команды Параметры из меню Файл.

6.2. РАЗРАБОТКА ЭЛЕКТРОННЫХ ТАБЛИЦ

6.2.1. Типы полей электронной таблицы

Результаты вычислений в Excel независимо от ее назначения чаще всего представляются в виде двумерных таблиц. Таблица обычно имеет четыре поля (рис. 6.20):

<table>
<tr><td colspan="4">1/1</td></tr>
<tr><td>1/2</td><td>2</td><td>3</td><td>4/2</td></tr>
<tr><td></td><td colspan="2">4/1</td><td></td></tr>
</table>

Рис. 6.20. Размещение полей электронной таблицы

1 – поле описания задачи, состоящее из клеток с текстовой информацией, отражающей наименование и назначение ЭТ; глобальные параметры, используемые при вычислениях в ячейках таблицы; описание строк и столбцов (левая и верхняя графы таблицы);

2 – поле исходных данных, содержащее ячейки с числовой информацией, не изменяющейся в процессе расчета таблицы;

3 – поле расчетных формул, содержащее формулы для выполнения промежуточных вычислений. Операндами в этих ячейках являются ссылки на ячейки с числовыми данными из полей 1 и 2;

4 – поле формирования результатов расчета таблицы, которое может как содержать ячейки с формулами конечных результатов, так и создаваться процедурой копирования при многовариантном расчете.

Расположение полей зависит от содержания решаемой задачи, объема вычислений, объема исходных данных и способа их ввода и ряда других факторов. При этом необходимо исходить из общего принципа: расположение полей должно обеспечивать *наглядность* представления материала и *доступность данных*, *удобство их ввода, использования и редактирования*.

Рассмотрим пример разработки таблицы для расчета деформации балки.

Пример 6.1. Рассчитать деформацию балки, закрепленной одним концом на вертикальной опоре (рис. 6.21). Тип балки – I-образная, ширина – 0,3 м, площадь сечения Z = 2 м^2, линейная плотность d = 80 кг/м, момент инерции I = 3,68 м^4, модуль упругости E = 5,3 кг/м^2. Нагрузка приложена на расстоянии L = 2,6 м от закрепленного конца, величина нагрузки W = 70 кг. Длина балки $L1$ = 3,6 м.

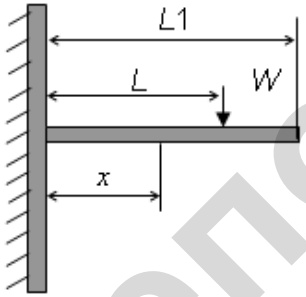


Рис. 6.21. К расчету деформации балки

Расчетные формулы:

Напряжение:

$$S = \frac{W}{Z}(L - x), \text{ если } x < L, \text{ и } S = 0, \text{ если } x \geq L. \quad (6.1)$$

Прогиб:

$$y = \frac{Wx^2}{6EI}(3L - x), \text{ если } x \leq L, \text{ и } y = \frac{WL^2}{6EI}(3x - L), \text{ если } x > L. \quad (6.2)$$

Пример разработки таблицы приведен на листинге 6.1.

Листинг 6.1. **Разработка таблицы**

	A	B	C	D	E	F	G
1	18 Марта 2011						
2	**Расчет деформации балки**						
3	**Исходные данные:**				**Расчет параметров**		
4	Тип балки	Обознач.	I-образная		x	Напряжение	Прогиб
5	Длина, м	L1	3,6		0	91	0
6	Положение нагрузки, м	L	2,6		0,3	80,5	0,403763
7	Нагрузка, кг	W	70		0,6	70	1,550451
8	Ширина, м	B	0,3		0,9	59,5	3,34316
9	Площадь сечения, м^2	Z	2		1,2	49	5,684988
10	Линейная плотность, кг/м^2	d	80		1,5	38,5	8,47903
11	Момент инерции, м^4	I	3,68		1,8	28	11,62838
12	Модуль упругости, кг/м^2	E	5,3		2,1	17,5	15,03615
13					2,4	7	18,60541
14	S=W/Z(L–x), если x<L, иначе 0				2,7	0	22,23988
15	y=Wx^2/(6EI)*(3*L–x), если x<L, иначе y=WL^2/(6EI)*(3*x–L)				3	0	25,87914
16					3,3	0	29,51839
17					3,6	0	33,15764

Порядок работы

1. Установите курсор в ячейку A1, щелкните по строке ввода – в строке ввода появится мигающий курсор. Введите дату, например: 18.03.05, и нажмите клавишу Enter или щелкните по кнопке ✓ в Строке формул. Измените формат представления даты командой ***Формат Ячеек*** контекстного меню или командами группы ***Число***

вкладки ***Главная,*** откройте вкладку ***Число***, выберите в списке ***Числовые форматы*** слово ***Дата***, а в списке ***Тип*** выберите нужный формат представления даты.

2. Введите наименование задачи в ячейку B2. Выделите ячейки A2:F2 и объедините их щелчком по кнопке ***Объединить и поместить в центре*** в группе ***Выравнивание***. Установите в группе ***Шрифт*** требуемый тип, размер и начертание шрифта. Выравнивание текста по горизонтали выполните с помощью команд в группе ***Выравнивание***.

3. Внесите аналогичным образом название таблицы Исходные данные в ячейку A3.

4. Внесите данные в ячейки A4:C12 согласно листингу 6.1. Выполните обрамление блока ячеек A4:C12, используя кнопку ***Границы*** группы ***Шрифт***.

5. Оформите аналогичным образом шапку таблицы Расчет параметров.

6. Внесите данные в ячейки E5:E17. Для ускорения ввода данных используйте маркер автозаполнения курсора. Введите в ячейку E5 значение 0, а в ячейку E6 – 0,3. Выделите ячейки E5:E6 мышью. Зацепите мышью за маркер автозаполнения и протащите по ячейкам E7:E17.

7. Внесите в ячейку F5 формулу (6.1) со ссылками на адреса ячеек:

=ЕСЛИ(E5<$C$6;$C$7/$C$9*($C$6-E5);0). (6.3)

Ссылки на ячейки C6, C7, C9 записаны в формуле с абсолютным адресом, так как эти данные не должны меняться при копировании.

Функция ЕСЛИ используется для выбора решения из двух альтернатив и имеет следующий синтаксис:

ЕСЛИ(<условие>;<выражение_истина>;<выражение_ложь>).

Функция работает следующим образом: программа проверяет условие, и если оно выполняется (истинно), то возвращает результат согласно Выражению_истина, в ином случае возвращается результат согласно Выражению_ложь.

Например. Требуется проверить равенство значений в ячейках A1 и B1 и вывести результат в ячейку C1.

Запишем в ячейку C1 формулу:

ЕСЛИ(A1=B1;”Выражение истинно”;”Выражение ложно”).

В зависимости от значений в ячейках A1 и B1 получим следующие результаты (листинг 6.2).

Листинг 6.2. **Использование функции ЕСЛИ**

	A	B	C	D
1	5	5	Выражение истинно	

	A	B	C	D
1	5	7	Выражение ложно	

8. Внесите в ячейку G5 формулу (6.2) со ссылками на адреса ячеек:

=ЕСЛИ(E5<$C$6;$C$7*E5^2/(6*$C$12*$C$11)*(3*$C$6-E5);
$C$7*$C$6^2/ (6*$C$12*$C$11)* (3*E5-$C$6)). (6.4)

9. Скопируйте формулы из ячеек F5, G5 в ячейки F6:G17, используя механизм автозаполнения.

6.2.2. Функции электронной таблицы

Excel имеет 11 категорий различных функций: математические/тригонометрические; инженерные; логические; текстовые; статистические; функции категории дата/время; функции для работы с базами данных/списками; финансовые; информационные и функции категории ссылки/массивы; функции проверки свойств и значений. Кроме того, Excel содержит большое число надстроечных функций, которые используются для создания компьютерных программ в Excel, а также имеется возможность создания пользовательских функций и программ на Visual Basic for Applications. Можно написать программы на других языках программирования высокого уровня, например, C, FORTRAN, и потом вызвать их в Excel.

Вызов функций осуществляется с помощью кнопки f_x строки ввода данных или команд вкладки ***Формулы***. В группе ***Библиотека функций*** (рис. 6.22) размещены списки категорий функций, что позволяет легко находить нужные функции. Список Автосумма содержит функции *Сумма*, *Среднее*, *Максимум*, *Минимум*, *Число*. Команда Число вызывает функцию *Счет*, которая подсчитывает число непустых ячеек в выделенной области.

Пример использования функции приведен на рис. 6.23. Если выделить имя функции и щелкнуть по нему мышкой, то появится справка по этой функции с примерами использования. Список ***Другие функции*** содержит категории специальных функций: *Статистические*, *Инженерные*, *Аналитические*, *Проверки свойств и значений*, *Совместимость*. Некоторые функции приведены в таблице 6.1.

С другими функциями будем знакомиться по мере надобности, а также с ними можно ознакомиться по технической документации или по справочной системе Excel.

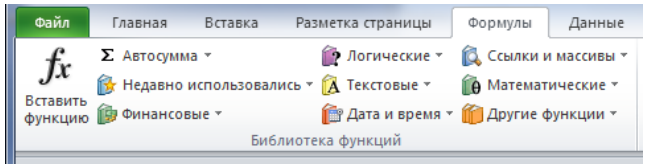


Рис. 6.22. Группа Библиотека функций вкладки Формула

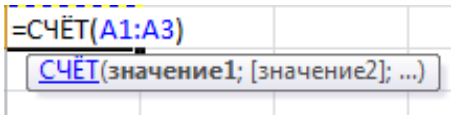


Рис. 6.23. Пример использования функции

Таблица 6.1

Функции электронной таблицы

Математические	
ABS(x)	Абсолютное значение числа
ФАКТР(n)	Факториал числа
ЦЕЛОЕ(x)	Число, округленное до ближайшего меньшего целого
ОСТАТ(x)	Модуль (остаток от деления двух чисел)
СЛЧИС(x)	Случайное число от 0 до 1
КОРЕНЬ(x)	Квадратный корень из числа
СУММА()	Сумма чисел в списке
СУММЕСЛИ()	Сумма значений в ячейках, соответствующих заданному критерию
СУММПРОИЗВ()	Сумма произведений элементов массивов
СУММКВ(x)	Сумма квадратов чисел в списке
СУММРАЗНКВ()	Сумма разностей квадратов элементов в двух массивах
СУММСУММКВ()	Сумма сумм квадратов элементов в двух массивах
СУММКВРАЗН()	Сумма квадратов разностей значений в двух массивах

Продолжение табл. 6.1

Логарифмические функции	
EXP(x)	Число *e*, возведенное в степень
LN(x)	Натуральный логарифм числа (основание *e*)
LOG(x, a)	Логарифм числа по заданному основанию LOG (число, основание)
LOG10(x)	Логарифм числа по основанию 10
Тригонометрические функции	
ПИ()	Возвращает значение числа π
COS(x)	Косинус числа
SIN(x)	Синус числа
TAN(x)	Тангенс числа
Обратные тригонометрические функции	
ACOS(x)	Арккосинус числа
ASIN(x)	Арксинус числа
ATAN(x)	Арктангенс числа от –π/2 до π/2
ATAN2(x)	Арктангенс отношения двух чисел (от –π до π)
Функции преобразования угла	
ГРАДУСЫ(x)	Показатель величины угла в градусах
РАДИАНЫ(x)	Показатель величины угла в радианах
Матричные функции	
МОПРЕД()	Определитель матрицы
МОБР()	Матрица, обратная заданной
МУМНОЖ()	Произведение двух матриц
ТРАНСП()	Транспонирование матрицы
Логические функции	
ЕСЛИ	Проверяет выполнение условия и возвращает результат согласно первому выражению, если условие истинно, и согласно второму выражению, если условие ложно: =ЕСЛИ(A1<B1;B1-A1;A1-B1)
ЕСЛИОШИБКА	Проверяет выражение на наличие ошибки и при ее наличии выполняет указанное действие, например, при $x > 1$ функция arcsin(x) выдаст сообщение об ошибке #ЧИСЛО!, а можно выдать сообщение об ошибке:

Окончание табл. 6.1

	=ЕСЛИОШИБКА(ASIN(A1);"Аргумент функции должен быть не более +1 и не менее –1")
И	Логическое умножение. Объединяет два условия, оба условия должны быть выполнены
ИЛИ	Логическое сложение. Возможно осуществление одного из двух условий или обоих условий
ИСТИНА	Возвращает значение ИСТИНА. Функция не имеет аргументов, используется в логических выражения, например: =ЕСЛИ(A1<B1=ИСТИНА;0;1)
ЛОЖЬ	То же, что и функция ИСТИНА
НЕ	Отрицание: =ЕСЛИ(НЕ(A1="ДОЖДЬ");"Хорошая погода";"Плохая погода")

Каждый список содержит команду ***Вставить функцию***, аналогичная команда имеется в группе ***Библиотека функций***. Эта команда вызывает окно диалога ***Мастер функций*** (рис. 6.24). Мастер функций содержит окно для поиска функции по ее краткому описанию, список категорий функций и окно выбора функции. При создании функций пользователя в списке категорий появляется группа ***Пользовательские***. При выделении функции в списке ***Выберите функцию*** ниже окна выбора выводится синтаксис данной функции и ее назначение. Тут же можно получить справку по функции, щелкнув мышкой по гиперссылке ***Справка по этой функции***.

В качестве примера использования мастера функций рассмотрим порядок ввода функции ЕСЛИ согласно выражению (6.2):

- Выделите ячейку G6, в которую надо поместить выражение. Введите команду ***Формулы, Вставить функцию*** или щелкните по одноименной кнопке f_x в строке ввода данных – откроется окно диалога Мастер функций (рис. 6.24).
- Найдите в списке функцию ЕСЛИ, выделите ее и щелкните по кнопке ОК – откроется окно диалога для ввода формул (рис. 6.25).
- В окне диалога имеется три строки ввода, в соответствии с синтаксисом функции. При выделении любой строки в нижней части таблицы выводится подсказка о назначении данной строки.

Первая строка служит для ввода логического условия, вторая – для ввода выражения, соответствующего истине, и третья строка – для ввода выражения, соответствующего отрицательному результату сравнения (ложь).

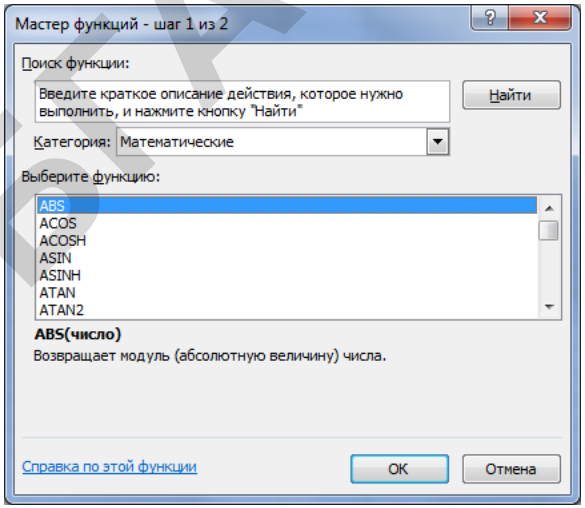


Рис. 6.24. Окно диалога Мастер функций

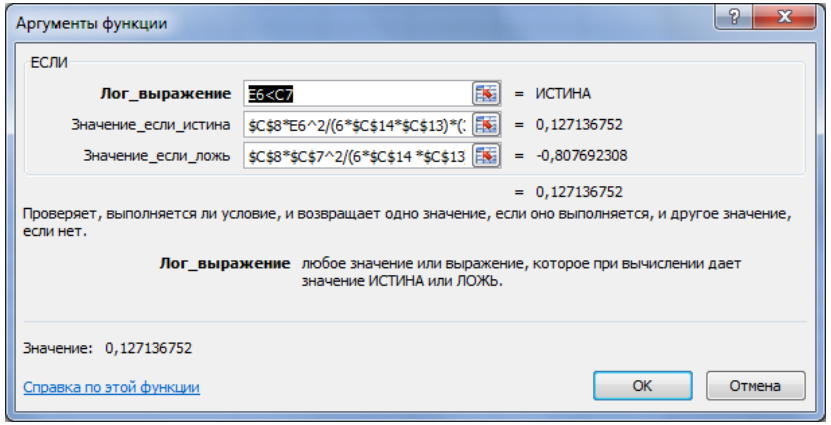


Рис. 6.25. Функция ЕСЛИ

- Введите в первую строку логическое условие E6<C7.

• Введите во вторую строку выражение для «истина»: $C$8*E6^2/(6*$C$14*$C$13)*(3*$C$7-E6).

• Введите в третью строку выражение для «ложь»: $C$8*$C$7^2/(6*$C$14*$C$13)*(3*E6-$C$7).

Если формулы введены правильно, то сразу же можно увидеть результат. Так как условие истинно, то функция возвращает результат согласно первому выражению (строка 2).

• Для завершения работы щелкните по кнопке ОК.

При достаточном навыке формулы можно вводить и без использования мастера функций. Достоинством использования мастера функций является то, что всегда можно получить оперативно подсказку по каждому полю ввода.

Примечание. Использованные функции сохраняются в списке адресов ячеек (Имя) на время текущего сеанса работы с таблицей. Поэтому для вызова ранее использовавшихся функций достаточно ввести команду = в строку формул и выбрать нужную функцию из списка Имя.

6.2.3. Генерирование данных

Часто бывает необходимо сгенерировать последовательность чисел, дат.

Для этой цели можно использовать механизм автозаполнения. Чтобы заполнить несколько ячеек прогрессией, необходимо записать в смежные ячейки данные, отличающиеся на величину шага, выделить эти ячейки и перетащить маркер автозаполнения выделенного диапазона ячеек. Можно также воспользоваться командой ***Заполнить*** из группы ***Редактирование*** вкладки ***Главная***. Данная команда имеет подменю: влево, вправо, вверх, вниз, прогрессия, выровнять.

Команды ***Влево***, ***Вправо***, ***Вверх***, ***Вниз*** позволяют заполнить выделенные ячейки одинаковыми данными или узором.

Команда ***Выровнять*** позволяет заполнить ячейку узором по ширине.

Команда ***Прогрессия*** позволяет заполнять ячейки рядами чисел и дат.

Внесите в ячейку начальное значение ряда чисел или дат; выделите область для заполнения, выберите команду ***Заполнить, Прогрессия***. На экран выводится диалоговое окно Прогрессия (рис. 6.26). Выберите *Тип* прогрессии, *Расположение* в соответствии

с выделенной областью (если выделен столбец – выберите *по столбцам*, если выделена строка – выберите *по строкам*), *Шаг* и щелкните по кнопке ОК. Если область не выделена, укажите *Предельное значение* – программа сама определит число строк/столбцов, необходимых для заполнения, и направление заполнения в соответствии с установкой режима в группе Расположение: по строкам – данные будут выведены в строку, по столбцам – данные будут выведены в столбец.

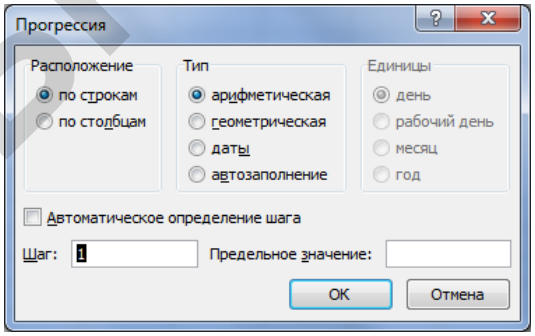


Рис. 6.26. Генерирование рядов данных

Если ввести в ячейки выделенного диапазона начальное и конечное значения ряда чисел, то шаг прогрессии определится автоматически.

Если в окне ввода Предельное значение указать нужное число или дату, программа автоматически определит требуемый диапазон.

При выборе типа Даты активизируется группа Единицы. Тогда можно вывести дни по порядку, рабочие дни в текущем месяце, число дней помесячно или число дней по годам.

Ряды чисел часто применяются для табулирования функций переменных. В этих случаях *целесообразнее* создать собственную программу генерирования ряда чисел с настраиваемым шагом (листинг 6.3). Для этого выполните следующее:

введите в ячейку A1 текст «Начальное значение», а в ячейку B1 – начальное значение ряда чисел;

введите в ячейку A2 текст «Шаг табуляции», а в ячейку B2 – значение шага табуляции (приращение аргумента);

запишите в ячейку A4 начальное значение ряда путем ссылки на ячейку B1: выделите ячейку A4 и запишите в нее формулу: =B1;

Листинг 6.3. **Табулирование функции**

	A	B
1	Начальное знач.	1
2	Шаг табуляции	0,5
3	Аргумент	Функция
4	=B1	=SIN(A4)
5	=A4+$B$2	=SIN(A5)
6	=A5+$B$2	=SIN(A6)
7	=A6+$B$2	=SIN(A7)
8	=A7+$B$2	= SIN(A8)
9	=A8+$B$2	=SIN(A9)
10	=A9+$B$2	=SIN(A10)

запишите в ячейку A5 формулу арифметической прогрессии A4+$B$2;

определите диапазон ячеек, куда необходимо скопировать формулу (номер начальной и конечной ячеек);

скопируйте формулу из ячейки A5 в остальные ячейки диапазона.

6.2.4. Табулирование функций

Табулирование функций с использованием операций копирования

Под табулированием понимают конструирование, вычисление и составление различных математических таблиц.

Пример табулирования функции одной переменной приведен на листинге 6.3. Для выполнения операции табулирования необходимо:

сгенерировать ряд значений аргумента на заданном интервале;

записать в соседний столбец справа расчетную формулу зависимости функции от аргумента;

скопировать расчетную формулу во все ячейки требуемого диапазона изменения аргумента.

Пример табулирования функции двух переменных $Z = 2X + Y^2$ приведен на листинге 6.4.

Порядок выполнения операции следующий:

запишите в ячейку A3 начальное значение аргумента X;

запишите в ячейку B2 начальное значение аргумента Y;

Листинг 6.4. **Табулирование функции двух переменных**

	A	B	C	D	E
1	Шаг 1-го аргумента	0,5	Шаг 2-го аргумента	0,2	
2	X \ Y	0,1	B2+$D$1	C2+$D$1	D2+$D$1
3	1	2*$A3+B$2^2	2*$A3+C$2^2	2*$A3+D$2^2	2*$A3+E$2^2
4	A3+$B$1	2*$A4+B$2^2	2*$A4+C$2^2	2*$A4+D$2^2	2*$A4+E$2^2
5	A4+$B$1	2*$A5+B$2^2	2*$A5+C$2^2	2*$A5+D$2^2	2*$A5+E$2^2
6	A5+$B$1	2*$A6+B$2^2	2*$A6+C$2^2	2*$A6+D$2^2	2*$A6+E$2^2
7	A6+$B$1	2*$A7+B$2^2	2*$A7+C$2^2	2*$A7+D$2^2	2*$A7+E$2^2
8	A7+$B$1	2*$A8+B$2^2	2*$A8+C$2^2	2*$A8+D$2^2	2*$A8+E$2^2
9	A8+$B$1	2*$A9+B$2^2	2*$A9+C$2^2	2*$A9+D$2^2	2*$A9+E$2^2

запишите в ячейки A4 и C2 формулы для генерирования рядов значений аргументов;

скопируйте в ячейки A5:A9 формулу для вычисления аргумента X из ячейки A4;

скопируйте в ячейки D2:E2 формулу для вычисления Y из ячейки C2;

запишите в ячейку B3 таблицы расчетную формулу с использованием смешанных адресов ячеек: у первого аргумента зафиксируйте столбец, а у второго аргумента – строку;

скопируйте формулу во все ячейки блока, используя маркер автозаполнения.

> *Общее правило при копировании формул со смешанными адресами:* если данные находятся в строке, то фиксируется номер строки, а если данные находятся в столбце, то фиксируется номер столбца.

Табулирование функции с использованием модуля Таблица данных

Для табулирования функций одной и двух переменных можно использовать средства Excel: команду ***Анализ «Что если»***, ***Таблица данных***[13] из группы ***Работа с данными*** вкладки ***Данные***. Однако, с точки зрения автора, алгоритмы табулирования функции с использованием этой команды не отвечают требованию массовости и не дают выигрыша во времени. Тем не менее, рассмотрим алгоритм использования модуля Таблица данных.

Пример 6.2. Протабулировать функцию SIN(x) на интервале от –ПИ/2 до ПИ/2 с шагом 0,5.

[13] Таблица подстановки – в предыдущих версиях программы Excel.

Решение:

введите в ячейки A1, A3, A5, C1 текст Ячейка ввода, Начальное значение, Шаг табуляции, Ячейка ввода формулы (листинг 6.5);

Листинг 6.5. **Использование модуля Таблица данных**

	A	B	C
1	Ячейка ввода		Ячейка ввода формулы
2	0		=SIN(A2)
3	Начальное значение	–1,5708	–1
4	–1,5708	–1,0708	–0,87758
5	Шаг табуляции	–0,5708	–0,5403
6	0,5	–0,0708	–0,07074
7		0,429204	0,416147
8		0,929204	0,801144
9		1,429204	0,989992
10		1,929204	0,936457

введите в Ячейку ввода A2 произвольное число, например, 0 (это число не влияет на результат табулирования);

введите в ячейку A4 начальное значение аргумента x: $-\pi/2$. Для ввода этого числа используем функцию ПИ();

введите в ячейку A6 значение шага 0,5;

введите в ячейку C2 формулу, например, SIN(x). В качестве аргумента указывается адрес ячейки ввода (A2);

сгенерируйте в столбце B, начиная с ячейки B3, ряд значений аргумента;

выделите область B2:C10 и введите команду ***Анализ «Что если», Таблица данных***;

в окне диалога ***Таблица данных*** (рис. 6.27) введите в окно ввода «*Подставлять значения по строкам в…*» адрес Ячейки ввода – A2 и щелкните кнопку OK. (Для ввода номера ячейки достаточно активизировать окно ввода щелчком мыши и щелкнуть по ячейке A2.) Работа завершена.

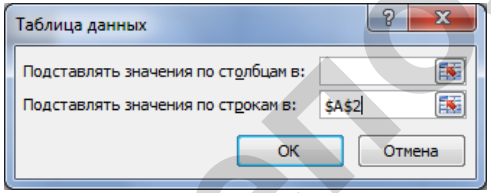


Рис. 6.27. Окно диалога Таблица данных

Пример 6.3. Протабулировать функцию $2x + y^2$ при x, изменяющемся от 0 до 1 с шагом 0,2, и y, изменяющемся от 1 до 4 с шагом 1, с использованием окна диалога Таблица данных.

Решение:

обозначьте ячейку A2 как ячейку ввода по строкам, а ячейку A4 как ячейку ввода по столбцам. Для этого внесите соответствующие записи в ячейки A1 и A3 (листинг 6.6);

Листинг 6.6. **Табулирование функции двух переменных**

A	B	C	D	E	F
По строкам	=2*A2+A4^2	1	2	3	4
1	0	1	4	9	16
По столбцам	0,2	1,4	4,4	9,4	16,4
1	0,4	1,8	4,8	9,8	16,8
	0,6	2,2	5,2	10,2	17,2
	0,8	2,6	5,6	10,6	17,6
	1	3	6	11	18

внесите в столбец B начиная с ячейки B2 значения аргумента x;

внесите в строку 1 начиная с ячейки C1 значения аргумента y;

внесите в ячейку B1 (ячейка на пересечении первого столбца и первой строки будущей таблицы) формулу $2x + y^2$, или с учетом ссылок на номера ячеек ввода – 2*A2+A4^2;

выделите область B1:F7 и введите команду ***Данные, Анализ «Что если», Таблица данных***;

внесите в строку ввода *Подставлять значения по столбцам в…:* номер ячейки A4, а в строку ввода *Подставлять значения по строкам в…:* – номер ячейки A2 и щелкните кнопку OK. Работа завершена.

Контрольные вопросы

1. Каким образом можно сгенерировать ряд чисел, используя маркер заполнения курсора таблицы?
2. Как сгенерировать ряд чисел с арифметической или геометрической прогрессией?
3. Как протабулировать функцию одной переменной?
4. Как протабулировать функцию двух переменных?
5. Опишите алгоритм табулирования функции одной переменной с использованием модуля Таблица данных.
6. Опишите алгоритм табулирования функции двух переменных с использованием модуля Таблица данных.

Заключение

Электронная таблица позволяет генерировать ряды данных: чисел, дат, времени и других последовательностей, заданных пользователем, с использованием маркера автозаполнения.

Во многих задачах имеется необходимость генерирования значений аргумента и соответствующих значений функции. Эта операция может выполняться с использованием как операции копирования, так и команды Таблица данных из меню Данные.

6.3. ГРАФИЧЕСКИЕ ВОЗМОЖНОСТИ ЭЛЕКТРОННОЙ ТАБЛИЦЫ

Диаграммы и графики позволяют представить числовые данные, результаты обработки таблиц в наглядной форме. При создании диаграммы можно выделить три этапа: создание таблицы, описание графика на бумаге, описание графика в электронной таблице и его использование.

Хорошо разработанная таблица содержит все элементы, необходимые для описания графика. При описании графика на бумаге необходимо установить соответствие между элементами таблицы и графиком (при достаточном навыке этот этап не обязателен). Если на таблице нет каких-то элементов, например, заголовков осей, заголовка таблицы, то их необходимо описать вне пределов таблицы.

График обычно включает следующие элементы: заголовок, обозначение осей, разметку по осям, описание меток (легенда), числовые данные на графике. Кроме того, необходимо определить тип диаграммы, наиболее подходящий для имеющихся данных.

Построение диаграмм в Excel 2010 значительно усложнено по сравнению с предыдущими версиями программы, когда все операции можно было выполнить за четыре шага.

Для построения графиков и диаграмм в электронной таблице используется группа ***Диаграммы*** вкладки ***Вставка***, а также вкладки ***Конструктор***, ***Макет*** и ***Формат*** вкладки *Работа с диаграммами*.

Непосредственно в группе Диаграммы можно выбрать тип диаграммы и ее вид. Мастер диаграмм позволяет использовать 11 стандартных типов диаграмм (рис. 6.28). Для построения графиков функций необходимо использовать ***Точечную*** диаграмму. Однако при первом использовании данного графика понадобится настройка параметров с целью установления зависимости функции от аргумента.

Тип ***График*** используется только для построения линейных диаграмм, так как он не позволяет связать функцию с аргументом. Выбранный вид диаграммы отображается на экране. После построения графика активизируется вкладка ***Работа с диаграммами***. Вкладка ***Конструктор*** (рис. 6.29) позволяет изменить тип диаграммы и сохранить его как шаблон, поменять местами строки и столбцы при построении графиков функций и диаграмм (команда *Строка/Столбец*), добавить, изменить или удалить данные (команда *Выбрать данные*), выбрать стиль оформления, то есть размещение заголовка, легенды, подписей осей, вывод сетки и данных на диаграмму (команда *Экспресс-макет*), выбрать стиль графика (команда *Экспресс-стили*), а также определить место размещения диаграммы: на текущем листе или на отдельном листе (команда *Переместить диаграмму*).

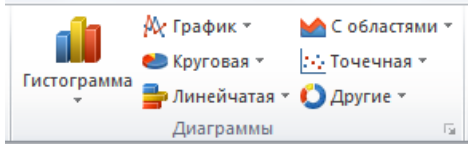


Рис. 6.28. Группа Диаграммы

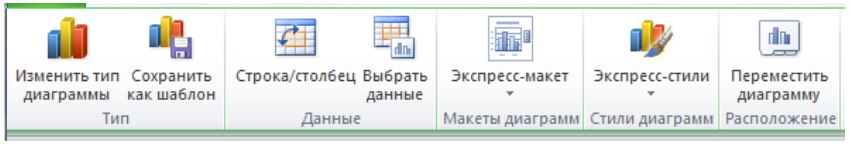


Рис. 6.29. Вкладка Конструктор диаграмм

Вкладка ***Макет*** (рис. 6.30) позволяет управлять оформлением диаграммы. То, что можно было сделать с помощью команды Экспресс-макет (макет диаграммы), на этой вкладке можно выбрать вручную, используя команды вкладки.

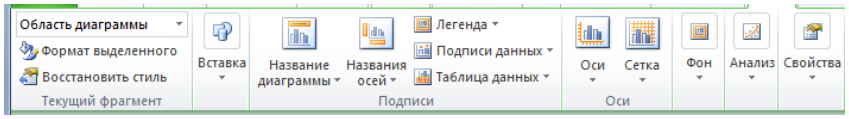


Рис. 6.30. Вкладка Макет настройки диаграммы

Вкладка ***Формат*** позволяет выбрать стиль оформления диаграммы, то есть то же, что было сделано с помощью команды Экспресс-стили вкладки Конструктор. Если отображаемые данные имеют разную размерность, например, стоимость и объем, то можно выделить график и построить вспомогательную ось командой ***Формат, Формат выделенного, Построить ряд По вспомогательной оси***.

Диаграмму можно перемещать по рабочему листу, изменять ее размеры, копировать. Если щелкнуть дважды мышью по какому-либо элементу диаграммы, то открывается окно диалога для настройки соответствующего элемента диаграммы (цвет линий, стиль линий, тип диаграммы, шрифт и так далее). При щелчке правой кнопкой мыши по элементу диаграммы вызывается контекстное меню для настройки соответствующего элемента.

Список *Область диаграммы* группы Текущий фрагмент содержит список всех элементов диаграммы, его удобно использовать для выделения нужных элементов при редактировании. Команды группы *Подписи* позволяют вывести местозаполнители для соответствующих названий и управлять их размещением на диаграмме. Кнопки группы Оси управляют выводом на диаграмму основных и вспомогательных осей.

Пример 6.4. Построение графика функции одной переменной.

Построить графики функций Sin(x) и Cos(x) на отрезке от –ПИ до ПИ. Отрезок разделить на десять равных частей. Пример построения графика приведен на рисунке 6.31.

Решение:

1. Протабулируйте функции на заданном отрезке (см. пример 6.1). Шаг табуляции определите как отношение длины отрезка табулирования функции к числу отрезков N: (ПИ() – (–ПИ))/N.

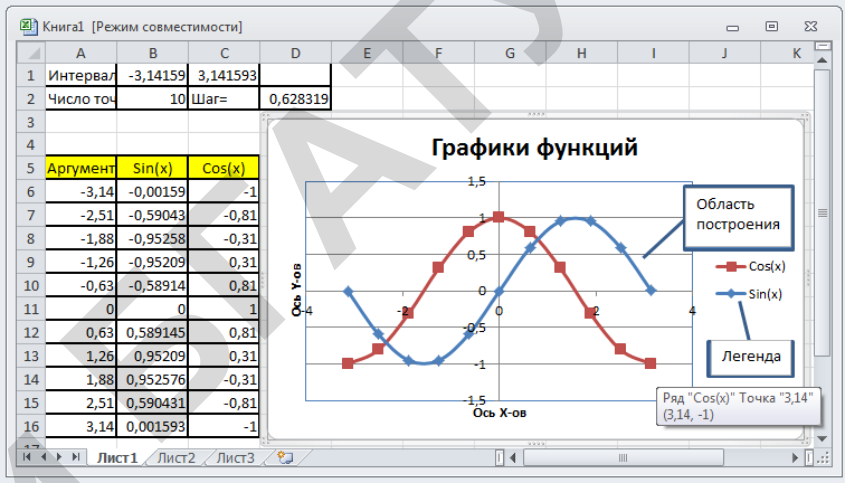
2. Выделите область исходных данных для построения графика функции Sin(x), включая заголовки: A5:C19.

3. Введите команду Точечная диаграмма из группы Диаграммы вкладки Вставка и выберите тип диаграммы *Точечная с гладкими кривыми и маркерами*.

4. Выделите диаграмму щелчком мыши по контуру диаграммы. Откройте вкладку Макет.

5. Установите границу области построения, если она отсутствует: ***Макет, Область построения, Показать область построения***.

6. Установите местозаполнители для названия диаграммы и названия осей командами ***Макет, Название диаграммы*** и ***Макет, Название осей*** и напишите названия осей и диаграммы.



	A	B	C	D
1	Интервал	-3,14159	3,141593	
2	Число точ	10	Шаг=	0,628319
3				
4				
5	Аргумент	Sin(x)	Cos(x)	
6	-3,14	-0,00159	-1	
7	-2,51	-0,59043	-0,81	
8	-1,88	-0,95258	-0,31	
9	-1,26	-0,95209	0,31	
10	-0,63	-0,58914	0,81	
11	0	0	1	
12	0,63	0,589145	0,81	
13	1,26	0,95209	0,31	
14	1,88	0,952576	-0,31	
15	2,51	0,590431	-0,81	
16	3,14	0,001593	-1	

Рис. 6.31. Табулирование функции одной переменной

7. Сохраните шаблон диаграммы: выделите диаграмму, введите команду ***Конструктор, Сохранить шаблон***.

При необходимости внести изменения в данные, добавить или удалить график выполните следующее.

Введите команду ***Конструктор, Выбрать данные*** – откроется окно диалога ***Выбор источника данных*** (рис. 6.32). Для удаления графика выделите нужную строку и щелкните по кнопке Удалить. Для добавления или изменения данных выберите соответствующую команду. Откроется окно диалога ***Изменение ряда*** (рис. 6.33). В строке Имя ряда укажите адрес ячейки, где записано название графика функции, в строке Значения X укажите область данных аргумента функции, а в строке Значения Y укажите область размещения значений функции.

6.4. ПРОВЕРКА ДАННЫХ

При вводе данных в ячейки в некоторых случаях необходимо контролировать данные, например, данные должны быть только целыми числами, или имеются ограничения на значения чисел, числа не могут быть отрицательными, или данные должны быть только текстовыми. Более сложные условия контроля возникают,

когда данные должны быть введены из ограниченного списка. В этом случае для помощи пользователю программа предлагает модуль ***Проверка данных*** из группы ***Работа с данными*** вкладки ***Данные***. После ввода команды открывается окно диалога ***Проверка вводимых значений*** (рис. 6.34). Окно диалога имеет три вкладки: *Параметры*, *Сообщение для ввода* и *Сообщение об ошибке*.

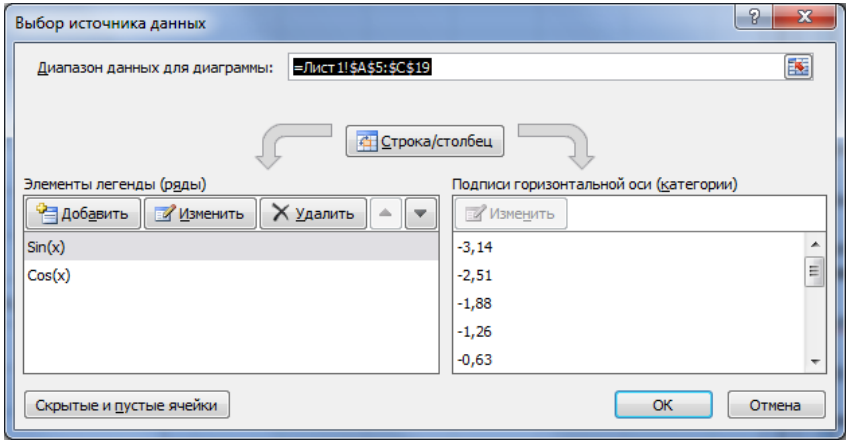


Рис. 6.32. Окно диалога Настройка ленты

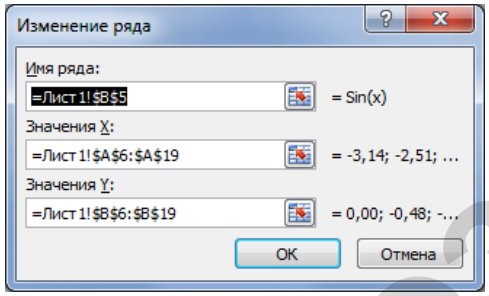


Рис. 6.33. Окно диалога Настройка ленты

Вкладка ***Параметры*** служит для установки контролируемых параметров вводимых данных. Список Тип данных предлагает следующие значения: Любое значение, Целое число, Действительное число, Список, Дата, Время, Длина текста, Другой. При выборе одного из типов значений открывается другое окно (рис. 6.35), которое

содержит список Значение для выбора знака отношения и два окна для ввода минимального и максимального значений числа.

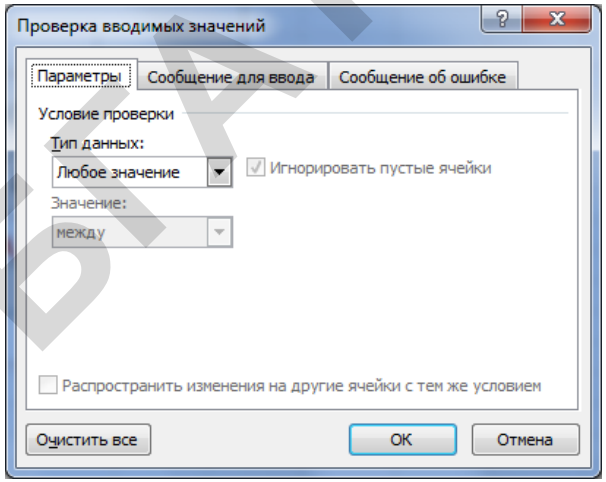


Рис. 6.34. Окно диалога Проверка вводимых значений

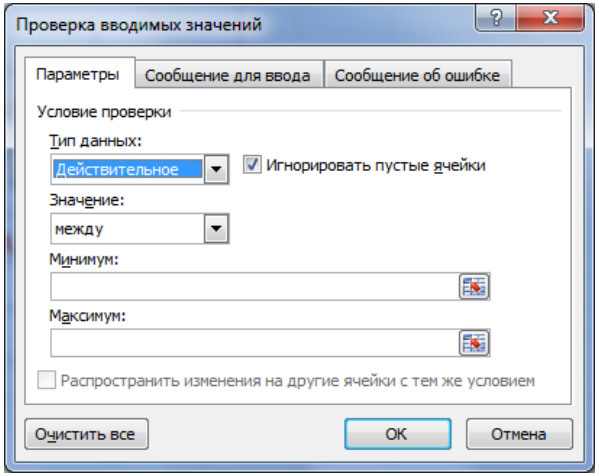


Рис. 6.35. Настройка параметров контроля вводимых значений

При выборе типа ***Список*** в строке Источник необходимо указать источник данных: область данных на рабочем листе, или нажать

клавишу *F3*, для выбора именованного списка. Следовательно, список данных должен быть подготовлен заранее на текущем листе или другом листе текущей книги.

В окне ***Сообщение для ввода*** (рис. 6.36) в строке *Заголовок* напишите название выполняемой операции, например, «Введите данные», «Укажите фамилию и инициалы» и т. д. В области *Сообщение* следует ввести напоминание о требованиях к вводу данных, например, «Данные должны быть целыми числами в диапазоне от 5 до 20».

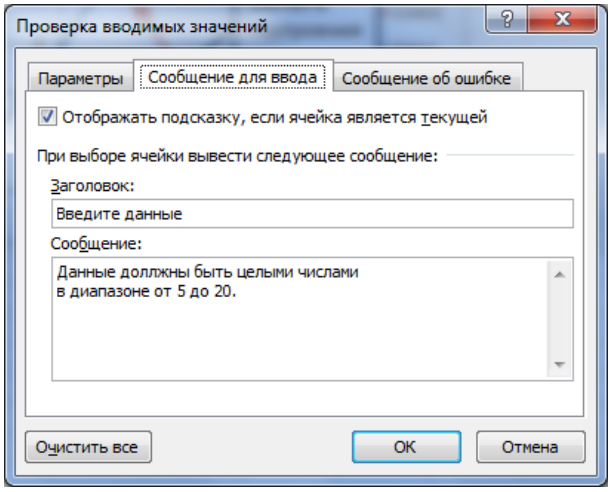


Рис. 6.36. Настройка сообщения для ввода

В окне ***Сообщение об ошибке*** (рис. 6.37) в списке *Вид* выберите действие, которое должно быть выполнено при возникновении ошибки: Останов, Предупреждение, Сообщение. В строке *Заголовок* укажите выполняемую операцию, при которой возникла ошибка, например, «Ввод данных», в области *Сообщение* следует написать, что произошло и какие действия необходимо выполнить для устранения ошибки, например, «Неправильный ввод данных! Повторите ввод».

Алгоритм создания контроля ввода данных

- Создайте на отдельном листе списки, которые могут понадобиться при вводе данных, и присвойте им имена.
- Выделите непрерывную область ячеек или несколько групп ячеек, подлежащих контролю.

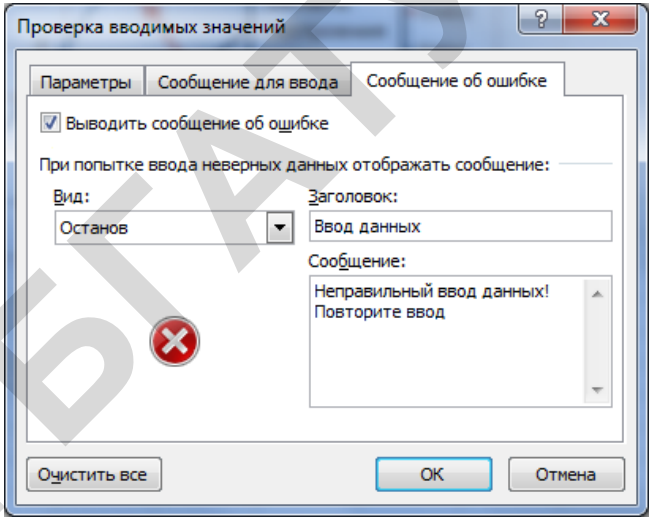


Рис. 6.37. Настройка сообщения об ошибке

- Введите команду Данные, Проверка данных.
- На вкладке Параметры установите контролируемые параметры.
- Заполните строки вкладки Сообщение для ввода.
- Заполните строки вкладки Сообщение об ошибке.
- Завершите операцию щелчком по кнопке ОК.

Для отмены настроек щелкните по кнопке ***Очистить все*** или установите на вкладке Параметры в списке Тип данных «Любое значение».

При выделении ячейки, для которой установлена проверка ввода данных, возле нее появляется сообщение, записанное ранее на вкладке Сообщение для ввода. Если в выделенную ячейку ввод данных предусмотрен из списка, то возле нее появляется кнопка открытия раскрывающегося списка. При возникновении ошибки ввода выдается звуковой сигнал и появляется всплывающее окно с предупреждением и советом, которые были записаны на вкладке Сообщение об ошибке.

Контрольные вопросы

1. Назовите основные элементы диаграммы.
2. Опишите порядок построения графиков функций.
3. Как добавить график функции на диаграмму?
4. Как изменить заголовок диаграммы или ее осей?

5. Как изменить стиль линий сетки?
6. Каким образом перемещается диаграмма по рабочему листу?
7. Как изменить размеры диаграммы?
8. Какой тип диаграммы больше подходит для построения графиков функций?
9. Как установить вспомогательную ось на диаграмму?
10. Для чего необходима проверка данных?
11. Как установить ограничения на ввод данных?
12. Как создать список для выбора значений из списка?

Заключение

Электронная таблица позволяет отображать данные в виде графиков и диаграмм. Для построения диаграмм используются команды группы Диаграмма вкладки Вставка, которые позволяют строить диаграммы 11 стандартных видов. Для построения графиков функций целесообразно использовать точечную диаграмму. Вкладки Конструктор, Макет и Формат позволяют производить различные настройки: оформление заголовков, сетки, таблицы данных, подписи данных, легенды. Для автоматического создания легенды рекомендуется перед построением диаграммы или графика выделить таблицу данных вместе с шапками таблицы (заголовками строк и столбцов). Электронная таблица позволяет редактировать графики и диаграммы и их отдельные элементы.

Для обеспечения контроля ввода данных целесообразно воспользоваться модулем Проверка данных из группы Работа с данными вкладки Данные. Для выбора данных из списка рекомендуется создать именованный список на отдельном листе текущей книги.

6.5. РАБОТА С МАТРИЦАМИ

6.5.1. Операции с матрицами

Электронная таблица позволяет выполнять линейные преобразования матриц: умножение, деление матриц на число, прибавление или вычитание чисел, а также операции над матрицами: сложение, умножение матриц, транспонирование, вычисление определителей. Средствами Excel можно решать и системы линейных алгебраических уравнений. Для этой цели электронная таблица имеет ряд функций для работы с матрицами:

МОБР(массив) – вычисление обратной матрицы;

МОПРЕД(массив) – вычисление определителя матрицы;

МУМНОЖ(массив; массив) – умножение матриц;

ТРАНСП(массив) – транспонирование матриц.

Примеры операций с матрицами приведены на листинге 6.7. Обратите внимание на разные результаты, получаемые при умножении матриц с использованием оператора умножения * и с использованием функции МУМНОЖ. В первом случае каждый элемент матрицы результата равен произведению соответствующих элементов сомножителей, во втором случае каждый элемент матрицы вычисляется по формуле:

$$C(i,k) = \sum_{j=1}^{m} A(i,j) * B(j,k),$$

где m – число столбцов в матрице A;
матрица C имеет размерность $P \times Q$;
P – число строк в матрице A;
Q – число столбцов в матрице B.

Листинг 6.7. **Примеры операций с матрицами**

	A	B	C	D	T	F	G	H	I	J	K	L
1	Прибавление числа к матрице				Умножение матрицы на число				Сложение матриц			
2												
3	a=	2,543			a1=	7,345			C=B+B1	3	6	
4										8	11	
5	B=	2	4		B1=	1	2					
6		5	7			3	4		Умножение матриц			
7												
8	B+a=	4,543	6,543		B1*a1=	7,345	14,69		D=B*B1	2	8	
9		7,543	9,543			22,035	29,38			15	28	
10												
11	Транспонирование матриц						Использование функции МУМНОЖ					
12												
13	ТРАНСП(B5:C6)			2	5		МУМНОЖ(B;B1)=			14	20	
14				4	7					26	38	

Алгоритм вычислений над матрицами сводится к следующим операциям:

выделить ячейку или область (если результатом выполнения операции будет матрица), куда будет помещаться результат вычисления (B8:C9);

ввести в строку ввода команду =;

ввести в строку ввода первый операнд, например, область матрицы *B* (B5:C6);

ввести в строку ввода символ операции, например, оператор сложения +;

ввести в строку ввода второй операнд, например, адрес числа *a* – B3. Для первого примера на листинге 6.7 получим выражение {=B5:C6+B3};

нажать комбинацию клавиш Ctrl + Shift + Enter для вставки формулы в выделенную область.

Наиболее удобно выполнять операции над матрицами, если им присвоены имена (как в Mathcad). В этом случае операции над матрицами выполняются так же, как и операции с простыми числами.

Внимание! Для ввода формулы в массив используется комбинация клавиш Ctrl + Shift + Enter.

6.5.2. Решение систем линейных алгебраических уравнений

Для решения систем линейных алгебраических уравнений применяют аналитические и численные методы.

Электронная таблица Excel не имеет функций для решения систем уравнений, формулы для вычисления матриц необходимо формировать самостоятельно, используя известные методы, например, метод Крамера или метод Гаусса (метод исключения переменных). Однако с помощью встроенных функций МОБР, МУМНОЖ и МОПРЕД эти операции выполняются достаточно легко. Например, можно воспользоваться формулой вычисления вектора неизвестных через обратную матрицу и вектор свободных членов: $\overline{X} = A^{-1} \cdot \overline{B}$.

Пример 6.5. Решить систему линейных алгебраических уравнений матричным методом (листинг 6.8):

$$\begin{cases} 65{,}18x + 36{,}31y + 23{,}76z = 86{,}46 \\ -17{,}98x + 23{,}89y + 27{,}54z = -38{,}07 \\ 23{,}75x + 13{,}94y + 48{,}12z = 53{,}97 \end{cases} \quad (6.5)$$

Решение:

1. Внесите в ячейки B6–D8 значения коэффициентов при неизвестных.

2. Внесите в ячейки F6–F8 значения свободных членов системы уравнений.

Листинг 6.8. **Решение системы линейных алгебраических уравнений матричным способом $X = A^{-1}B$**

	A	B	C	D	E	F	G
5		Матрица коэффициентов				Вектор свободных членов	
6		65,18	36,31	23,76		86,46	
7		–17,98	23,89	27,54		–38,07	
8		23,75	13,94	48,12		53,97	
9							
10			Обратная матрица			Результат	
11							
12		0,009	–0,017	0,005		1,67504	
13		0,018	0,030	–0,026		–1,01011	
14		–0,010	–0,001	0,026		0,58746	

3. Выделите диапазон ячеек B12:D14 и введите формулу МОБР(B6:D8), для завершения операции ввода нажмите комбинацию клавиш Ctrl + Shift + Enter.

4. Выделите диапазон ячеек F12:F14 и введите формулу МУМНОЖ(B12:D14;F6:F8). Для завершения ввода формулы нажмите комбинацию клавиш Ctrl + Shift + Enter.

В ячейках F12–F14 появятся значения корней системы уравнений.

Пример 6.6. Решите систему линейных алгебраических уравнений (6.5) методом Крамера (листинг 6.9).

Решение:

1. Внесите в таблицу расширенную матрицу, то есть запишите в ячейки A2:D4 электронной таблицы коэффициенты при неизвестных и свободные члены.

2. Сформируйте три дополнительных матрицы путем копирования матрицы коэффициентов:

выделите матрицу коэффициентов и скопируйте ее в буфер обмена;

вставьте матрицу коэффициентов из буфера в ячейки A6:C8; A10:C12; A14:C16, используя команду ***Специальная вставка, Вставить связь*** вкладки ***Главная***.

3. Замените в дополнительных матрицах коэффициенты при неизвестных на вектор свободных членов. При этом так же, как и в пункте 2, следует воспользоваться командой ***Специальная вставка, Вставить связь***.

Листинг 6.9. **Решение систем линейных алгебраических уравнений методом Крамера**

	A	B	C	D	E	F	G
1	Расширенная матрица				Гл. опред.		
2	65,18	36,31	23,76	86,46	85635		
3	–17,98	23,89	27,54	–38,07			
4	23,75	13,94	48,12	53,97			
5	Первый дополнительный определитель					***Результат***	
6	86,46	36,31	23,76		143443	X=	1,67504
7	–38,07	23,89	27,54				
8	53,97	13,94	48,12				
9	Второй дополнительный определитель						
10	65,18	86,46	23,76		–86501	Y=	–1,01011
11	–17,98	–38,07	27,54				
12	23,75	53,97	48,12				
13	Третий дополнительный определитель						
14	65,18	36,31	86,46		50308	Z=	0,58746
15	–17,98	23,89	–38,07				
16	23,75	13,94	53,97				

4. Запишите напротив первой строки расширенной матрицы в ячейку E2 расчетную формулу для вычисления главного определителя: МОПРЕД(A2:C4).

5. Скопируйте расчетную формулу из ячейки E2 в ячейки E6, E10, E14.

6. Запишите формулы для вычисления неизвестных как отношение соответствующих дополнительных определителей к главному определителю в ячейки G6, G10 и G14.

Контрольные вопросы

1. Что такое расширенная матрица коэффициентов системы линейных уравнений?

2. Что является решением системы линейных алгебраических уравнений?

3. Какие методы применяются для решения СЛАУ в электронной таблице?

4. Напишите алгоритм решения СЛАУ методом Крамера.

5. Напишите алгоритм решения СЛАУ матричным методом.

6. Назовите функции электронной таблицы для работы с матрицами.

Заключение

В настоящем разделе мы познакомились с возможностями электронной таблицы по работе с векторами и матрицами. Excel имеет несколько функций для работы с массивами: МОБР, МОПРЕД, МУМНОЖ, ТРАНСП. Эти функции позволяют выполнять элементарные преобразования матриц, а также решать системы линейных алгебраических уравнений известными методами.

6.6. ЭЛЕМЕНТЫ МАТЕМАТИЧЕСКОГО АНАЛИЗА

6.6.1. Вычисление производных численными методами

Если функция задана в виде таблицы, то производные от функции в любой точке могут быть вычислены численными методами по известным приближенным разностным формулам (см. раздел 3.3.1 пример 3.5):

$$\frac{dy}{dx} = \frac{y(x+h) - y(x-h)}{2h}, O(h); \qquad (6.6)$$

$$\frac{d^2y}{dx^2} = \frac{y(x+h) - 2y(x) + y(x-h)}{h^2}, O(h^2), \qquad (6.7)$$

где h – шаг табуляции;

$O(h)$ и $O(h^2)$ – ошибки вычисления производных.

С целью уменьшения ошибки, вызванной нелинейностью функции для вычисления производной, целесообразно применять трехточечную схему (рис. 6.38).

При вычислении производной численными методами возникают ошибки двух видов: ошибки усечения и ошибки округления.

Ошибка усечения пропорциональна величине шага для первой производной и квадрату величины шага для второй производной. С уменьшением шага ошибка усечения уменьшается, но одновременно возрастает ошибка округления. Во избежание получения ошибочных результатов необходимо контролировать, чтобы разность двух смежных значений функции была не меньше точности вычислений. Например, на компьютере, вычисляющем с точностью до 14 знаков, разность не должна быть меньше 10^{-14}.

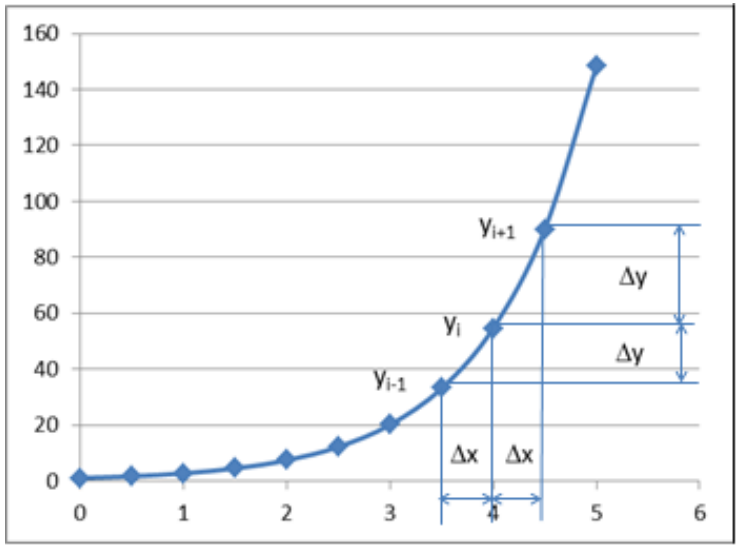


Рис. 6.38. Определение производной по трехточечной схеме

6.6.2. Вычисление определенного интеграла численными методами

Известно, что определенный интеграл численно равен площади, ограниченной подынтегральной функцией, осью OX и вертикальными прямыми $x = a$ и $x = b$, где a и b – границы интервала интегрирования функции (см. раздел 3.3.3).

При численном интегрировании интеграл заменяют суммой конечного числа элементарных площадок, вычисленных тем или иным способом:

$$\int_a^b f(x)dx = \sum_{i=1}^{n} hf(x), \qquad (6.8)$$

где h – шаг табулирования функции.

Поэтому в Excel определенный интеграл легко вычислить, протабулировав выражение под знаком суммы (6.8) и подсчитав сумму значений функции на отрезке табулирования. Другой путь – создать функцию пользователя (см. раздел 6.8).

6.6.3. Определение коэффициентов эмпирических формул методом наименьших квадратов

Нередко при обработке результатов наблюдений встречаются со следующей задачей: в результате проведенных опытов получен ряд значений переменных x и y, однако характер функциональной зависимости между ними остается неизвестным. Требуется по полученным данным найти аналитическое выражение зависимости между ними. Формулы, полученные в результате решения задач подобного рода, называются эмпирическими.

Задача о построении эмпирической формулы состоит в следующем.

Пусть в результате экспериментов получены данные, представленные таблицей:

x_1	x_2	...	x_k	...	x_n
y_1	y_2	...	y_k	...	y_n

где x_i – значения аргументов, изменяющихся с постоянным шагом и расположенных в порядке возрастания их значений;

y_i – экспериментальные значения функции, соответствующие данным значениям аргументов.

Требуется найти эмпирическую формулу $y = f(x_i,a_1,a_2,\ldots,a_m)$, где функция f зависит не только от значения аргумента x_i, но и от некоторых ***параметров*** a_j, такую, для которой сумма квадратов отклонения для данного многочлена окажется минимальной.

Разности

$$f(x_i,a_1,a_2,\ldots,a_m) - y_i = e_i, \ (I = 1, 2, 3, \ldots, m) \qquad (6.9)$$

называются отклонениями или погрешностями, здесь

x_i – числа из первой строки таблицы;

y_i – числа из второй строки данной таблицы;

$f(x_i,a_1,a_2,\ldots,a_m)$ – значения функции при соответствующих значениях аргумента x_i.

Параметры a_j эмпирической формулы $y = f(x,a_1,a_2,\ldots,a_m)$ необходимо подобрать таким образом, чтобы отклонения e_i оказались наименьшими на всем множестве данных.

Наиболее распространенным критерием поиска решения является критерий, лежащий в основе ***метода наименьших квадратов***: параметры функции выбирают так, чтобы сумма квадратов отклонений оказалась минимальной:

$$S = \min \sum_{i=0}^{n} e_i^2 = \sum_{i=0}^{n} [f(x_i, a_1, a_2, ...) - y_i]^2. \qquad (6.10)$$

Минимум функции находят, приравнивая нулю частные производные по переменным параметрам a_j:

$$\frac{dS}{da_1} = 0, \frac{dS}{da_2} = 0, \ \ldots \qquad (6.11)$$

Полученные соотношения образуют систему уравнений для определения коэффициентов a_j, $j = 1, 2, \ldots, m$.

Пусть функция $f(x)$ является многочленом степени m, т. е.:

$$f(x) = a_0x^m + a_1x^{m-1} + ... + a_ix^{m-i} + ... + a_{m-2}x^2 + a_{m-1}x + a_m$$

(a_0 не равно 0).

Задача ставится следующим образом: подобрать коэффициенты многочлена так, чтобы сумма квадратов отклонения для данного многочлена оказалась минимальной.

В случае $m = 1$ имеем линейное приближение функции по методу наименьших квадратов, в случае $m = 2$ – квадратичное приближение.

В случае линейной зависимости предполагается, что все точки лежат на некоторой прямой:

$$y = ax + b, \qquad (6.12)$$

где a и b – некоторые постоянные параметры, подлежащие определению. Для нахождения их требуется решить систему уравнений:

$$\begin{cases} a\Sigma x_i^2 + b\Sigma x^i = \Sigma x^i y^i, \\ a\Sigma x_i + bn = \Sigma y_i. \end{cases} \qquad (6.13)$$

В случае квадратичной зависимости предполагается, что все точки лежат на некоторой параболе. В этом случае естественно предположить, что между ними существует квадратичная зависимость, т. е.:

$$y = ax^2 + bx + c, \qquad (6.14)$$

где a, b, c – постоянные параметры, подлежащие определению. Они находятся из системы уравнений:

$$\begin{cases} a\Sigma x_i^4 + b\Sigma x_i^3 + c\Sigma x_i^2 = \Sigma x_i^2 y_i, \\ a\Sigma x_i^3 + b\Sigma x_i^2 + c\Sigma x_i = \Sigma x_i y_i, \\ a\Sigma x_i^2 + b\Sigma x_i + cn = \Sigma y_i. \end{cases} \qquad (6.15)$$

Решение системы уравнений осуществляется любым известным методом.

Для общего случая можно записать следующее выражение для составления системы уравнений:

$$\begin{cases} \sum_i \left[(f(x_i) - y_i) \cdot \frac{df(x_i)}{d(a_1)} \right] = 0, \\ \ldots \\ \sum_i \left[(f(x_i) - y_i) \cdot \frac{df(x_i)}{d(a_m)} \right] = 0. \end{cases} \qquad (6.16)$$

После того как найдены значения коэффициентов систем уравнений (6.13) и (6.15), вычисляют значения выражений (6.12) и (6.14) при заданных значениях аргументов. Каждое из полученных уравнений удовлетворяет условию (6.10).

Из двух полученных функций предпочтительной является функция, для которой сумма квадратов отклонений будет наименьшая.

Из приведенного алгоритма видно, что нахождение коэффициентов аппроксимирующей функции – достаточно трудоемкая задача. К счастью, электронная таблица Excel имеет ряд технологий для решения подобных задач: использование линий тренда на графиках функций и использование встроенных функций.

6.6.4. Оценка параметров выбранной модели регрессии

Для оценки параметров выбранной модели регрессии и оценки самих данных применяется ряд коэффициентов: *стандартная ошибка коэффициентов, стандартная ошибка оценки y, число степеней свободы, F-статистика, регрессионная сумма квадратов и остаточная сумма квадратов.*

Стандартная ошибка для оценки y (выборочное стандартное отклонение остатков) является несмещенной оценкой дисперсии σ^2 нормальной случайной величины ε вычисляется по формуле:

$$S_{y-x} = \sqrt{\frac{\sum_{i=1}^{n} (y_i - y(x_i))^2}{p}}, \qquad (6.17)$$

где p – число степеней свободы;

y_i – заданное значение функции;
$y(x_i)$ – вычисленное значение функции.

Число степеней свободы p равняется разности числа точек и числа коэффициентов в уравнении регрессии. Например, уравнение прямой линии имеет два коэффициента, соответствующие углу наклона прямой и сдвигу по оси *y*. Если число заданных точек равно 10, то число степеней свободы будет равно 8.

Стандартные ошибки выборочных коэффициентов регрессии S_A и S_B вычисляются по формулам:

$$S_A = \sqrt{\frac{1}{n} + \frac{\langle x \rangle^2}{\sum_{i=1}^{n}(x_i - \langle x \rangle)^2}} S_{y-x}, \text{ где } \langle x \rangle = \frac{\sum_{i=1}^{n} x_i}{n}. \qquad (6.18)$$

F-статистика – расчетное (выборочное) значение статистики *F*. Используется совместно с *F-значениями* для оценки вероятности того, что данные действительно описываются этими функциями, а не вызваны случайными флуктуациями.

F-значения вычисляются с помощью функции FРАСПОБР(). Если значения *F-статистики* больше соответствующего значения *F-значения*, то это означает, что совпадение обусловлено реальной корреляцией, а не случайными флуктуациями.

Коэффициент детерминации вычисляется по формуле:

$$r^2 = 1 - \frac{SS_{\text{ост}}}{SS_{\text{общ}}}. \qquad (6.19)$$

Общая сумма отклонений – сумма квадратов отклонений фактических значений *y* от ее выборочного среднего вычисляется по формуле:

$$SS_{\text{общ}} = \sum_{i=1}^{n}\left(y_i - \langle y_i \rangle\right)^2, \text{ где } \langle y_i \rangle = \left(\sum_{i=1}^{n} y_i\right)/n. \qquad (6.20)$$

Остаточная сумма квадратов (сумма наименьших квадратов) вычисляется по формуле:

$$SS_{\text{ост}} = \sum_{i=1}^{n}\left[y_i - y(x_i)\right]^2. \qquad (6.21)$$

Регрессионная сумма квадратов вычисляется по формуле:

$$SS_{рег} = \sum_{i=1}^{n}\left[y(x_i) - \langle y_i \rangle\right]^2, \text{ где } \langle y_i \rangle = \left(\sum_{i=1}^{n} y_i\right)/n. \qquad (6.22)$$

Между тремя суммами: общей, регрессионной и остаточной – имеется определенное соотношение:

$$SS_{\text{общ}} = SS_{\text{рег}} + SS_{\text{ост}}. \qquad (6.23)$$

Коэффициент детерминации может принимать значения от 0 до 1. Чем больше этот коэффициент, тем ближе располагаются точки линии тренда к экспериментальным точкам на графике. Приближение считается хорошим, если R^2 больше 0,9. Если $R^2 = 1$, то это означает полное совпадение прогнозируемых и фактических данных.

6.6.5. Вычисление коэффициентов эмпирических формул «вручную»

Если составлена система уравнений для определения коэффициентов эмпирических формул (6.13) или (6.15), то коэффициенты эмпирических формул можно вычислить вручную, используя уже известные приемы работы с матрицами. Одновременно познакомимся и с некоторыми другими функциями.

Составим таблицу (листинг 6.10), куда включим все коэффициенты, которые необходимо определить, и дополнительно среднее значение, которое понадобится нам при вычислении коэффициента детерминации (6.19).

Вычислим значения x^2 и xy.

Вычислим суммы значений параметров таблицы. Для вычисления сумм можно воспользоваться кнопкой ***Автосумма*** в группе ***Редактирование*** вкладки ***Главная*** или функцией ***СУММ***: =СУММ(A3:A12).

Полученные суммы представляют собой коэффициенты системы уравнений. Составим матрицу коэффициентов и вектор свободных членов согласно (6.13).

Решим систему линейных алгебраических уравнений матричным способом (с использованием обратной матрицы).

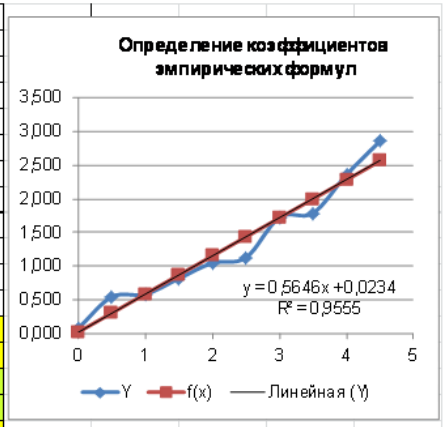
Вычислим значение $f(x) = ax + b$ (6.12).

Вычислим среднее значение для вектора *y* с помощью функции СРЗНАЧ: =СРЗНАЧ($B$3:$B$12) – и скопируем это во все ячейки таблицы.

Листинг 6.10. **Определение коэффициентов эмпирических формул**

X	Y	X^2	X*Y	f(x)	Среднее
0	0,078	0	0,000	0,023	1,294
0,5	0,545	0,25	0,273	0,306	1,294
1	0,584	1	0,584	0,588	1,294
1,5	0,817	2,25	1,226	0,870	1,294
2	1,049	4	2,098	1,153	1,294
2,5	1,128	6,25	2,820	1,435	1,294
3	1,728	9	5,184	1,717	1,294
3,5	1,789	12,25	6,262	2,000	1,294
4	2,360	16	9,440	2,282	1,294
4,5	2,860	20,25	12,870	2,564	1,294
22,5	12,938	71,25	40,756	12,938	12,938

Матрица			Вектор		SSобщ
71,3	22,5		40,76		6,8810
22,5	10		12,94		SSост
Обратная матрица			Коэфф.		0,3061
0,05	-0,109	a=	0,565		R^2
-0,1	0,3455	b=	0,023		0,9555



Вычислим значения $SS_{общ}$ и $SS_{ост}$, для чего воспользуемся функцией СУММКВРАЗН: =СУММКВРАЗН(B3:B12;F3:F12) – для общей суммы (6.20) и =СУММКВРАЗН(B3:B12;E3:E12) – для остаточной суммы (6.21).

Вычислим значение коэффициента детерминации согласно (6.19).

По полученным данным построим графики значений y_i и $f(x)$. Для сравнения на диаграмме построена и линия тренда, на диаграмму выведено также значение R^2 и уравнение регрессии. Коэффициенты уравнения регрессии и величина достоверности аппроксимации совпадают с рассчитанными параметрами.

6.6.6. Подбор аппроксимирующей функции по графику

Коэффициенты эмпирических формул можно определить, подбирая вид аппроксимирующей функции по графику:

составьте таблицу значений экспериментальных данных X и Y (рис. 6.39);

постройте график функции (точечный или график);

щелкните правой кнопкой мыши по линии графика – открывается контекстное меню;

выберите в этом меню команду ***Добавить линию тренда***. Открывается окно диалога Линия тренда (рис. 6.40);

X	Y
0,1	**2,56**
0,2	**3,45**
0,3	**2,78**
0,4	**2,12**
0,5	**2,98**
0,6	**3,65**
0,7	**4,01**
0,8	**5,8**

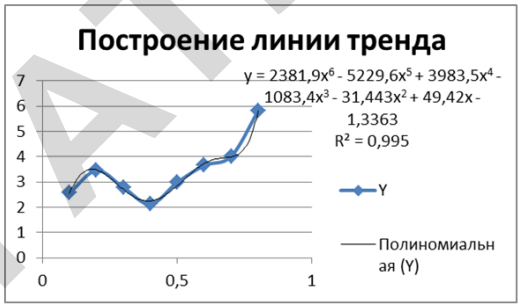


Рис. 6.39. Построение линии тренда

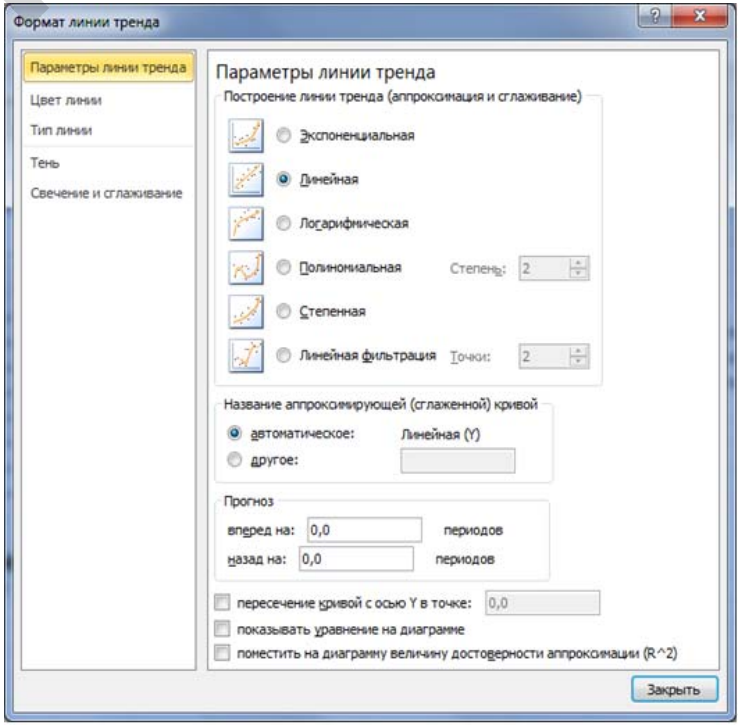


Рис. 6.40. Настройка параметров линии тренда

выберите в этом окне подходящую функцию, в данном примере полиномиальную, и выберите степень функции таким образом, чтобы она наиболее точно описывала экспериментальные данные. Установите флажки *Показывать уравнения на диаграмме* и *Поместить на диаграмму величину достоверности аппроксимации* (R^2).

6.6.7. Использование встроенных функций Excel для определения коэффициентов эмпирических формул

Электронная таблица Excel располагает встроенными средствами для определения коэффициентов эмпирических формул – это функции ЛИНЕЙН, ЛГРФПРИБЛ, ТЕНДЕНЦИЯ, РОСТ. Все эти функции возвращают множество точек аппроксимирующей кривой. Функции ЛИНЕЙН и ЛГРФПРИБЛ возвращают, кроме того, и коэффициенты уравнений регрессии.

Для поиска коэффициентов эмпирических формул можно использовать и возможности пакета Анализ.

Использование функции ЛИНЕЙН

Функция **ЛИНЕЙН** использует модель многомерной линейной регрессии

$$y(x_{1,i}, x_{2,i},\ldots) = A + Bx_{1,i} + Cx_{2,i} + \ldots$$

Функция имеет следующий синтаксис:

ЛИНЕЙН(Y – массив; X – массив; конст; статистика),

где *Y – массив* – ссылка на массив данных *Y*;

X – массив – ссылка на один или несколько массивов данных *x*;

конст – логическое значение, определяющее константу сдвига. Она может принимать два значения: ИСТИНА (1) и ЛОЖЬ (0). Если *конст* равна 1, то коэффициент A вычисляется обычным образом, иначе коэффициент A будет равен 0 для функции ЛИНЕЙН и 1 для функции ЛГРФПРИБЛ;

статистика – логическое значение, которое указывает, требуется ли вернуть дополнительную статистику регрессии: *стандартная ошибка коэффициентов*, *стандартная ошибка оценки y*, *число степеней свободы*, *F-статистика*, *регрессионная сумма квадратов* и *остаточная сумма квадратов*.

Если вектор X один, то уравнение регрессии будет представлять уравнение прямой линии $y = a + bx$.

Пример использования функции ЛИНЕЙН приведен на листинге 6.11.

Листинг 6.11. **Линейная аппроксимация**

	A	B	C	D	E	F	G	H
	Линейная аппроксимация							
1	***X***	***Y***	***Ожидаемое***					
2	250	0,445	0,3616		***Таблица регрессии***			
3	300	0,362	0,3363			B	A	Комментарий
4	350	0,302	0,3111			–0,00050	0,487797	Коэффициенты
5	400	0,256	0,2858		S_A, S_B	0,00006	0,033366	Стд. ошибка коэф.
6	450	0,223	0,2606		R^2	0,87542	0,038741	Коэф. детерминации
7	500	0,197	0,2353		F	77,29486	11	Степени свободы
8	550	0,176	0,2101		Сумма кв.	0,11601	0,01651	
9	600	0,158	0,1848			регрес-сионная	остаточ-ная	

Оформите таблицу регрессии в соответствии с листингом 6.11.

Введите в столбцы A и B экспериментальные значения X и Y.

Введите в ячейки F3 и G3 обозначение коэффициентов регрессии – символы B и A. Присвойте ячейкам F4 и G4 имена B и A соответственно (чтобы не использовать абсолютный адрес при копировании формул).

В ячейку C2 введите формулу A+B*A2 (то есть используется простейшая формула линейной аппроксимации y = A + Bx) и скопируйте эту формулу в соответствующие ячейки колонки C.

Выделите блок F4:G8. Введите в первую ячейку выделенного блока функцию ЛИНЕЙН(B2:B9;A2:A9,1,1) и вставьте ее во весь блок комбинацией клавиш Ctrl + Shift + Enter.

Результат: y = –0,00050x +0,487797.

Имеет место достаточно высокое совпадение результатов регрессионного анализа с исходными данными, так как R^2 = 0,87542.

Использование функции ЛГРФПРИБЛ

Функция **ЛГРФПРИБЛ** реализует следующую модель:

$$y(x_1, x_2, \ldots) = A(B^{x_1})(C^{x_2}) \ldots$$

Синтаксис функции:

ЛГРФПРИБЛ(Y – массив; X – массив; конст; статистика).

Пример использования функции ЛГРФПРИБЛ приведен на листинге 6.12.

Оформите таблицу регрессии в соответствии с листингом 6.12.

Листинг 6.12. **Логарифмическое приближение**

	A	B	C	D	E	F	G	H	I
1		**Логарифмическое приближение**							
2	***X1***	***X2***	***Y***	***Ожидаемое***					
3	250	1	0,445	0,3833	**Таблица регрессии**				
4	300	1,5	0,362	0,3408		C	B	A	
5	350	2	0,302	0,3031		0,338465	1,0085	0,135841	Коэффициенты
6	400	2,5	0,256	0,2695		0	0	0	Станд. ошибка коэф.
7	450	3	0,223	0,2396	R^2	0,975031	0,08022	#Н/Д	Стд. ошибка оценки Y
8	500	3,5	0,197	0,2130	F	195,2452	10	#Н/Д	Степени свободы
9	550	4	0,176	0,1894	Суммы кв.	2,51316	0,06435	#Н/Д	
10	600	4,5	0,158	0,1684		↑ регрессионная	↑ остаточная		
11	650	5	0,144	0,1497					
12	700	5,5	0,132	0,1331					

Функция логарифмического приближения применяется аналогично функции линейного приближения. Если имеется два вектора X, то в качестве блока аргумента *x* указывать область A2:B12. Векторы X1 и X2 не должны совпадать.

Введите в столбцы A, B и C заданные значения X1, X2 и Y.

Введите в ячейки F3, G3 и H3 обозначение коэффициентов регрессии – символы C, B и A. Присвойте ячейкам F5, G5 и H5 имена C, B и A соответственно.

В ячейку D3 введите формулу $A(B^{x_1})(C^{x_2})$, то есть ограничимся двумя векторами X, и скопируем эту формулу в соответствующие ячейки колонки D.

Выделите блок F5:H9. Введите в первую ячейку выделенного блока функцию ЛГРФПРИБЛ(C3:C12;A3:B12,1,1) и вставьте ее во весь блок.

Результат: $y = 0{,}135841 \cdot (1{,}0085^{X1}) \cdot (0{,}338465^{X2})$.

Степенная регрессия

Функции Excel не рассчитаны на выполнение степенной регрессии, но функцию ЛИНЕЙН можно легко приспособить для вычисления коэффициентов эмпирических формул с использованием степенной регрессии $y = A + Bx + Cx^2 + \ldots$. Для этого в выражении

множественной регрессии вводят следующие замены: $x_{1,i} = x_i$; $x_{2,i} = x_i^2$; $x_{3,i} = x_i^3 \ldots$

Пример использования степенной регрессии приведен на листинге 6.13.

Листинг 6.13. **Степенная регрессия**

	A	B	C	D	E	F	G	H
1		**Степенная регрессия**						
2	X	X^2	X^3	Y	Ожидаемое			
3	250	62500	15625000	0,445	0,4399478			
4	300	90000	27000000	0,362	0,3661319			
5	350	122500	42875000	0,302	0,3062602			
6	400	160000	64000000	0,256	0,2586284			
7	450	202500	91125000	0,223	0,2215317			
8	500	250000	125000000	0,197	0,1932657			
9	550	302500	166375000	0,176	0,1721259			
10	600	360000	216000000	0,158	0,1564076			
11	650	422500	274625000	0,144	0,1444063			
12	700	490000	343000000	0,132	0,1344176			
13	750	562500	421875000	0,121	0,1247368			
14	800	640000	512000000	0,112	0,1136593			
15	850	722500	614125000	0,103	0,0994808			
16								
17	**Таблица регрессии**							
18		D	C	B	A	Комментарий		
19		–2E–09	4,83432E–06	–0,0036	1,0778511	Коэффициенты		
20		2E–10	3,59808E–07	0,00019	0,0303105	Стд. ошибка коэф.		
21	R^2	0,999	0,003893183	#Н/Д	#Н/Д	Стд. ошибка оценки Y		
22	F	2911,4	9	#Н/Д	#Н/Д	Степень свободы		
23	Сумма кв.	0,1324	0,000136412	#Н/Д	#Н/Д			

Контрольные вопросы

1. Дайте постановку задачи для определения коэффициентов эмпирических формул методом наименьших квадратов.
2. В чем заключается суть метода наименьших квадратов?
3. Запишите уравнения линейной и квадратичной зависимостей.
4. Как построить графики функций линейной и квадратичной зависимостей?
5. Как построить линию тренда и вывести на график уравнение регрессии и значение коэффициента детерминации?
6. Какие функции Excel могут использоваться для определения коэффициентов эмпирических формул?
7. Запишите уравнения регрессии для линейной и логарифмической функций.

8. Запишите уравнение квадратичной (степенной) регрессии и поясните, как ее можно реализовать с помощью функции ЛИНЕЙН.

Заключение

Программа Excel имеет большое число встроенных функций, а также специальный Пакет анализа для обработки данных. В настоящем разделе мы ознакомились с некоторыми возможностями программы. Для определения коэффициентов эмпирических формул по данным, представленным в виде таблицы зависимости функции от аргумента, могут использоваться функции ЛИНЕЙН, ЛГРФПРИБЛ, а также графические средства, в частности линия тренда графика функции.

6.7. РЕШЕНИЕ АЛГЕБРАИЧЕСКИХ И ТРАНСЦЕНДЕНТНЫХ УРАВНЕНИЙ

6.7.1. Методы, основанные на табулировании функций

Процесс нахождения корней состоит, как известно, из двух этапов: *отделения корней* и *уточнения значения корней* на отрезках отделения с заданной точностью (см. раздел. 3.3.3).

Отделением корней называется процесс выделения из области определения функции f отрезков $[a; b]$, в каждом из которых содержится один и только один корень уравнения $f(x) = 0$.

Корни уравнения могут находиться на интервалах, определяемых переменой знака функции, между критическими точками. К критическим относятся точки, в которых производная от функции $f(x)$ обращается в нуль, а также граничные точки.

В электронной таблице отделение корней можно выполнить путем табулирования функции с некоторым, достаточно малым, шагом. Областями отделения корней будут значения аргументов, между которыми происходит смена знака функции. Можно воспользоваться также графическим методом: построить график функции на заданной области определения функции и выделить отрезки, на которых функция меняет знак.

Под **уточнением значения** корня с заданной точностью ε понимают сужение границ отрезка $[a; b]$ до длины, не превосходящей h. Уточнение значения корня на отрезке отделения осуществляется различными методами одномерной поисковой оптимизации: простых

итераций, деления отрезка пополам, касательных, хорд, наискорейшего спуска и др.

Для уточнения значения корня в Excel могут применяться все названные методы. В настоящем пособии рассмотрены три метода:

простое табулирование;

метод простых итераций;

метод Ньютона (метод касательных).

Метод простого табулирования не имеет ограничений, но малоэффективен, требует большого числа ручных операций. Условием окончания процедуры поиска является достижение функцией $f(x)$ заданного значения: $f(x_i) \le \varepsilon$, где ε – заданные требования к точности поиска корня. Можно использовать и другое условие окончания поиска: $h \le \varepsilon$, где h – шаг табулирования функции. Это правило можно применить, когда функция медленно меняется и функция принимает малое значение при относительно большом шаге табулирования.

Метод простых итераций более эффективен. Он сходится, если $f'(x) < 0$. Для его реализации необходимо функцию $f(x) = 0$ преобразовать к рекуррентному виду:

$$x_{i+1} = \varphi(x_i). \qquad (6.24)$$

Табулировать необходимо правую часть выражения (6.24). Для первой формулы в качестве аргумента используется начальное приближение x_0 (как правило, одна из границ отрезка отделения), для последующих формул – значение корня на предыдущем шаге, т. е. $f(x_i)$. Начальное приближение выбирается произвольно. Условие окончания процедуры вычисления: $\varphi(x_{i+1}) - \varphi(x_i) \le \varepsilon$.

Метод Ньютона – самый эффективный метод. Он обеспечивает сходимость за минимальное число шагов. Однако этот метод накладывает серьезные ограничения на вид функции. Функция должна быть непрерывной на отрезке отделения и дважды дифференцируемой. Для поиска корня в этом методе, так же, как и в методе простых итераций, составляется рекуррентная формула:

$$x_{i+1} = x_i - f(x_i)/f'(x_i). \qquad (6.25)$$

Табулировать необходимо правую часть выражения (6.25). Начальное приближение выбирается на одной из границ отрезка отделения корня. В качестве начального приближения x_0 выбирается граница $\boldsymbol{b}$, если $f'(x)f''(x) > 0$, и граница $\boldsymbol{a}$, если $f'(x)f''(x) < 0$. Для первой формулы в качестве аргумента используется начальное

приближение x_0, для последующих формул – значение корня на предыдущем шаге, т. е. x_i. Условием окончания процедуры уточнения корня является достижение функцией значения, меньшего или равного заданному, $-f(x_{i+1}) \le \varepsilon$.

Приближенное значение производных можно вычислить численно с помощью формул (6.6), (6.7).

Значение Δx можно принять в интервале от 0,0001 до 0,00001.

Однако проще построить график функции и определить начальное приближение по виду функции, применяя известные правила:

- Если функция возрастает, то первая производная положительная, а если убывает, то первая производная отрицательная.
- Если функция выпуклая вниз (вогнутая), то вторая производная положительная, а если выпуклая вверх, то вторая производная отрицательная.

Все указанные методы могут быть реализованы с помощью функций пользователя, разработанных с помощью встроенного языка программирования Visual Basic for Application (VBA).

Пример 6.7. Пример отделения и уточнения корня

Пусть требуется найти корни уравнения $\boldsymbol{y = 2^x + 2x - 5}$ с точностью 0,001.

Загрузите электронную таблицу.

1. Отделите корни уравнения. Для этой цели протабулируйте функцию на значительном отрезке с большим шагом и зафиксируйте границы смены знака функции. После нескольких шагов получим отрезок отделения [1,28; 1,285]. Выберите шаг 0,0005 и протабулируйте функцию вновь. Получен новый интервал, где функция меняет свой знак [1,283; 1,2835] (листинг 6.14). На границе отрезка при x = 1,283 значение функции равно 0,000555. То есть функция меньше заданной точности. Эту точку и примем в качестве значения корня заданной функции. ***Результат***: x = 1,283, $f(x)$ = 0,000555.

2. Уточните значение корня на отрезке отделения [1,283; 1,2835] методом итераций:

напишите рекуррентную формулу. В заданном выражении это обеспечивается просто путем переноса второго слагаемого в левую часть уравнения и делением на коэффициент при неизвестной:

$$x = (5 - 2\text{^}x)/2; \qquad (6.26)$$

запишите в ячейку C11 формулу (6.26). В качестве начального значения x примите значение функции на левой границе отрезка отделения, то есть 1,28 (ячейка D4);

Листинг 6.14. **Решение алгебраических и нелинейных уравнений**

	A	B	C	D	E	F
1	***Решение нелинейных уравнений y=2^x+2x–5***					
2	1. Простое табулирование		2. Метод итераций		3. Метод Ньютона	
3	Начальное значение	0	Нач. знач.	1,28	Нач. знач.	1,28
4	Шаг табуляции	0,0005	Шаг таб.	нет	Шаг таб.	нет
5	Точность	0,001	Точность	0,001	Точность	0,001
6	Формула y=2^x+2x–5		y=2^x+2x–5		y=2^x+2x–5	
7			Рекуррентная формула x=(5–2^x)/2		x=x–(2^x+2x–5)/(2^x* *LN(2)+2)	
8	X	Функция f(x)	Функция $\varphi(x_i)$	Разность $ABS(x_{i+1}-x_i)$	Функция $x_i-f(x_i)/f'(x_i)$	Функция f(x)
9	1.28	–.011610	1,285805		1,80719	2,11396
10	1.2805	–.009768	1,28091	0,0048955	1,32953	0,17228
11	1.281	–.007926	1,285039	0,0041297	1,28349	0,00127
12	1.2815	–.006084	1,281557	0,0034828	1,28315	0,00000
13	1.282	–.004241	1,284494	0,0029379		
14	1.2825	–.002399	1,282017	0,0024778		
15	1.283	–.000555	1,284107	0,0020901		
16	1.2835	.0012882	1,282344	0,0017628		
17	1.284	.0031320	1,283831	0,0014869		
18	1.2845	.0049761	1,282577	0,0012541		
19	1.285	.0068205	1,283635	0,0010578		
20			1,282742	0,0008922		

запишите в ячейку C12 формулу (6.26), но в качестве x примите значение ячейки C11, то есть значение корня на предыдущем шаге;

запишите в ячейку D12 формулу ABS(C11–C12);

скопируйте формулы из ячеек C12, D12 в нижележащие ячейки, пока в колонке D не будет выполнено требование к разности двух смежных значений аргументов.

Результат: x = 1,282742, $f(x)$ = 0,00089.

3. Уточните значение корня на отрезке отделения [1,28; 1,285] методом Ньютона:

найдите первую производную для заданного выражения:

$$f'(x) = (2 \cdot 2\text{^}x \cdot \mathrm{Ln}(2) + 2); \qquad (6.27)$$

$$x_{i+1} = x_i - (2\text{^}x_i + 2x_i - 5)/(2 \cdot 2\text{^}x_i \cdot \mathrm{Ln}(2) + 2); \qquad (6.28)$$

запишите в ячейку E11 формулу (6.28). В качестве начального значения x примите значение функции на левой границе отрезка отделения, то есть 1,28 (ячейка F4);

запишите в ячейку E12 формулу (6.28), но в качестве *x* примите значение ячейки E11, то есть значение корня на предыдущем шаге;

запишите в ячейку F11 формулу $\mathbf{2^x + 2x - 5}$ со ссылкой на ячейку E11;

скопируйте формулы из ячеек E12 и F11 в нижележащие ячейки, пока в колонке F не будет выполнено требование к точности значения функции.

Результат: x = 1,283141; $f(x)$ = –3,5E–05.

Из приведенных примеров наглядно видно, что метод Ньютона (метод касательных) обеспечивает гораздо более быструю сходимость, чем метод простых итераций.

6.7.2. Использование встроенных процедур

Другим способом решения линейных и нелинейных уравнений является использование возможностей программы Excel по оптимизации решений. Для этой цели служат модули ***Подбор параметра*** и ***Поиск решения*** из вкладки ***Данные*** ленты. Модуль ***Подбор параметра*** находится в группе ***Работа с данными***, команда ***Анализ «Что если»***. Команда ***Поиск решения*** находится в группе ***Анализ***. Если команды Поиск решения нет в группе Анализ, то ее необходимо загрузить (см. раздел 6.1.9). Модуль Поиск решения может использоваться для решения различных задач оптимизации, в том числе и задач линейного программирования, транспортных задач и др.

Использование возможностей Excel по оптимизации рассмотрим на примерах.

Пример 6.8. Решим квадратное уравнение $\mathbf{3x^2 + 2x - 5}$ с точностью 0,0001.

Результаты решения обоими методами приведены на листинге 6.15.

Порядок решения уравнения с использованием модуля Подбор параметра

Введите в ячейку B11 формулу 3*C11^2+2*C11–5. В ячейку C11 введите начальное значение x = –1. Результат приведен на листинге в столбцах B и C.

Введите команду ***Данные, Анализ «Что если», Подбор параметра*** – открывается одноименное окно диалога (рис. 6.41).

Введите в строку «Установить в ячейке» адрес ячейки, содержащей формулу, – B11, в строку «Значение» введите значение 0, а в строку «Изменяя значение ячейки» – адрес ячейки, содержащий значение X, – C11 с абсолютным адресом. Щелкните по кнопке ОК. В ячейке B11 отобразится значение точности поиска решения, а в ячейке C11 – значение корня. Для сравнимости результатов на листинге 6.15 в левой таблице показано исходное состояние, а в правой – результаты. Результат решения зависит от начального значения корня. Сравните результаты в строках 12 и 13. Функция Подбор параметра позволяет найти только один корень уравнения. Для получения значения второго корня необходимо изменить начальное значение. То есть предварительно надо исследовать функцию, например, графическим методом и отделить корни уравнения. В строке 13 приведен результат поиска второго корня.

Листинг 6.15. **Решение уравнений**

	A	B	C			A	B	C
8		**Подбор параметра**			8		**Подбор параметра**	
9		**Исходное состояние**			9		**Результат решения**	
10		Функция	Аргумент		10		Функция	Корень
11	1	–4,0000E+00	–1,0000		11	1	–8,4213E–04	–1,6666
12	2	6,0000E+01	–5,0000		12	2	4,9306E–04	–1,6667
13	3	8,0000E+01	5,0000		13	3	1,5199E–05	1,0000
14		**Поиск решения**			14		**Поиск решения**	
15		**Исходное состояние**			15		**Результат решения**	
16		Функция	Аргумент		16		Функция	Корень
17	1	–4,0000E+00	–1,0000		17	1	3,2259E–07	–1,6667
18	2	6,0000E+01	–5,0000		18	2	–4,1624E–07	–1,6667
19	3	8,0000E+01	5,0000		19	3	0,0000E+00	1,0000
20	4	6,0000E+01	–5,0000		20	4	0,0000E+00	1,0000

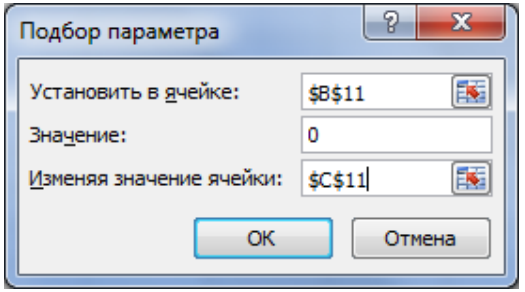


Рис.6.41. Окно диалога модуля Подбор параметра

Порядок решения уравнения с использованием модуля Поиск решения

Введите в ячейку B17 формулу 3*C17^2+2*C17–5. В ячейку C17 введите начальное значение x = –1. Результат приведен на листинге 6.16 в столбцах B и C.

Введите команду ***Данные, Поиск решения*** – открывается одноименное окно диалога (рис. 6.42).

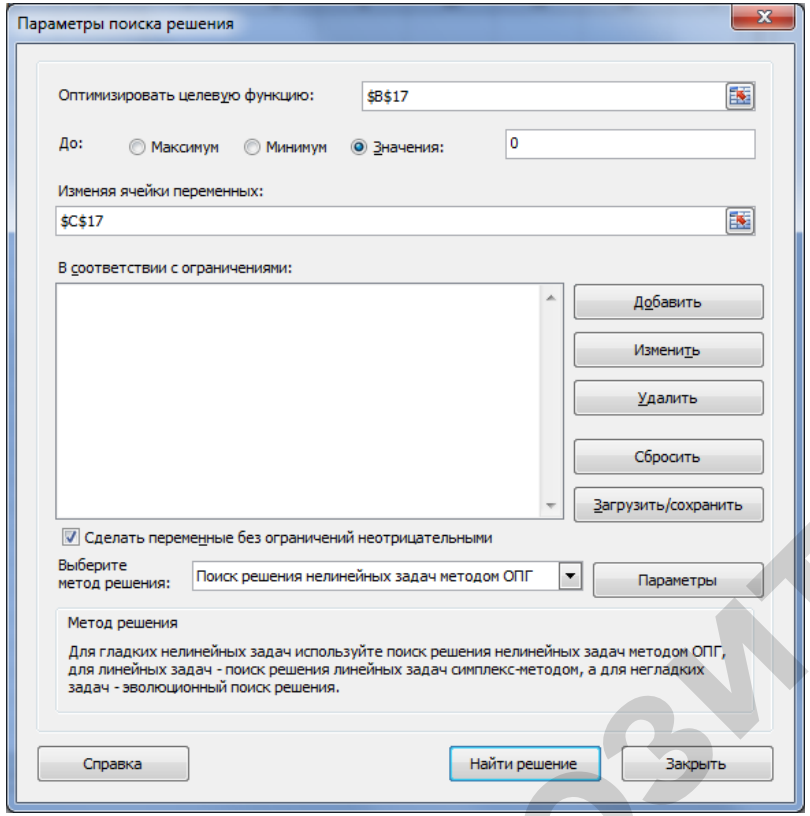


Рис. 6.42. Поиск решения

Введите в строку «*Оптимизировать целевую функцию*» адрес ячейки, содержащей формулу, – B17, установите переключатель «*Значения*» и в строку ввода «значения» введите значение 0, а в строку «*Изменяя ячейки переменных*» введите адрес ячейки, содержащий значение X – C17 с абсолютным адресом. Щелкните по кнопке ***Найти решение***. Открывается окно диалога

«Результаты поиска решения» (рис. 6.43). Для получения результата щелкните по кнопке OK. В ячейке B17 отобразится значение точности поиска решения, а в ячейке C17 – значение корня. Для сравнимости результатов на листинге 6.15 приведены исходные состояния и результаты. Результат решения не зависит от выбранного начального значения корня. Сравните результаты в строках 17 и 18. Однако модуль Поиск решения, так же, как и модуль Подбор параметра, позволяет найти только один корень уравнения.

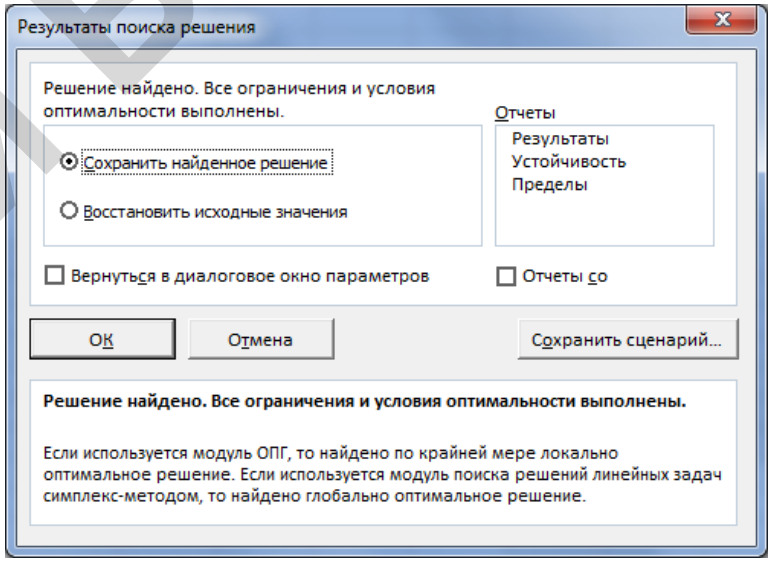


Рис. 6.43. Окно диалога Результаты поиска решения

Для получения значения второго корня необходимо изменить начальное значение. В строке 19 приведен результат поиска второго корня. Второй корень можно найти при любом начальном значении X, если установить ограничения на его значение. Щелкните по кнопке ***Добавить*** в окне диалога Поиск решения (рис. 6.42). Открывается окно диалога «*Добавление ограничения*» (рис. 6.44). Выберите в списке «*Ссылка на ячейки*» адрес ячейки, содержащей значение *x*, например, C20, в среднем списке – знак отношения >=, а в списке «*Ограничение*» введите значение ограничения, например, 0, и щелкните по

кнопке ОК. Результат будет записан в окне «*В соответствии с ограничениями*» окна диалога Поиск решения (рис. 6.42). Пример решения приведен в строке 20 листинга 6.15.

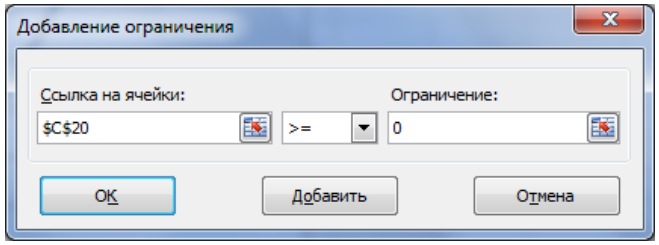


Рис. 6.44. Окно диалога Добавление ограничений

Из анализа результатов на листинге 6.15 можно сделать следующие выводы. Модуль Поиск решения дает более точные результаты по сравнению с модулем Подбор параметра, при этом результат не зависит от начального приближения. Функция может найти только одно решение, поэтому поиск значения корня необходимо вести на отрезке отделения.

Пример 6.9. Задача о пожарном ведре. Дана заготовка из жести в виде круга диаметром R = 0,75 м (рис. 6.45). Требуется выкроить из него конусообразное ведро таким образом, чтобы объем ведра был наибольший.

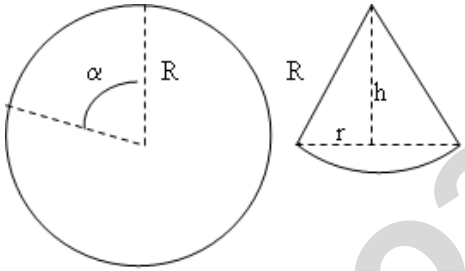


Рис. 6.45. К примеру 6.9

Разработаем математическую модель.

Объем пожарного ведра $Q = 1/3hS_{осн}$, $S_{осн} = \pi r^2$, где r – радиус основания конуса; h – высота ведра; $h = \sqrt{R^2 - r^2}$

Радиус основания зависит от угла вырезки α. Длина окружности основания ведра:

$$L = (2\pi - \alpha)R, \text{ или } L = 2\pi r, \text{ отсюда } r = \frac{(2\pi - \alpha)R}{2\pi},$$

$$Q = \frac{1}{3}\sqrt{R^2 - \left[\frac{(2\pi - \alpha)R}{(2\pi)}\right]^2} \cdot \pi \left[\frac{(2\pi - \alpha)R}{(2\pi)}\right]^2. \qquad (6.29)$$

Объем ведра Q должен максимизироваться. Выражение (6.29) называют целевой функцией. Оптимизируемым параметром является угол α. Ограничение в данном выражении одно: угол α должен быть больше 0, но меньше 2π. Решение приведено на листинге 6.16.

Листинг 6.16. **Оптимизация**	
Задача о пожарном ведре	
Q=1/3*Корень(R^2–(((2π–α)R)/(2π))^2)*π*(((2π–α)R)/(2π))^2	
Исходные данные:	
R=	0,75
Угол альфа	Целевая функция
1,84	0,046875

Контрольные вопросы

1. В чем отличие алгебраического уравнения от трансцендентного уравнения?
2. Что называется корнем уравнения?
3. Что такое отделение корня? Для какой цели оно производится?
4. Какими способами можно отделить корни уравнения в электронной таблице?
5. Что такое уточнение значения корня, какими методами оно осуществляется?
6. Поясните, в чем заключается метод простых итераций?
7. Поясните принцип использования метода касательных для уточнения значения корня.
8. Что такое рекуррентная формула, как она получается?
9. Какие команды Excel могут использоваться для решения линейных и нелинейных уравнений?
10. Поясните порядок решения уравнения с помощью модуля Подбор параметра.
11. Поясните порядок решения уравнений с помощью модуля Поиск решения.

Заключение

Для решения алгебраических и трансцендентных уравнений в электронной таблице могут использоваться численные методы решения уравнений, основанные на различных математических методах одномерной поисковой оптимизации: простых итераций, деления отрезка пополам, Ньютона и др. Электронная таблица имеет встроенные модули для решения нелинейных и линейных уравнений Подбор параметра и Поиск решения. Однако эти процедуры находят только одно решение, ближайшее к заданному начальному приближению. Поэтому функцию необходимо предварительно исследовать и отделить корни уравнения, например, путем табулирования или построения графика функции.

Процедуры Подбор параметра и Поиск решения позволяют также решать задачи одномерной поисковой оптимизации.

6.8. РЕШЕНИЕ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ ПЕРВОГО ПОРЯДКА С НАЧАЛЬНЫМИ УСЛОВИЯМИ

Постановка задачи

Пусть некоторый процесс описывается дифференциальным уравнением первой степени:

$$\frac{du}{dx} = f(u, x)\cdot \qquad (6.30)$$

Известно значение процесса в некоторый момент времени и требуется оценить значение этого процесса в произвольный момент времени. Для приближенного решения этой задачи необходимо проинтегрировать данное уравнение от одного конца интервала с известными граничными значениями до другого конца интервала, на котором они неизвестны. Такие задачи получили название задачи Коши (см. раздел 2.2).

Задачи такого типа возникают обычно для уравнений с производными по времени. Для решения задач Коши могут использоваться разные методы: метод рядов Тейлора, метод Эйлера, модифицированный метод Эйлера и метод Рунге-Кутты. Эти методы отличаются сложностью используемых выражений, скоростью сходимости и точностью получаемых результатов.

Рассмотрим два метода: метод Эйлера и метод Рунге-Кутты.

Метод Эйлера

В методе Эйлера значение функции на следующем шаге вычисляется в соответствии с выражением:

$$u(x + h) = u(x) + hu'(x). \qquad (6.31)$$

Пример 6.8. Найти решение дифференциального уравнения:

$$\sqrt{1+x^2}\,\frac{du(x)}{dx} + u(x) = x \qquad (6.32)$$

при $x = 0{,}2$ с начальным условием $x(0) = 0$, $u(0) = 0$.

Решение. Приведем выражение (6.32) к виду выражения (6.30):

$$\frac{du(x)}{dx} = \frac{x - u(x)}{\sqrt{1+x^2}}, \qquad (6.33)$$

тогда согласно (6.31) получим следующее выражение:

$$u(x+h) = u(x) + h\frac{x - u(x)}{\sqrt{1+x^2}}\cdot \qquad (6.34)$$

Теперь для получения результата достаточно сгенерировать ряд значений аргумента x с некоторым шагом и протабулировать правую часть выражения (6.33) на отрезке от 0 до 0,2.

Для примера (6.32) известно аналитическое выражение для вычисления значения функции в произвольной точке:

$$u(x) = 0{,}5\left[x - \frac{\ln\left(x + \sqrt{1+x^2}\right)}{x + \sqrt{1+x^2}}\right], \qquad (6.35)$$

поэтому можно вычислить относительную погрешность вычисления по формуле:

$$\Delta = \frac{u(x)_a - u(x)}{u(x)_a}, \qquad (6.36)$$

где $u(x)_a$ – значение функции, вычисленное по аналитической формуле.

Порядок работы:

внесите в ячейки A6 и B6 (листинг 6.17) начальные значения аргумента и функции;

Листинг 6.17. **Решение дифференциального уравнения методом Эйлера**

	A	B	C	D
5	x	y	Аналитика	Погрешность
6	0	0,0000	0,0000	#ДЕЛ/0!
7	0,01	0,0000	0,0000	1
8	0,02	0,0001	0,0002	0,49664
9	0,03	0,0003	0,0004	0,32885
…	...			
24	0,18	0,0144	0,0152	0,04918
25	0,19	0,0160	0,0168	0,04624
26	0,2	0,0177	0,0186	0,04359

сгенерируйте в столбце A ряд значений аргумента от 0 до 0,2 с шагом 0,01;

запишите в ячейку B7 правую часть формулы (6.33) со ссылками на адреса ячеек:

B6+(A7–A6)*(A6–B6)/КОРЕНЬ(1+A6^2);

запишите в ячейку C6 правую часть формулы (6.35) со ссылками на адреса ячеек:

0,5*(A6–LN(A6+КОРЕНЬ(1+A6^2))/(A6+КОРЕНЬ(1+A6^2)));

запишите в ячейку D6 правую часть формулы (6.36) со ссылками на адреса ячеек: (C6–B6)/C6;

скопируйте формулу из ячеек B7, C6, D6 в нижележащие ячейки.

Результаты:

Методом Эйлера: $u(0,2) = 0,0177$.

По аналитической зависимости: $u(0,2) = 0,0186$.

Погрешность около 4 процентов: $\Delta = 0,04359$.

Метод Рунге-Кутты

Одним из наиболее распространенных методов решения дифференциального уравнения вида $y' = f(x,y)$ на заданном отрезке $[X_{нач}, X_{кон}]$ с начальными условиями является метод Рунге-Кутты (см. раздел 2.7.2). При решении данного уравнения точное значение y заменяют его приближенным значением:

$$Y_{i+1} = Y_i + (K_1 + 2K_2 + 2K_3 + K_4)/6, \qquad (6.37)$$

где значения коэффициентов на i-м шаге вычисляются по формулам:

$$\begin{aligned} K_1 &= hf(x_i,y_i);\\ K_2 &= hf(x_i + 0,5h,y_i + 0,5K_1);\\ K_3 &= hf(x_i + 0,5h,y_i + 0,5K_2);\\ K_4 &= hf(x_i + h,y_i + K_3). \end{aligned} \qquad (6.38)$$

Здесь $x_{i+1} = x_i + h$.

Значение шага при переходе к следующей точке можно изменять. Правильность выбора шага проверяется по формуле:

$$T = |(K_2 - K_3)/(K_1 - K_2)|, \qquad (6.39)$$

величина T не должна превышать нескольких сотых.

Для проверки точности необходимо сделать второй проход с шагом $h/2$. Если разница между Y_n на $k + 1$ и k проходах будет меньше требуемой точности, то процесс вычисления прекращается. Значение точности выбирают в интервале от 0,001 до 0,00001.

Грубую оценку погрешности метода проводят по формуле:

$$Y_k - Y(x_i) = |Y_{k+1} - Y_k|/15, \qquad (6.40)$$

где $Y(x_i)$ – значение точного решения уравнения.

Для решения задачи необходимо вычислить значения коэффициентов K_1, K_2, K_3, K_4 при начальных условиях, выбрать начальный шаг и вычислить значения X_i и Y_i. Затем увеличить значение x и повторить процедуру вычисления.

Пример 6.11. Решить дифференциальное уравнение первого порядка $dY/dx = 2(x^2 + Y)$ методом Рунге-Кутты при $0 \le x \le 1$, $Y(0) = 1$, шаг $h = 0,1$.

Для этой задачи известно аналитическое выражение:

$$Y = 1,5e^{2x} - x^2 - x - 0,5. \qquad (6.41)$$

Порядок работы:

- Оформите таблицу согласно листингу 6.18.
- Запишите исходные данные и шапку таблицы в строки 2 и 3.
- Сгенерируйте в ячейки A4:A14 значения аргумента.
- Запишите в ячейку B4 начальное значение Y(0).
- Запишите в ячейки расчетные формулы со ссылками на ячейки таблицы:

ячейка C5 – коэффициент K1: $B$2*2*(A4^2+B4);

ячейка D5 – коэффициент K2: $B$2*2*((A4+$B$2/2)^2+(B4+C5/2));

ячейка E5 – коэффициент K3: $B$2*2*((A4+$B$2/2)^2+(B4+D5/2));

ячейка F5 – коэффициент K4: $B$2*2*((A4+$B$2)^2+(B4+E5));

Листинг 6.18. **Решение дифференциального уравнения методом Рунге-Кутты**

	A	B	C	D	E	F	G
1	Исходные данные						
2	h=	0,1	Xn=	0,00000	Y(0)=	1,00000	Контроль выбора шага
3	Xn=	Yn=	K1=	K2=	K3=	K4=	
4	0,00	1,0000					
5	0,10	1,2221	0,2	0,2205	0,2225	0,2465	0,10
6	0,20	1,4977	0,2464	0,2735	0,2762	0,3076	0,10
7	0,30	1,8431	0,3075	0,3428	0,3463	0,3868	0,10
8	0,40	2,27829	0,3866	0,4318	0,4363	0,4879	0,10
9	0,50	2,8274	0,4877	0,5449	0,5507	0,6158	0,10
10	0,60	3,5201	0,6155	0,6875	0,6947	0,7764	0,10
11	0,70	4,3927	0,7760	0,8661	0,8751	0,9771	0,10
12	0,80	5,4894	0,9765	1,0887	1,0999	1,2265	0,10
13	0,90	6,8643	1,2259	1,3650	1,3789	1,5357	0,10
14	1,00	8,5834	1,5348	1,7069	1,7245	1,9177	0,10

ячейка B5 – формула для Y: B4+(C5+2*D5+2*E5+F5)/6.

- В ячейку G5 запишите формулу (6.40) для контроля выбора шага: (D5–E5)/(C5–D5).
- Скопируйте формулы из ячеек B5:G5 в нижележащие ячейки.

Сравнительные результаты решения дифференциального уравнения примера 6.10 методом Рунге-Кутты, методом Эйлера и по аналитической формуле (6.41) приведены на листинге 6.19. Из таблицы видно, что метод Рунге-Кутты позволяет получить результаты с высокой точностью во всем диапазоне изменения значения аргумента, чего нельзя сказать о методе Эйлера.

Контрольные вопросы

1. Сформулируйте постановку задачи Коши для решения дифференциального уравнения первого порядка.
2. Поясните порядок решения дифференциального уравнения методом Эйлера.
3. Поясните порядок решения дифференциальных уравнений методом Рунге-Кутты.

Заключение

Электронная таблица Excel не имеет встроенных функций для решения дифференциальных уравнений, однако позволяет решать их с использованием классических методов, например, метода Эйлера

или Рунге-Кутты. Метод Рунге Кутты, несмотря на его громоздкость, при той же величине шага дает более точные результаты по сравнению с методом Эйлера. Для получения результатов с заданной точностью необходимо использовать метод двойного прохода.

Листинг 6.19. **Сравнение методов решения дифференциальных уравнений**

x	Метод Эйлера	Метод Рунге–Кутты	Точное решение	x
0,0000	1,0000	1,0000	1,0000	0,0000
0,1000	1,2000	1,2221	1,2221	0,1000
0,2000	1,4420	1,4977	1,4977	0,2000
0,3000	1,7384	1,8432	1,8432	0,3000
0,4000	2,1041	2,2783	2,2783	0,4000
0,5000	2,5569	2,8274	2,8274	0,5000
0,6000	3,1183	3,5201	3,5202	0,6000
0,7000	3,8139	4,3927	4,3928	0,7000
0,8000	4,6747	5,4894	5,4895	0,8000
0,9000	5,7377	6,8643	6,8645	0,9000
1,0000	7,0472	8,5834	8,5836	1,0000

6.9. ПОИСК ОПТИМАЛЬНЫХ РЕШЕНИЙ СРЕДСТВАМИ MICROSOFT EXCEL

В области экономики, управления часто возникают задачи, требующие принятия решения при известных условиях, но зависящие от многих факторов. Математическая модель таких задач представляется системой линейных уравнений или неравенств.

Уравнение относительно переменных x_1, x_2, …, x_n называется линейным, если его можно записать в виде:

$$a_1x_1 + a_2x_2 + \ldots + a_nx_n = b, \qquad (6.42)$$

где a_1, a_2, …, a_n – произвольные вещественные числа, называемые *коэффициентами уравнения*;

b – произвольное вещественное число, называемое *свободным членом уравнения*.

Упорядоченная совокупность n вещественных чисел $(\alpha_1, \alpha_2, \ldots, \alpha_n)$ называется решением уравнения (6.42), если в результате подстановки этих значений вместо соответствующих переменных уравнение

обращается в арифметическое тождество. M уравнений вида (6.42) образуют систему уравнений. Способы решения систем линейных алгебраических уравнений были рассмотрены в разделе 6.5.

Кроме того, в экономике и управлении часто возникают задачи, сводящиеся к системе неравенств вида:

$$\begin{aligned} &a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \le b_1,\\ &a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \le b_2,\\ &\ldots\\ &a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \le b_m. \end{aligned} \qquad (6.43)$$

Решением неравенства (6.43) называется множество значений переменных $(\alpha_1, \alpha_2, \ldots, \alpha_n)$, при которых каждое неравенство системы становится верным числовым неравенством.

Несколько иного вида задача возникает, когда необходимо найти наилучшее решение для одного из уравнений системы при ограничении на другие уравнения системы. Такие задачи получили название задач многомерной оптимизации. Уравнение, для которого ищется наилучшее решение (максимизации или минимизации цели), называется целевой функцией, другие уравнения называются функциональными ограничениями. Кроме этого на значения параметров x_i также могут налагаться ограничения вида неотрицательности, целочисленности или диапазона значений. В результате приходим к следующей математической модели задачи.

Целевая функция:

$$F = a_1x_1 + a_2x_2 + \ldots + a_nx_n \Rightarrow (\max, \min, \text{константа}).$$

Ограничения функциональные:

$$\begin{aligned} &q_{11}x_1 + q_{21}x_2 + \ldots + q_{31}x_n \le b_1,\\ &q_{21}x_1 + q_{22}x_2 + \ldots + q_{2n}x_n \le b_2,\\ &\ldots\\ &q_{m1}x_1 + q_{m2}x_2 + \ldots + q_{mn}x_n \le b_m. \end{aligned}$$

Ограничения на значения параметров (граничные условия):

$$x_i \ge 0;\ i = 1, 2, \ldots, n.$$

Такие задачи получили название задач линейного программирования.

Целевая функция, или критерий оптимальности, показывает, в каком смысле решение должно быть наилучшим. При этом возможно три вида назначения целевой функции: максимизация, минимизация, назначение заданного значения.

Ограничения устанавливают зависимости между переменными. Они могут быть как односторонними, так и многосторонними:

$q_i(x_j) \le b_i$ – одностороннее ограничение,

$c_i \le q_i(x_j) \le b_i$ – двухстороннее ограничение.

Граничные условия показывают, в каких пределах могут быть значения искомых переменных в оптимальном решении. Можно потребовать, чтобы переменные были неотрицательными, например, выпуск продукции. Можно потребовать, чтобы переменные были целочисленные, например, трудовые ресурсы, готовая продукция (станки, оборудование и т. д.), можно также потребовать, исходя из конкретных условий, чтобы значения переменной находились в определенном диапазоне значений.

Решение задачи, удовлетворяющее всем ограничениям и граничным условиям, называется допустимым. Задача может иметь множество допустимых решений, но только одно из них будет наилучшим или оптимальным в свете выбранного критерия оптимальности.

Важной характеристикой задачи оптимизации является ее размерность, определяемая числом переменных n и числом ограничений m.

При $n < m$ задача не имеет решения, при $n = m$ задача имеет единственное решение, выбора нет, при $n > m$ задача имеет бесконечное множество решений.

Таким образом, для задач оптимизации обязательным требованием является выполнение условия $n > m$. В экономических задачах в качестве целевой функции чаще всего принимают минимум затрат, минимум простоя оборудования, максимум прибыли. Ограничениями выступают ресурсы: трудовые, материальные, сырьевые. Граничными условиями принимают неотрицательность и целочисленность переменных.

В зависимости от исходных данных, переменных и зависимости между переменными задачи оптимизации можно разделить на следующие группы:

1. В зависимости от исходных данных:
детерминированные;
случайные (стохастические).
2. В зависимости от исходных переменных:
непрерывные;
дискретные.
3. По зависимости между переменными:
линейные;
нелинейные.

Сочетания различных элементов модели образуют различные классы задач оптимизации, которые требуют применения различных методов решения (табл. 6.2).

Таблица 6.2

Классы задач оптимизации

Исходные данные	Искомые переменные	Зависимости между переменными	Классы задач
Детерминированные	Непрерывные	Линейные	Линейного программирования
	Целочисленные	Линейные	Целочисленного программирования
	Непрерывные, целочисленные	Нелинейные	Нелинейного программирования
Случайные	Непрерывные	Линейные	Стохастического программирования

Электронная таблица Excel имеет мощные средства для решения задач оптимизации. Для этой цели, как уже отмечалось ранее, имеется надстройка ***Поиск решения***, которая вызывается одноименной командой из группы ***Анализ*** вкладки ***Данные***.

Решение задач линейного программирования

Решение задач линейного программирования рассмотрим на примере.

Задача. Для изготовления штучной продукции 4-х видов требуются ресурсы трех видов: трудовые, сырьевые, финансовые. Известны нормы расхода трудовых и сырьевых ресурсов для выпуска продукции каждого вида, а также прибыль, получаемая от реализации продукции каждого вида, представленные в таблице 6.3.

Таблица 6.3

Исходные данные

Ресурсы	Продукт 1	Продукт 2	Продукт 3	Продукт 4	Знак	Наличие
Прибыль	60	70	120	130	max	
Трудовые	1	1	1	1	≤	16
Сырье	6	5	4	3	≤	110
Финансы	4	6	10	13	≤	100

Требуется определить план выпуска продукции, определяющий максимальную прибыль.

Решение:

Разработаем математическую модель решения задачи.

Обозначим выпускаемую продукцию x_i, $i = 1, 2, 3, 4$.

Целевая функция представляет собой сумму произведений выпускаемой продукции на прибыль, получаемую от реализации продукции каждого вида. Прибыль должна максимизироваться:

$$F = 60x_1 + 70x_2 + 120x_3 + 130x_4 \geq MAX.$$

Ограничения функциональные связывают объемы выпускаемой продукции, нормы расхода и запасы сырья:

$$x_1 + x_2 + x_3 + x_4 \leq 16;$$
$$6x_1 + 5x_2 + 4x_3 + 3x_4 \leq 110;$$
$$4x_1 + 6x_2 + 10x_3 + 13x_4 \leq 100.$$

Ограничения на значения параметров.

Так как продукция штучная, то, во-первых, она должна быть неотрицательной, а во-вторых, – целочисленной:

$$x_i \geq 0;\ i = 1, 2, 3, 4.$$

Решение задач линейного программирования в Excel, независимо от их содержания, сводится к использованию шаблона, приведенного на листинге 6.20.

Листинг 6.20. **Шаблон для решения задач линейного программирования**

	A	B	C	D	E	F	G	H
1	**Переменные**							
2	Имя	Продукт 1	Продукт 2	Продукт 3	Продукт 4			
3	Значение	1	1	1	1			
4	Нижняя гр.	0	0	0	0			
5	Верхняя гр.					Формула	MAX	
6	Коэф. в ЦФ	60	70	120	130	1320		
7								
8			**Ограничения**					
9	Вид	Продукт 1	Продукт 2	Продукт 3	Продукт 4	Формула	Знак	Запасы
10	Трудовые	1	1	1	1	4	<=	16
11	Сырье	6	5	4	3	18	<=	110
12	Финансы	4	6	10	13	33	<=	100

Перенесем данные из таблицы исходных данных (табл. 6.3) в шаблон электронной таблицы. Начальным значениям объемов выпуска продукции присвоим значение 1.

Нижняя граница объема выпуска продукции равна 0. Максимальный объем выпускаемой продукции не ограничен. Формулу для вычисления целевой функции запишем в ячейку F6. Для ввода формулы воспользуемся встроенной функцией СУММПРОИЗВ: =СУММПРОИЗВ(B3:E3;B6:E6).

Аналогично введем формулы для вычисления ограничений в ячейки F10, F11, F12. Для наглядности в ячейки G10:G12 введем знаки отношений, а значения имеющихся запасов введем в ячейки H10:H12.

Оптимизация решения

Введите команду ***Поиск решения*** из группы ***Анализ*** вкладки ***Данные***. Открывается окно диалога ***Параметры поиска решения*** (рис. 6.46). В окне ввода *Оптимизировать целевую функцию* укажите адрес ячейки с целевой функцией F6 с абсолютным адресом. Активизируем переключатель *Максимум*. В окне ввода *Изменяя ячейки переменных* укажите ячейки B3:E3 – объем выпуска продукции также с абсолютным адресом. В окне *В соответствии с ограничениями* введите ограничения. Для ввода ограничений введите команду Добавить – откроется окно диалога Добавление ограничений (рис. 6.47). В левом окне *Ссылка на ячейку* укажите адрес контролируемой ячейки, например, $F$10, в окне Условия выберите символ знака отношения (меньше или равно), а в списке Ограничения укажите значение ограничения или адрес ячейки, содержащей это значение, – $H$10.

После ввода всех ограничений щелкните по кнопке Найти решение в окне диалога Параметры поиска решения. Открывается очередное окно диалога ***Результаты поиска решения*** (рис. 6.48). Данное окно позволяет сохранить найденное решение, восстановить исходное значение или вернуться в диалоговое окно параметров. При активации флажка Отчеты на отдельном листе сформируются отчеты: Результаты, Устойчивость решения и Пределы, а также Структура сценария. Сценарий можно сохранить.

Результаты оптимизации приведены на листинге 6.21.

Из анализа результатов видно, что максимальное значение прибыли достигается при выпуске продукции только двух видов: Продукта 1 и Продукта 3. При этом полностью использованы трудовые и финансовые ресурсы, а запасы сырья израсходованы только на 76,4 %.

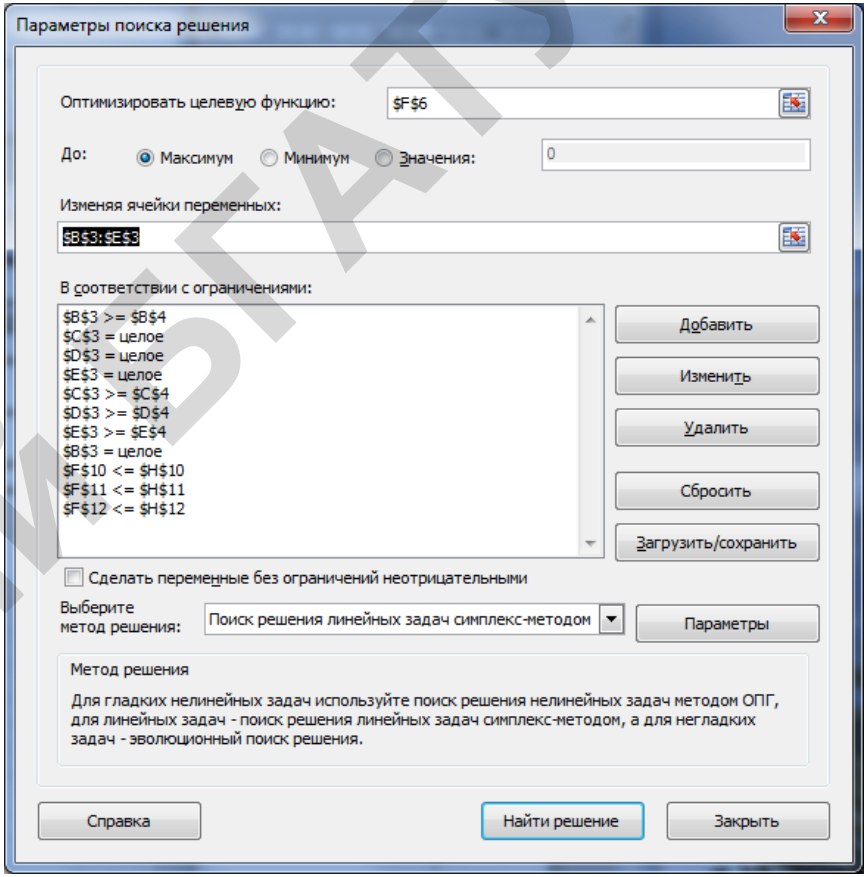


Рис. 6.46. Решение задачи линейного программирования

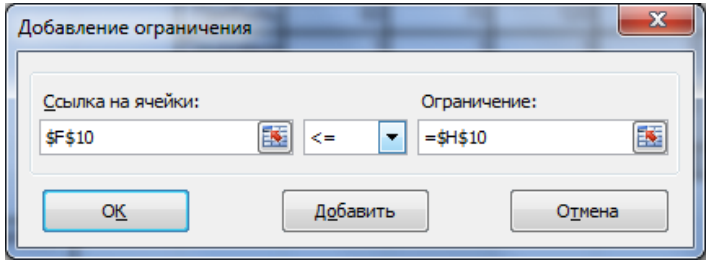


Рис. 6.47. Добавление ограничений

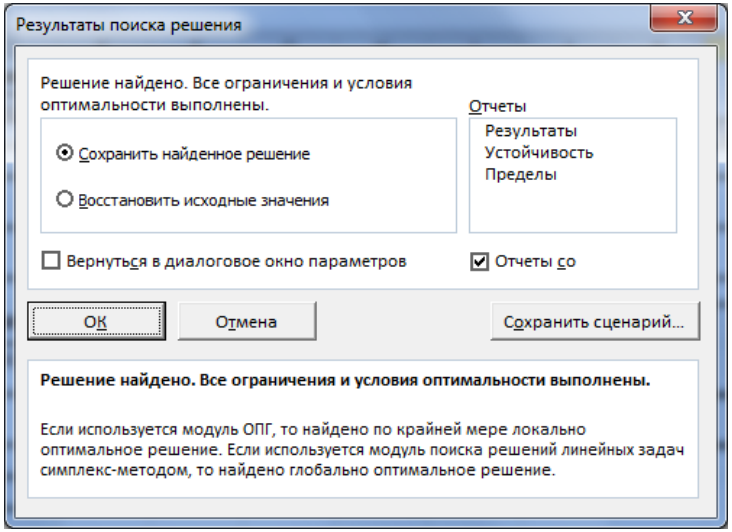


Рис. 6.48. Результаты поиска решения

Листинг 6.21. **Результаты поиска решения**

Переменные							
Имя	Продукт 1	Продукт 2	Продукт 3	Продукт 4			
Значение	10	0	6	0			
Нижняя гр.	0	0	0	0			
Верхняя гр.					Формула	MAX	
Коэф. в ЦФ	60	70	120	130	1320		
Ограничения							
Вид	Продукт 1	Продукт 2	Продукт 3	Продукт 4	Формула	Знак	Запасы
Трудовые	1	1	1	1	16	<=	16
Сырье	6	5	4	3	84	<=	110
Финансы	4	6	10	13	100	<=	100

Можно посмотреть, как изменится прибыль, если увеличить финансовые ресурсы на 10 %, 20 % и т. д. Результаты такого анализа приведены на рисунке 6.49 и листинге 6.22.

Из таблицы (листинг 6.22) и графиков (рис. 6.49) видно, что с увеличением финансовых затрат прибыль вначале растет, а затем рост прекращается. Идет перераспределение плана выпускаемой продукции: сначала предпочтительнее было выпускать продукцию видов 1 и 3, потом объем выпуска продукции П1 стал уменьшаться,

а затем и вовсе стал равен нулю, но выгоднее стало выпускать продукцию вида 4, которая заместила в конце концов и выпуск продукции третьего вида. При этом расходы сырья постоянно уменьшаются. Но интересно взглянуть, а какова эффективность дополнительных финансовых вложений? Оказывается, что она постоянно уменьшается. При уменьшении финансовых вложений до определенного размера эффективность вложений будет расти. При 64 единицах финансовых ресурсов выгодно выпускать только продукцию первого вида, при полном использовании трудовых ресурсов и максимальной эффективности капиталовложений. Дальнейшее уменьшение финансовых ресурсов ведет к сокращению рабочих и уменьшению выпуска продукции.

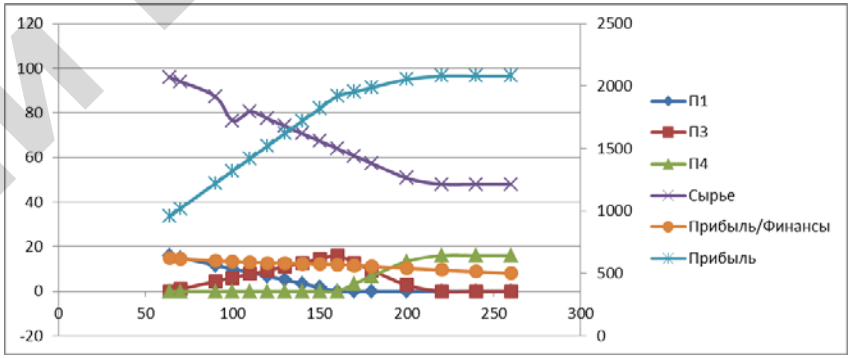


Рис. 6.49. Сценарии оптимизации решения задачи линейного программирования

Листинг 6.22. **Анализ влияния финансовых затрат на прибыль**

Финансы	П1	П3	П4	Сырье	Прибыль	Прибыль/Финансы
64	16	0	0	96	960	15
70	15	1	0	94	1020	14,57143
90	11,6	4,3	0	87,3	1220	13,55556
110	8,3	7,6	0	80,6	1420	12,90909
130	5	11	0	74	1620	12,46154
150	1,7	14,3	0	67,3	1820	12,13333
170	0	12,7	3,3	60,6	1953	11,48824
200	0	2,7	13,3	50,7	2053	10,265
220	0	0	16	48	2080	9,454545
240	0	0	16	48	2080	8,666667
260	0	0	16	48	2080	8

Аналогичным образом можно проанализировать, как будет влиять на план выпуска продукции увеличение числа рабочих.

Сохраненные сценарии можно просмотреть в диспетчере сценариев: вкладка ***Данные***, группа ***Работа с данными, Анализ «Что если», Диспетчер сценариев***.

6.10. ИНФОРМАЦИОННАЯ ТЕХНОЛОГИЯ СЦЕНАРНОГО ПОДХОДА

В электронной таблице имеется возможность сохранять наборы значений параметров моделей в виде сценариев. Сценарий – именованный набор значений указанных ячеек листа рабочей книги. Сценарный подход обеспечивает решение задач типа «ЧТО ЕСЛИ». Число одновременно изменяемых параметров для каждого набора не ограничивается. Сценарий используется для подстановки значений параметров в ячейки таблицы и вычисления зависящих от них формул.

Максимальное возможное число сценариев в отчете – 251. Число сценариев на одном рабочем листе не ограничено.

Создаются сценарии командой ***Диспетчер сценариев*** из группы ***Работа с данными, Анализ «Что если»*** вкладки ***Данные***. Команда Сценарий позволяет подготовить *Отчет* и *Структурную таблицу*.

Один из примеров использования сценариев приведен в разделе 6.8.

Диспетчер сценариев только позволяет готовить отчет на основе введенных данных.

Порядок создания сценариев рассмотрим на примере функции Будущая стоимость.

1. Подготовим таблицу (листинг 6.23). Введем в ячейки B2:B5 исходные данные, а в ячейку B6 функцию =БС(B2;B3;B4;B5;1).

Листинг 6.23. **Исходные данные для построения сценария**

	A	B
1		
2	Процентная ставка	0,170
3	Число периодов	5
4	Платежи	100
5	Первоначальный взнос	1000
6	Будущая стоимость	3'013,13р.

Структура сценария | Лист1 | Лист2 | Лист3

2. Исследуем зависимость Будущей стоимости от числа периодов:

- Введите команду Диспетчер сценариев. Откроется одноименное окно диалога.
- Щелкните мышкой по кнопке ***Добавить***. Открывается окно диалога Добавление сценария (рис. 6.50). Введите название сценария – Б1". В окне *Изменяемая ячейка* укажите адрес изменяемой ячейки B3 и щелкните по кнопке ОК. Программа возвращается в окно Диспетчера сценария.

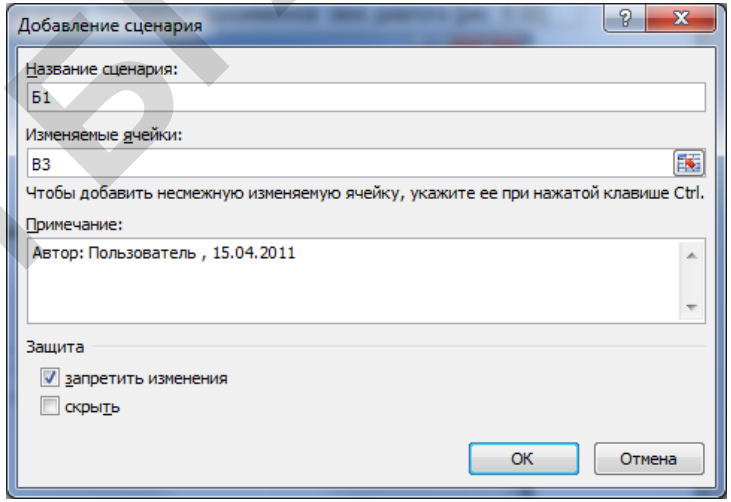


Рис. 6.50. Создание сценария шаг 2

- Снова щелкните по кнопке ***Добавить***. Открывается окно диалога *Диспетчер сценария*. Укажите имя Б2, адрес изменяемой ячейки и щелкните по кнопке ОК. Открывается окно диалога ***Значения ячеек сценария*** (рис. 6.51).

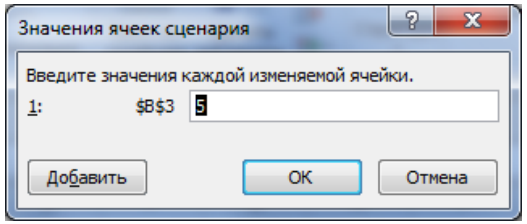


Рис. 6.51. Окно диалога Значения ячеек сценария

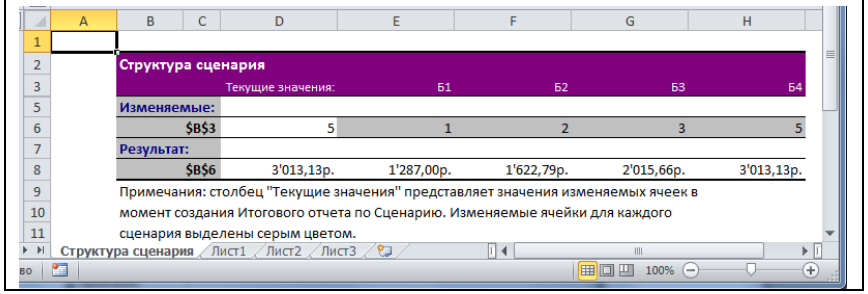
• Введите в окне диалога новое значение изменяемой ячейки и щелкните по кнопке ***Добавить***.

Повторите операцию, вводя каждый раз новое имя сценария и новое значение числа периодов, нужное количество раз. И для завершения операции щелкните по кнопке ОК.

Программа возвращается в окно *Диспетчера сценариев*.

При щелчке по кнопке ***Вывести*** результат последнего вычисления появится в ячейке B6. Для получения отчета щелкните по кнопке ***Отчет***. В окне диалога Отчет по сценарию укажите адрес ячейки результата B6 и щелкните по кнопке ОК. Открывается новый лист Структура сценария с результатами вычислений (листинг 6.24).

Листинг 6.24. **Результаты работы программы Диспетчер сценария. Структура отчета**

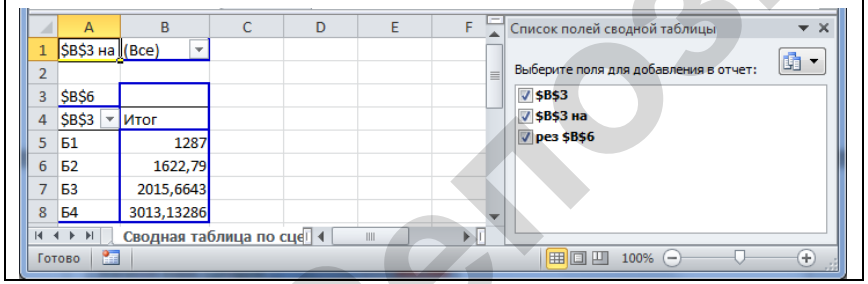


	Структура сценария				
	Текущие значения:	Б1	Б2	Б3	Б4
Изменяемые:					
$B$3	5	1	2	3	5
Результат:					
$B$6	3'013,13р.	1'287,00р.	1'622,79р.	2'015,66р.	3'013,13р.

Примечания: столбец "Текущие значения" представляет значения изменяемых ячеек в момент создания Итогового отчета по Сценарию. Изменяемые ячейки для каждого сценария выделены серым цветом.

Имеется возможность получить Сводную таблицу по сценарию. Вызовите снова Диспетчер сценариев, выделите имя сценария Будущая стоимость и введите команду Отчет. Программа выведет на экран окно диалога с предложением выбрать тип отчета: Структура или Сводная таблица. При выборе Сводная таблица откроется новый лист ***Сводная таблица по сценарию*** с отчетом (листинг 6.25).

Листинг 6.25. **Сводная таблица по сценарию**



$B$3 на	(Все)
$B$6	
$B$3	Итог
Б1	1287
Б2	1622,79
Б3	2015,6643
Б4	3013,13286

Список полей сводной таблицы

Выберите поля для добавления в отчет:

$B$3

$B$3 на

рез $B$6

Контрольные вопросы

1. Напишите математическую модель задачи линейного программирования.
2. Что такое целевая функция?
3. Какие виды ограничений могут быть в задачах линейного программирования?
4. Приведите пример шаблона для решения задачи линейного программирования.
5. Какие надстройки электронной таблицы используются для решения задач линейного программирования?
6. Как сохранить сценарий оптимизации решения?
7. Как можно просмотреть сценарий решения задачи оптимизации?
8. Поясните порядок работы по созданию сценария.
9. Как сформировать отчет и сводную таблицу по сценарию?

Заключение

Электронная таблица имеет надстройку Поиск решения, которая позволяет решать задачи линейного программирования. В процессе решения имеется возможность сохранять сценарии и анализировать влияние отдельных параметров модели на результаты решения с целью оптимизаций принимаемых решений. Эта возможность обеспечивается мощным механизмом создания сценариев – Диспетчер сценариев.

6.11. АВТОМАТИЗАЦИЯ ВЫЧИСЛЕНИЙ В EXCEL

6.11.1. Создание макросов

Для автоматизации выполнения часто повторяющихся операций, например, форматирования выделенных ячеек, оформления шапок таблиц и т. п. в Excel можно создавать макрокоманды – макросы (см. также раздел 4.7).

Удобнее всего макросы создавать путем записи. Правда, при этом в макрос записываются все действия пользователя, в том числе и ошибочные. Однако, если алгоритм работы продуман, то проблем не возникает.

Порядок создания макроса

Введите команду ***Вид, Макросы, Начать запись***. Открывается окно диалога (рис. 6.52), в котором необходимо указать имя макро-

са, назначить, при необходимости, сочетание клавиш для запуска макроса, указать место хранения макросов и щелкнуть по кнопке Ок. После этого необходимо выполнить нужные действия. Для окончания записи макроса введите команду ***Вид, Макросы, Остановить запись***.

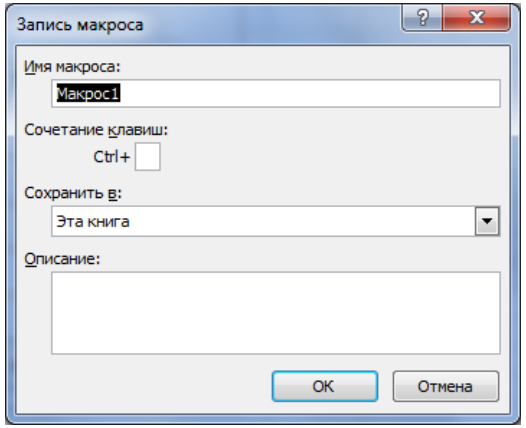


Рис. 6.52. Начало записи макроса

В списке *Сохранить в* (рис. 6.52) предлагается три варианта сохранения макросов: Личная книга макросов, Новая книга, Эта книга. Если выбрать *Личная книга макросов*, то макрос будет доступен во всех документах пользователя, при выборе *Эта книга* макрос будет доступен в текущей книге, а при выборе *Новая книга* макрос будет доступен во вновь создаваемой книге.

Аналогично макрос можно создать командами из группы ***Код*** вкладки ленты ***Разработчик***.

Если вкладка Разработчик отсутствует на экране, выполните следующие действия:

введите команду ***Файл, Параметры, Настройка ленты***;

в списке ***Основные вкладки*** установите флажок ***Разработчик***;

закройте окно диалога щелчком по кнопке ОК.

При использовании макросов в Excel необходимо сразу позаботиться о защите от заражения вирусами, распространяющимися через макросы программ Word, Excel и других приложений Windows. Для этого установите режим ***Отключить макросы с предупреждением*** (см. раздел 6.1).

Для установки уровня безопасности, временно разрешающего выполнение всех макросов, выполните следующие действия:

на вкладке ***Разработчик*** в группе ***Код*** выберите команду ***Безопасность макросов***;

в категории ***Параметры макросов*** в группе ***Параметры макросов*** нажмите переключатель ***Включить все макросы (не рекомендуется, возможен запуск опасной программы)***, а затем нажмите кнопку ОК;

для предотвращения запуска потенциально опасного кода по завершении работы с макросами рекомендуется ***вернуть*** параметры, отключающие все макросы.

Выполнение макросов

Существует несколько способов выполнения макроса в Microsoft Excel:

- Введите команду ***Макросы*** из группы ***Макросы*** вкладки ***Вид***. Откроется окно диалога Макрос (рис. 6.53). Выделите в окне диалога нужный макрос и щелкните по кнопке Выполнить.

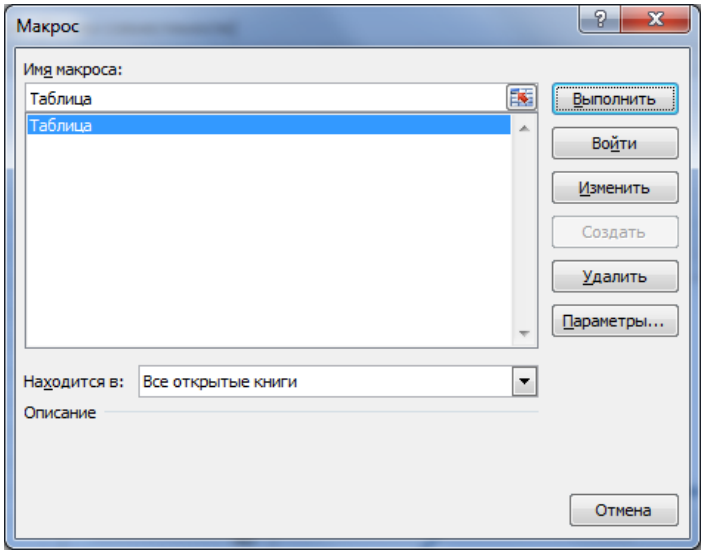


Рис. 6.53. Выполнение макроса

- Аналогично можно выполнить макрос с помощью команды **Макросы** из группы **Код** на вкладке **Разработчик**.

- Если при создании макроса было установлено сочетание клавиш для запуска макроса, воспользуйтесь сочетанием клавиш быстрого запуска.
- Удобно запускать макросы кнопками с панели быстрого доступа.
- Кроме того, макросы можно запускать автоматически при открытии книги.

Назначение макросу комбинации клавиш

Если при создании макроса комбинация клавиш ему не была назначена, это легко исправить:

введите команду Макрос, Макросы;

в окне диалога Макрос (рис. 6.53) выделите нужный макрос и введите команду Параметры – откроется окно диалога Параметры макроса, подобное окну диалога Запись макроса;

введите в окно символ, который будет использоваться в сочетании с клавишей Ctrl – [Ctrl + символ]. При вводе латинских символов при нажатой клавише Shift назначается сочетание клавиш [Ctrl + Shift + символ];

введите описание макроса и завершите работу щелчком по кнопке ОК.

Назначение макроса кнопке панели быстрого доступа

Для присвоения макроса кнопке панели быстрого доступа выполните следующее:

введите команду ***Файл***, выберите команду ***Параметры, Панель быстрого доступа***;

в раскрывающемся списке ***Выбрать команды из*** выберите элемент ***Макросы***;

найдите в списке созданный макрос, выделите его и нажмите кнопку ***Добавить***;

чтобы изменить изображение на кнопке макроса, выделите макрос в правом списке, в который он был добавлен, и нажмите кнопку ***Изменить***;

в поле Символ выберите нужное изображение для кнопки;

в этом же окне диалога можно изменить имя макроса, которое будет отображаться при наведении указателя мыши на кнопку: в поле *Отображаемое имя* введите имя, которое желаете назначить макросу;

нажмите кнопку ОК, и кнопка макроса будет добавлена на панель быстрого доступа.

Макрос можно назначить также любой картинке, рисунку, объекту SmartArt.

6.11.2. Создание процедур и функций пользователя с помощью VBA

В приложение Excel интегрирована неполная версия языка программирования Visual Basic под названием Visual Basic, Application Edition или Visual Basic for Applications, которая обладает частью стандартных функциональных возможностей. Кроме того, эта специальная версия поддерживает объекты, которые позволяют обращаться к содержимому ячеек и управлять приложением Excel.

Процедура или функция представляют собой фрагмент кода, выполняемый как один блок, и имеют обязательно заголовок и завершающую инструкцию, между которыми и находится собственно выполняемый текст программы (код).

Синтаксис заголовка процедуры:

Sub Имя_процедуры(аргументы)
Тело процедуры
End Sub

Синтаксис функции пользователя:

Function Имя_функции(аргументы)
Тело процедуры
End Function

Для создания функции пользователя необходимо ввести команду ***Редактор Visual Basic*** из группы ***Код*** вкладки ***Разработка***. В редакторе VB выбрать команду ***Вставка, Модуль***, а затем ***Вставка, Процедура***. Написать программный код и сохранить программу командой ***Файл (File), Сохранить и выйти из Excel*** (*Close and Return to Microsoft Excel*).

Функции пользователя используются так же, как и встроенные функции Excel. Для ввода функций пользователя можно воспользоваться Мастером функций, категория ***Определенные пользователем***.

Пример 6.12. Создать функцию пользователя для табулирования функции одной переменной $y = x^2$ (листинг 6.26).

Порядок выполнения:

1. Введите исходные данные для табулирования функции в ячейки A1:B4.
2. Внесите текст Аргумент и Функция в ячейки A5 и B5.
3. Введите команду ***Разработчик, Visual Basic***.
4. В редакторе VB выберите команду ***Вставка, Модуль***, а затем ***Вставка, Процедура***.

Листинг 6.26. **Табулирование функции одной переменной**

	A	B	C	D
1	*Исходные данные*			**Текст программы**
2	Число шагов N=	7		Public Function Tab1perem(n As Byte, _
3	Нач. значение X=	1		x As Single, dx As Single) As Variant
4	Число шагов Dx=	0,5		Dim i As Byte
5	*Аргумент*	*Функция*		Dim a(50,1) As Single
6	1,0	1,00		For i = 0 To n – 1
7	1,5	2,25		a(i, 0) = x: a(i, 1) = x * x
8	2,0	4,00		x = x + dx
9	2,5	6,25		Next i
10	3,0	9,00		Tab1perem = a
11	3,5	12,25		End Function
12	4,0	16,00		

5. Установите в окне диалога переключатели ***Function*** и ***Public***, запишите в строке ввода ***Name*** имя функции: *Tab1perem*.

6. Напишите текст программы, приведенный на листинге 6.26.

7. Сохраните программу командой ***File, Save*** и вернитесь в программу Excel командой ***Close and Return to Microsoft Excel***.

Из данного примера видно также, что в функциях можно использовать массивы. Массивы могут быть объявлены как статические (см. листинг 6.26) либо как динамические. Для объявления динамического массива объявите его сначала без указания размерности в разделе *General* (Общие), а затем в теле функции с указанием размерности с помощью оператора ReDim, например:

Dim A() As Single ‘объявление массива в разделе Общие

_____

Public Function Tab2perem(n As Byte, m As Byte, x As Single, dx As Single) As Variant

ReDim A(n,m) As Single ' объявление массива в теле функции

…

End Function

Применение функции пользователя:

внесите в таблицу исходные данные N, X, Dx, как показано на листинге 6.26;

выделите область ячеек A6:B12;

вызовите мастер функций командой ***Вставить функцию*** (*fx*) строки формул или командой ***Формулы, Вставить функцию***, выберите в списке ***Категории*** категорию ***Определенные пользователем***, выберите функцию ***Tab1perem***;

внесите адреса ячеек данных в окне диалога и нажмите комбинацию клавиш Ctrl + Alt + Enter (рис. 6.54).

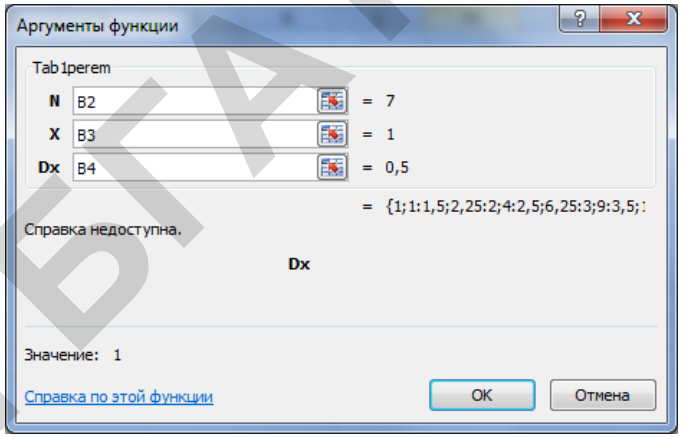


Рис. 6.54. Использование функции пользователя

Контрольные вопросы

1. Опишите порядок создания макроса.
2. Как назначить макрос кнопке панели инструментов?
3. Как назначить макросу комбинацию клавиш?
4. Как изменить надпись и значок на кнопке панели инструментов, предназначенной для управления макросом?
5. Как создать функцию пользователя?
6. Как используется функция пользователя?
7. Как установить/изменить защиту от макровирусов?

Заключение

Макросы являются удобным средством автоматизации типовых операций в электронной таблице. Другим способом повышения эффективности работы является создание процедур и функций пользователя. При создании функций пользователя они автоматически помещаются в список функций категории «Определенные пользователем» и используются так же, как и встроенные функции.

Для защиты от макровирусов следует обязательно устанавливать защиту. Рекомендуется устанавливать режим Отключить все макросы с уведомлением. Допускается включать все макросы на время работы с надежными источниками, с последующим восстановлением защиты.

6.12. ОСНОВНЫЕ СВЕДЕНИЯ О VISUAL BASIC FOR APPLICATION

6.12.1. Общие принципы объектно-ориентированного программирования

Объектно-ориентированное программирование (ООП) – вид программирования, при котором используются понятие «класс» и связанные с ним объекты. Или иначе, ООП можно описать как методику анализа, проектирования и написания приложений с помощью объектов.

Объектно-ориентированное программирование – это программирование, основанное на использовании объектов.

Техника ООП выгодна тем, что позволяет многократно и эффективно использовать созданные пользователем или другими программистами программные компоненты. Поддержка ООП в Visual Basic базируется на использовании модулей классов.

Основными принципами объектно-ориентированного программирования являются *инкапсуляция*, *наследование*, *полиморфизм.*

Инкапсуляция (сокрытие) – сокрытие от пользователя структур данных, процедур и функций для их обработки. Для пользователя объект представляется как «черный ящик». Пользователь знает входные и выходные данные, а что и как происходит внутри объекта – ему не известно. Обращение к данным и функциям выполняется через вызов соответствующих методов или свойств. Сокрытие информации позволяет разработчику объекта изменять внутренние принципы его функционирования, не оказывая при этом никакого влияния на пользователя объекта.

Наследование – способность создавать классы, зависящие от других классов. Цель наследования – сделать текст программы проще, используя уже готовые объекты и их свойства. Новый объект можно определить на основе уже существующих объектов. При этом новый объект может содержать те же свойства и методы, которые были у исходного объекта (родителя), а также новые свойства и методы. То есть свойства и методы родителя «наследуются» новым объектом. Таким образом, образуется иерархия классов. Классы, находящиеся на нижних уровнях, могут использовать все свойства и методы, определенные в классах вышележащих уровней. Механизм наследования позволяет переопределять и дополнять данные и методы их обработки. Возможность

изменения функциональности метода или свойства обусловлена принципом полиморфизма.

Полиморфизм (многоликий) – способность объекта реагировать на запрос (вызов метода) сообразно своему типу, при этом одно и то же имя свойства или метода может использоваться для различных классов объектов. Суть его заключается в том, что при написании программы, посылающей объекту сообщение, не нужно знать, к какому классу принадлежит этот объект. Все, что нужно, – это имя сообщения и его параметры. Например, оператор + может использоваться как для строковых, так и для числовых переменных. Однако это будут совершенно разные операции. В первом случае будут объединяться строки символов, во втором – будет вычисляться сумма числовых переменных. Или другой пример: метод Print формы и метод Print объекта Debug. Первый выводит информацию на форму, а второй в окно непосредственного наблюдения, хотя синтаксис у них одинаковый: <имя_объекта>.Print<список выражений>. Программисту не надо будет задумываться, как будет работать программа при выводе информации на форму или в окно непосредственного наблюдения. Программа сама определит, что надо делать в соответствии с классом объекта.

Ключевыми понятиями ООП являются *класс*, *объект*, *свойство*, *метод*, *событие.*

Класс – это абстрактный тип объекта, шаблон или план, по которому создаются объекты определенного класса. Класс содержит внутренние переменные, недоступные пользователю, внешние переменные, доступные пользователю через свойства класса, внутренние (закрытые) и внешние (открытые) процедуры и функции. Внутренние процедуры и функции недоступны пользователю, а внешние процедуры и функции доступны пользователю в виде методов и обработчиков событий.

Класс – это абстрактный тип объекта, шаблон или план, по которому создаются объекты определенного класса. Класс содержит свойства, методы и события, общие для всех объектов данного класса.

Объект – это экземпляр определенного класса. Например, есть *класс форма*, при загрузке на основе этого класса программа автоматически создает и помещает в конструктор форм экземпляр этого класса – объект Form1, для которого пользователь может установить требуемые свойства из набора уже известных свойств объекта класса форма. Или другой пример: есть *класс текстовое поле*, по-

мещая на форму текстовое поле, мы создаем объект типа текстовое поле. Поместив на форму пять текстовых полей, мы создадим тем самым пять объектов класса текстовое поле. На основе данного класса могут создаваться подклассы, в которых используется механизм наследования. VB 6.0 *не поддерживает* в полной мере механизм наследования.

Объект содержит инструкции (программный код), определяющие действия, которые может выполнять объект, и обрабатываемые данные.

Объект – это экземпляр класса, характеризующийся определенным набором свойств, методов, событий.

Метод – программа действий над объектом или его свойствами. Метод рассматривается как программный код, который связан с определенным объектом и осуществляет преобразование свойств, изменяет поведение объекта.

Методы – это функции или процедуры, определенные в классе.

Методы могут быть объявлены как глобальные (внешние, открытые) или как локальные (внутренние, закрытые). Глобальные процедуры, определенные в классе, работают так же, как и обычные подпрограммы и функции, но они доступны только пользователям объекта. Методы, объявленные как глобальные, образуют программный интерфейс класса, и программы работают с ними без всяких ограничений. Методы, объявленные как локальные, используются классами для выполнения внутренних операций, недоступных за пределами класса. Локальные методы объясняют суть принципа инкапсуляции. Программист может изменить локальную подпрограмму или функцию, определенную в классе, и это никак не отразится на его программном интерфейсе, доступном пользователю.

Методы классов могут пересекаться. Например, метод Print формы выполняет те же действия, что и метод Print объекта Picture (рисунок). В то же время метод Print объекта Printer выполняет другие функции – выводит данные на принтер, а не на форму. Здесь проявляется принцип полиморфизма – реагировать сообразно типу объекта.

Свойство – характеристика объекта, его параметр. Все объекты наделены определенными свойствами, которые в совокупности выделяют объект из множества других объектов, т. е. придают ему качественную определенность. Свойства объектов разных классов, так же, как и методы, могут пересекаться.

Свойство – характеристика объекта, его параметр.

Свойства могут быть открытыми (внешними) и закрытыми (внутренними). Открытые свойства могут использоваться процедурами, не входящими в данный класс (например, свойство Text объекта Text). Открытые свойства образуют часть интерфейса класса. Закрытые свойства используются для хранения информации, не входящей в интерфейс. Они предотвращают намеренное или случайное изменение данных, обеспечивающих нормальную работу класса.

Чем отличается метод от свойства? Свойство определяет внешний вид объекта и его местоположение. Свойства задаются на этапе разработки приложения, а также могут изменяться программным путем, например, положение объекта определяется значением свойств Top и Left (положение левого верхнего угла объекта). Методы – это команда объекту выполнить какие-то действия. Методы могут приводить к тем же результатам, что и изменение свойств, например, метод Move при выполнении программы также перемещает объект в требуемое положение: Move X,Y. Методы могут использоваться не только при выполнении программы, но и на этапе проектирования, а некоторые методы можно вызвать, даже не выделив память под объект, например, метод Load формы.

Событие – это свойство объекта процедурного типа. События позволяют классу обмениваться информацией с приложением при соблюдении определенных условий.

Событие – это свойство объекта процедурного типа.

События могут генерироваться системой – внутренние события – или пользователем – внешние события. Например, щелчок мышью или нажатие клавиши на клавиатуре – внешние события; загрузка или выгрузка формы – внутренние события. Каждому событию объекта соответствует ***процедура***. Если в эту процедуру поместить программу, то она будет выполняться при наступлении данного события. Использование механизма события избавляет программиста от необходимости многократной проверки значения некоторой величины до тех пор, пока не будет выполнено некоторое условие. Цикл активного опроса поглощает ресурсы процессора и замедляет выполнение программы.

Например, изменение значения текстового поля есть событие *Change*. Для обработки этого события предусмотрена процедура:

```
Private Sub Text1_Change () – ‘ заголовок процедуры
. . .
End Sub – ‘ конец процедуры
```

Шаблон этой процедуры входит в состав программного кода объекта TextBox. Предположим, что на форму установлен объект TextBox, имя объекта – Text1. Если в окне программы в списке объектов выделить объект Text1, а в списке процедур выбрать событие Change, то в окно программы автоматически вставляется шаблон этого события.

Чтобы эта процедура работала, в нее надо поместить текст программы, например:

```
Private Sub Text1_Change ()
    Длина=Val(Text1.Text)
End Sub.
```

Тогда при каждом изменении значения свойства Text объекта Text1 переменной Длина будет присваиваться новое значение.

Программа, созданная с помощью инструментальных средств ООП, содержит объекты с их характерными свойствами, для которых разработан графический интерфейс пользователя. Как правило, работа с приложением осуществляется с помощью *экранных форм* с объектами управления – *окон диалога*. Экранные формы[14] используются также для выполнения заданий и перехода от одной части программы к другой. При разработке приложений, для объектов управления уточняется перечень событий и создается пользовательский метод обработки – текст программы на языке программирования в виде событийных процедур.

В объектно-ориентированном программировании используется следующий синтаксис операторов для обращения к методам и свойствам объектов:

обращение к методу:

ИмяОбъекта.Метод

ИмяФормы.ИмяОбъекта.Метод

изменение свойства объекта в процессе выполнения программы:

ИмяОбъекта.Свойство =”значение”

использование свойства объекта для присвоения значения переменной:

Переменная = ИмяОбъекта.Свойство

[14] Формы, представленные на экране монитора, не имеют материального аналога, они существуют только в виде программ на диске или в оперативной памяти компьютера, а также в виде визуального изображения на экране монитора. Поэтому их можно называть «экранные формы». В дальнейшем по тексту будем писать просто «формы».

Например:

UserForm1.Show – показать форму. Здесь UserForm1 – имя объекта, Show – метод.

Text1.Text =”Учебное пособие” – свойству Text объекта Text1 присваивается значение «Учебное пособие». Text1 – объект, Text – свойство объекта, «Учебное пособие» – значение свойства Text.

Художник = Text2.Text – переменной «Художник» присваивается значение свойства Text объекта Text2.

6.12.2. Основные понятия языка программирования VBA

Язык программирования VBA является объектно-ориентированным языком программирования.

Основными объектами VBA являются:

Application – приложение Windows;

Workbooks(Workbook) – рабочая книга. Рабочие книги образуют коллекцию – рабочие книги. Объекты в коллекции пронумерованы, первый номер – 1. Поэтому обратиться, например, к первому листу можно командой Worksheets(1) или по имени листа – Worksheets(Лист1);

Worksheets(Worksheet) – рабочий лист. Листы также образуют коллекцию;

Cell – ячейка. ActiveCell – активная ячейка;

Range – ранг (диапазон);

Chart – диаграмма;

UserForm – форма. Основной конструктивный элемент при разработке окон диалогов.

Объекты в VBA имеют определенную иерархию. Для обращения к ним необходимо соблюдать следующий синтаксис:

Приложение. РабочаяКнига("Имя книги"). РабочийЛист("Имя листа"). Диапазон("A1:B6")

Application. Workbooks("Книга1"). Worksheets("Лист1"). Range("A1:B8")

Для активного объекта его имя можно не указывать в команде. Например, для активного листа рабочей книги для обращения к некоторому диапазону достаточно указать только имя диапазона: *Range("A1")*.

Каждый объект VBA имеет *свойства*, *методы* и *события*.

Свойства – характеристика объекта.

Методы – программа, действие, выполняемое над объектом.

События – реакция на внешние или внутренние воздействия, распознаваемые объектом.

Синтаксис обращения к методам:
Объект.Метод
например, *Application.Quit* – выход из приложения.

Методы могут применяться ко всем объектам коллекции.

Синтаксис обращения к свойствам:
Объект.Свойство = ЗначениеСвойства
например, *Application.Caption="База данных"*

В VBA особое место занимают активные объекты. К ним можно обращаться без указания имени:

ActiveWindow – возвращает активное окно Excel;

ActiveWorkbook – возвращает активную книгу;

ActiveSheet – возвращает активный лист рабочей книги;

ActiveDialog – возвращает активное окно диалога;

ActiveChart – возвращает активную диаграмму активного рабочего листа;

ActiveCell – возвращает активную ячейку активного рабочего листа;

Selection – возвращает выбранный объект.

6.12.3. Среда разработки проекта

Разработка проектов, процедур и функций пользователя осуществляется в среде разработки проекта Visual Basic for Application (VBA) (рис. 6.55), с которым мы частично познакомились в разделе Создание процедур и функций пользователя. Окно программы представляет собой стандартное приложение Windows предыдущих версий операционной системы и поэтому имеет все основные компоненты: строку заголовка, главное меню, панели инструментов, окна диалога.

В строке заголовка указывается наименование программы и имя текущей книги Excel, кнопка системного меню – слева и кнопки свертывания, развертывания и закрытия окна – справа.

Ниже главного меню расположена стандартная панель инструментов и окно документа. VBA имеет многооконный интерфейс.

Строка Меню содержит список команд, предназначенных для управления разработкой проекта, пункты меню содержат несколько уровней вложения.

File (Файл) – содержит команды для работы с файлами создаваемых приложений: сохранения, импорта, экспорта, вывода на печать, удаления листов.

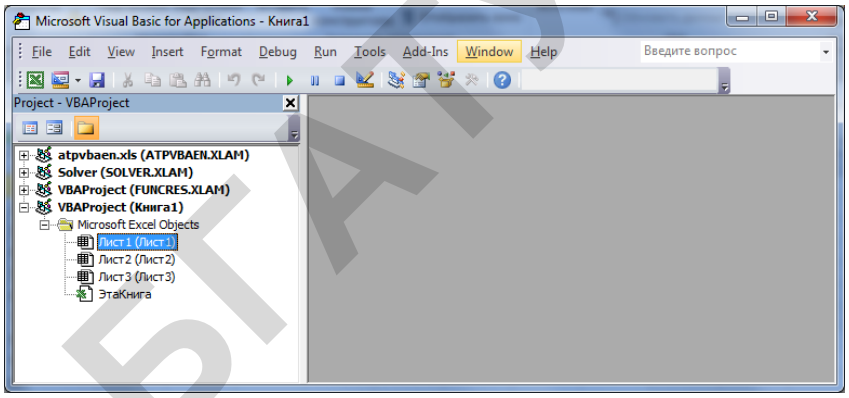


Рис. 6.55. Интерфейс программы VBA

Edit (Правка) – содержит команды редактирования, предназначенные для создания исходного текста программы, включая средства поиска и замены.

View (Просмотр) – обеспечивает доступ к различным частям приложения и среды разработки VB, часть команд выведена на панель инструментов стандартную.

Insert (Вставка) – предназначен для добавления новых объектов VB к разрабатываемым проектам: процедур, модулей, классов, форм, файлов.

Format (Формат) – предоставляет средства для управления размещением элементов управления на формах.

Debug (Отладка) – содержит средства, предназначенные для отладки программ или поиска ошибок.

Run (Выполнение) – служит для запуска, перезапуска и остановки программ непосредственно из среды разработки.

Tools (Инструменты) – обеспечивает доступ к работе с процедурами и меню внутри приложения. Этот пункт меню имеет важную команду ***Options***, которая открывает одноименную диалоговую панель с закладками ***Options***, где настраивается практически вся среда разработки Visual Basic.

Add-Ins (Модули) – открывает доступ к инструментам, которые могут быть добавлены к окружению VBA: мастера, ActiveX – элементы и другое. По умолчанию в него включена команда Add-In Manager. Add-In Manager – менеджер модулей – служит для выбора других надстроек, включаемых в модули.

Window (Окно) – используется для работы с окнами в среде разработки.

Help – справочная система.

Выбор пунктов меню осуществляется мышью или с помощью клавиатуры. При управлении с помощью клавиатуры для входа в меню используется клавиша ***Alt***. Выбор пунктов меню осуществляется с помощью клавиш управления курсором или с использованием клавиатурных команд (комбинация клавиши Alt и символа, подчеркнутого в командах меню: [***Alt + клавиша***]).

Панели инструментов

VBA имеет четыре стандартные панели инструментов: *Standard* (стандартная) – стандартная, *Edit* (правка) – редактирование, *Debug* (отладка) – отладка и *UserForm* (редактор форм) – редактор форм. Вывод панелей инструментов осуществляется следующим образом: выберите пункты меню ***View***, ***Toolbars*** и щелкните в открывшемся меню третьего уровня мышью по наименованию нужной панели инструментов или выберите опцию ***Настройка*** и установите флажки у тех панелей, которые нужно отобразить на рабочем столе.

Стандартная панель инструментов

Стандартная панель инструментов (рис. 6.56) обычно выведена в рабочее окно по умолчанию. Она содержит команды, дублирующие основные команды меню.

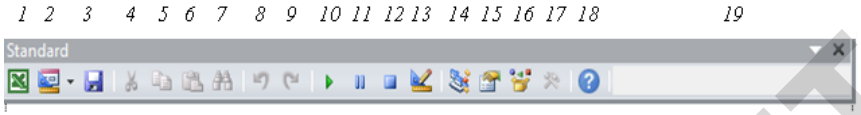


Рис. 6.56. Панель инструментов стандартная:
1 – переход в электронную таблицу; *2* – новая форма; *3* – сохранить; *4* – вырезать; *5* – копировать; *6* – вставить; *7* – поиск; *8* – отменить ввод; *9* – повторить ввод; *10* – старт; *11* – пауза; *12* – стоп; *13* – конструктор; *14* – окно проекта; *15* – окно свойств; *16* – окно просмотра объектов; *17* – панель элементов управления; *18* – справка; *19* – параметры панелей инструментов

Окно проекта

Окно проекта (рис. 6.57) открыто по умолчанию и выводится в левом верхнем углу окна документа. Оно служит для отображения всех элементов приложения: стандартные элементы приложения atpvbaen.xls, Solver, VBAProjectp (объекты защищены паролем); текущий проект. Эти объекты имеют кнопки для

свертывания/развертывания их структуры. Текущий проект содержит ряд объектов: листы, формы, модули, классы и т. п., сгруппированные по категориям. Текущий проект сохраняется вместе с текущей книгой.

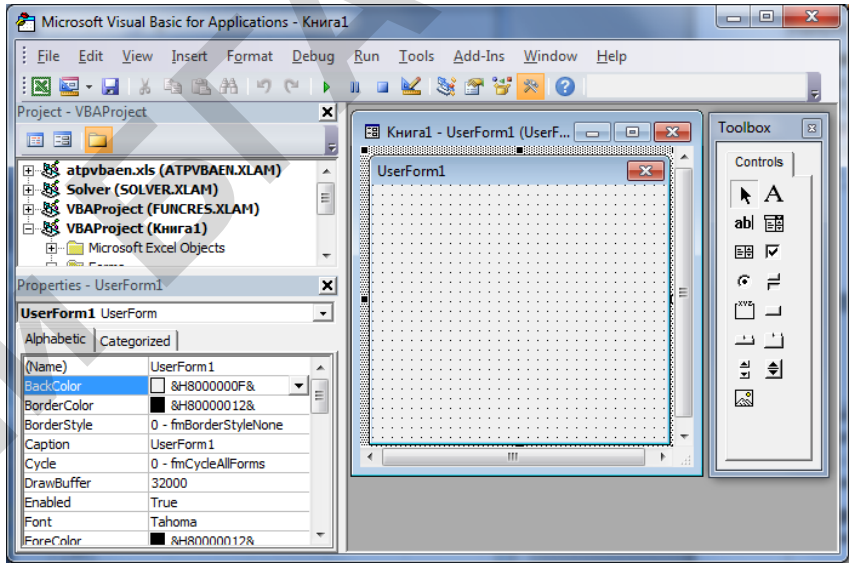


Рис. 6.57. Окно проекта, Окно свойств, Конструктор форм, Панель элементов управления

В заголовке окна проекта имеется три кнопки: View Code, View Object, Toggle Folders. Кнопка Toggle Folders служит для свертывания или развертывания структуры объектов проекта. Кнопка View Object открывает окно формы, а кнопка View Code открывает окно модуля формы (окно Code).

Окно свойств

В окне Свойств (Properties) (рис. 6.57) устанавливаются свойства выбранного элемента управления. В строке заголовка окна свойств рядом с текстом Properties указывается имя формы, которой принадлежит элемент управления. Поле со списком под строкой заголовка позволяет выбрать требуемый элемент управления, расположенный на форме. В списке, расположенном ниже, перечислены свойства данного элемента управления. Эти свойства могут быть упорядочены в алфавитном порядке (Alphabetic) или расположены

по категориям (Categorized). Набор свойств зависит от класса элемента управления.

Как установить свойства объекта? Имеется несколько способов:

ввести значение в поле справа от свойства;

выбрать из списка, который открывается щелчком мыши по полю;

установить с помощью окна диалога. При щелчке мышью по полю появляется кнопка (…) – многоточие, или эллипсис. При щелчке по кнопке многоточие появляется окно диалога для настройки соответствующего свойства;

загрузить из файла через стандартное окно диалога.

Форма

Форма (рис. 6.57) предназначена для разработки окон диалога. Форма является контейнером, это значит, что в нее можно помещать другие элементы управления. Для удобства размещения элементов управления на форме установлена сетка. Выравнивать элементы управления на форме можно с помощью меню Формат. По умолчанию форме присваиваются Имя и Название ***UserForm1***. Эти названия хранятся в свойствах Name и Caption. Свойства формы устанавливаются с помощью окна свойств формы (Properties – UserForm1), а также окна диалога, вызываемого командой ***Параметры*** (Option) меню ***Инструменты***. С формой связан модуль Программы (Code) (рис. 6.58).

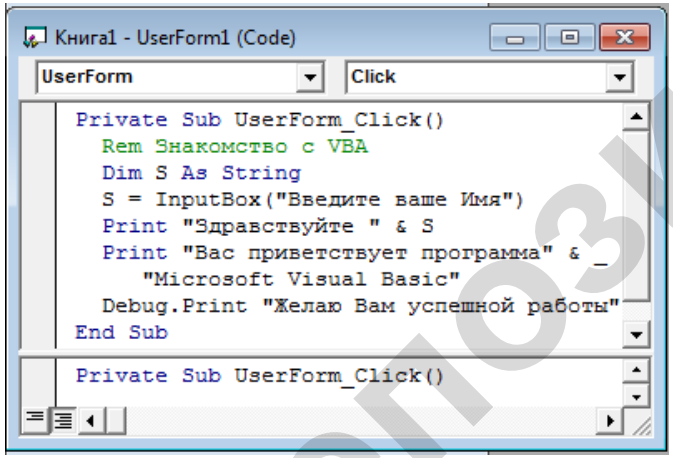


Рис. 6.58. Окно программы

Окно Программы (Code)

Окно Программы предназначено для ввода текста программы разрабатываемого приложения. Следует различать Окно программы формы (Code) и Модуля (Module). Модуль предназначен для размещения текста пользовательских процедур, функций и макросов, а также объявления глобальных переменных, доступных во всех формах приложения. Текст программы в VBA разделяется на части – процедуры и записывается в процедуры обработки событий или процедуры пользователя, поэтому он, как правило, связан с определенным элементом управления. Это позволяет открыть окно диалога двойным щелчком по соответствующему элементу формы или по самой форме. Кроме того, окно программы можно открыть щелчком по кнопке *View Code* в окне *Project*.

В верхней строке окна программы (рис. 6.58) располагается строка заголовка, в которой указаны имя проекта и управляющие кнопки (системного меню, закрытия окна, свертывания и развертывания окна). Ниже строки заголовка расположены два раскрывающихся списка: список объектов (левый) и список процедур (правый).

Список объектов содержит все объекты текущей формы. С каждым объектом связан список процедур – событий данного объекта, который появляется в правом списке при выделении объекта. Кроме того, в списке объектов имеется раздел General (Общий). В этом разделе размещаются объявления глобальных переменных, доступных во всех процедурах и функциях формы.

Список процедур содержит список всех событий, распознаваемых текущим объектом. Если для объекта уже написаны процедуры обработки событий, то они выделяются здесь жирным шрифтом. Если выбрать любой пункт данного списка, то VBA выведет соответствующую процедуру либо ее шаблон в окно программы и поместит в нее курсор.

Окно программы можно разделить на две части (рис. 6.58) с помощью кнопки, расположенной над вертикальной полосой прокрутки окна программы. В каждом из окон можно просматривать различные части текущей программы, копировать и редактировать текст программы. Для установки линии раздела зацепите эту кнопку мышкой и переместите в нужное положение. Для удаления разделительной линии зацепите ее мышкой и переместите к верхнему краю окна.

Внизу окна Программа размещены две кнопки, предназначенные для изменения режима просмотра текста программы. Левая кнопка

предназначена для просмотра текущей процедуры. Правая кнопка включает режим просмотра всех процедур модуля.

Панель элементов управления ToolBox

Панель элементов управления ***ToolBox*** появляется сразу же при открытии формы. Она может перемещаться в любое место экрана, ее размеры также могут устанавливаться пользователем. На ней размещены следующие элементы управления (по порядку слева направо и сверху вниз).

Выбор объектов. Кнопка активна по умолчанию. Позволяет выделять объекты щелчком мышки или обводкой.

Надпись (Label) – предназначена для ввода текста. Обычно это текст, сопровождающий окна ввода, или другие краткие пояснения.

Текстовое поле (Text Box) – предназначено для ввода в программу текстовой и числовой информации, даты, времени, а также для вывода информации на форму.

Раскрывающийся список (Combo Box) – список, свернутый в строку ввода, с кнопкой развертывания окна. Служит для хранения списка данных, возможно дополнение данных в процессе работы программы.

Список (List Box) – постоянно открытый список. Служит для хранения данных, возможно добавление данных в процессе работы.

Флажок (Check Box) – используется для включения или выключения процедур.

Переключатель (Options) – то же, что и флажок, но все объекты типа Переключатель по умолчанию группируются в пределах контейнера, и поэтому включенным в группе может быть только один переключатель. Флажки, напротив, независимы друг от друга.

Toggle Button (Выключатель) – этот элемент управления создает кнопку с двумя состояниями: включено, выключено.

Рамка (Frame) – является контейнером, предназначена для группирования элементов управления по функциональному назначению.

Кнопка (Command Button) – предназначена для управления приложением. В обработчик события кнопки записывается текст программы, которая должна быть выполнена при щелчке мышкой по кнопке.

Набор вкладок (Tab Strip) – служит для создания окон диалога с вкладками.

Набор страниц (Multi Page) – служит для создания многостраничного документа.

Линейка прокрутки (Scroll Bar) – используется для управления просмотром документов, а также для изменения параметров регулируемых объектов и других целей.

Счетчик (Spin Button, в Visual Basic 6.0 этот объект имеет имя Up Down). Используется для ввода данных. Самостоятельного значения не имеет. Используется в комбинации с другими объектами, чаще с объектом TextBox.

Картинка (Image) – служит контейнером для ввода картинок, объектов с расширением .bmp, .cur и др.

Окно ввода (Ref Edit) – служит для ввода данных из электронной таблицы путем выделения требуемой области или указания области данных в окне ввода. Этого элемента нет среди стандартных элементов управления в Excel 2010. Но его можно добавить с помощью команды ***Добавление контрольных элементов*** контекстного меню *Панели элементов* (Additional Controls).

Свойства объектов

Каждый объект характеризуется определенным набором свойств. Свойства придают объекту качественную определенность, выделяют его из совокупности других объектов. Некоторые из свойств объекта характерны для всех объектов, другие присущи только данному объекту. Общими свойствами объектов являются такие свойства, как *имя*, *надпись*, *положение объекта* и его *размеры*, *цвет фона*, *цвет текста*, *индекс*, *индекс обхода*.

При выделении объекта его свойства отображаются в окне свойств (Properties). Однако в окне свойств отображаются только те свойства, которые могут быть установлены пользователем на этапе разработки проекта. При выделении группы объектов в окне свойств отображаются только общие свойства, которые присущи всем выделенным объектам.

Имя элемента (Name). Каждому элементу управления имя присваивается по умолчанию при установке его на форму, например, форме – *UserForm*, метке – *Label*, текстовому полю – *Text* и т. д. Так как однотипных объектов может быть много, то программа нумерует их: *Label1*, *Label2* и т. д. Однако, такие имена малоинформативны, поэтому каждому объекту необходимо присвоить оригинальное, достаточно информативное имя. Текстовым полям можно присваивать имена в соответствии с элементом данных: длина, ширина, успеваемость. Кнопкам – в соответствии с их функциональным назначением: стоп, пуск, пауза. Имена объектов используются в программе для обращения к ним, поэтому имя объекту необходимо при-

сваивать до написания программы. Имя объекта необходимо писать латинскими символами. В имени объекта не допускается использование пробелов.

Примите за правило: создал объект – присвой ему имя.

Рекомендуется имя объекта начинать с префикса. Например, для формы – frm, для текстового поля – txt, для метки – lbl, для кнопки – cmd и т. д.: frmForm1, txtDlina, lblDlina, cmdStart. При этом префикс начинается со строчной буквы, а базовое имя (Dlina) – с прописной буквы. При таком подходе можно использовать одинаковые имена, например, для метки и текстового поля. Программа различит их, так как префиксы у них разные. Использование префиксов в имени объекта облегчает анализ программы при поиске ошибок и поиске в списке объектов окна программы.

Надпись (Caption) – это текстовое сообщение, которое выводится на соответствующий объект в форме. Каждому объекту программа автоматически присваивает имя и аналогичную надпись. Надпись на элементе управления может совпадать с именем объекта. В отличие от имени, надпись можно писать на любом языке, т. к. она не используется в программе. Надпись на объекте можно создавать в процессе разработки программы, а также при ее выполнении. Текст программы в этом случае представляет собой строку следующего вида:

ИмяФормы.ИмяОбъекта.Свойство =''<текст>''

frmUserForm1.cmdExit1.Caption = ''Резервная кнопка''

– свойству Caption кнопки cmdExit1 формы frmUserForm1 присваивается значение «Резервная кнопка». Для текущей формы имя формы не указывается:

cmdExit1.Caption = ''Резервная кнопка''.

Положение объекта. Положение объекта в форме определяется координатами верхнего левого угла (рис. 6.59). Для этого используются свойства Top и Left:

Top – расстояние от верхнего края формы;

Left – расстояние от левого края формы. Все расстояния задаются в «твипах».

Твип – экранно-независимая единица измерения. Один твип равен 1/20 точки принтера. Это гарантирует независимость отображения элементов приложения от разрешения дисплея (есть и другие определения размерности Твип).

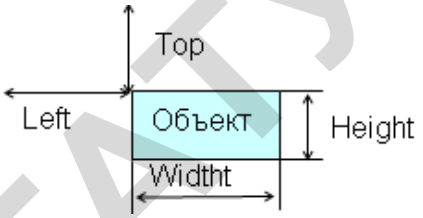


Рис. 6.59. Положение объекта в форме

Размеры объекта определяют также два свойства: ***Height*** и ***Width*** (высота и ширина).

Цвет. Управление цветовым оформлением элементов осуществляется с помощью свойств ***BackColor***, ***FillColor*** и ***ForeColor***. Этим свойствам по умолчанию назначаются стандартные цвета Windows.

Свойство ***BackColor*** определяет цвет фона. При проектировании цвет выбирают из палитры в диалоговом окне настройки, а в программе цвета задаются либо с использованием цветовой схемы RGB, либо константами библиотеки VBRUN.

Свойство ***ForeColor*** определяет цвет текста. Для формы оно определяет также цвет сетки.

Свойство ***FillColor*** определяет цвет заполнения рисованных объектов.

Appearance – позволяет изменять внешний вид объекта, имеет два значения: *Flat* – плоский, *3D* – объемный.

Enabled – доступность элемента управления. Может принимать два значения: *True* – истина и *False* – ложь. Если значение свойства установлено в *False*, то элемент управления будет недоступен пользователю. Такой элемент управления при выполнении программы подсвечивается блеклым серым цветом. Данное свойство используется для управления объектами. Например, кнопка «Расчет» может быть отключена до тех пор, пока не будут введены все исходные данные.

Font – это свойство позволяет оформлять шрифты, используя все возможности Windows.

Index – индекс. Определяет номер элемента в массиве элементов управления.

TabIndex – индекс обхода. Присваивается программой автоматически при помещении элементов управления на форму в порядке их создания. Этот индекс определяет порядок перемещения *фокуса* по элементам управления при нажатии клавиши Tab. Порядок об-

хода может быть изменен пользователем на этапе разработки программы. Если объект выделен, то говорят, что на него установлен фокус. Фокус может устанавливаться автоматически в порядке обхода элементов управления, в соответствии с возрастанием значения свойства TabIndex, или устанавливаться программным путем.

ToolTipText – позволяет ввести текст, который отображается в подсказке, появляющейся при зависании указателя мыши на элементе управления.

Visible – видимость элемента. Это свойство также имеет два значения: *True* и *False*. Если установлено значение False, то элемент управления будет невидимым.

Методы объектов

Объекты имеют различное число методов, например, объект типа надпись имеет 10 методов, объект типа форма – 22 метода. Общими для этих объектов являются методы ***Move*** – перемещать, ***OLE Drag*** – перемещать с использованием механизма OLE (вставка и внедрение), ***Refresh*** – перерисовать объект, ***SetFocus*** – установка фокуса.

Одним из важных понятий при обращении к элементам управления в Windows является понятие ***фокус***. При нажатии клавиш на клавиатуре управление получает активный элемент, то есть элемент, имеющий фокус. Например: при нажатии клавиши Enter выполняются те команды, которые связаны с командной кнопкой, имеющей фокус; текст вводится в поле ввода, на которое установлен фокус, и т. д. Не все объекты могут получать фокус, например, Надпись. Если элемент получает фокус, то он выделяется особым образом, например, текстовое поле отмечается мигающим курсором, командная кнопка – пунктирной рамкой по периметру и т. д. С установкой и потерей фокуса связаны два события и один метод: ***GotFocus*** – событие, возникающее при установке фокуса; ***LostFocus*** – событие, связанное с потерей фокуса; ***SetFocus*** – метод, обеспечивающий установку фокуса.

Фокус на элемент управления может быть установлен в процессе разработки программы путем присвоения свойству ***Default*** значения True. Только один элемент управления на форме может иметь значение Default. Фокус на элемент управления может быть установлен также и в процессе выполнения программы с помощью метода ***SetFocus*** командой следующего вида:

<имя объекта>.SetFocus,

например:

txtText1.SetFocus.

События объектов

Внешние события объектов связаны с мышью и клавиатурой.

Click. Событие Click вызывается, как только пользователь выполнит щелчок мышью на элементе управления.

DblClick. Событие DblClick вызывается двойным щелчком мыши на элементе управления. Временной интервал между двумя щелчками устанавливается в панели управления Windows.

KeyDown – нажатие клавиши.

KeyUp – отпускание клавиши.

KeyPress – возвращает код ASCII нажатой клавиши.

После нажатия клавиши события наступают в такой последовательности: KeyDown, KeyPress, KeyUp.

KeyPreview. Это свойство присуще только форме. Оно определяет порядок обработки событий, связанных с нажатием клавиш. Если свойству KeyPreview формы присвоить значение True, то событие, связанное с клавиатурой, передается сначала форме, а затем текущему элементу управления.

GotFocus – событие, возникающее при установке фокуса.

LostFocus – событие, возникающее при потере фокуса.

Окно просмотра объектов

Все свойства, события и методы объекта можно просмотреть в окне просмотра объектов *Object Browser* (рис. 6.60), который содержит каталог объектов. Для вывода окна диалога необходимо ввести команду ***View***, ***Object Browser***. Выбрать в этом меню в верхнем поле списка необходимую библиотеку либо объекты из всех библиотек. Стандартные элементы управления содержит библиотека VBA. В списке *Classes* перечислены все классы объектов VBA. Здесь *UserForm* – класс объекта типа Форма, *UserForm1* – объект, экземпляр данного класса, остальные элементы – константы объекта *UserForm1*. ***Константы*** – это данные, значения которых не меняются при выполнении программы. Для каждого объекта имеется свой набор констант.

После выбора объекта в списке *Members of* выводятся все свойства и методы, относящиеся к выбранному объекту, в данном случае к форме. В этом окне отображаются свойства (Caption, Cycle, DrawBuffer, Enabled), события (Click, DblClick) и методы (Copy, Cut). Свойства обозначаются характерной пиктограммой серого

цвета – карточка с рукой. События выделяются желтым цветом, методы – зеленым цветом. Нажав клавишу F1 или кнопку ?, можно вызвать справочную информацию. Чтобы найти конкретное свойство или метод, можно воспользоваться кнопкой поиска (бинокль). Введите во второй строке ввода имя объекта, например, UserForm, и щелкните по кнопке поиска. После окончания поиска в окне Результаты поиска (Search Results) появится имя библиотеки и искомый объект, а в окне Members – свойства и методы этого объекта.

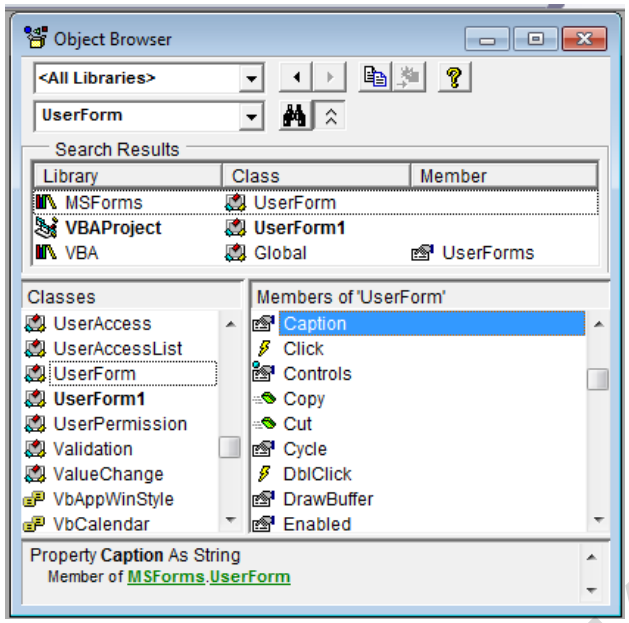


Рис. 6.60. Окно просмотра свойств объектов

В нижней части окна дается краткое описание выделенного объекта.

6.12.4. Основы программирования в VBA

В синтаксисе языка Visual Basic используются операторы, функции, переменные и константы.

Операторы – синтаксические конструкции, которые управляют вычислительным процессом. Операторы образуют текст программы.

Функции – вычислительные процедуры, предназначенные для выполнения наиболее часто используемых вычислительных или логических операций.

Переменные – именованные области памяти, предназначенные для хранения данных. Значения переменных могут изменяться в процессе выполнения программы.

Константы – постоянные величины, или поименованные области памяти, предназначенные для хранения данных. Значения констант не изменяются в процессе выполнения программы.

Переменные

Имя переменной

По определению переменная – это именованная область памяти, предназначенная для хранения данных. Поэтому каждой переменной присваивается идентификатор – имя. Имя переменной начинается с буквы и может содержать до 255 символов. В имени не допускаются пробелы, нельзя использовать также символы . и &. Имя переменной рекомендуется начинать с односимвольного ***префикса*** (табл. 6.4), характеризующего тип хранимых данных. Тип переменной устанавливается при объявлении переменной или определяется программой автоматически по содержанию присвоенной ей информации. Тип переменной задает определенный формат и размер содержимого переменной.

Таблица 6.4

Типы переменных

Суффикс	Префикс	Тип переменной	Объявление по умолчанию	Занимаемая память в байтах, байт/символ	Описание
1	2	3	4	5	6
$	s	String	DefStr	1	Строковая переменная переменной длины, 2 миллиарда символов для динамически изменяемых срок
%	n	Integer	DefInt	2	Целое, одинарной длины (–32 768, +32 767)

Окончание табл. 6.4

1	2	3	4	5	6
&	l	Long	DefLng	4	Целое, двойной длины (от –2 147 483 648 до +2 147 483 647)
!	f	Single	DefSng	4	Вещественное, одинарной точности, +/–(1,4E–45…3,4E38)
#	d	Double	DefDbl	8	Вещественное, двойной точности, +/–(4,94E–324…1,79E308)
–	b	Boolean	DefBool	2	Логическое, используется для представления логических переменных, имеет два значения: True и False
–	v	Date	DefDate	8	Дата и время, от 1.1.100 до 12.31. 9999; от 0:0:0 до 23:59:59
–	–	Object	–	4	Экземпляр класса; объект типа OLE
–	–	Variant	DefVar	16 + 1	Вариантный

Чтобы не было путаницы с типами данных и проблем при поиске ошибок, рекомендуется обязательно объявлять тип переменной.

Типы переменных

VBA имеет большое число типов переменных – 14. Основные типы переменных приведены в табл. 6.4. Не допускается использование в программе двух одинаковых имен, отличающихся только типом.

Особо необходимо отметить переменные типа Variant. Если тип переменной не указан, она будет объявлена программой как Variant. В переменной типа Variant можно хранить переменные любого типа. Преобразование типов переменных в этом случае осуществляется автоматически. Но при этом в некоторых случаях могут возникать ошибки. Кроме того, использование переменной типа Variant приводит к перерасходу оперативной памяти, а также этот тип данных неприемлем для использования в качестве аргументов для тех функций, параметры которых описаны явным

образом. Поэтому рекомендуется с самого начала объявлять типы переменных явно.

Объявление типов переменных явно считается хорошим стилем программирования.

В языке программирования VBA переменная типа вариант при использовании в заголовках процедур и функций имеет особое значение: ей ***можно присваивать массивы*** в качестве входных параметров.

Способы объявления переменных

Тип переменной можно объявить несколькими способами:

1. Неявно с помощью специального символа – ***суффикса***, записанного после имени переменной (табл. 6.4). Спецсимвол следует указывать только при первом использовании переменной:
C$ = “текст”: a! = 1.769: B% = 12674.

2. По умолчанию с помощью операторов вида ***DefType***. Этот оператор используется только в разделе Главная. Синтаксис оператора:
DefType <список символов>

Например:
DefStr C
DefInt I-L

В данном примере первый оператор объявляет, что все переменные, имя которых начинается с символа C, являются переменными строкового типа, второй оператор объявляет, что все переменные, имена которых начинаются с символов I, J, K, L, являются переменными целого типа одинарной точности.

3. Явное объявление с помощью операторов ***Dim***, ***Private***, ***Static***, ***Public***, ***Global***. При объявлении переменной указываются ее имя и тип. Синтаксис операторов объявления переменных:
Dim <имя_переменной> As <тип_переменной> [,<имя_переменной> As <тип_переменной>]

Внимание! *В VBA, в отличие от VB6.0, тип переменной можно указать сразу для нескольких переменных. В одном операторе Dim можно объявлять несколько типов переменных, разделяя их запятыми и указав для каждой группы переменных ключевое слово As и тип переменной.*

Например:
Dim i,j As Integer, a,b As Single
Dim nCount As Integer, I As Integer, J As Integer

Publuc sText As String, NameFile As String
Global dFiz As Double

Соглашение о типах переменных, принятых по умолчанию, можно изменить, используя суффикс или оператор Dim.

Контроль типов переменных

С целью исключения ошибок, связанных с неправильным объявлением имен переменных, рекомендуется объявлять все переменные явно. Контроль типов переменных можно установить с помощью оператора ***Options Explicit***. Этот оператор необходимо поместить в разделе Общие (General) данной формы. Если переменная не была объявлена, то при запуске программы выдается сообщение об ошибке.

Другой способ обеспечить контроль типов переменных – установить флажок в опции ***Require Variable Declaration*** на вкладке ***Editor*** в меню ***Tools, Options***. В этом случае оператор ***Options Explicit*** будет автоматически вставляться в раздел Общие формы при ее создании.

Область определения (видимости) переменных

В VBA есть три вида областей определения, характеризующих доступность или видимость переменной: локальная, контейнера (формы или модуля), глобальная.

Локальная переменная доступна только в текущей процедуре или функции. Локальные переменные объявляются оператором ***Dim***.

Переменная контейнера доступна только в текущей форме, модуле или классе. Объявляются оператором ***Dim*** в разделе Общие (General) формы или модуля. Переменная модуля объявляется оператором ***Private***.

Глобальные переменные доступны во всем проекте. Они объявляются в разделе Общие модуля программы оператором ***Global*** или ***Public***.

Время жизни переменных

Переменные, объявленные как локальные, при выходе из процедуры удаляются из памяти, а при новом вызове инициализируются заново.

Переменные уровня формы инициализируются при первом открытии формы и сохраняют свои значения до окончания работы проекта. С целью экономии оперативной памяти переменные уровня формы могут быть уничтожены при закрытии формы командой ***Set <имя формы>=Noting***.

Глобальные переменные при выходе из программы сохраняют свои значения до окончания работы проекта.

Статические переменные

В некоторых случаях необходимо, чтобы при выходе из процедуры переменные сохраняли свое значение. Например, при использовании процедуры для печати страниц номер страницы должен увеличиваться при каждом вызове процедуры. Поэтому при выходе из процедуры переменная, хранящая значение текущего номера страницы, должна сохранить свое значение, чтобы при последующем вызове процедуры можно было увеличить номер страницы на единицу.

С целью обеспечения сохранности значений переменных при выходе из процедуры они могут быть объявлены в данной процедуре как статические оператором ***Static***:

Static nPageNumber As Integer

Статические переменные являются локальными для процедуры, в которой они используются, но их значения сохраняются до повторного вызова процедуры.

Чтобы объявить статическими все переменные процедуры, ключевое слово Static следует записать в заголовке процедуры:

Static Sub <имя_процедуры>(аргументы)
Static Sub ПечатьФормы (x)

Константы

Основное отличие констант от переменных состоит в том, что их значения нельзя изменить в процессе выполнения программы. Они всегда сохраняют значение, присвоенное при разработке. Области видимости для констант определяются так же, как и для переменных. Константы бывают локальные, уровня контейнера и глобальные. При объявлении констант используется ключевое слово ***Const***. Глобальная константа объявляется как ***Public Const***.

Глобальные константы объявляются только в модуле приложения.

Одновременно с объявлением константе можно присвоить и значение:

Public Const <Имя константы> = <Значение>
Private Const <Имя константы> = <Значение>

Например:

Public Const Pi = 3.1415926538897932
Private Const G = 9.81

Public Const nName = «Лев Толстой»
Const ПлотностьМатериала = 2.25
*Const Масса = ПлотностьМатериала * Высота * Ширина * Длина*

Информацию о константах, их значениях и применении можно получить, обратившись к соответствующим разделам справки или воспользовавшись каталогом объектов: войти в меню ***View, Object Browser***, щелкнуть кнопку ***Object Browser*** на панели инструментов или нажать клавишу ***F2***.

Константы можно объявлять и с указанием типа данных:

[Public/Private] Const <Имя константы> As <Тип_данных> = <Значение>

Public Const Pi As Double = 3.1415926538897932

Для указания типа констант используются те же ключевые слова, что и при объявлении переменных.

Visual Basic имеет большое число встроенных констант. В виде констант в VBA определены коды цветов, коды клавиш, флажки, константы, задающие тип пиктограмм и набор кнопок в окне сообщения Message Box? и др. Встроенные константы начинаются с префикса vb, например: vbRed, vbCascade.

Стандартные функции языка VBA

Visual Basic имеет большое число встроенных функций. Можно выделить следующие категории функций:

математические;
работы с массивами;
преобразования типов данных;
работы со строковыми переменными;
работы с датами и временем;
финансово-математические;
работы с файлами;
управления компьютером;
обработки ошибок/

В программе на VBA могут использоваться как функции языка VBA, так и функции Excel. При обращении к функциям VBA указывается приложение, например:

Application.Sum(x)

Математические функции в основном совпадают с функциями, используемыми в MathCad (табл. 2.1). Список встроенных функций можно просмотреть с помощью окна диалога ***Browser*** (рис. 6.60): откройте окно диалога щелчком мышки по кнопке на панели инструментов, выберите в верхнем списке библиотеку VBA, введи-

те в строке поиска имя любой функции, например, Cos, и щелкните по кнопке Поиск (Search). В частности, из библиотеки VBA доступны функции: Abs, Atn, Cos, Exp, Log (натуральный логарифм), Round, Sgn, Sqr, Tan. Из библиотеки Excel доступны, например, логарифмические функции Ln, Log, Log10, обратные тригонометрические функции Acos, Asin.

Стандартные функции для работы с массивами:

Count (Счет) – количество числе в массиве;

CountA (Счет3) – количество элементов в массиве;

Sum (Sum) – сумма элементов массива;

SumSq (СуммКв) – сумма квадратов элементов массива;

SumXmY2 (СуммКвРазн) – сумма квадратов разностей элементов двух массивов;

SumX2mY2 (СуммРазнКв) – сумма разностей квадратов элементов двух массивов;

SumProduct (СуммПроизв) – сумма произведений;

Mmult (Умнож) – произведение двух матриц;

Minverse (Мобр) – обратная матрица;

Transpose (Трансп) – транспонирование;

Mdeterm (Мопред) – определитель матрицы.

Ввод данных

Данные в программу могут быть введены несколькими способами:

присвоением начальных значений переменным непосредственно в программе с помощью оператора ***Let***;

вводом данных с клавиатуры в режиме диалога с помощью элемента управления ***TextBox***;

с помощью диалогового окна ***InputBox***.

Присвоение начальных значений переменным

Присвоение начальных значений переменным осуществляется с помощью оператора ***Let***. Этот оператор не обязателен и может быть опущен.

Синтаксис оператора:

[Let] <имя_переменной> = <выражение>

Например:

Let a = 3.754: b = Sin(x) +exp(x):
Let c$ = "Зимние каникулы": c1$ = "Новый год"

Ввод данных с помощью элемента управления TextBox

В режиме диалога пользователя с программой данные можно ввести с клавиатуры непосредственно в строку ввода элемента

управления TextBox. Текстовое поле используется для ввода самой разнообразной информации. При вводе данных с клавиатуры в активное текстовое поле программа не делает различий между буквами и цифрами: все данные вводятся как текст. Поэтому для перевода текста в числа и обратного перевода чисел в текст используются функции преобразования строковых переменных:

Val(C) – преобразование текста в число;

Str(N) – преобразование числа в строковую переменную или **Str$(N)** – для преобразования в строковую переменную переменных типа Variant.

<переменная_символьного_типа> = txtText1.Text

<переменная_числового_типа> = Val(txtText1.Text)

Например:

Длина = Val(Text1.Text) ‘присвоение числовой переменной Длина значения

‘свойства Text объекта Text1

Text2.Text = Str(Длина) ‘присвоение свойству Text объекта Text2 значения

‘числовой переменной Длина

Ввод данных с помощью окна диалога InputBox

В режиме диалога данные можно ввести в программу с помощью функции ***InputBox*** (рис. 6.61). Функция InputBox используется тогда, когда необходимо ввести только краткую информацию. Она имеет следующий синтаксис:

<Переменная> = InputBox (prompt [,title] [,default] [,xpos] [,ypos] [,helpfile], context])

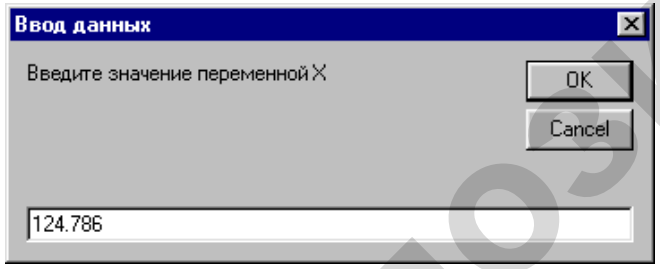


Рис. 6.61. Окно диалога функции InputBox

Параметр: *Prompt* определяет текст, отображаемый в окне диалога, как приглашение («Введите значение переменной X»).

Title – отвечает за надпись заголовка («Ввод данных»). Если этот параметр не указан, то отображается название приложения.

Default – определяет значение по умолчанию, отображаемое в строке ввода (124.786).

Параметры *XPos* и *YPos* используются совместно и указывают координаты верхнего левого угла окна. По умолчанию окно отображается посередине экрана.

Функция InputBox имеет еще два необязательных параметра *HelpFile* и *Context*, которые позволяют открывать определенные файлы справочной системы. При нажатии кнопки ОК функция InputBox возвращает строку, введенную пользователем. При нажатии кнопки Cancel возвращается пустая строка.

Например:

x = Val(InputBox("Введите значение переменной X",124.786, "Ввод данных")

В приведенном примере:

- «Введите значение переменной X» – текстовое сообщение, выводимое в окно ввода.
- 124.786 – значение переменной, принимаемое по умолчанию.
- «Ввод данных» – текст, выводимый в строку заголовка.
- Значение координат XPos и YPos принимается по умолчанию, то есть окно ввода будет выведено посередине экрана.

Из-за громоздкости окна ввода его рекомендуется использовать только при отладке программы. В готовой программе рекомендуется использовать текстовое поле TextBox.

Вывод данных

Вывод данных осуществляется несколькими способами:

с помощью оператора *Print*;

с помощью *текстового поля*;

с помощью окна сообщений *MessageBox*;

с помощью свойства *Print* объекта *Debug*.

Оператор Print

Оператор Print может выводить информацию непосредственно в форму или на печать:

Print [“текстовое сообщение”] [;_ ,] <список выражений> [; ,]

В операторе Print в качестве управляющих операторов могут использоваться символы [;], [,] и пробел. Символ [;] в списке выражений означает, что данные выводятся без пробелов, при выводе чисел оставляется место для знака числа. Символ [;] в конце списка означает, что курсор ввода остается на месте и следующий опера-

тор будет выводить текст с текущей позиции. Символ [,] в списке выражений означает, что данные будут выводиться по зонам. Вся строка разбивается на пять зон по 14 символов, если данные не умещаются в одной зоне, то занимается соседняя зона. Отсутствие символов в конце списка означает, что курсор ввода переводится в начало строки и очередной оператор Print будет выводить информацию с новой строки.

Для позиционирования точки вывода используются свойства ***CurrentX*** и ***CurrentY*** объекта, а также функции ***Tab(N) и Spc(N)***. Например:

CurrentX = 100: CurrentY = 50
Print x,y
или
Print Tab(100); x; Spc(10); y

Свойства CurrentX и CurrentY перемещают точку вывода в указанные координаты X и Y. Расстояния задаются в твипах.

Функции Tab и Spc используются в операторе Print. Они перемещают точку вывода на заданное число позиций. Отличие в использовании функций Tab и Spc состоит в том, что функция Tab перемещает точку ввода относительно края формы, а функция Spc – относительно текущей позиции (рис. 6.62).

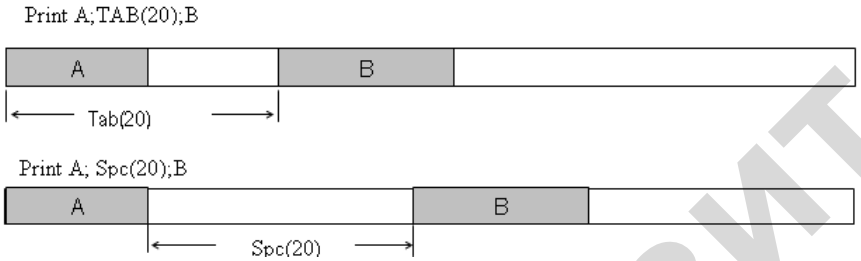


Рис. 6.62. Использование функций Tab и Spc

Текстовое поле TextBox

Текстовое поле TextBox может использоваться для вывода информации в любом месте формы. Позиция элемента управления может устанавливаться как при разработке формы, так и в процессе выполнения программы, например:

Text1.Top = Y: Text1.Left = X
Text1.Text = "<выводимый текст>"

Окно диалога MessageBox

Окно MessageBox подобно окну InputBox и служит для вывода информации. Этот компонент широко используется для вывода различных сообщений в приложениях Windows.

MessageBox можно вызвать как команду и как функцию.

Синтаксис команды:

MsgBox Prompt [,Buttons] [,Title] [,Helpfile, Context]

Синтаксис функции:

<Переменная> = MsgBox (Prompt [,Buttons] [,Title] [,Helpfile, Context])

Параметры *Prompt*, *Title*, *Helpfile*, *Context* имеют то же значение, что и в функции InputBox. В отличие от синтаксиса команды, в синтаксисе функции аргументы заключаются в скобки.

Параметр *Buttons* определяет состав кнопок в окне. Он формируется из нескольких частей:

$$Buttons = Button + Icon + Default + Modal + Extras,$$

где *Button* – количество кнопок и их состав;
Icon – вид пиктограммы;
Default – определяет, какая кнопка активна по умолчанию;
Modal – вид диалогового окна (окно приложения или системы). Модальность означает, что выполнение приложения возможно только после закрытия окна;
Extras – дополнительные свойства.

По умолчанию выводятся кнопки ОК и Отмена.

Код, возвращаемый функцией, зависит от того, какая кнопка нажата пользователем:

1 – OK, 2 – Cancel, 3 – Abort, 4 – Retry, 5 – Ignore, 6 – Yes, 7 – No.

Это позволяет анализировать код нажатой клавиши и управлять программой в зависимости от реакции пользователя.

Пример 6.13. Приветствие (рис. 6.63):

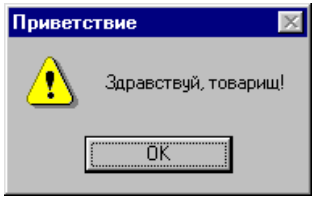


Рис. 6.63. Окно MsgBox

MsgBox “Здравствуй, товарищ!”, vbExclamation, “Приветствие”.

Функция:

КодВозврата = MsgBox (“Конец игры?”, vbCritical, “Игра”)
If КодВозврата = 2 Then End ‘Если КодВозврата равен 2 Тогда Выход из программы

Пример 6.13. Формирование параметра Buttons.

Type = 4 + 32 + 256 + 1
MsgBox “Сообщение”, Type, “Заголовок”.

При данном коде в окне отображаются кнопки Yes и No (код 4), пиктограмма Warning Query (код 32), активна кнопка No (код 256), окно модальное (код 1).

Пример 6.15. Выведите на экран результаты вычисления X и Y.

Если текст длинный или желательно разместить результаты вычислений в несколько строк, используется константа Visual Basic ***vbCr*** – возврат каретки. Для сложения символьных переменных используется символ & или +.

MsgBox “X=” & Str(x) & VbCr & “Y=” & Str(y).

Для управления выводом текста может использоваться также константа VBA ***vbTab***.

Операторы для управления вычислительным процессом

Операторы управления вычислительным процессом Visual Basic приведены в разделе 3. Это условный оператор If/Then/Else, оператор выбора Select Case, операторы циклов For/Next, While/Wend, Do/Loop.

Кроме того в Visual Basic для работы с перечисляемыми переменными используется оператор цикла ***For/Each***. Синтаксис оператора:

For Each <переменная цикла> In объект
[операторы]
Next переменная цикла.

Пример 6.16.

```
Dim nNumber As Integer
Dim MyArray As Variant
MyArray = Array (3,6,9,9,5,2,3,1)
For Each nNumber In MyArray
    If nNumber > 5 Then Debug.Print nNumber
Next nNumber
```

Здесь MyArray – массив, nNumber – текущий индекс элемента массива. Пока nNumber принадлежит массиву, цикл выполняется. Пользователю не надо заботиться о контроле числа элементов

в массиве. Обязательное требование – переменная цикла должна быть переменной целого типа.

Пример 6.17. Процедура очистки объектов TextBox

```
Option Explicit
Dim ctl As Control
Private Sub cmdTextBoxClear_Click()
For Each ctl In Controls
If TypeOf ctl Is TextBox Then
ctl.Text = ""
End If
Next ctl
End Sub
```

Процедура предназначена для очистки элементов управления TextBox.
Объявлена переменная *ctl* типа *Control*.
Программа проверяет все объекты, установленные на форме, и если текущий объект является текстовым полем, очищает его.

В данном примере переменная ***ctl*** является переменной типа объект. В примере используется также функция ***TypeOf***, которая проверяет принадлежность текущего объекта *ctl* заданному типу объекта:

If TypeOf ctl Is TextBox,
если *ctl* является объектом *TextBox*, тогда выполнить действия.

Примеры использования операторов управления вычислительным процессом

Примеры управления вычислительным процессом достаточно подробно рассмотрены в разделе 3. В настоящем разделе рассмотрим еще ряд примеров, учитывающих специфику вычислений в электронной таблице.

Пример 6.18. Решение квадратного уравнения.

Найти корни квадратного уравнения $ax^2 + bx + c = 0$. Для управления проектом поместите на форму кнопку Command1. Ввод данных в программу выполнить с помощью функции InputBox. Результат вывести с использованием функции MsgBox.

Порядок работы:

- Поместите на форму командную кнопку:

щелкните мышью по значку кнопки на панели ToolBox (кнопка выделяется светлым тоном);

переместите указатель мыши на форму, нажмите левую клавишу мыши и, протягивая мышь, установите требуемые размеры кнопки.

- Откройте окно программы двойным щелчком мыши по форме и выберите в списке объектов (левый список) объект Command1.
- Откройте список свойств объектов и выберите свойство Click этого объекта. Если щелкнуть дважды мышью по кнопке Command1, то сразу попадаете в обработчик события Click кнопки.

- Подготовьте контрольные данные для тестирования программы.
- Запишите в обработчик события Click текст программы:

```
Private Sub CommandButton1_Click( )
a = Val(InputBox (“Введите значение коэффициента a”))
b = Val(InputBox (“Введите значение коэффициента b”))
c = Val(InputBox (“Введите значение коэффициента c”))
D = b^2 – 4 * a * c
If D < 0 then MsgBox “Нет действительных корней”, vbCritical: Exit Sub
x1 = (–b + Sqr(D))/(2 * a) : x2 = (–b – Sqr(D))/(2 * a)
MsgBox “x1 = ” & Str(x1) & vbCr & “x2 = ” & Str(x2),,”Корни уравнения:”
End Sub
```

- Запустите программу.
- Щелкните мышью по кнопке Command1 для выполнения расчетов.
- Проверьте правильность работы программы.
- Сохраните программу на диске.

Примеры для проверки:
1) a = 2, b = 5, c = –4;
2) a = 2, b = 5, c = 4.
Результаты теста приведены на рисунке 6.64.

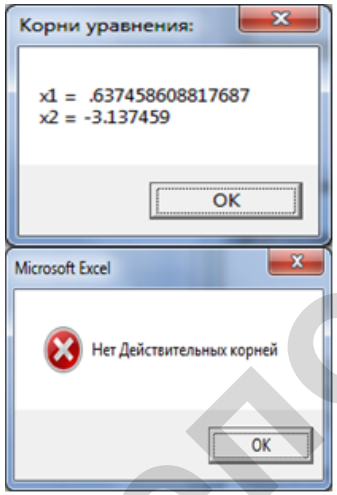


Рис. 6.64. Результаты тестов

Пример 6.19. Пример на использование функций для работы с массивами.

Разработать функцию пользователя для вычисления выражения:

$$s = \left(2\sum_{i=1}^{n} x_i + \left(\sum_{i=1}^{m}\sum_{j=1}^{m} b_{ij}c_{ij} \right)^2 \right) \Bigg/ \left(1 + \sum_{i=1}^{n} x_i^2 \right),$$

где x – вектор, содержащий три элемента;
b и c – матрицы размером 2×2;
$n = 3$, $m = 2$.

Введем в ячейки A2:A4 компоненты вектора x, в ячейки C2:D3 – элементы матрицы b, в ячейки F2:G3 – элементы матрицы c.

	A	B	C	D	E	F	G
1	Вектор x		Массив b			Массив c	
2	1,5		3	7		2	4
3	0,7		5	8		6	9
4	–2,8						
5	=Q(A2:A4;C2:D3;F2:G3)						

Первый способ. Использование базовых структур программирования – циклов.

```
Option Base 1
Function Q(x,b,c As Variant) As Double
    Dim s1, s2, s3 As Double, i, j, n, m As Integer
    n = x.Rows.Count ‘ вычисление размерности массивов
    m = b.Rows.Count ‘ вычисление размерности массивов
    s1 = 0: s2 = 0: s3 = 0 ‘ начальные значения сумм
    ‘ Вычисление суммы элементов вектора X
    For i = 1 to n
        s1 = s1 + x(i)
    Next i
    ‘ Вычисление суммы произведений элементов массивов B и C
    For i = 1 To n
        For j = 1 To m
            s2 = s2 + b(i,j) * c(i,j)
        Next j
    Next i
    s3 = 0
    ‘ Вычисление сумм квадратов вектора X
```

```
    For i = 1 to n
        s3 = s3 + x(i)^2
    Next i
    ' Вычисление значения функции
    Q = (2 * s1 + s2^2)/(1 + s3)
End Function
```

Применение функции: в ячейку результата введите формулу: =Q(A2:A4;C2:D3;F2:G3).

Примечание. В данной процедуре аргументы функции *x*, *b*, *c* объявлены как переменные типа *Variant*, в результате этого они, при вводе данных в виде массивов, стали по умолчанию массивами со свойствами *Rows* – строки и *Cols* – колонки, которые, в свою очередь, имеют свойство *Count* – количество. Это позволило определить размерности массивов *n* и *m* для организации операций с массивами.

Второй способ. Использование функций Excel:

```
=(2*СУММ(A2:A4)+СУММПРОИЗВ(C2:D3;F2:G3)^2)/(1+СУММКВ(A2:A4))
```

Третий способ. Использование функций VBA:

```
Function Q(x,b,c As Variant) As Double
    Dim s1, s2, s3 As Double
    s1=Application.Sum(x)
    s2=Application.SumProduct(b, c)
    s3=Application.SumSq(x)
    Q=(2*s1+s2^2)/(1+s3)
End Function
```

Для определения числа элементов массива можно использовать следующую команду:

```
n=Application.Count(x)
```

Оценим эти алгоритмы. Первый и третий алгоритмы оформлены в виде функций пользователя и позволяют использовать данные, расположенные в любом месте таблицы. При этом третий алгоритм намного компактнее, использует встроенные функции Visual Basic. Второй алгоритм можно использовать только в той таблице, где он записан. При изменении местоположения данных функцию каждый раз надо переписывать заново.

Ввод данных из электронной таблицы и вывод данных в электронную таблицу

В рассмотренных ранее примерах ввод и вывод данных осуществлялся на уровне формы, без связи с рабочими листами электронной таблицы. Но при работе в Excel очень часто данные размещены

в ячейках электронной таблицы, и выводить данные также необходимо в электронную таблицу.

Можно указать несколько способов ввода данных:

использование объектов TextBox;

присвоение значений переменным из ячеек электронной таблицы, например: a=Range("A1");

использование объекта RefEdit.

Для вывода данных также можно указать два способа:

использование функций пользователя. Данные выводятся в выделенные ячейки;

использование оператора Selection.

Примеры различных способов ввода данных в процедуры

1. Ввод данных в окно диалога. Результат выводится в заданный заранее участок таблицы.

На форме размещены три окна ввода TextBox, три объекта Label – надписи напротив окон ввода и две кнопки CommandButton (рис. 6.65).

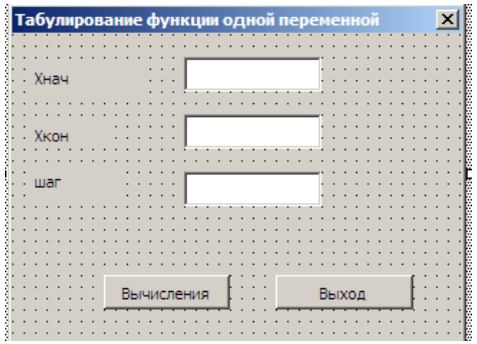


Рис. 6.65. Форма для ввода данных

```
Public Function Groop1(a, b, h)
    ' Функция пользователя для табулирования функции одной переменной
    Dim i As Single, x As Single
    Dim n As Integer
    n = Int((b – a)/h)+ 1
    ReDim ay(n, 1) As Single
    x = a
    For i = 0 To n – 1
        ay(i, 0) = x: ay(i, 1) = Round(Sin(x), 3)
```

```
        x = x + h
    Next i
    Groop1 = ay
End Function

Private Sub CommandButton1_Click() ' кнопка вычисления
    a = Val(TextBox1.Text) ' ввод данных в переменные с помощью окна ввода TextBox
    b = Val(TextBox2.Text)
    s = Val(TextBox3.Text)
    Range("A5:B20").Select ' выделение области для вывода данных
    Selection = Groop1(a, b, s) ' вывод данных из функции пользователя в указанную область
End Sub
```

Для использования формы необходимо установить на лист кнопку и в обработчик события Click этой кнопки написать команду <Имя_фомы>.Show, например, *UserForm1.Show*. Данные будут помещены в область A2:B20. Если число данных будет недостаточно, то в остальные ячейки будет выведено сообщение #н/д – недействительные данные.

2. Табулирование функции одной переменной. Аргументы функции вводятся из выделенного массива. Данные помещаются в переменную *xm* типа Variant. Данная переменная имеет свойство Rows (Строки), которое, в свою очередь, имеет свойство Count (количество) (см. пример 6.19).

```
Public Function Tab2(xm As Variant) As Variant
    Dim i As Integer ' объявление локальной переменной
    n = xm.Rows.Count ' определение числа строк в массиве входных данных
    ReDim z(n, 1) As Variant ' объявление динамического массива
    For i = 1 To n ' цикл для ввода данных в массив
    z(i, 0) = xm(i): z(i, 1) = xm(i) * 2 ' вычисленные данные присваиваются второму столбцу массива z
    Next i
    Tab2 = z ' присвоение имени функции вычисленного значения
End Function
```

Применение. Выделите два столбца с требуемым числом строк, введите функцию Tab2, на запрос программы выделите входной массив *xm*, выходной массив формируется в выделенной области (вектор *x* и *y*).

Достоинство данной процедуры: данные могут находиться в любом месте и результаты вычислений можно поместить также в любое место.

3. Табулирование функции 2-х переменных с вводом данных из массивов.

```
Option Base 1 ' Нумерация элементов массивов будет начинаться с единицы
' Оператор Option Base должен быть первым оператором, записанным в разделе Общие
Dim z() As Variant ' объявление динамического массива
----------------------------------------------------------------------------------
Public Function Tab22(xm As Variant, ym As Variant) As Variant
    Dim i As Integer, j As Integer, n As Integer, m As Integer ' объявление локальных переменных
    n = xm.Rows.Count ' определение числа строк в массиве входных данных
    m = ym.Rows.Count
    ReDim z(n, m) As Variant ' переобъявление динамического массива
    For i = 1 To n ' два вложенных цикла
        For j = 1 To m
            z(i, j) = xm(i) + 2 * ym(j)
        Next j
    Next i
    Tab22 = z ' присвоение имени функции вычисленного значения
End Function
```

Использование функции:

создайте в электронной таблице два одномерных массива X и Y. Размерности массивов могут быть разные;

выделите область с числом строк, равным размерности массива X, и числом столбцов, равным размерности массива Y;

введите в массив функцию Tab22;

введите в окно диалога функции области, занимаемые массивами X и Y;

для завершения операции нажмите комбинацию клавиш Ctrl + Shift + Enter.

4. Табулирование функций одной переменной. Исходные данные хранятся в ячейках таблицы в определенном месте. Результаты могут быть помещены в любое место.

```
Option Base 1
Public Function Tab3() As Variant
    Dim a As Single, b As Single, s As Single
    Dim i As Integer, x As Single
```

```
    a = Range("A1").Value ' ввод данных из ячеек
    b = Range("A2").Value
    s = Range("A3").Value
    n = Int((b – a)/s) + 1
    ReDim ay(n, 2) As Single
    x = a
    For i = 1 To n
        ay(i, 1) = Round(x, 2): ay(i, 2) = Round(Sin(x), 2)
        x = x + s
    Next i
    Tab3 = ay
End Function
```

Контрольные вопросы

1. Опишите среду разработки проекта.
2. Опишите среду разработки программы.
3. Дайте определение понятиям «переменные», «константы».
4. Какие типы переменных используются в VBA?
5. Что такое область видимости переменных?
6. Что такое время жизни переменных?
7. Как объявляются типы переменных?
8. Приведите примеры ввода данных в программу.
9. Как вывести результаты вычислений из функций пользователя и процедур пользователя?
10. Какие операторы используются в VB для управления вычислительным процессом?
11. Приведите синтаксис условных операторов.
12. Приведите синтаксис операторов цикла For/Next и For/Each.
13. Приведите синтаксис и схему алгоритма оператора While/Wend.
14. Приведите синтаксис оператора Do/Loop.
15. Приведите синтаксис и поясните принцип работы оператора Select Case.

6.13. ОСНОВЫ СОЗДАНИЯ И ИСПОЛЬЗОВАНИЯ СПИСКОВ В ЭЛЕКТРОННОЙ ТАБЛИЦЕ

6.13.1. Общие сведения, понятия и определения

Списки или базы данных в электронной таблице Excel представляют собой подобие информационно-справочных систем, известных как системы управления базами данных (СУБД).

СУБД – один из классов программных средств, предназначенных для создания, ведения и использования баз данных, справочных, информационно-поисковых систем. Основными компонентами информационной системы являются: база данных, система управления базами данных, прикладная программа и интерфейс. База данных содержит интересующую пользователя информацию, а также описание структуры хранимых данных. СУБД выполняет типовые процедуры управления данными, осуществляет взаимодействие с прикладной программой. Прикладная программа реализует требуемый алгоритм ведения диалога пользователя с информационной системой, ввода и контроля запросов, организации информационного поиска, выборки и представления данных в виде справок и отчетов. Взаимодействие между прикладной программой и СУБД осуществляется с помощью специальных операторов или команд языка управления базой данных.

Возможны три модели баз данных: *сетевые*, *иерархические* и *реляционные*. Сетевые и иерархические СУБД получили наибольшее распространение на больших и мини-ЭВМ. На ПК используется преимущественно реляционная модель данных. Сетевые СУБД используют модель представления данных в виде произвольного графа. В иерархических СУБД данные представляются в виде древовидной структуры. Реляционная модель ориентирована на представление данных в виде таблицы. В Excel реализована реляционная база данных.

Таблица реляционной БД (рис. 6.66) представляет собой двумерный массив и обладает следующими свойствами: каждый элемент таблицы – это один элемент данных, повторяющиеся группы отсутствуют; все столбцы (колонки) в таблице однородные. Это означает, что все элементы одного столбца имеют одинаковую природу, например: марка автомобиля или размер заработной платы. Столбцам присвоены уникальные имена; в таблице нет одинаковых строк; в операциях с таблицей ее строки и столбцы могут просматриваться в любом порядке и в любой последовательности безотносительно к их информационному содержанию и смыслу.

Каждая строка таблицы – это запись, каждая колонка таблицы – поле записи. Размещение в одной строке таблицы определенных элементов данных означает установление между ними связи или отношения. Например, если база данных содержит сведения о запасных частях к автомобилям, то в одной строке могут быть помещены сведения о запасных частях к автомобилю конкретной марки.

То есть данные в одной строке связаны между собой тем, что принадлежат одной марке автомобиля. Строка таблицы с этими данными представляет собой один конкретный экземпляр отношения данного типа, или его *кортеж*, а всю таблицу в целом называют *отношением*. Таким образом, при описании реляционной модели данных отношением называют всю таблицу в целом как совокупность конкретных экземпляров отношения. Слова «отношения» и «реляционный» (от латинского relation – отношение) представляют собой синонимы.

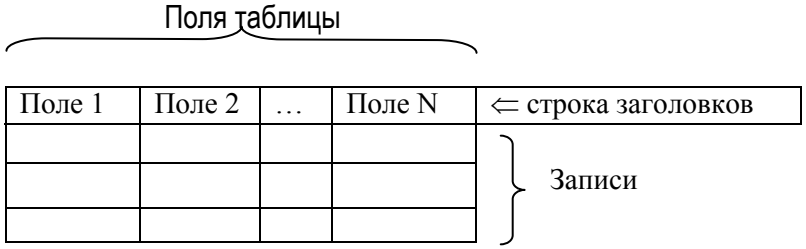


Рис. 6.66. Структура базы данных

Совокупность значений элементов данных, размещенных в одном столбце таблицы и определяющих некоторую характеристику или свойство объектов, описываемых строками таблицы, называют атрибутом отношения. Количество элементов данных в кортеже (количество столбцов в таблице) определяет степень отношения. Если таблица включает n столбцов, то она представляет собой отношение степени n. Количество кортежей в отношении (число строк в таблице) определяет его мощность – m. Тогда общее количество элементов данных в отношении степени n будет равно $n \times m$.

Атрибут, значение которого идентифицирует кортеж, то есть позволяет однозначно выделить его из других кортежей данного отношения, называется ***ключевым атрибутом***, или просто ***ключом***. Ключ может включать несколько атрибутов – составной ключ или представлять собой только часть значения атрибута – частичный ключ. В приведенном выше примере в качестве ключа может быть марка автомобиля, что позволяет однозначно выделить кортеж из всего отношения.

Программа Excel позволяет импортировать и обрабатывать данные из других баз данных. Excel позволяет создавать собственные

однотабличные базы данных, устанавливать между ними связи с помощью функций ПРОСМОТР, ВПР, ГПР, а также OLE-автоматизации, осуществлять сортировку, поиск и извлечение данных.

6.13.2. Создание баз данных

Базу данных рекомендуется создавать на отдельном листе. В этом случае программа быстрее использует команды сортировки и фильтр, а также исключается возможность испортить другие данные.

Для создания списка может использоваться любой диапазон ячеек. Тогда каждый столбец диапазона считается полем, которое может содержать строку длиной до 255 символов. Соответственно каждая строка диапазона будет считаться записью. Для создания списков и работы с ними Excel имеет специальные команды и функции.

Щелкните правой клавишей мыши по ярлычку номера страницы и введите название базы данных (по ее содержанию).

В первую строку диапазона, отведенного для создания списка, записываются имена полей. Имена полей должны быть, по возможности, простыми, краткими и описательными, при этом они не могут занимать более одной строки таблицы. В последующие строки записываются данные. Первая запись не должна ничем отделяться от строки заголовков, в списке не должно быть одинаковых записей, пустых строк и колонок. Пример базы данных приведен на листинге 6.27.

Листинг 6.27. **Пример базы данных**

	A	B	C	D	E	F	G
10	Дата	Откуда	Вид	*Количество*	*Объем*	*Цена*	Стоимость
11	Сентябрь	Братск	Бумага	*22500*	*45*	*3500*	78750000
12	Сентябрь	Братск	Ватман	*15600*	*31*	*2400*	37440000
13	Сентябрь	Вологда	Цемент	*13600*	*27*	*5800*	78880000
14	Сентябрь	Тюмень	Клей столярный	*11000*	*22*	*1200*	13200000
15	Октябрь	Братск	Картон	*12000*	*48*	*5600*	67200000
16	Октябрь	Вологда	Плитка облицовочная	*13500*	*27*	*3200*	43200000

Для работы со списками используются команды ***Сортировка, Фильтр***, которые находятся в группе Сортировка и фильтр на вкладке ***Данные***.

Команда ***Сортировка*** позволяет отсортировать выделенный диапазон по значениям нескольких полей (рис. 6.67). В списке

Сортировать по необходимо указать поле, по которому будет осуществляться сортировка. Сортировку можно осуществлять по Значению поля, Цвету ячейки, Цвету шрифта или Значку ячейки. Сортировку можно проводить по возрастанию или убыванию значения соответствующего поля. В случае сортировки базы данных столбцы будут фигурировать под названиями полей, поэтому необходимо установить флажок *Мои данные содержат заголовки* (установлен по умолчанию). Если флажок снят, то сортировка будет осуществляться по имени столбца. Кнопка ***Добавить уровень*** позволяет добавить новое поле для сортировки, а команда ***Удалить уровень*** – удаляет уровень. Кнопка ***Параметры*** позволяет указать дополнительные условия сортировки: учитывать ли регистр, сортировать строки диапазона или столбцы диапазона.

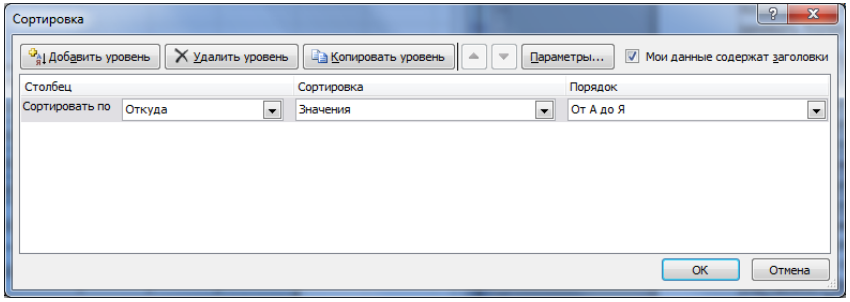


Рис. 6.67. Окно диалога Сортировка

Для выбора данных по заданным критериям в электронной таблице предусмотрено два фильтра: *автофильтр* и *расширенный фильтр*.

Автофильтр вызывается командой ***Фильтр***. При выборе команды на каждом поле устанавливается раскрывающийся список (рис. 6.68). Если щелкнуть мышью по кнопке раскрывающегося списка, то открывается список параметров (рис. 6.69). По умолчанию все флажки установлены. Для вывода на экран требуемых данных снимите ненужные флажки или активизируйте строку поиска и введите название искомого объекта (можно ввести часть названия).

Команда *Текстовые фильтры/Числовые фильтры* выводит на экран меню условий отбора значений для текстовых или числовых полей по контексту. При выборе нужного условия или команды ***Настраиваемый фильтр*** на экран выводится окно ***Пользовательского***

автофильтра, которое позволяет объединить два параметра по логическому условию И или ИЛИ (рис. 6.70). В левых раскрывающихся списках выбираются логические условия для выбора числовой или текстовой информации, а в правых раскрывающихся списках выбираются из списка или вводятся значения параметров для отбора. При указании значений параметров допускается использование маски: вопросительный знак или звездочка. Вопросительный знак заменяет один символ в текущей позиции. Звездочка заменяет все слово или его часть. После применения фильтра все записи, не удовлетворяющие заданным критериям, убираются с экрана. Для отображения всех записей необходимо удалить фильтр командой в списке (рис. 6.69) или щелкнуть по кнопке ***Очистить*** в группе ***Сортировка и фильтр*** вкладки ***Данные***.

Дата	Откуда	Вид
Сентябрь	Братск	Бумага
Сентябрь	Братск	Ватман
Октябрь	Братск	Картон
Сентябрь	Вологда	Цемент
Октябрь	Вологда	Плитка облицовочная
Сентябрь	Тюмень	Клей столярный

Рис. 6.68. Установка фильтра

Кнопка ***Дополнительно*** в группе ***Сортировка и фильтр*** вызывает окно диалога ***Расширенного фильтра*** (рис. 6.71).

Расширенный фильтр предоставляет пользователю дополнительные возможности по выбору критериев и формированию результатов: список можно фильтровать на месте или скопировать результат в указанный диапазон; условия для выбора задаются в отдельном диапазоне рабочего листа; можно использовать при выборе только уникальные записи.

Блок критериев для расширенного фильтра содержит условия для поиска и выборки данных. Он может располагаться в любом месте электронной таблицы. Однако рекомендуется блок критериев для расширенного фильтра задавать выше базы данных (списка), отделяя его от базы данных одной строкой.

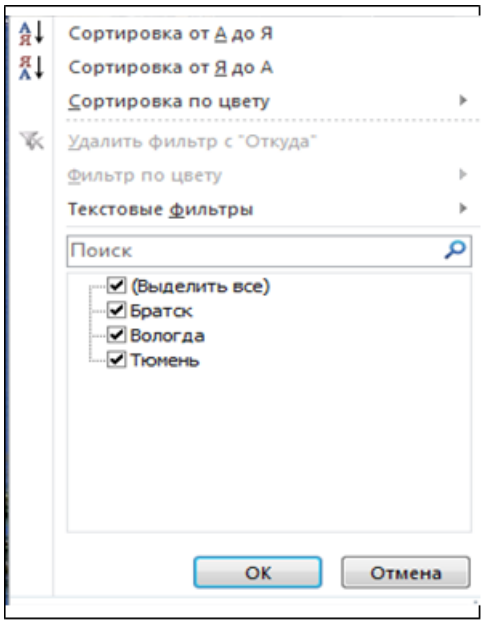


Рис. 6.69. Меню фильтра

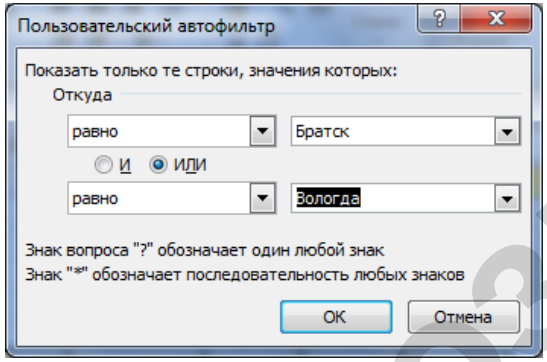


Рис. 6.70. Окно просмотра свойств объектов

Блок критериев состоит из двух или более строк. В первой строке задаются имена полей, а в последующих строках – значения критериев поиска. Значением критерия может быть текстовая или числовая константа, логическая функция или логическое выражение.

Если критерий содержит несколько строк, то считается, что эти строки связаны функцией ИЛИ. Если строка критерия содержит несколько полей в одной строке, то считается, что эти поля связаны функцией И. Если блок критерия содержит несколько полей, но значения полей находятся в разных строках, то это значит, что критерии объединены по схеме ИЛИ. Примеры блоков критериев для расширенного фильтра приведены на листинге 6.28.

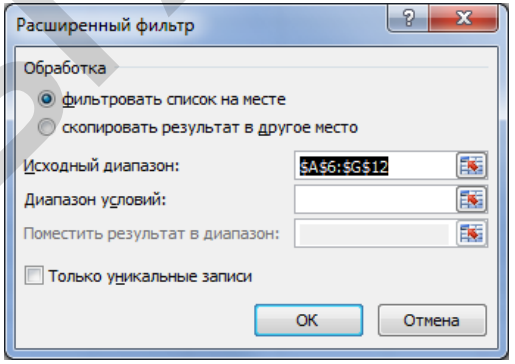


Рис. 6.71. Окно диалога расширенного фильтра

Листинг 6.28. **Примеры блоков критериев**

	B	C	D	E	F
1	**Простой Критерий**	**Схема И**		**Схема ИЛИ-И**	
2	**Дата**	**Вид**	**Количество**	**Вид**	**Цена**
3	Октябрь	Ватман	>10000	Ватман	<2500
4				Картон	<6000
5	**Схема ИЛИ**		**Схема ИЛИ**		
6	**Откуда**		**Дата**	**Откуда**	
7	Братск		Сентябрь		
8	Тюмень			Братск	

Простой критерий представляет собой совокупность имени поля и значения (блок B2:B3).

В блоке критериев B6:B8 два наименования объединены по ***схеме ИЛИ*** – имя поля Откуда, значения Братск и Тюмень. То есть при наличии на складе указанных товаров все они будут включены в выходной список.

В блоке критериев C2:D3 два параметра объединены по ***схеме И***, имена полей Вид и Количество. Это значит, что для выбора товара должны быть выполнены оба условия: в базе данных будет отыскиваться ватман, поступивший на склад в количестве больше 10 000 кг.

В блоке критериев E2:F4 два разных критерия объединены по схеме ИЛИ-И, то есть будут отыскиваться ватман по цене меньше 2500 руб ИЛИ картон по цене меньше 6000 руб.

В блоке D6:E8 критерии объединены по схеме ИЛИ, то есть на экран будут выведены все товары, поступившие в сентябре, и все товары, поступившие из Брянска.

В поле критерия для текстовых данных могут использоваться шаблоны ?, *. Символ ? заменяет один знак в указанной позиции. Например, «ию?ь» совпадает с «июнь» и «июль». Символ * заменяет все слово или его часть. Например, «авто*» соответствует «автомобиль», «автокар» и т. д.

Для получения выборки с помощью расширенного фильтра необходимо:

1) создать блок критериев;

2) выделить базу данных (для выделения базы данных достаточно установить курсор в любую ячейку этой базы данных);

3) ввести команду ***Данные, Дополнительно***;

4) указать в окне диалога Расширенный фильтр (рис. 6.71) исходный диапазон и диапазон условий;

5) если предполагается получить выборку на месте, то следует нажать клавишу ОК. Если предполагается поместить выборку в другое место, то следует активизировать флажок ***Скопировать результат в другое место*** и указать в строке ввода ***Поместить результаты в диапазон*** начальную ячейку выходного блока данных.

При обработке баз данных полезными являются функции обработки данных:

ДСРЗНАЧ() – среднее значение элементов базы данных, соответствующих заданному критерию;

БСЧЕТ() – количество записей в базе данных, удовлетворяющих заданному критерию;

ДМАКС() – максимальное значение записей, соответствующих критерию, заданному в поле;

ДМИН() – минимальное значение записей, соответствующих критерию, заданному в поле;

БДСУМ() – сумма значений записей, соответствующих критерию, заданному в поле; и другие функции.

Все функции базы данных имеют одинаковый синтаксис:

<Имя_функции>(*база_данных, поле, критерий*).

С помощью аргумента *база данных* в функцию передается диапазон ячеек, подлежащих обработке. Можно ссылаться на имя Базы данных.

Аргумент *поле* идентифицирует поле базы данных, с которым предполагается проводить вычисления. Для обозначения поля можно использовать как номер столбца в базе данных, так и имя поля базы данных.

Аргумент *критерий* соответствует ссылке на диапазон условий. Диапазон условий задается так же, как и при формировании блока критериев для расширенного фильтра.

Например, для вычисления суммы количества ватмана, поступившего от поставщиков (листинг 6.27), следует ввести в ячейку формулу:

БДСУММ(A10:G16;D10;C2:D3)

или

БДСУММ(ПокупныеИзделия;"Количество";C2:D3).

Здесь ПокупныеИзделия – имя базы данных, «Количество» – имя поля, C2:D3 – блок критериев.

6.13.3. Анализ данных

Для анализа данных можно использовать команды ***Структура, Промежуточные Итоги*** вкладки ***Данные*** и ***Сводная таблица*** на вкладке ***Вставка***.

Команда ***Промежуточные итоги*** позволяет получать сводные данные по числовым параметрам: сумму, минимум, максимум, среднее значение и другие статистические данные. Предварительно необходимо определить, по какому параметру требуется группировать итоги, и *отсортировать* данные по этому параметру. В примере на рисунке 6.72, листинге 6.29 в качестве такого параметра выбрана дата поступления.

Затем необходимо выделить базу данных и ввести команду ***Данные, Структура, Промежуточные итоги***. На экране появится окно диалога Промежуточные итоги (рис. 6.72). Выберите в списке *При каждом изменении в:* параметр *Дата*, в списке *Операция* выберите вид операции *Сумма*. В списке *Добавить итоги по:* установите флажок для параметра *Стоимость*. Установите также флажки *Заменить текущие итоги* и *Итоги под данными*.

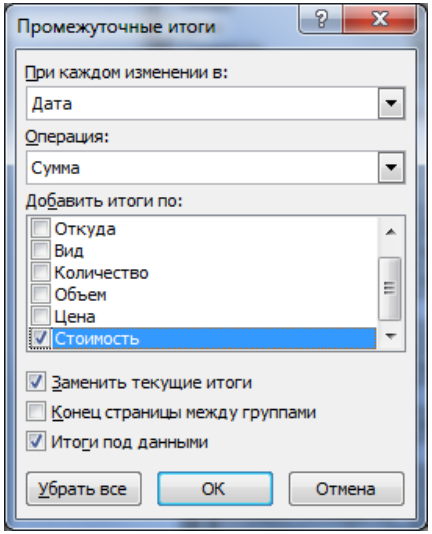


Рис. 6.72. Окно диалога Промежуточные итоги

Листинг 6.29. **База данных после выполнения команды Итоги**

	A	B	C	D	E	F	G
10	Дата	Откуда	Вид	Количество	Объем	Цена	Стоимость
11	Октябрь	Братск	Картон	12000	48	5600	67200000
12	Октябрь	Вологда	Плитка облицовочная	13500	27	3200	43200000
13	**Октябрь Итог**						110400000
14	Сентябрь	Братск	Бумага	22500	45	3500	78750000
15	Сентябрь	Братск	Ватман	15600	31	2400	37440000
16	Сентябрь	Тюмень	Клей столярный	11000	22	1200	13200000
17	Сентябрь	Вологда	Цемент	13600	27	5800	78880000
18	**Сентябрь Итог**						208270000
19	**Общий итог**						318670000

Если флажок *Итоги под данными* не установлен, то итоговые данные выводятся сразу же после шапки таблицы, в ином случае итоговые данные будут размещены в конце таблицы.

Флажок *Конец страницы между группами* позволяет разбить данные по датам и напечатать их на отдельных листах. При малом объеме данных это делать нецелесообразно.

Кнопка ***Убрать все*** удаляет все итоги и переводит базу данных в исходное состояние.

После установки всех параметров щелкните по кнопке ОК – база данных преобразуется к виду, представленному на листинге 6.29. Под каждым месяцем помещены промежуточные итоги, а внизу таблицы – Общий итог. Слева от таблицы появляются кнопки для управления уровнем просмотра. При щелчке по кнопке 2 на экране остаются только промежуточные и общие итоги, при щелчке по кнопке 1 – только общие итоги.

Структуры данных

В группе ***Структуры данных*** вкладки ***Данные*** имеются еще две полезные команды: ***Группировать*** и ***Разгруппировать***. Эти парные команды позволяют управлять представлением данных на экране. Предположим, что мы имеем данные по успеваемости студентов в группах. Для анализа успеваемости нужен полный список. Однако в учебном отделе интересуются только общим баллом успеваемости студентов. Поэтому при представлении данных об успеваемости можно сократить список, свернув ненужные строки или столбцы. Пример структурирования данных приведен на листинге 6.30.

Листинг 6.30. **Группировка данных**

	A	B	C	D	E	F	G
59							
60							
61	Группа	Фамилия, Инициалы	Оценки по предметам обучения				Средний балл
62			Физика	Математи	Белорусс	История	
63	A1						
64	A1						
65	A1						
66	A1						
67	A1						
68	Средний балл						
69	A2						
70	A2						
71	A2						
72	A2						
73	Средний балл						

В таблице на листинге 6.30 можно выполнить группировку столбцов оценок по предметам обучения, а также группировку по группам. Для выполнения группировки оценок по предметам обучения выделим ячейки с наименованиями предметов обучения и введем команду ***Данные, Структура, Группировать***. На запрос программы о способе группировки ответим: ***По столбцам***. Над номерами столбцов появляются кнопки управления. Кнопки 1 и 2 изменяют уровень просмотра: кнопка 1 показывает верхний уровень, то же самое выполняет кнопка Свернуть (–); кнопка 2 показывает все, то же самое выполняет кнопка Развернуть (+). Аналогично выполним группировку строк в пределах каждой группы, но на запрос программы о способе группировки ответим: ***По строкам***.

Пример таблицы со свернутыми списками групп приведен на листинге 6.31.

Листинг 6.31. **Список со свернутыми списками групп**

		A	B	C	D	E	F	G
	60							
	61	Группа	Фамилия, Инициалы	Оценки по предметам обучения				Средний балл
	62			Физика	Математи	Белорусс	История	
+	68	Средний балл						
+	73	Средний балл						

6.13.4. Сводные таблицы

Команда Сводные таблицы позволяет формировать новые таблицы на базе существующих таблиц в различных разрезах и проводить анализ полученных результатов.

Создание сводных таблиц осуществляется с помощью команды ***Сводная таблица*** вкладки ***Вставка*** или с помощью Мастера сводных таблиц (по умолчанию данная команда отсутствует на ленте, но ее можно добавить, при необходимости, на панель быстрого доступа). Команда Сводные таблицы позволяет создавать как собственно сводную таблицу, так и сводную диаграмму.

После ввода команды открывается окно диалога (рис. 6.73). В этом окне, прежде всего, необходимо выбрать источник данных: таблицу Excel или внешний источник данных. Если источником данных

является электронная таблица, то укажите в соответствующем окне диапазон исходных данных. А также укажите, куда поместить сводную таблицу: на текущий лист или на новый лист. После нажатия клавиши ОК откроется макет сводной таблицы (рис. 6.74). При этом открывается и новая вкладка Работа со сводными таблицами.

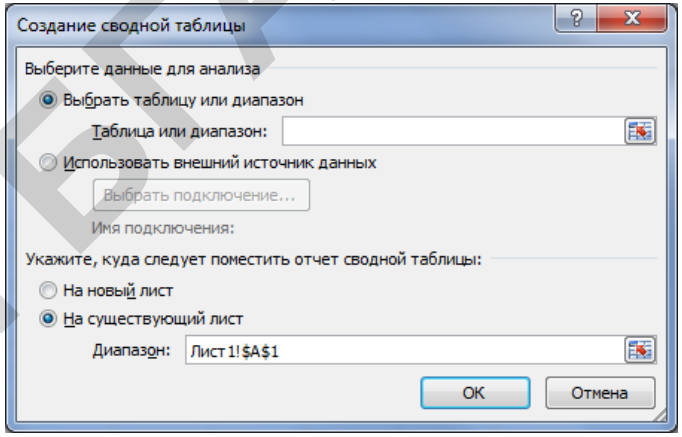


Рис. 6.73. Окно диалога Создание сводной таблицы

Справа в макете находится список полей исходной таблицы, в котором размещены также четыре окна, отражающие структуры сводной таблицы. Видом этой области можно управлять с помощью списка *Разделы полей и областей*, расположенного в правом верхнем углу области. Слева размещен шаблон полей сводной таблицы. Перетащите туда мышью нужные поля, например, в область фильтра отчета переместите поле Дата, в область строк поместите поле Откуда, в область столбцов перетащите поле Вид, а в область Значения – поле Цена. В область Значения помещаются, как правило, числовые поля, но можно помещать и текстовые поля, тогда для обработки их используется функция *Count*.

По мере ввода полей формируется структура сводной таблицы (рис. 6.75). В области Фильтр создается раскрывающийся список, который распределяет данные таблицы по датам. По умолчанию показаны все данные. Для сортировки данных по датам снимите ненужные флажки. Список Вид содержит список товаров по видам. По умолчанию все флажки также активизированы. Список Вид по структуре соответствует списку автофильтра базы данных (рис. 6.69).

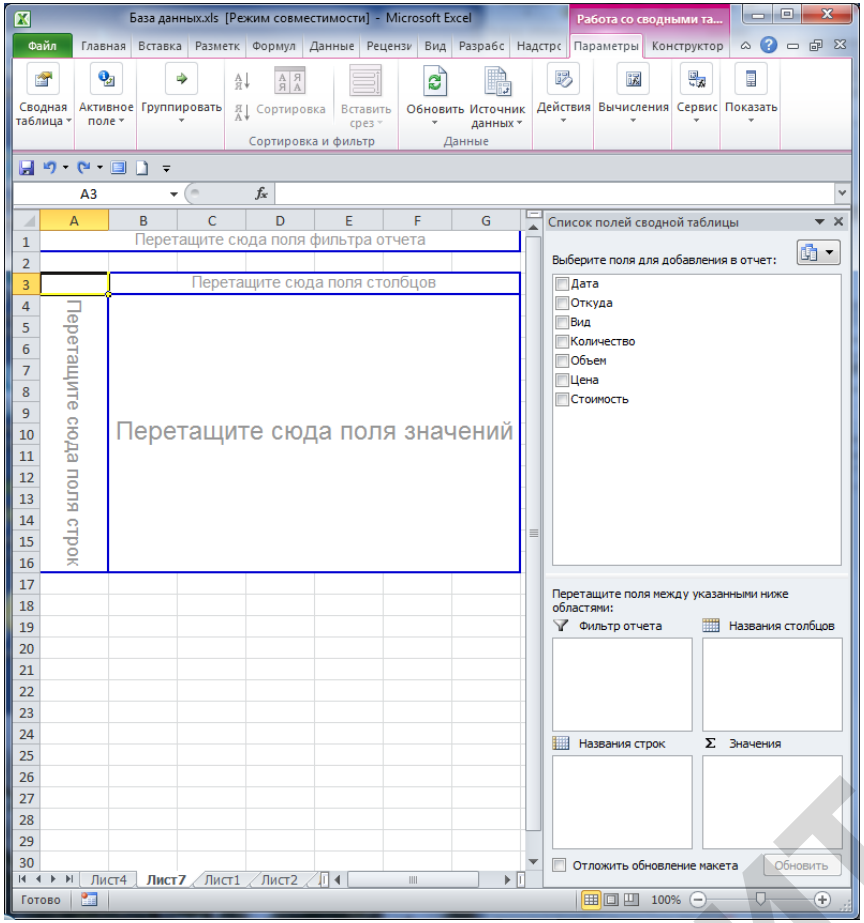


Рис. 6.74. Создание сводной таблицы

	A	B	C	D	E	F	G	H
1	Дата	(Все)						
2								
3	Сумма по полю Цена	Вид						
4	Откуда	Бумага	Ватман	Картон	Клей столярный	Плитка облицовочная	Цемент	Общий итог
5	Братск	3500	2400	5600				11500
6	Вологда					3200	5800	9000
7	Тюмень				1200			1200
8	Общий итог	3500	2400	5600	1200	3200	5800	21700

Рис. 6.75. Сводная таблица

Мастер сводных таблиц формирует таблицу за четыре шага.

На первом шаге предлагается выбрать источник данных и вид отчета: сводная таблица или сводная диаграмма. В качестве источника данных может быть база данных Excel или внешняя база данных, например, база данных Microsoft Access, Visual FoxPro, dBase-Word. На втором шаге предлагается указать диапазон, содержащий исходные данные. На третьем шаге можно указать место для сводной таблицы (на текущем листе или на новой странице). На следующем шаге открывается макет сводной таблицы (рис. 6.74).

6.13.5. Консолидация

Еще одной операцией с базами данных является ***консолидация***. Консолидация означает совместную обработку данных двух и более диапазонов, например, требуется сложить данные из столбцов двух диапазонов. При этом диапазоны могут находиться в одной таблице, на одном и том же листе, на нескольких листах рабочей книги и даже в разных книгах. Консолидация может проводиться двумя способами:

1. Консолидация по расположению ячеек – в этом случае состав и порядок расположения данных во всех диапазонах должен быть одинаков.

2. Консолидация по категориям – в этом случае консолидация проводится на основании одинаковых подписей строк и столбцов диапазонов.

Введем команду ***Консолидация*** из группы ***Работа с данными*** вкладки ***Данные*** – на экране появится окно диалога Консолидация (рис. 6.76). Список ***Функция*** позволяет выбрать операцию над консолидируемыми данными. При консолидации доступны все функции статистических итогов (сумма, минимум, максимум и т. д.). Строка ввода ***Ссылка*** позволяет указать или выбрать диапазон консолидируемых данных. Список диапазонов хранит списки диапазонов консолидируемых данных. Кнопка ***Добавить*** позволяет добавлять данные в список диапазонов через строку Ссылка.

Флажки в группе ***Использовать в качестве имен*** используются при втором способе консолидации. При установке флажка ***Создавать связи с исходными данными*** в случае изменения исходных данных одновременно будет изменяться и результат. Связи нельзя использовать, если исходная область и область назначения находятся на одном листе. После установки связей нельзя добавлять

новые исходные области и изменять исходные области, уже входящие в консолидацию.

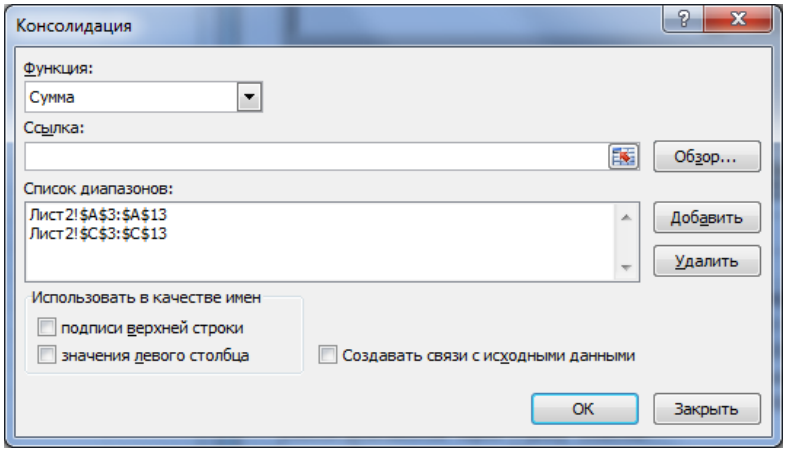


Рис. 6.76. Консолидация данных

Кнопка ***Обзор*** служит для выбора данных в других книгах.

Пример 6.20. Пусть требуется найти среднее значение для каждой строки данных в столбцах A и B. Результаты поместить в столбец C.

Алгоритм работы:

выделите диапазон ячеек, равный высоте столбца A без учета заголовков;

введите команду ***Данные, Консолидация***;

активизируйте строку ввода *Ссылка*, выделите первый диапазон в столбце A и щелкните по кнопке *Добавить*;

активизируйте строку ввода *Ссылка*, выделите второй диапазон в столбце B и щелкните по кнопке *Добавить*;

выберите функцию *Среднее* в списке *Функция*;

щелкните по кнопке OK.

Состояние окна диалога Консолидация после ввода данных и выбора функции приведено на рисунке 6.76, а результат – на листинге 6.32.

Контрольные вопросы

1. Что такое база данных?
2. Какие типы баз данных существуют?

Листинг 6.32. **Консолидация данных**

A		B		C
1		2		1,5
2		3		2,5
3		4		3,5
4		5		4,5
5		6		5,5
6		7		6,5
7		8		7,5
8		9		8,5

3. Какой тип базы данных реализован в электронной таблице?
4. Какие требования необходимо соблюдать при создании базы данных?
5. Что такое автофильтр, как он создается?
6. Что такое расширенный фильтр, как его создать?
7. Как создать блок критериев: а) простой; б) по схеме ИЛИ; в) по схеме И?
8. Для чего предназначена команда Итоги?
9. Что такое сводные таблицы, как они создаются?
10. Для чего необходима консолидация, как она осуществляется?

Заключение

Электронная таблица Excel позволяет создавать и вести базы данных. При этом под ведением базы данных подразумевается редактирование базы данных, добавление и удаление записей, поиск и извлечение данных по заданному критерию.

Каждую базу данных рекомендуется создавать на отдельном листе.

Добавление, удаление записей осуществляются так же, как и в обычной таблице. Поиск информации по заданному критерию осуществляется с помощью фильтров в режиме автофильтр.

Выборка и извлечение данных осуществляются с помощью фильтров: автофильтр и расширенный фильтр. Автофильтр позволяет создавать критерии для выборки данных по соответствующему полю по схеме И и ИЛИ на месте базы данных. Расширенный фильтр позволяет осуществлять выборку и извлечение данных как на месте базы данных, так и в другое место в пределах рабочего листа. Для использования расширенного фильтра необходимо создавать отдельный блок критериев.

Для подведения промежуточных итогов используется команда Итоги. Перед использованием команды база данных должна быть отсортирована по тому полю, которое будет использоваться для группировки данных.

Команда Сводные таблицы позволяет создавать на основе существующей базы данных новые базы данных, упорядоченные по строкам и полям. Сводные таблицы используются для анализа информации, содержащейся в базах данных.

При необходимости использования и совместной обработки данных из разных таблиц или диапазонов одной таблицы можно воспользоваться командой Консолидация.

7. ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ И СЕТИ

Ключевые слова: вычислительный комплекс, вычислительная сеть, вычислительная система, веб-страница, Интернет, компьютерная сеть, модем, протокол, трафик.

7.1. ОСНОВНЫЕ ПОНЯТИЯ

Под *вычислительной системой* понимают совокупность взаимосвязанных и взаимодействующих процессов или ЭВМ, периферийного оборудования, программного обеспечения, предназначенных для автоматизации приема, хранения, обработки и выдачи информации. В зависимости от способа комплексирования различают вычислительные комплексы (многомашинные системы), вычислительные системы (многопроцессорные системы), вычислительные сети, сети ЭВМ (компьютерные сети).

Вычислительные комплексы представляют собой совокупность двух и более вычислительных машин, каждая из которых имеет собственную оперативную память и работает под управлением своей операционной системы. Каждая машина использует другую машину как канал или устройство ввода-вывода, каким бы общим ни был их интерфейс. Обмен информацией между машинами происходит в результате взаимодействия их операционных систем. Информационное взаимодействие компьютеров в многомашинных вычислительных системах может быть организовано на уровне процессоров, оперативной памяти или каналов связи. Вычислительные комплексы применяются чаще всего для создания надежных автоматизированных систем управления технологическими процессами (АСУТП) и техническими системами, например, вычислительный комплекс на борту космического корабля.

Вычислительные системы (многопроцессорные вычислительные системы) представляют собой совокупность процессоров, оперативной памяти, внешних устройств. Причем все процессоры ра-

ботают под управлением одной операционной системы. В этом случае достигается более быстрый обмен информацией между процессорами, чем между ЭВМ в вычислительном комплексе, и более высокая суммарная производительность системы. Поэтому вычислительные системы применяются для повышения вычислительной мощности и скорости вычислений. Информационное взаимодействие процессоров осуществляется на уровне регистров процессорной памяти или на уровне оперативной памяти. Последний тип взаимодействия используется чаще всего, так как организуется проще. Примером многопроцессорных вычислительных систем могут быть серверы крупных сетей, работающие под управлением многопроцессорной операционной системы, например, Windows XP Server, а также суперкомпьютеры.

Известны три разновидности высокопараллельных многопроцессорных вычислительных систем (МПВС):

магистральные (конвейерные) МПВС. В этих системах процессор одновременно выполняет разные операции над одним последовательным потоком обрабатываемых данных. По принятой классификации их относят к системам с многократным потоком команд и однократным потоком данных (MISD);

векторные МПВС, у которых все процессоры одновременно выполняют одну команду над различными потоками данных – однократный поток команд с многократным потоком данных (SIMD);

матричные МПВС, в этих системах процессор одновременно выполняет разные операции над последовательными потоками обрабатываемых данных – многократный поток команд с многократным потоком данных (MIMD).

В суперкомпьютерах наибольшую эффективность показывает параллельно-векторная модификация, то есть векторная структура обработки данных (рис. 7.1).

Вычислительной сетью называется система взаимосвязанных и распределенных по фиксированной территории вычислительных центров или ЭВМ, ориентированных на коллективное использование общесетевых ресурсов: аппаратных, программных и информационных. Основное назначение сети – обеспечение удобного и надежного доступа пользователей к распределенным по территории общесетевым ресурсам и организации коллективного их использования.

Компьютерные сети представляют собой дальнейшее развитие разобщенной вычислительной сети.

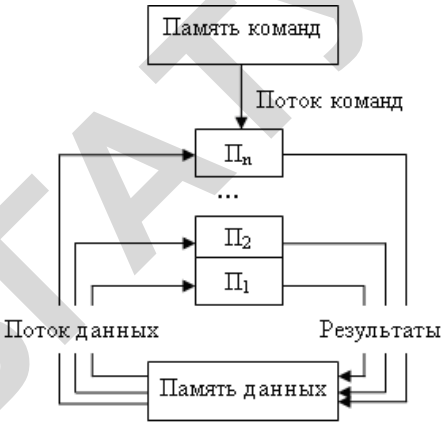


Рис. 7.1. Структура многопроцессорной вычислительной системы типа SIMD

7.2. КОМПЬЮТЕРНЫЕ СЕТИ

Компьютерной сетью или сетью электронных вычислительных машин называется совокупность взаимосвязанных и распределенных по некоторой территории ЭВМ.

Компонентами компьютерной сети являются компьютеры, коммутирующие устройства, каналы связи.

Организация обмена информацией между компьютерами осуществляется с помощью протоколов. *Протокол* – это соглашение о том, как абоненты взаимодействуют друг с другом.

Методы передачи сообщений: симплексный, дуплексный, полудуплексный. *Симплексный* метод – передача информации осуществляется только в одном направлении. *Полудуплексный* метод – передача ведется в обоих направлениях поочередно. *Дуплексный* метод передачи информации – передача ведется одновременно в обоих направлениях.

Передача информации осуществляется в виде сообщений.

Поток сообщений канала, или его нагрузка, называется ***трафиком***. Пропускная способность канала измеряется в ***бодах***. Для простых сигналов 1 бод = 1 бит/с. Если изменение сигнала (фаза и амплитуда) могут быть не единичными, то 1 бод > 1 бит/с.

Типы доступа к сети: On-Line и Off-Line. Тип доступа On-Line обеспечивает немедленное получение ответа на запрос пользовате-

ля. Такой режим обеспечивается при работе, например в сети ИНТЕРНЕТ. Тип Off-Line позволяет получить ответ через неопределенное время. Этот режим характерен для электронной почты.

Классификация сетей

В зависимости от протяженности компьютерные сети делятся на глобальные, региональные и локальные, а по платности услуг – на коммерческие и не коммерческие.

Локальные сети имеют небольшую протяженность (до 10 км) и предназначены для объединения компьютеров в пределах одного здания или нескольких зданий, распределенных на небольшой территории.

Региональные сети объединяют компьютеры на значительной территории (район, область). Эти же сети могут иметь принадлежность к какой-либо отрасли народного хозяйства, министерства, ведомства, доступ к ресурсам которых ограничен. Такие сети называются *корпоративными*. Например, вычислительная сеть министерства обороны, министерства путей сообщений и т. п.

Глобальные сети объединяют компьютеры на большой территории независимо от ведомственной принадлежности.

Линии и каналы передачи информации

Линии связи – это физическая среда, по которой передаются информационные сигналы. В одной линии связи может быть организовано несколько каналов связи путем временного, частотного, кодового и других видов разделения. Если канал полностью монополизирует линию связи, то он может называться физическим каналом и в этом случае совпадает с линией связи.

Каналы связи могут быть классифицированы по нескольким признакам:

по физической природе:	механические, акустические, оптические, электрические
по форме представления передаваемой информации:	аналоговые, дискретные
в зависимости от возможных направлений передачи информации:	*симплексные* – информация передается только в одном направлении; *полудуплексные* – информация передается в обоих направлениях, но попеременно;

	дуплексные – передача может вестись одновременно в обоих направлениях
по наличию коммутации:	коммутируемые КС, выделенные КС
по пропускной способности:	*низкоскоростные КС*, скорость передачи от 50 до 200 бит/с – это телеграфные каналы связи; *среднескоростные КС*, скорость передачи от 300 до 9600 бит/с, а согласно новым стандартам v.90–v.92 Международного консультативного комитета по телеграфии и телефонии (МККТТ) – и до 56 000 бит/с; *высокоскоростные КС*, обеспечивают скорость передачи информации свыше 56 000 бит/с

Аналоговые каналы связи могут передавать информацию со скоростью до 9600 бит/с при простом кодировании и до 56 000 бит/с при сложном кодировании сигнала.

Цифровые каналы связи, организованные на базе телефонных линий связи, могут обеспечить скорость передачи сигналов до 64 кбит/с, а при мультиплексировании нескольких цифровых каналов в один в таком составном канале связи скорость передачи можно удваивать, утраивать и т. д., существуют подобные каналы связи со скоростями в десятки и сотни мегабит в секунду. В настоящее время наиболее распространенной и активно развивающейся является цифровая сеть с интеграцией услуг (ISDN), опирающаяся на цифровые абонентские каналы. В 1990 году компания Bellcore предложила технологию NDSL, которая при использовании двух или трех пар проводов обеспечивает скорость передачи данных от 1,544 до 2,048 Мбит/с. Существующие в настоящее время стандарты и технологии обеспечивают скорость передачи до 8 Мбит/с. При использовании оптоволоконных линий связи в цифровых магистралях с синхронно-цифровой иерархией SDN ожидается достижение скоростей передачи данных до 10 Гбит/с.

В качестве физической среды передачи информации используется эфир, оптоволоконный кабель, коаксиальный кабель, витая пара и различные виды плоских кабелей.

В *эфире* может быть образовано множество каналов передачи сообщений, где используются разные несущие частоты. В зависи-

мости от используемой полосы частот различают радиоканал, инфракрасный канал, ультракоротковолновый канал, микроволновый канал.

Радиоканал характеризуется большой скоростью передачи информации – до 50 Мбит/с, большой дальностью передачи, но подвержен влиянию радиопомех.

Инфракрасный канал удобен для передачи информации по разветвленным каналам связи в пределах прямой видимости. Благодаря инфракрасным частотам, канал не подвержен влиянию радиопомех, электромагнитным и промышленным помехам. Может с успехом применяться на промышленных предприятиях.

Ультракоротковолновый канал использует ультракоротковолновый диапазон радиочастот, приемопередающую радио- и телевизионную аппаратуру. Этот канал позволяет использовать для передачи информации маломощную приемо-передающую аппаратуру, обладает повышенной помехоустойчивостью к амплитудно-модулированным электромагнитным помехам.

Для повышения скорости передачи информации может использоваться спутниковая система связи.

Микроволновый канал – диапазон видимой части спектра электромагнитных излучений. Для организации микроволнового канала можно использовать две среды: атмосферу и световоды. В атмосфере микроволновый канал может быть образован лишь в пределах прямой видимости. Световоды позволяют создать высокоскоростной и помехоустойчивый канал связи. Каналы связи на основе световодов известны как оптоволоконные линии связи (множество световодов, объединенных в один кабель).

Диапазон радиоволн приведен в таблице 7.1.

Таблица 7.1

Диапазон радиоволн

Сверхдлинные волны	3–30 кГц
Длинные волны	30–300 кГц
Средние волны	300–3000 кГц
Короткие волны	3–30 МГц
Ультракороткие волны	30–300 МГц
Субмиллиметровые волны	300–6000 МГц

Оптоволоконные кабели представляют собой набор стеклянных или пластиковых волокон диаметром 8–10 мкм для однолучевых

волокон и 50–60 мкм для многолучевых волокон, окруженных твердым заполнителем и помещенных в защитную оболочку диаметром 125 мкм. В одном кабеле может содержаться от одного до нескольких сотен таких волокон. Частота пропускания оптоволоконного кабеля – до сотен гигагерц (10^{11} Гц), а скорость передачи достигает 1000 Мбит/с.

Коаксиальный кабель представляет собой голый медный провод, окруженный диэлектриком, экраном и наружной оболочкой. Различают толстый коаксиальный кабель (проводник диаметром 2,17 мм, наружный диаметр – 12,5 мм) и тонкий коаксиальный кабель (диаметр проводника – 0,9 мм, наружный диаметр – 5–6 мм). Каналы связи на основе коаксиального кабеля обеспечивают достаточно высокую скорость передачи информации (до 10 Мбит/с для тонкого кабеля и до 50 Мбит/с для толстого кабеля) и хорошую помехоустойчивость.

Витая пара – пара скрученных проводов (для повышения помехоустойчивости к электромагнитным помехам) – самое распространенное для массового пользователя средство соединения компьютеров в сеть. В частности, используются неэкранированные (UTP) и экранированные (STP) витые пары из медных проводов. Выделяют 5 категорий витых пар. Они допускают передачу данных со скоростью до 16, 25 и 155 Мбит/с, а при использовании технологии Gigabit Ethernet и до 1000 Мбит/с.

Плоский кабель представляет собой совокупность проводников, объединенных общей экранирующей сеткой и изолированных друг от друга. Их конструкция ориентирована на передачу информации в параллельном коде по 8, 16 и 32 бита.

Показатели потребительских качеств компьютерной сети

К основным показателям компьютерной сети, определяющим ее потребительские качества, относятся следующие показатели:

операционные возможности;
производительность;
пропускная способность;
достоверность;
надежность работы;
время доставки сообщения;
стоимость предоставляемых услуг.

Операционные возможности – это перечень услуг, предоставляемых сетью пользователю. Этот перечень зависит от установленного на сервере программного обеспечения. Компьютерные сети

должны обеспечивать пользователя всеми традиционными видами услуг: средствами автоматизации программирования, доступом к пакетам прикладных программ, базам данных, обеспечивать полную доступность сетевых ресурсов всем пользователям, возможности параллельной обработки данных, построения распределенных баз данных, дублирование баз данных для повышения защищенности их от повреждения компьютерными вирусами, отказами по питанию и др.

Производительность – это среднее количество запросов пользователей сети, исполняемых за единицу времени.

Пропускная способность системы (канала) передачи информации – наибольшее теоретически достижимое количество информации, которое может быть передано по системе в единицу времени. Она определяется физическими свойствами канала связи и сигнала.

Основными характеристикам канала связи являются: полоса пропускания канала связи – полоса частот, которую может пропустить канал, не внося заметного нормированного затухания сигнала; динамический диапазон, равный отношению максимально допустимого уровня сигнала в канале к уровню помех, нормированного для этого типа каналов; время, в течение которого канал используется для передачи данных.

Основными характеристиками сигнала являются: ширина спектра частот, занимаемая сигналом; динамический диапазон, представляющий собой отношение средней мощности сигнала к средней мощности помехи в канале, длительность сигнала, то есть время его существования.

Организация компьютерных сетей

Требования к компьютерной сети

Базовыми требованиями, предъявляемыми к вычислительной сети, являются:

открытость *(масштабируемость)* – возможность расширения путем подключения дополнительных компьютеров, терминалов, узлов связи;

живучесть – способность сохранять работоспособность при изменении структуры в результате выхода из строя отдельных узлов связи, каналов связи, компьютеров);

адаптивность – допустимость изменения типов компьютеров, терминалов, линий связи, операционных систем;

эффективность – обеспечение требуемого качества обслуживания пользователей при минимальных затратах.

Указанные требования обеспечиваются модульной организацией управления процессами в сети, реализуемой по многоуровневой схеме. В основе этой схемы лежат следующие понятия:

процесс;

уровень управления;

интерфейс;

протокол.

Процесс – динамический объект, реализующий собой целенаправленный акт обработки данных. Процесс порождается программой или пользователем и связан с данными, поступающими извне. Ввод и вывод данных осуществляется в виде сообщения. Ввод сообщений в процесс и вывод сообщений из процесса осуществляется через логические (программно организованные) «точки», которые называются *портами*. Интервал времени, в течение которого взаимодействуют процессы, называется *сеансом* (*сессией*).

Система управления организуется по многоуровневой схеме. Число уровней и распределений функций между ними влияет на сложность программного обеспечения сети. Использование многоуровневой системы управления призвано обеспечить независимость от средств, реализующих сеть. В сетях ЭВМ международной организацией OSI (организация открытых систем) установлено семь уровней протоколов. В сети Интернет используется четырехуровневая система протоколов.

Протоколы **–** это соглашения по организации взаимодействия процессов в сетях ЭВМ.

Архитектура клиент-сервер

Взаимодействие компьютеров в сети организуется по схеме клиент-сервер. *Сервер* – компьютер, предоставляющий свои ресурсы другому компьютеру. *Клиент* – компьютер, запрашивающий ресурсы у другого компьютера.

Структура информационных вычислительных сетей

Структура информационных вычислительных сетей определяется принципом управления и типом связей ЭВМ (*топологией*). Топологии компьютерных сетей отличаются большим разнообразием:

шинные (линейные);

радиальные (звездообразные);

кольцевые (петлевые);

распределенные радиальные (сотовые);

иерархические (древовидные);

сетевые (полносвязные);

смешанные (гибридные).

Примеры некоторых характерных топологий сетей приведены на рисунке 7.2. Простейшей схемой объединения компьютеров в сеть является линейная структура. В этом случае все компьютеры подключены к общей шине и могут связываться непосредственно друг с другом. Они могут быть построены как с выделенным сервером, так и без выделенного сервера (децентрализованные). В первом случае одна станция является ведущей, остальные подчиненные. Во втором случае все станции равноправны.

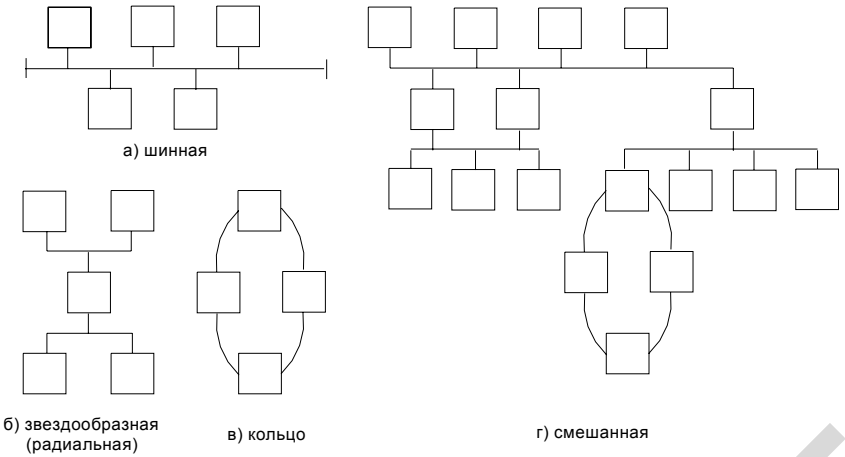


Рис. 7.2. Топология сетей

При радиальной топологии в компьютерной сети имеется один сервер, который соединен с каждой рабочей станцией по выделенным каналам связи. Такая сеть малонадежна. В случае выхода из строя канала связи связь с рабочей станцией прерывается.

При кольцевой схеме все компьютеры объединены в кольцо. Такая сеть может работать как с выделенным сервером, так и без выделенного сервера. Достоинством кольцевой топологии является повышенная надежность. При обрыве канала связи имеется возможность передать информацию в другом направлении.

При сетевой топологии все компьютеры соединены каналами связи по схеме каждый с каждым. Этим обеспечивается высокая

устойчивость сети к возникновению неисправностей. Однако в этом случае значительно усложняются протоколы обмена информацией и затраты на организацию сети.

Протоколы передачи информации

Передача информации в сетях осуществляется в соответствии с протоколами.

Протокол – это соглашение о правилах взаимодействия компьютеров в сети. Они устанавливают также требования к параметрам сети и ее компонентам.

В целях унификации требований к протоколам передачи информации в начале 80-х годов была разработана Модель взаимодействия открытых сетей (OSI). Эта модель предусматривает семь уровней протоколов: прикладной, представительский, сеансовый, транспортный, сетевой, канальный и физический (рис. 7.3).

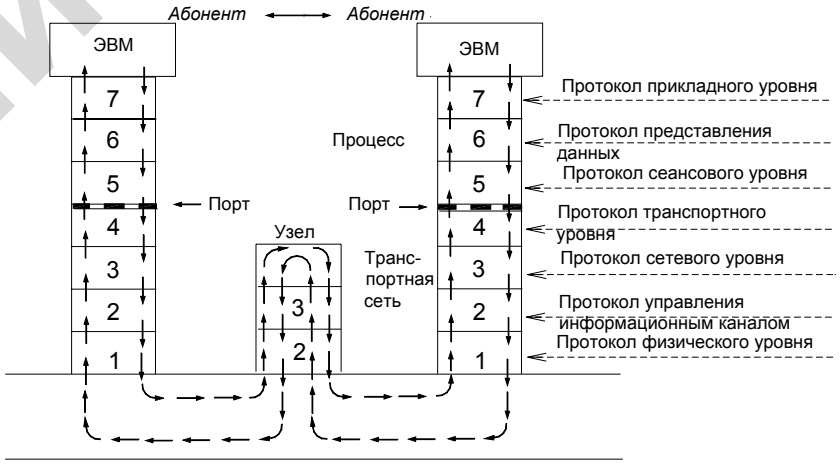


Рис. 7.3. Семиуровневая структура протоколов передачи информации

Протокол *прикладного* уровня обеспечивает доступ пользователей к распределенным ресурсам сети (файлы, принтеры, факсы, сканер, гипертекстовые страницы). На этом уровне обеспечивается предоставление пользователю различных услуг, связанных с запуском его программ, передачей данных и т. п. Например, протоколы работы электронной почты. Единицей информации на этом уровне является сообщение.

На *представительском* уровне осуществляется преобразование формы представления данных, трансляция и интерпретация программ с разных языков, шифрование данных, преобразование их в форму, необходимую для восприятия информации прикладным уровнем другой системы. На практике многие функции этого уровня реализованы на прикладном уровне.

Протокол *сеансового* уровня решает вопросы обеспечения синхронизации передачи данных. Кроме того, в длинных передачах устанавливаются специальные контрольные точки для возможного отката в случае сбоев не в начало блока передачи данных, а в контрольную точку.

Протокол *транспортного* уровня управляет сегментированием данных и сквозной передачей данных от источника информации к потребителю. Поэтому одной из важнейших задач протоколов данного уровня является обеспечение надежности передачи данных, обнаружение и исправление ошибок передачи. Модель OSI определяет три класса сервиса, которые определяются качеством предоставляемых услуг и надежностью.

Сетевой протокол решает вопросы объединения сетей с разными топологиями, разными принципами передачи данных. Решает вопросы обмена данными между сетями. На этом уровне осуществляется структуризация данных – разбивка их на пакеты и присвоение им сетевых адресов. Обмен данными осуществляется пакетами.

Протоколы *канального* уровня решают две задачи: определение доступности среды передачи данных; определение механизма обнаружения и коррекции ошибок. Обмен данными осуществляется *кадрами*.

Кадр передачи данных канального уровня включает *Заголовок*, *Тело* и *Замыкатель*. Заголовок содержит служебную информацию и сведения о маршруте передачи данных. Тело кадра содержит передаваемую информацию (пакет данных). Замыкатель содержит контрольную сумму и сведения о методе вычисления контрольной суммы. Протоколы канального уровня реализуются сетевыми адаптерами и их драйверами.

Протоколы *физического* уровня обеспечивают пересылку информации. Они устанавливают характеристики электрических сигналов, физического канала связи, типа разъемов с назначением каждого контакта. Однако не во всех сетях используется полный набор протоколов. Состав протоколов некоторых сетей приведен в таблице 7.2.

Таблица 7.2

Примеры использования протоколов в сетях ЭВМ

Название сети	Уровни протоколов						
	Прикладной	Представления	Сеансовый	Транспортный	Сетевой	Канальный	Физический
Сеть МККТТ	CMIP, FTAM	X.226	X.225	X.224	X.25	LAB-B	Не нормируется
Сеть Интернет	HTTP, FTP, SMTP, DNS, TFTP, Telnet	–	–	TCP, UDP	IP	IEEE 802	Не нормируется
Локальная сеть	NCP, IW, SAP	–	NETBEUI	SPX, PEP	IPX	IEEE 802	X.21

Сети и сетевые технологии нижних уровней

В настоящее время применяется несколько технологий, используемых также и для обозначения сетей. Каждая из них имеет особенности, определяющие область их применения.

Сеть и технология ISDN. Цифровая сеть с интеграцией услуг, использует цифровые каналы связи в режиме коммутации каналов. Адресация в сети строится по телефонному принципу. Номер ISDN состоит из 15 десятичных цифр и включает в себя код страны, код сети и код местной подсети. Код страны такой же, как и в обычной телефонной сети. По коду сети выполняется переход в заданную сеть ISDN. Внутри подсети для адресации используется 35 десятичных знаков, что позволяет идентифицировать любое устройство. Основным достоинством сети является то, что она позволяет осуществлять различные виды связи: передачу аудио-, видеоданных, текста, графики и т. д. Скорость передачи данных, реализуемых сетью: 64 кбит/с, 128 кбит/с, а также до 2 Мбит/с и до 155 Мбит/с в мощных и широкополосных каналах связи.

Сеть и технология X.25. Является классической полнопротокольной сетью, разработанной международной организацией по стандартизации (ISO). Эта сеть является базой информационного обмена в региональных и иных корпоративных структурах. Она ориентирована на использование малых и больших компьютеров,

распределенных на большой территории. Информационный обмен в сети X.25 во многом похож на аналогичный процесс в сетях ISDN. На физическом уровне используется протокол X.21, а на канальном – протокол LAP-B. Он обеспечивает передачу данных в виде кадров переменной длины. Начало и конец кадра помечаются специальной последовательностью битов, называемой флагом. Данный протокол описывает взаимодействие соседних узлов как процедуру с установлением и подтверждением соединения. Скорость передачи данных – от 56 кбит/с до 64 кбит/с. Данная технология предназначена для работы в сетях с большим уровнем помех, обеспечивает гарантированную доставку информации.

Сеть и технология Frame Realy. Данная технология ориентирована на использование в сетях с коммутацией пакетов. То есть она предполагает использование выделенных каналов связи. Сама технология охватывает только физический и канальный уровни OSI, предназначена для работы в сетях с высокой помехозащищенностью. Скорость передачи находится в диапазоне от 56 кбит/с до 44 Мбит/с, но без гарантии достоверности доставки. Применяется в корпоративных и региональных сетях.

Сеть и технология ATM. Сеть ATM – одна из перспективных технологий построения высокоскоростных каналов связи. Обеспечивает асинхронный режим передачи данных, может использоваться в сетях любого класса. В качестве транспортного механизма ATM лежит технология широкополосной ISDN. Передача информации в сетях ATM происходит после предварительного установления соединения, выполненного высокоскоростными коммутаторами ATM. Коммутаторы создают широкополосный физический канал, в котором динамически можно формировать более узкополосные виртуальные подканалы. Передаются по каналу не кадры и не пакеты, а ячейки, представляющие собой очень короткие последовательности байтов – 53 байта, включая заголовок длиной 5 байт. Размер ячейки длиной в 48 байт выбран исходя из особенностей передачи голосового сигнала. Время заполнения ячейки в 48 байт составляет примерно 6 мс, что является пределом временной задержки, заметно не искажающей голосовой трафик. Технология ATM сочетает в себе технологии коммутации пакетов и коммутации каналов. Скорость передачи сигналов лежит в пределах от 155 Мбит/с до 2200 Мбит/с.

7.3. ЛОКАЛЬНАЯ ВЫЧИСЛИТЕЛЬНАЯ СЕТЬ

Локальные вычислительные сети предназначены для обработки данных и обмена информацией между компьютерами, размещенными на небольших расстояниях друг от друга (в лаборатории, цехе, на предприятии). Связь локальной вычислительной сети с Интернет осуществляется через *хост-компьютер*, в качестве которого может использоваться *веб-сервер* или *сервер-шлюз* (*прокси-сервер*), имеющий специальное программное обеспечение для непосредственной работы в Интернет.

Локальные вычислительные сети могут быть классифицированы по нескольким признакам, например: уровню управления, назначению, топологии, однородности, протяженности линий связи. Подробная классификация в настоящем пособии не рассматривается.

В зависимости от используемого оборудования и функциональных возможностей различают малые, средние и большие локальные вычислительные сети (ЛВС).

Малые ЛВС располагаются на небольшой территории. Протяженность канала связи – от 100 до 500 м. Средние ЛВС имеют протяженность до 1 км, а большие ЛВС – от 1 до 10 км. Кроме компьютеров, средств коммутации и канала связи ЛВС включает и программное обеспечение, обеспечивающее работу компьютера в сети.

Наибольшее распространение в локальных вычислительных сетях получили шинная (рис. 7.4) и радиальная топологии. Они отличаются простотой методов управления, возможностью расширения и изменения конфигурации сети без заметного усложнения средств управления сетью, высокой эффективностью использования каналов связи. При шинной или кольцевой топологии ЛВС возможно как централизованное, так и децентрализованное управление.

При децентрализованном управлении все ЭВМ в сети равноправны (одноранговые). При централизованном управлении одна ЭВМ является ведущей и называется *сервером* или *файловым сервером***,** другие машины вычислительной сети называются рабочими станциями.

Сервер – это персональная ЭВМ, имеющая некоторые распределяемые ресурсы, доступные пользователям сети.

Файловый сервер – это персональная ЭВМ, на которой установлена сетевая операционная система и которая содержит центральное устройство внешней памяти (обычно это жесткие диски). Файловый сервер может быть выделенным (работает исключительно

как файловый сервер) и не выделенным (работает одновременно как файловый сервер и как рабочая станция).

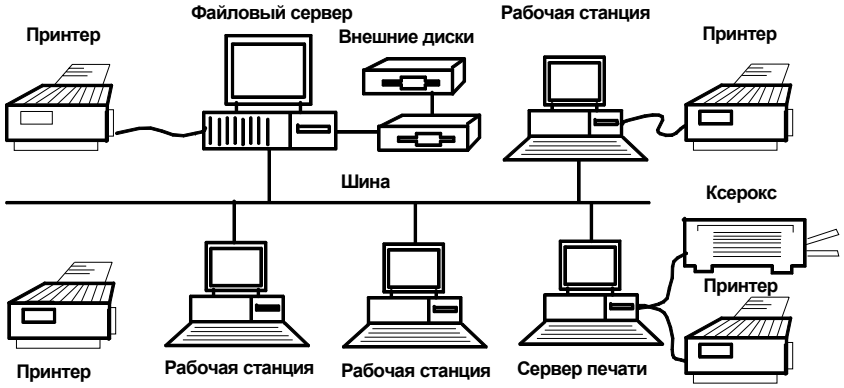


Рис. 7.4. Локальная вычислительная сеть с шинной топологией

Скорость передачи зависит также от используемых модемов.

Модемы и ***сетевые карты*** (производное от двух слов: модулятор и демодулятор) – это устройства, предназначенные для сопряжения компьютера с каналом связи. Модем обеспечивает прямое и обратное преобразование сигнала к виду, принятому для использования в определенном канале связи. Сетевые карты обычно являются встраиваемыми в компьютер. Модемы могут быть как внутренними, так и внешними.

При прохождении сигнала по каналу связи за счет влияния распределенной емкости и индуктивности происходит затухание сигнала, а также искажение его формы, поэтому длина канала связи ограничена. Чтобы обеспечить передачу сообщений на большие расстояния, устанавливаются специальные устройства расширения – повторители, коммутаторы (мосты), маршрутизаторы, шлюзы. Участок сети между узлами расширения принято называть *сегментом* сети.

Повторители служат для усиления или регенерации сигнала. Повторители не осуществляют развязки каналов связи. Описываются протоколами канального уровня.

Коммутаторы (или ***мосты***) служат для объединения сегментов сети, а также обеспечивают развязку присоединенных к ним сегментов, то есть поддерживают одновременно несколько процессов

обмена данными между каждой парой станций разных сегментов. Описываются протоколами сетевого уровня.

Маршрутизаторы обеспечивают переключение каналов в зависимости от адреса назначения сообщений. Они обеспечивают соединение сегментов сети независимо от того, работают они по одному или по разным протоколам. Выполняют свои функции на уровне транспортных протоколов модели открытых систем OSI.

Шлюзы служат для сопряжения каналов, использующих разные протоколы обмена данными. Они обеспечивают протокольное преобразование всех семи уровней управления моделями OSI. Кроме функций маршрутизаторов они выполняют также и преобразование формата информационных пакетов и их перекодирование в случае, если сети не однородные.

Мосты, маршрутизаторы и шлюзы в локальной вычислительной сети – это, как правило, выделенные компьютеры со специальным программным обеспечением и дополнительной связной аппаратурой.

Для организации связи в локальных сетях используются витая пара, коаксиальный кабель или оптоволоконные линии связи. Эти линии связи отличаются устойчивостью к помехам и достаточно высокой скоростью передачи информации. Наибольшей помехоустойчивостью и скоростью передачи обладают оптоволоконные линии связи, но они достаточно дороги.

В локальных вычислительных сетях основная роль отводится протоколам канального и физического уровней. Различают два подуровня: ***логической передачи данных*** (LLC) и ***управления доступом к сети*** (MAC). К основным протоколам локальной вычислительной сети относятся протоколы **IEEE 802.2** – протокол LLC; **IEEE 802.3**, **Eternet**, **IEEE 802.4** – **Token Bus**; **IEEE 802.5** – **Token Ring** и др. – протоколы MAC.

Подуровень логической передачи данных у всех протоколов нижнего уровня практически одинаков.

Примером протокола уровня логической передачи данных является стандарт **IEEE 802.3**/***Ethernet DIX*** на основе коаксиального кабеля:

10Base-5 – использует толстый коаксиальный кабель, диаметр 0,5 дюйма, сегмент до 500 м, максимальное число станций в сегменте – 100;

10Base-2 – использует тонкий коаксиальный кабель, диаметр 0,25 дюйма, сегмент до 185 м, максимальное число станций в сегменте – 30;

10Base-T – использует неэкранированную витую пару. Применяется при звездообразующей топологии, длина сегмента не более 100 м с максимальным числом рабочих станций в сегменте – 1024;

10Base-F-использует волоконно-оптический кабель, длина сегмента от 1000 до 2000 м. Пропускная способность – до 10 Мбит/с, максимальное число станций в сегменте – 1024.

Протоколы ***Gigabit Eternet***. Этот стандарт обеспечивает высокую скорость передачи данных и надежность. Для него используется топология «Общая шина». Протокол реализует метод множественного доступа. То есть сообщение, отправляемое одной рабочей станцией, принимается всеми остальными станциями, подключенными к общей шине. Но поскольку сообщение содержит адрес станции отправителя и адресата, то другие станции это сообщение игнорируют. Так как сообщение могут передать одновременно несколько станций, имеются средства контроля занятости сети. Перед началом передачи станция проверяет, занят канал или нет. Если канал свободен, то она устанавливает флажок о занятости канала и начинает передачу. По окончании передачи сообщения флажок снимается.

Стандарт ***Token Ring*** рассчитан на кольцевую топологию, поддерживает экранированную и не экранированную витую пару, оптоволоконный кабель. Максимальная длина кольца – 4000 м. Максимальное количество узлов – 260. Скорость передачи – 100, 155 Мбит/с. Для исключения конфликтов в сети, связанных с одновременным обращением нескольких станций к каналу, используется маркерный способ или иначе «эстафетная палочка», но при этом имеется возможность назначить разные приоритеты разным рабочим станциям. При маркерном методе маркер перемещается по кольцу, давая последовательно расположенным на нем компьютерам право на передачу. Если компьютер получает пустой маркер, он может заполнить его сообщение кадром любой длины, однако лишь в течение того промежутка времени, который отводит специальный таймер для нахождения маркера в одной точке сети. Кадр перемещается по сети, и каждая ЭВМ регенерирует его, но только принимающая его ЭВМ копирует этот кадр в свою память и отмечает его как принятый, однако не выводит сам кадр из кольца. Эту функцию выполняет передающая станция, когда ее сообщение возвращается к ней обратно. Тем самым обеспечивается подтверждение факта передачи сообщения.

Технология ***ARCNet*** получила распространение в силу простоты и дешевизны оборудования. Она используется в ЛВС со

звездообразной топологией и поддерживает различные линии связи. Здесь также используется маркерный способ для исключения конфликтов при обращении к занятому каналу. Одна из станций сети создает специальный маркер, который последовательно передается от одной станции к другой. Если станция передает сообщение другой станции, то она должна дождаться маркера, поместить в него сообщение и дополнить адресами отправителя и назначения. Когда пакет дойдет до станции назначения, сообщение будет отделено от маркера и передано станции. Технология ARCNet обеспечивает скорость передачи данных 4 Мбит/с, а в конфигурации ARCNet Plus – 20 Мбит/с.

Технология FDDI поддерживает волоконно-оптический интерфейс, во многом базируется на технологии Token Ring. Обеспечивает передачу данных по кольцу длиной до 100 км с максимальным числом узлов 500 и скоростью 100 Мбит/с. Для доступа в сеть используется детерминированный маркерный метод доступа без выделенных приоритетов. Для повышения надежности организуется два ориентированных навстречу друг другу кольца. В случае отказа одного кольца передача данных ведется по объединенному первому и второму кольцам с исключением сбойного сегмента кольца.

Один из самых простых способов объединения ПК в вычислительную сеть – соединить компьютеры через последовательные порты. В этом случае имеется возможность копировать файлы с жесткого диска одного компьютера на другой, если воспользоваться сервисной оболочкой Total Commander или другой аналогичной программой. Для получения прямого доступа к жесткому диску другого компьютера разработаны специальные сетевые платы (адаптеры) и программное обеспечение. В простых локальных сетях функции выполняются не на серверной основе, а по принципу соединения рабочих станций друг с другом. Каждый компьютер в такой сети может выполнять функции как сервера, так и рабочей станции.

Основным разработчиком сетевых программных продуктов для сервера ЛВС является фирма Novell. Фирма выпустила несколько вариантов сетевой операционной системы Net Ware. Они поддерживают пять верхних протоколов OSI и работают со многими сетевыми протоколами нижнего уровня.

В России известны также отечественные ЛВС «Руслан», «Эстафета», «КвантС», «Сибирь», «Курьер», «Орбита», «Диалог», «Невод», «Лаура», «Ива» и др.

В последние годы все большую популярность приобретают локальные вычислительные сети на базе операционной системы Windows. Операционная система Windows 7 имеет средства для создания «домашней» локальной сети.

Модем

Модем – это устройство сопряжения компьютера с телефонной линией, воспринимающее сигналы от компьютера и преобразующее их в форму, пригодную для передачи по телефонным линиям связи, и наоборот. Модемы бывают разные, но в первую очередь они делятся на аналоговые и цифровые.

Модуляторы могут осуществлять преобразование выходного сигнала по фазе, по амплитуде или по частоте в зависимости от входного – модулирующего сигнала. В современных модуляторах встречаются четыре вида модуляции: частотная, фазовая, импульсно-кодовая и квадратурная амплитудная.

При *частотной* модуляции изменяется частота в соответствии с текущими значениями модулирующего сигнала при неизменной его амплитуде. Например, 0 соответствует частоте 980 Гц, а единица – 180 Гц. Помехоустойчивость высокая.

При *фазовой* модуляции модулируемым параметром является фаза сигнала при неизменной частоте и амплитуде входного сигнала. Помехоустойчивость высокая.

При *импульсно-кодовой* модуляции при изменении модулирующего сигнала меняется количество импульсов. Помехоустойчивость высокая.

При *амплитудной* модуляции помехоустойчивость низкая. Поэтому для повышения помехоустойчивости применяют более сложную *квадратурную амплитудную* модуляцию, при которой в такт передаваемым данным изменяется одновременно и фаза, и амплитуда сигнала.

Свойства модемов определяются большим числом специфических характеристик, которые так или иначе отражаются в маркировке модема. Одна из основных характеристик – скорость передачи данных, которая измеряется в бодах (бит/с). Указанный параметр модема определяется его типом, например, для низкоскоростных каналов: V21 – скорость передачи 300 бит/с, V22 – скорость передачи 1200 бит/с, V32 – скорость передачи 9600 бит/с, V32 бис. – скорость передачи 14 400 бит/с, V34 – скорость передачи 28 800 бит/с. ADSL-модемы обеспечивают скорости 1,5–6 Мбит/с при приеме информации и 64–384 кбит/с при передаче. Модемы

VDSL при передаче на короткие расстояния до 300 м обеспечивают скорость 2,3 Мбит/с, а при приеме – 51,8 Мбит/с. Кабельные модемы для цифровых каналов связи обеспечивают скорость передачи данных до 100 Мбит/с.

Другой важной характеристикой модемов является возможность обнаружения ошибок. Для защиты передаваемых данных от ошибок применяют помехоустойчивое кодирование. В некоторых моделях модемов предусмотрена возможность тестирования состояния сети на помехоустойчивость перед началом передачи и, в зависимости от этого, изменения длины пакета передаваемых данных и скорости передачи. С целью уменьшения времени передачи иногда применяют сжатие данных. Это отражается в маркировке, например, v. 42 bis и MNP5.

Модемы могут поддерживать разные методы передачи информации: симплексный, дуплексный или полудуплексный.

Сетевые карты

Вместо модема в локальных сетях можно использовать сетевые адаптеры (сетевые карты). При использовании адаптеров большую часть работы по преобразованию сигналов выполняет процессор, поэтому сетевые карты проще и дешевле модемов. Основными характеристиками сетевых карт являются:

установленная микросхема контроллера (микрочип);

разрядность: выпускаются 8-, 16-, 32-, 64-разрядные сетевые карты;

скорость передачи – от 10 Мбит/с до 1 Гбит/с.

Тип подключаемого кабеля – коаксиальный толстый, коаксиальный тонкий, неэкранированная витая пара или волоконно-оптический кабель;

поддерживаемый протокол передачи данных.

Беспроводные компьютерные сети

К беспроводным компьютерным сетям относят линии связи, использующие для передачи сигналов радиоканалы и линии электропередач.

Основное отличие этих каналов связи от проводных заключается в используемых интерфейсах. Беспроводные интерфейсы применяются для передачи данных по радиоканалам в диапазоне радиочастот на расстояниях от нескольких десятков сантиметров до нескольких километров. Основные характеристики некоторых стандартов беспроводных интерфейсов приведены в таблице 7.3.

Таблица 7.3

Характеристики некоторых стандартов беспроводных интерфейсов

Интерфейс	Дата создания	Область применения	Дальность действия, км	Частотный диапазон, ГГц	Поддержка мобильности	Пропускная способность, Мбит/с
Bluetooth	–	LAN	0,1	2,5	Нет	0,1
WiUSB	–	LAN	0,01	3,1–10,6	Нет	12
WiFi (IEEE802.11g)	2000	LAN	0,3	2,4 и 5	Есть	108
WiMax (IEEE 802.16)	2001	WLAN	5	10–66	Нет	134
WiMax (IEEE 802.16a)	2003	WLAN	50	2–11	Нет	75
WiMax (IEEE 802.20)	2007	WLAN	>100	<3,5	Есть	6
WiBro	2005	WLAN	5	2,3	Есть	50

Интерфейс Bluetooth был очевидно первым по времени разработки, используется для создания локальных сетей и управления различными устройствами в небольших группах и в домашних условиях.

Интерфейс WiUSB был предложен как альтернатива проводному интерфейсу USB.

Интерфейс WiFi имеет несколько модификаций, в которых разработчики стремятся найти компромисс между дальностью действия и пропускной способностью. Применяется для доступа к групповым ресурсам и для создания локальных сетей. Обеспечивает доступ к сети Интернет.

Интерфейсы WiMax и WiBro обеспечивают беспроводной доступ к региональным и глобальным сетям, сетям сотовой телефонии и персональной спутниковой связи.

Интерфейс WiBro используется в сетях, а также обеспечивает доступ по каналам сотовой телефонии с технологиями 3G, GPRS – сервис пакетной передачи данных по радиоканалу.

Для работы в сети WLAN компьютер должен быть укомплектован своей мобильной **точкой доступа** – беспроводной сетевой

интерфейсной картой (NIC), а также необходима локальная сеть, имеющая в своем кабельном сегменте **точки доступа**. Точки доступа являются стационарными устройствами, выполняющими роль моста между беспроводными и кабельными сегментами сети. Точки доступа содержат приемо-передатчики, контроллер проводного сетевого интерфейса и необходимое программное обеспечение. Количество пользователей в сетях должно быть не больше 10–15 на одну точку доступа, так как при одновременном обращении пользователей к ресурсам сети полоса пропускания для каждого из пользователей уменьшается.

7.4. СРЕДСТВА ЭЛЕКТРОННОЙ СВЯЗИ

К традиционным средствам связи (почта, телефон, телеграф) на современном этапе добавилась факсимильная связь. Факсимильный аппарат включает в себя телефон, автоответчик и факс.

Телефон представляет собой обычный телефонный аппарат, снабженный некоторыми дополнительными функциями, например, автоматический набор номера, наличие радиотелефонной трубки, возможность ведения разговора по абонентской линии, наличие встроенного динамика для прослушивания разговора и др.

Автоответчик предоставляет возможность записать сообщение и оповестить абонента об отсутствии «хозяина».

Факс представляет собой сочетание нескольких устройств: принтера, сканера, модема.

Связь с факсимильным аппаратом может осуществляться или автоматически, или вручную.

Международным консультативным комитетом по телеграфии и телефонии CCITT установлены четыре стандарта для факсов: Group 1, Group 2, Group 3, Group 4. Эти стандарты отличаются скоростью передачи и, соответственно, качеством. Стандарт Group 1 был разработан в конце 60-х годов, скорость передачи – 300 бод, для передачи одной страницы текста затрачивалось до 6 мин. Стандарт Group 2 был разработан в середине 70-х годов. Скорость передачи – в два раза выше, чем у Group 1. Group 3 был введен в 1980 году. Скорость передачи – 9600 бод (время на передачу одной страницы сократилось до 15–20 секунд). Факс, при плохих линиях связи, может вести передачу с меньшей скоростью – 7200 бод или 4800 бод.

Стандарт Group 4 введен в 1988 году, он позволяет передавать до 30 страниц в минуту.

7.5. ГЛОБАЛЬНАЯ СЕТЬ ИНТЕРНЕТ

7.5.1. Общие сведения

Интернет – это глобальная сеть компьютерных сетей: научных, учебных, правительственных, военных, соединенных каналами связи и передающих информацию друг другу по определенным правилам, называемым протоколами. В качестве каналов связи могут использоваться обычные телефонные линии. В этом случае для подключения компьютера к сети используется устройство, называемое модемом. Предоставляют доступ к сети Интернет фирмы или организации, называемые провайдерами (от английского provide – обеспечивать). Провайдер – это организация или лицо, реализующее услуги Интернет в данном регионе.

Рождение Интернет относят к 60–70 годам. Первоначально Министерство Обороны США создало сеть, которая называлась ARPANet. Эта сеть создавалась в военно-промышленной сфере для поддержки научных исследований, в частности, для исследования методов построения сетей, устойчивых к частичным повреждениям, получаемым, например, при бомбардировке авиацией, и способных в таких условиях продолжать нормальное функционирование. В 80-х годах к Интернету подключились образовательные учреждения, государственные организации, американские и иностранные фирмы различного направления деятельности.

В России начало Интернету положило создание в 1990 году на базе института атомной энергии им. Курчатова компьютерной сети Relcom. Уже к концу 1990 года к ней подключилось более 30 локальных сетей разных организаций, что позволило осуществить ее официальную регистрацию и подключение к мировой сети.

У сети Интернет нет высших руководящих органов. Однако направление ее развития определяет «Общество Интернета» (ISOC) – организация, действующая на общественных началах, целью которой является содействие глобальному обмену информацией через Интернет. Оно назначает совет старейшин, который отвечает за

техническое руководство и ориентацию Интернета. Совет старейшин формируется из приглашенных лиц, которые добровольно изъявили желание принять участие в его работе. Совет старейшин собирается регулярно, чтобы утверждать *стандарты* и распределять *ресурсы*. Интернет работает благодаря наличию стандартных способов взаимодействия компьютеров и прикладных программ друг с другом.

При работе в Интернет должны соблюдаться правовые нормы. Прежде всего, речь идет о правовых нормах, касающихся интеллектуальной собственности и лицензий, а также при экспорте информации за границу следует руководствоваться законами государства, регулирующими эти нормы.

Основу Интернета составляют высокоскоростные телекоммуникационные магистральные сети. К магистральной сети через точки сетевого доступа NAP присоединяются автономные системы, каждая из которых уже имеет свое административное управление, свои внутренние протоколы маршрутизации.

Основными ячейками Интернета являются локальные вычислительные сети, а также локальные компьютеры, самостоятельно подключенные к Интернету. Компьютеры сетевые или локальные, непосредственно подключенные к Интернету, называются ***хост-компьютерами***.

7.5.2. Принцип функционирования Интернет

Организация обмена информацией в сети Интернет осуществляется в соответствии с протоколами. Все протоколы сети Интернет разбиты на четыре уровня (табл. 7.2):

1-й уровень – прикладной, служит для организации взаимодействия с пользователем. К нему относятся протоколы *HTTP* – доступ к удаленным гипертекстовым базам данных во всемирной паутине, *Gopher* – доступ к ресурсам всемирного пространства, *WAIS* – распределенная информационно-поисковая система, *FTP* – копирование файлов, *Telnet* – эмуляция терминала и другие;

2-й уровень – протокол управления передачей *TCP*;

3-й уровень – протокол межсетевого взаимодействия *IP*;

4-й уровень – уровень доступа к сети не регламентируется. Здесь могут использоваться любые протоколы в зависимости от конфигурации локальной сети (табл. 7.4).

Таблица 7.4

Протоколы обмена информиацией Интернет

Пользователь	Прикладной уровень					
I	HTTP, Gopher, WAIS	SNMP	FTP	Telnet	SMTP	TFTP
II	Протокол управления передачей (TCP), выполняет функции транспортного уровня модели OSI					
III	Протокол межсетевого взаимодействия (IP), выполняет функции сетевого уровня модели OSI					
IV	Уровень доступа к сети Ethernet, Token Ring, FDDI, X.25, SLIP, PPP					

Протокол IP организует разбиение сообщений на электронные пакеты (IP-дейтаграммы), маршрутизирует отправляемые пакеты и обрабатывает получаемые пакеты. Дейтаграммы – общее название единиц информации (пакетов, кадров, ячеек, сегментов), с которыми оперируют протоколы в сетях без установления предварительного соединения.

Протокол TCP является типичным протоколом транспортного уровня. Он управляет потоком данных, обрабатывает ошибки и гарантирует, что все информационные пакеты получены и собраны в нужном порядке. Пиковая скорость передачи данных, поддерживаемая протоколом, составляет 620 Мбит/с. В 2003 году принят новый протокол Fast TCP, который поддерживает пиковую скорость 8,5 Гбит/с.

Протоколы IP и TCP тесно связаны между собой, поэтому их приводят обычно одним названием TCP/IP. На их базе разработаны многие прикладные сервисные протоколы, приведенные в таблице 7.4:

FTP – протокол передачи файлов;

Telnet – протокол удаленного доступа, то есть дистанционного исполнения команд на удаленном компьютере;

SMTP – протокол пересылки электронной почты;

HTTP – протокол передачи гипертекста (hyper text transfer protocol – гипертекстовый транспортный протокол);

NNTP – протокол передачи новостей, телеконференций;

WAIS, Gofer – распределенные информационно-поисковые системы.

Эти протоколы формируют в сети соответствующие им прикладные процессы, а задача протокола TCP – обеспечить передачу

данных между этими процессами. Одновременно в сети может выполняться множество процессов, и чтобы протокол TCP мог их опознать, они идентифицируются номерами, носящими названия номеров портов. За некоторыми процессами жестко закреплены номера портов. Например, порт 21 – процесс передачи файлов FTP.

Типы соединения с Интернет

Подключение к Интернету производится посредством сетевого адаптера или другого сетевого устройства, например, модема или платы ISDN (цифровая сеть с интеграцией сервиса). Скорость передачи информации в Интернете выражается в битах в секунду (бод) (табл. 7.5).

Таблица 7.5

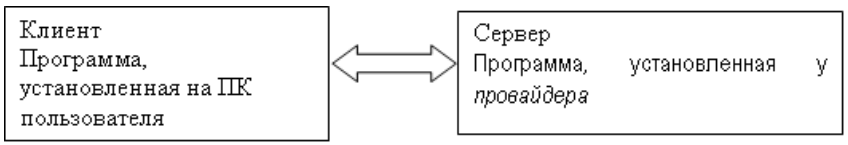
Типы соединений, используемые в сетях Интернет

Тип соединения	Максимальная скорость передачи	Приблизительное число пользователей
Выделенный PPP/SLIP	Скорость модема	2–3
56 К (Frame Relay)	56 000 бод	10–20
ISDN	128 000 бод	10–50
T1	1 540 000 бод	100–500
Дробная T1	В зависимости от необходимости	В зависимости от необходимости
T3	45 000 000	5000 и выше

Модемные соединения имеют различную скорость передачи: от 9600 бод до 50 000 бод (см. раздел 7.2).

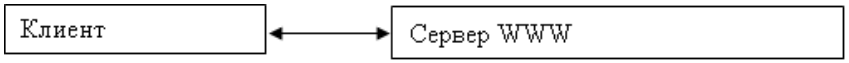
Модель клиент-сервер как основа построения информационных сервисов Интернет

В основу взаимодействия компонентов информационных сервисов сети Интернет в большинстве случаев положена модель клиент-сервер. В качестве клиента выступает программа, которая установлена на компьютере пользователя, а в качестве сервера – программа, установленная у провайдера.

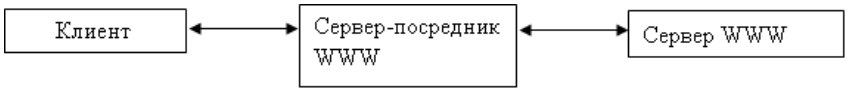


Существует несколько моделей доступа пользователя к ресурсам сети:

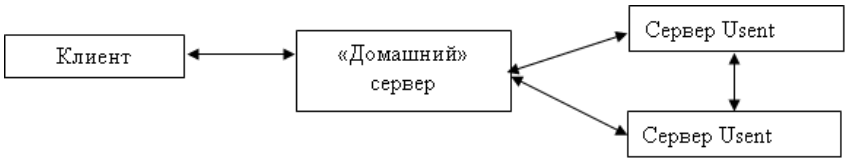
1) каждый с каждым – каждый пользователь может напрямую, без посредников, взаимодействовать с каждым сервером. Недостаток этой схемы – большой трафик в сети;



2) использование службы доменных имен. Можно настроить сервер для работы через другой сервер или непосредственно с программами-клиентами;



3) пользователь взаимодействует только со своим сервером и не может обратиться непосредственно к произвольному серверу – схема Usent.



Принципиальная разница между схемой с посредником и схемой Usent состоит в том, что при наличии посредника работа по доступу к ресурсу перекладывается на его плечи. Посредник будет устанавливать соединение с каждым сервером сети. По схеме Usent этого делать не обязательно, так как информацию в принципе можно получить с любого сервера.

Система адресации в Интернет

IP-адрес

Как уже упоминалось, для обмена данными между абонентами используется набор правил, который получил название «протокол». Набор протоколов, используемых в Интернет, существует под общим названием Протоколы управления передачей/Протоколы Интернет (Transmission Control Protocols/Internet Protocols – TCP/IP). Отсюда происходит понятие IP-адрес.

Каждый компьютер в Интернете имеет свой номер, так называемый уникальный IP-адрес (Internet Protocol – IP). Это адрес, которым руководствуются компьютеры, общаясь между собой. Существует две разновидности адресов: ***адреса компьютеров*** (узлов Интернет); ***адреса информационных ресурсов***, расположенных на

этих компьютерах. Адреса компьютеров назначаются организацией InterNIC, которая передает их провайдерам. Провайдеры в свою очередь распределяют адреса среди компьютеров своего участка сети. Адрес содержит полную информацию, необходимую для идентификации компьютера.

IP-адрес версии v.4 содержит четыре группы чисел по 8 бит, которые можно записать в десятичном виде (каждое в пределах от 0 до 255), разделенных точками. Структура адреса зависит от класса сети. Зарезервировано 5 классов сетей:

класс A: 1-й байт – адреса подсетей, номера от 1 до 126, три байта – номера узлов. Возможное число узлов в подсети – 16 777 214. Адрес класса A используется в крупных сетях общего пользования и позволяет создавать сети с большим количеством узлов;

класс B: два байта – номера подсетей со 128 до 191, два байта – номера узлов. Возможное число узлов в подсети – 65 534. Адреса класса B используются в корпоративных сетях средних размеров;

класс C: три байта – номера подсетей с 192 до 223, один байт – номер узла. Возможное число узлов в подсети – 254. Адреса класса C используются в локальных сетях небольших предприятий;

класс D: группы компьютеров номера с 224 до 239, число узлов – 2–28. Класс D предназначен для обращения к группам машин;

класс E: номера с 240 до 247, число узлов – 2–27. В настоящее время не используются.

Любой подключенный к Интернету компьютер, независимо от его назначения, называется ***хост***.

Пример полного IP-адреса компьютера:

120.158.212.42

Здесь 42 – номер компьютера в сети. Например, 212.193.2.201 – адрес сервера Санкт-Петербургского отделения института «Открытое общество».

Всего в Интернете может быть использовано 4 294 967 296 IP-адресов.

Учитывая ограниченность 32-х битового IP-адреса, в настоящее время ведется разработка модернизированного протокола IP-адресации. Основой этого протокола являются 128-разрядные адреса. Кроме увеличения количества адресов этот протокол решает и такие задачи, как повышение пропускной способности сети, обеспечение защиты информации и др. IP-адрес будет содержать уже не четыре, а пять групп адресов: две группы для провайдеров сети и три для абонента.

Доменный адрес

Для удобства IP-адресам поставлены в соответствие *символьные* или *доменные* адреса – ***DNS*** (Domain Name System). Домен – это множество компьютеров, имеющих общую часть имени. Составные части доменного адреса называются сегментами: <сегмент> . <сегмент> . … . <сегмент>. Сегменты образуют иерархическую систему. Самый правый сегмент является доменом верхнего уровня. Различают домены двух видов: географические и тематические.

Географические доменные имена верхнего уровня – двухбуквенные. Они определяют принадлежность владельца имени к сети конкретной страны, например: by – Беларусь, ru – Россия, fr – Франция, us – Соединенные Штаты Америки, de – Германия, uk – Великобритания, es – Испания.

Тематические адреса дают возможность представить сферы деятельности их владельцев: com – коммерческие фирмы (например, http://www.microsoft.com/), edu – образовательные учреждения, gov – государственные, mil – военные, net – сетевые организации, org – прочие организации.

В доменном имени крайняя правая часть обозначает домен верхнего уровня, крайняя левая – имя собственно компьютера, остальные, справа налево, – набор вложенных друг в друга поддоменов, каждый следующий домен является частью предыдущего домена. Например, sky.inp.nsk.ru: ru – Россия, nsk – Новосибирск, inp – Институт ядерной физики (Institute for Nuclear Physics), sky – имя компьютера.

URL-адрес

Информационным ресурсам Интернет ставят в соответствие ***URL-адреса*** (от англ. Uniform Resource Locator – универсальный адрес ресурса). URL-адрес состоит из двух частей, разделяемых двоеточием и двумя символами обратный слэш (://). Первая (левая) часть указывает на то, к какому протоколу принадлежит ресурс и как получить к нему доступ, т. е. определяет конкретный сетевой протокол. Вторая часть URL-адреса сообщает, где расположен искомый ресурс, то есть его доменное имя. Таким образом, URL содержит информацию о том, где данный ресурс расположен и как к нему следует обращаться. Пример URL-адреса: http://www.icape.nw.ru/.spb.osi.ru. Указывать префикс http:// не обязательно. Браузер сам добавит эти символы, подразумевая, что пользователя интересуют, скорее всего, гипертекстовые ресурсы и, стало быть, к ним нужно обращаться по протоколу

HTTP, оптимизированному для работы с гипертекстовыми ресурсами Интернет. Персональный компьютер, рабочая станция, сервер или иное устройство, подключенное к сети и обеспечивающее пользователю связь с другими компьютерами, подключенными к Интернету, имеет IP-адрес, доменное имя и, следовательно, может быть найдено средствами поиска в Интернет.

Широкое использование компьютерных сетей пользователями-неспециалистами стало возможным благодаря разработке специальных программ-клиентов, называемых браузерами. Браузеры обладают «дружественным» интерфейсом. Они не требуют от пользователя больших знаний, поэтому легко осваиваются и детьми.

Схемы адресации ресурсов в Интернет

За основу при написании доменных адресов принята нотация *Unix*. В настоящее время рассмотрено 8 схем адресации в Интернет и синтаксис двух схем адресации находится в стадии обсуждения.

Схема http:

http://www.citmgu.ru/user/data/letters.html # Marka1

Здесь:

http:// – протокол на основе языка гипертекстовой разметки документов;

www.citmgu.ru/ – номер сервера;

user/data/letters.html – спецификация файла;

 # Marka1 – метка внутри файла.

http://www.citmgu.ru/cgi-bin/proc?corn

В данном примере:

cgi-bin/proc – спецификация файла;

?corn – контекст для поиска.

Наличие в адресе www свидетельствует о том, что на данном компьютере установлен веб-сервер.

Схема FTP

Эта схема позволяет адресовать файловые архивы из программ-клиентов WWW. При этом программа должна поддерживать протокол FTP. В этой схеме можно указывать идентификатор пользователя и даже его пароль:

ftp://polyn.net.kial.su/pub/oindex.txt

ftp://nobody:password@polyn.net.kial.su/user/local/pub

Здесь *nobody:password* – идентификатор и пароль пользователя.

Схема Gopher

Схема Gopher используется для ссылки на ресурсы информационной системы Gopher:

gopher:/ gopher.kail.su:70:/7/kuku

Здесь *su:70:/7/* – порт для поиска.

Схема MAIL TO предназначена для отправки почты по стандарту RFC-822:

mailto:paul@quest.polyn.kial

Схема TELNET – по этой схеме осуществляется доступ к ресурсам удаленного терминала: telnet://quest:password@apollo.polyn.kial.su.

Схема FILE – схема для локального режима:

file:///CI/text/html/inaex.htm – обращение к локальному документу на ПК.

Основные ресурсы Интернет

Интернет обладает большим набором сервисных функций, значительно увеличивающих к нему интерес различных категорий пользователей, среди которых следует выделить следующие:

различные службы;

электронная доска объявлений;

электронная конференция;

чат;

форум;

e-mail – электронная почта и т. д.

Службы Интернета

FTP – система файловых архивов – это огромное распределенное хранилище всевозможной информации, накопленной за годы работы Интернет. Протокол передачи файлов позволяет пересылать файлы с одного компьютера на другой. С помощью него можно осуществлять процесс обмена данными – текстовыми и программными файлами.

Gopher – распределенная информационная система Интернет. Широко применяется как поисковая машина в других информационных серверах.

WAIS – распределенная информационно-поисковая система ИНТЕРНЕТ. Широко применяется как поисковая машина в других информационных серверах.

LISTSERV – система почтовых списков сети BIT-NET (сеть образовательных учреждений). Это один из популярных ресурсов в глобальных компьютерных сетях, и в ИНТЕРНЕТ существуют шлюзы к нему. Данная программа специально предназначена для работы в качестве электронной почты.

WHOIS – служба, которая содержит сведения о пользователях сети, их электронные и обычные номера, идентификаторы и реальные имена. WHOIS – распределенная система.

TRICKLE – обеспечивает доступ по почте к архивам FTP, который организован через специальный шлюз. Этот шлюз имеет специальные навигационные средства для поиска нужной информации в сети, пользователь может вести с ним своеобразный диалог по почте, выбирая нужную информацию путем ввода специальных команд.

Telnet – программа работы с удаленным компьютером. Программа разработана под UNIX, но поддерживается всеми версиями Windows, позволяет устанавливать связь с удаленным компьютером и использовать его в интерактивном режиме. Telnet и более современная программа Remote Access эффективно работают при организации «домашних офисов», то есть для надомной работы специалистов.

WWW – распределенная гипертекстовая информационная система. WWW – всемирная паутина – самая популярная информационно-поисковая система. Информация в WWW представляется в виде текста, звука, трехмерных изображений, возможна передача мультимедийных документов. Документы, передаваемые в сети Интернет, называются ***веб-страницами***. Страницы хранятся на компьютерах, называемых ***веб-серверами***, которые имеют специальный комплекс программ, и являются частью Интернет.

Работа в WWW осуществляется по принципу клиент-сервер. То есть по запросу клиента осуществляется связь с одним из серверов, который возвращает ему затребованную веб-страницу. На веб-странице могут потребоваться разъяснения, которые находятся на другой веб-странице. Такие пункты или элементы веб-страницы выделены цветом или подчеркиванием и называются *гиперссылками*. Чтобы вызвать описание этого элемента, надо щелкнуть по нему мышью.

Проект WWW возник в 1989 г. в Европейской лаборатории физики элементарных частиц. Основное назначение проекта – предоставить пользователям-непрофессионалам Online (непосредственный) доступ к информационным ресурсам. Результатом проекта WWW является предоставление пользователям сетевых компьютеров достаточно простого доступа к самой разнообразной информации. Используя популярный программный интерфейс, проект WWW изменил процесс просмотра и создания информации. Идея заключалась в том, что по всему миру хаотично разбросаны тысячи компьютеров и любой компьютер, подключенный к Интернету в режиме Online, можно преобразовать в сервер и начинить его ин-

формацией. С любого другого компьютера, подключенного к Интернету, можно свободно установить соединение с таким сервером и получить от него информацию.

Информационный веб-сервер использует гипертекстовую технологию. Для записи документа используется специальный, но очень простой язык HTML – язык гипертекстовой разметки документа, который позволяет структурировать документы, управлять шрифтами, отступами, вставлять цветные иллюстрации, поддерживает вывод звука и анимации. В стандарт языка входит также поддержка математических формул.

Внешне гипертекст отличается от обычного текста тем, что в нем часть слов или целые строки, будучи выделенными особым шрифтом или цветом, оказываются чувствительными к появлению на них указателя мыши. Указатель мыши при этом меняет свою форму. Щелчок мыши по такому полю приводит к инициализации некоторого нового события: загрузка и просмотр нового документа, переход на другую страницу и т. д. В результате появляется возможность самому выбирать последовательность просмотра тех или иных страниц, двигаясь по перемежающимся между собой нитям – паутина ссылок.

Электронная доска объявлений (англ. www-board, синонимы: веб-доска, веб-борд) – это, с одной стороны, прикладная программа, установленная на головной машине, оборудованной модемом. На диске этой машины выделяется область, куда каждый пользователь может записать свою информацию или считать имеющуюся информацию. Связь между узлами сети осуществляется по двухточечной схеме с помощью специальной программы – «почтальона». С другой стороны, электронная доска объявлений – это сайт, где можно поместить свое объявление, она аналогична доске объявлений или газете бесплатных объявлений в обычной жизни. Как правило, такие доски являются бесплатными, тематически организованными. На такой доске можно легко поместить свое объявление, указав его тематику и срок хранения, и оно появится на доске практически сразу после отправки.

Телеконференции USENT – это та же электронная доска объявлений, но пользователи разбиты на группы по интересам. Чтобы обмениваться с коллегами по интересам, достаточно подписаться на соответствующую конференцию. После этого на Ваш компьютер будет поступать нужная Вам информация.

Службы прямого общения

Веб-Chat (Чат) – одновременный разговор нескольких абонентов через сеть. Обычно это диалог в текстовом режиме, который возможен как при использовании браузера, так и через специального клиента. Доступ по адресу http://www.chat.ru. Требуется выбрать раздел для общения и зарегистрировать в этом разделе имя, под которым Вы будете фигурировать в чате.

Служба IRC предназначена также для прямого общения пользователей в режиме реального времени. Эту службу называют также чат-конференцией. В отличие от телеконференций, открытых всему миру, в чате общение происходит только между пользователями одного канала. Все IRC-каналы имеют имена идентификаторы, начинающиеся с символа #. Имена русских чатов – #russia и #russian.

Служба ISQ предназначена для обмена короткими текстовыми сообщениями между пользователями, одновременно находящимися на связи. На время связи компьютерам присваиваются временные адреса. Если пользователь зарегистрировался на ISQ-сервере, то ему присваивается персональный идентификационный номер (UIN), который пользователь должен сообщить своим партнерам по контактам.

Форум – это инструмент для общения на сайте. Сообщения в форуме в чем-то похожи на почтовые сообщения: каждое из них имеет автора, тему и собственно содержание. Но для того, чтобы отправить сообщение в форум, не нужна никакая дополнительная программа – нужно просто заполнить соответствующую форму на сайте.

Электронная почта – это способ пересылки сообщений в Интернете. Для передачи сообщения надо знать электронный адрес человека или организации, на который поступает корреспонденция. Каждый клиент получает у своего провайдера адрес почтового ящика. Сообщение посылается по сети и содержит адрес получателя, адрес отправителя и собственно сообщение. При этом пользователю, по его желанию, могут предоставляться дополнительные услуги: уведомление о вручении; уведомление о недоставке.

Работа электронной почты осуществляется в режиме Offline.

Электронная коммерция (англ. e-commerce, синоним – е-коммерция). В самом широком смысле – коммерческая деятельность с использованием Интернета. Как правило, имеются в виду механизмы, позволяющие упростить работу и продавцов и покупателей.

Подключение компьютера для работы в Интернете

Услуги, связанные с доступом к Интернету, предоставляются фирмами, называемыми провайдерами. Провайдер – это организация или частное лицо, которое ведет (поддерживает) информационные ресурсы. Он предоставляет клиентам доступ к сети по телефонным линиям. Кроме того, провайдер предоставляет такие услуги, как аренда пространства на сервере и создание *веб-страниц*. При выборе провайдера основными критериями могут служить местоположение, цена, надежность и набор предоставляемых услуг.

Подключение к Интернету может быть осуществлено несколькими способами:

постоянное соединение по выделенной линии;

сеансовое соединение по коммутируемой линии;

дистанционный терминальный доступ к хост-компьютеру;

сеансовый доступ по спутниковым каналам связи.

Чаще всего используется подключение по выделенной линии.

В качестве выделенной линии могут использоваться обычные телефонные линии связи тональной частоты, обеспечивающие скорость передачи 48–56 кбит/с; цифровые выделенные линии по двухпроводной линии связи, скорость передачи данных по которым составляет от 64 кбит/с до 2 Мбит/с. При использовании выделенной волоконно-оптической линии связи скорость передачи данных может составлять до 622 Мбит/с.

Стратегия поиска информации в сети

Просмотр информации в Интернет осуществляется с помощью специальных программ – ***браузеров***. Основная задача браузера – сделать общение с Интернет простым и удобным. За время существования Интернета в сети накоплено огромное количество информации. Так, по данным фирмы Alexa Internet объем веб-страниц в 1998 году составлял около 12 Тбайт и ежедневно увеличивался на полтора миллиона веб-страниц. Поэтому найти нужную информацию в современном Интернете не так уж просто. Именно эту задачу и помогают решить браузеры или специальные поисковые машины. Наиболее известным браузером, применяемым в операционных системах Windows, является Internet Explorer. Однако наряду с ним имеется еще большое число браузеров, например, Netscape Navigator, Mozilla, Opera, Safari, известна также разработка московской фирмы Advanced Multimedia System Design под названием Ariadna. Браузеры не только позволяют просматривать веб-страницы,

но и предоставляют пользователю различные справки, тематические подборки, а также средства поиска по запросам.

Вследствие большого объема информации, найти необходимые данные можно только с использованием специальных поисковых систем. Чтобы получить доступ к системам поиска, необходимо нажать кнопку Поиск на панели инструментов приложения Internet Explorer. При появлении панели поиска ввести ключевое слово или фразу в поле поиска и нажать кнопку Поиск.

Поиск нужной информации можно проводить с помощью веб-страницы поисковой системы. Сегодня в Интернет насчитывается более 200 поисковых систем, которые признаны популярными; общее же их число превышает 1000. Самыми распространенными русскоязычными поисковыми системами являются следующие: all.by, yandex.ru (Яndex.ru), rambler.ru, aport.ru, google.ru, google.by, yahoo.com, excite.com, altavista.com, lycos.com. Самым новым, быстрым и стабильным браузером ряд авторов считают Google Chrome.

Поисковые машины представляют собой мощные информационно-поисковые системы, размещенные на серверах свободного доступа, специальные программы которых непрерывно в автоматическом режиме сканируют информацию сети на основе заданных алгоритмов, проводя поиск и индексацию документов. В последующем поисковые машины предоставляют пользователю, на основе созданных баз данных, доступ к распределенной на узлах сети информации через выполнение поискового запроса в рамках собственного интерфейса.

Различают четыре режима поиска информации: сканирование, индексирование, классификация и обслуживание.

Сканирование – осуществляется непрерывно и круглосуточно программами в автономном режиме.

Индексирование – предполагает формирование базы данных поисковой машины, организованной по определенному принципу. В результате этой операции для каждого документа формируется набор ключевых слов, по которым затем на стадии обслуживания поискового запроса пользователю выдаются адреса заиндексированных ресурсов.

Классификация – дополнительная функция поисковой машины, которая предполагает, например, присвоение при индексировании пометки о принадлежности данного информационного объекта к определенному типу.

Поисковые машины принято различать по области сканирования:

гипертекстовые базы данных ВЕБ;

ресурсы всемирного пространства Gopher Space, FTP-архивы.

На сегодня в Интернете достаточное число поисковых машин. Ссылки на большинство из них присутствуют на специальной поисковой страничке компании NetsCape Communications.

http://home.netscape.com/escapes/search/ntsrchrun-2.html

Поисковая машина AltaVista – это наиболее полная реализация поисковых возможностей Интернета. Благодаря наличию межпротокольных шлюзов машина располагает адресами ресурсов, доступных по протоколам, отличным от http. Имеется версия, позволяющая искать информацию во всех русскоязычных кодировках. Доступ к ней возможен по адресу:

http://www.altavista.telia.corn/

Страна для поиска выбирается из меню.

Широко известны такие русские поисковые машины, как:

- Yandex – реализована на базе данных сервера Издательского дома «Открытые системы»:

http://win/www.osp.ru

- Rambler – позволяет вести поиск как в Веб, так и в системе телеконференций при распознавании всех кодировок кириллицы и с применением обычных логических коннекторов:

http://rambler.ru

Другой способ поиска – использование **каталогов**. В каталогах информация упорядочена по темам, что облегчает ее поиск. Одним из таких качественных каталогов является так называемые «Желтые страницы Интернета» – http://yp.piter.com.

Обслуживание пользователя той или иной поисковой машиной строится на разработке информационно-поискового языка, естественным образом связанного со структурой базы данных.

Способы поиска: по дереву каталогов; формирование поискового запроса.

В качестве общих рекомендаций по поиску информации можно указать следующие:

- Четко формулировать цель и тему запроса.
- Тщательно подбирать ключевые слова с исключением слов общего характера, предлогов, союзов и вспомогательных слов.
- Подбирать к ключевым словам максимальное количество синонимов и альтернативных слов (как при разгадывании кроссвордов).

Можно осуществлять расширенный поиск с использованием логических операторов:

- && – логическое И (ключевые слова должны находиться в пределах одного документа);
- пробел или & – краткое логическое И (ключевые слова должны находиться в пределах одного абзаца);
- , или | – логическое ИЛИ;
- ( ) – группирование слов;
- ~ (тильда) – оператор И НЕ (в пределах одного абзаца);
- ~~ (двойная тильда) – оператор И НЕ (в пределах одного документа);
- /(nm) – расстояние в словах (- назад, = вперед);
- &&/(nm) – расстояние в абзацах (- назад, = вперед).

Для указания зоны поиска можно использовать следующие команды:

- $Title – поиск в заголовках документов;
- $A – поиск в ссылках и др.

В поисковых системах поиск ведется в гигантской базе данных URL. Для осуществления успешного поиска важно правильно сформулировать запрос путем задания фразы, одного или нескольких ключевых слов, которые лучше всего описывают предмет поиска. Некоторые поисковые системы чувствительны к регистру поиска. Поэтому при вводе ключевых слов и фраз необходимо учитывать эту особенность. В поисковых системах можно использовать следующие способы поиска.

Поиск по одному слову. В поле запроса вводится одно или несколько слов, которые могут характеризовать содержание документа. Например, ввести слово «компьютер», после чего запускается процесс поиска.

Поиск по группе слов. Результат зависит от того, как эти слова введены в конкретной поисковой системе. Например, по запросу *монтаж компьютер* в системе поиска будут разыскиваться документы, в которых встречаются введенные слова. Среди результатов такого поиска возможны нестрогие соответствия, например: …монтаж компьютера…; …после монтажа в компьютере… и т. д.

Поиск в найденном. В системах Rambler, Aport и Yandex после выполнения поиска можно задать другое ключевое слово (или слова) и включить флажок «Искать в найденном». В этом случае поиск по новым ключевым словам производится только среди ранее найденных документов.

Логическая операция AND (И). Связывает два элемента запроса. Будут найдены те страницы, на которых есть оба элемента.

Логическая операция OR (ИЛИ). Связывает два элемента запроса. Ведется поиск страниц, на которых есть хотя бы один из этих элементов.

После перехода на веб-страницу можно найти на ней определенный текст, выбрав команду ***Правка, Найти на этой странице***.

Защита информации в Интернет

Почему нужна защита?

При передаче информации в сети возникает ряд проблем, связанных с ее защитой, при этом возможны следующие виды нарушений:

перехват (нарушение конфиденциальности информации);

модификация (искажение или подмена);

подмена авторства.

В соответствии с этими проблемами под безопасностью подразумевается совокупность трех характеристик:

аутентификация – процесс распознавания пользователя системы и предоставление ему прав и полномочий;

целостность – состояние данных, при котором они сохраняют свое информационное содержание и однозначность интерпретации;

секретность – предотвращение несанкционированного доступа к информации (предотвращение перехвата).

Основные способы защиты: *криптография*, *электронная подпись*.

Криптография, или ***шифрование***. В основе шифрования лежат алгоритм и ключ. Различают шифрование с секретным ключом (симметричное), шифрование с открытым ключом (асимметричное).

При симметричном шифровании и отправитель, и получатель владеют одним и тем же ключом. Недостаток этого способа:

необходимость передачи секретного ключа;

необходимость хранения ключа;

невозможность определить личность отправителя.

При асимметричном шифровании и отправитель, и получатель имеют два разных ключа. При помощи одного ключа послание зашифровывается, а при помощи другого расшифровывается. При этом нет надобности в передаче секретного ключа. Послание зашифровывается с помощью открытого ключа, но расшифровать его может только владелец личного ключа. Недостатком этого способа

является необходимость использования длинных ключей, чтобы обеспечить требуемую степень безопасности.

Электронная подпись. При шифровании остаются нерешенными проблемы модификации или подмены исходного сообщения.

Для исключения этих недостатков предлагается передавать получателю краткое представление передаваемого сообщения (контрольная сумма или дайджест сообщения). Одним из способов передачи контрольной суммы является включение ее в электронную подпись.

Электронная подпись создается шифрованием контрольной суммы и дополнительной информации при помощи личного ключа отправителя. При этом можно расшифровать подпись, используя открытый ключ, но корректно создать подпись может только владелец личного ключа.

Аудентификация – определение права доступа к сети. Для этого используются пароли. Рекомендуется использование одноразовых паролей. Для генерации одноразовых паролей используются как программные, так и аппаратные генераторы, представляющие собой устройства, вставляемые в слоты компьютера. Для приведения в действие данного устройства необходимо задание секретного слова.

Защита сетей

Для защиты корпоративных информационных сетей используются брандмауэры. ***Брандмауэр*** – это система или комбинация систем, позволяющая разделить сеть на две или более частей и реализовать набор правил, определяющих условия прохождения пакетов из одной части в другую. Брандмауэры могут реализоваться как аппаратными, так и программными средствами.

Различают брандмауэры двух типов: пакетные файлы и серверы прикладного уровня.

Пакетные файлы осуществляют фильтрацию IP-пакетов средствами фильтрующих маршрутизаторов.

Серверы прикладного уровня блокируют доступ к определенным серверам в сети.

7.6. СТРУКТУРА БРАУЗЕРА INTERNET EXPLORER

Браузер, или, в переводе, *обозреватель*, – программное средство, обеспечивающее просмотр веб-страниц, формирование запросов,

получение и предоставление документов в сети Интернет. Среди браузеров можно выделить: Internet Explorer, Opera, Netsscape Navigator. Наиболее распространенным браузером в настоящее время является Internet Explorer, интерфейс которого приведен на рисунке 7.5.

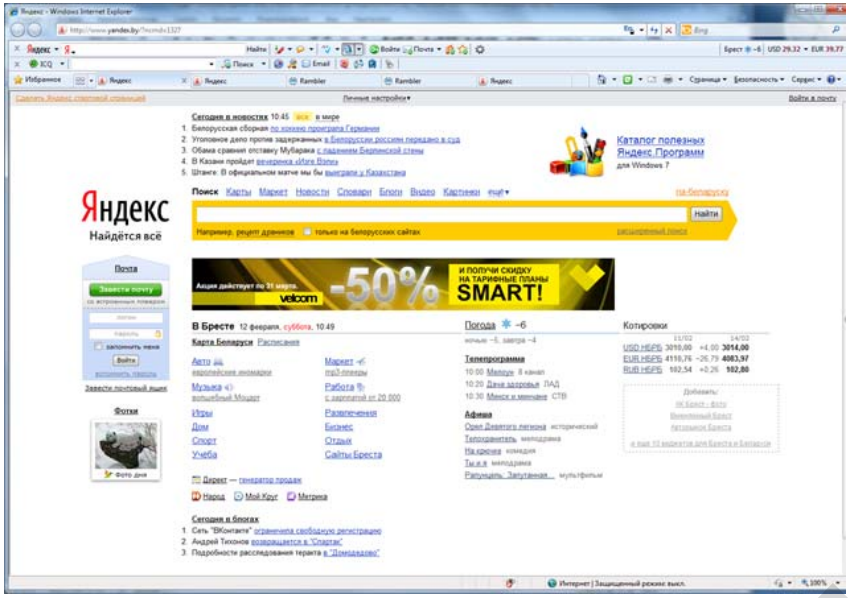


Рис. 7.5. Интерфейс браузера Internet Explorer

Как и любое приложение Windows, он имеет строку заголовка, строку адреса, меню, панели инструментов, строку состояния, окно и строку поиска.

В строке заголовка указано имя текущего поисковика (сайта) и наименование программы, а также три кнопки управления свертыванием, развертыванием/восстановлением и закрытием окна (рис. 7.6).

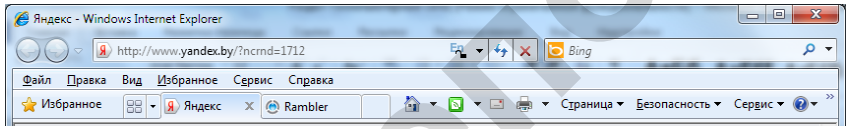


Рис. 7.6. Заголовок и панели инструментов браузера Internet Explorer

Для изменения состава панелей инструментов вызовите контекстное меню панелей инструментов и установите/снимите нужные флажки.

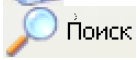
Строка адреса служит для ввода URL-адреса требуемого узла. Раскрывающийся список справа содержит список ранее посещенных узлов. Этот список также можно отредактировать. Для этого введите команду ***Сервис, Свойства обозревателя, Общие***, добавьте узлы в список и щелкните кнопку ***Применить***. Сделанные изменения вступают в силу после перезагрузки браузера.

Кнопки управления. Слева от строки ввода расположены кнопки навигации Вперед и Назад для перемещения по уже посещенным страницам. Эти кнопки позволяют наиболее быстро вернуться к уже просмотренным страницам, так как эти данные хранятся в особой папке – кэше. Ниже приведены другие кнопки, видимые на рисунке 7.6:

Остановить – прерывает загрузку веб-страницы (например, если ожидание появления страницы занимает много времени).

Обновить – служит для обновления содержимого кэш-памяти браузера.

Домой – возврат на стартовую страницу.



Поиск – выводит панель поиска для поиска веб-страниц по ключевому слову или фразе.

Почта – выводит меню с командами для работы в приложении Outlook Express.

Печать – вывод текущей веб-страницы на принтер.

Меню. Меню содержит все команды, доступные для управления браузером.

Строка состояния отображает состояние браузера. Существует несколько основных моментов: 50 % – сайт загружен на 50 процентов; Готово – сайт загружен полностью.

Некоторые места в тексте веб-страницы подчеркнуты и выделены другим цветом на фоне остального текста, указатель мыши над ними принимает форму руки. Такой текст называется гиперссылкой (или просто ссылкой) и является указателем перехода на другие документы. После щелчка по ссылке она изменит свой цвет, а потом содержимое окна исчезнет и будет загружена запрашиваемая страница. При повторной загрузке страницы все ранее использо-

ванные ссылки, как правило, выделяются особым цветом, что значительно экономит время и позволяет не просматривать одни и те же материалы несколько раз.

Internet Explorer имеет свои собственные средства поиска (рис. 7.6). По умолчанию в качестве поисковика используется программа Bing. Поисковик можно заменить, открыв список, размещенный справа от строки поиска.

В нижней строке панели инструментов расположен ряд кнопок и списков. Рассмотрим их по порядку.

Список «Избранное». В этот список можно поместить ссылки на те странички или сайты, которые вам понравились. Эти ссылки называются ***закладками***. Теперь к ним легко будет открыть доступ одним щелчком мыши при очередном сеансе работы.

Список вкладок. Содержит список вкладок часто посещаемых сайтов для быстрого перехода между ними.

Вкладки сайтов. Последняя пустая вкладка служит для создания новой вкладки.

Список «Домой». Служит для перехода к домашней страничке или к выбранной страничке из списка.

Добавление Веб-фрагментов. Получение нового содержимого этой веб-страницы на панели Избранное.

Список «Печать». Открывает доступ к настройкам параметров страниц, просмотра страниц перед печатью и печати веб-страниц.

Страница. Служит для настройки параметров веб-страницы.

Безопасность. Данный список предназначен для настройки и контроля параметров безопасности при работе в сети.

Сервис. Служит для настройки некоторых параметров Internet Explorer.

Справка. Получение справки о программе, технической поддержке и др.

7.7. РАБОТА С ЭЛЕКТРОННОЙ ПОЧТОЙ

Пользователи Интернет имеют возможность обмениваться информацией друг с другом с помощью электронной почты – E-mail – одного из самых популярных и дешевых сервисов. Электронная почта осуществляет передачу сообщений между зарегистрированными адресами. Адрес электронной почты имеет вид adress@tut.by. Справа от символа @ находится доменное имя –

обычный Интернет-адрес компьютера или веб-сайта (www.tut.by), на котором зарегистрирован пользователь. Слева от символа @ помещается имя почтового ящика пользователя. Для получения собственного почтового ящика необходимо: выбрать сайт, предоставляющий почтовые услуги (например, http://www.mail.ru/); зарегистрироваться в качестве нового пользователя, сообщив некоторые сведения о себе. Результатом регистрации является создание личного почтового ящика, доступ к которому обеспечивается через выбранные Вами Имя пользователя (Login) и Пароль.

В Интернет для работы с электронной почтой используются прикладные протоколы SMTP и POP3. Протокол SMTP (Simple Mail Transfer Protocol – простой протокол передачи почты) поддерживает передачу сообщений между почтовыми серверами Интернет. Протокол SMTP допускает использование различных сетевых транспортных служб и почтовых серверов, позволяет группировать сообщения в адрес одного получателя и размножать копии e-mail сообщения для передачи в разные адреса.

Почтовый сервер отвечает за прием всех сообщений, идущих на определенное доменное имя. Для каждого пользователя (почтового ящика) на сервере заводится специальный файл, в который помещаются поступающие сообщения в ожидании того, когда пользователь соединится с сервером для их получения. Также сервер производит рассылку адресатам сообщений, поступивших от его пользователей. Предварительно он выясняет, какой сервер отвечает за прием сообщений на доменное имя адресата (например, сообщения с адресами кто-либо@tut.by принимаются сервером tut.by), после чего устанавливает с этим сервером связь по протоколу SMTP.

POP3 (Post Office Protocol) дает пользователю доступ к пришедшим ему электронным сообщениям, т. е. осуществляет связь между компьютером пользователя и почтовым сервером, на котором зарегистрирован почтовый ящик пользователя. При конфигурировании почтовой программы нужно указать Интернет-адрес (доменное имя) почтового сервера для входящих (POP-server) и исходящих (SMTP-server) сообщений (как правило, это один и тот же сервер) и имя пользователя на сервере (login name, POP-user login, учетная запись). При установлении соединения будет запрошен пароль. Имя пользователя и пароль предоставляются администрацией сервера. Пользователь может изменить пароль.

Получение электронной почты можно осуществлять с любого компьютера. Нужно только, чтобы компьютер был подключен

к Интернету. Для работы с электронной почтой созданы специальные программы, например, Netscape Mail, Eudora, MS Outlook. Многие поисковики также имеют свои почтовые службы.

Для подключения к почтовому серверу необходимо получить пароль поставщика услуг.

Контрольные вопросы

1. Что такое Интернет?
2. Что такое HTTP и HTML?
3. Что понимается под веб-сайтом?
4. Что включает IP-адрес?
5. Что такое домен, доменный адрес?
6. Что такое URL-адрес?
7. Что называется хостом?
8. Что понимается под браузером?
9. Какие браузеры получили наибольшее распространение?
10. Назовите основные кнопки интерфейса Internet Explorer и их назначение.
11. Перечислите наиболее распространенные русскоязычные поисковые системы.
12. Как осуществляется поиск информации в сети?
13. Какие логические операции используются при поиске информации?
14. Какую структуру имеет адрес электронной почты?
15. Как осуществляется доступ к личному почтовому адресу?
16. Какие прикладные протоколы используются для работы с электронной почтой?

7.8. ЯЗЫК ГИПЕРТЕКСТОВОЙ РАЗМЕТКИ ДОКУМЕНТОВ HTML

Язык описания гипертекстовой разметки документов HTML (Hyper Text Markup Language) был разработан сотрудником Европейской лаборатории элементарных частиц Тимом Бернерс-Ли в 1989 году. Он же реализовал и первую программу для просмотра HTML-документов. Предпосылкой для разработки языка стала необходимость обмениваться множеством различных документов при помощи быстро развивающейся сети Интернет. И в настоящее время он является общепризнанным языком для создания веб-страниц.

HTML-документ – это обычный текстовый документ, созданный в любом текстовом редакторе и оформленный в соответствии с правилами языка. Обычно для написания HTML-документа используют стандартное приложение Windows – Блокнот. Кроме того, практически все приложения Windows позволяют создавать веб-странички, а несколько веб-страничек, объединенных единой системой навигации, – это уже сайт. Файл, содержащий HTML-документ, должен иметь расширение .htm или .html.

Основным элементом конструкции языка HTML является ***тэг***. Тэг представляет собой команду языка HTML, заключенную в скобки, – <команда>. Тэги бывают парными и непарными. Парный тэг состоит из открывающего тэга и закрывающего тэга, например: *<P> Привет! </P>*.

Здесь <P> – открывающий тэг, </P> – закрывающий тэг.

Непарный (одиночный) тэг не имеет закрывающего тэга.

Тэги кроме команд могут содержать атрибуты и значения, которые помещаются в открывающем тэге перед закрывающей скобкой, количество пар атрибут/значение в открывающем тэге не ограничивается. Пары атрибут/значение разделяются в тэге пробелом: <BODY bgcolor = “#FF0000” text = ”#0000FF”>.

В соответствии с принятыми правилами команды HTML записываются в верхнем регистре (для удобства чтения).

Тэги могут быть вложенными, при этом не допускается пересечение тэгов.

Структура HTML-файла

HTML-документ в общем случае состоит из трех частей: данные о версии используемого HTML, заголовок документа, тело документа (рис. 7.7).

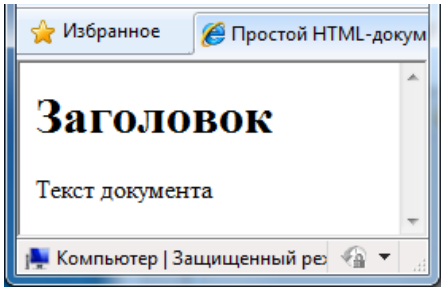


Рис. 7.7. Пример простого HTML-документа

Пример 7.1. Пример простейшего HTML-документа:

```
<HTML version = “4.01”> ‘ тэг HTML-документа
   <TITLE> Простой HTML-документ </TITLE>
   <BODY> ‘ тэг тела документа
      <H1> Заголовок </H1>
         Текст документа
   </BODY>
</HTML>
```

Первый тэг указывает используемую версию языка программирования, второй тэг – заголовок страницы, отображаемый браузером, третий тэг содержит тело документа.

Данные о версии используемого языка

HTML-документ должен начинаться с указания версии используемого HTML, например:

<!DOCTYPE HTML PUBLIC “ – //W3C/ /DTD HTML 4/01//EN”> – использовать строгое определение HTML версии 4.01, в которое не включаются нежелательные для версии 4.01 элементы и атрибуты, а также фреймы.

За указанием версии документа следует ключевой элемент структуры HTML-документа – элемент HTML.

Элемент HTML является корневым элементом структуры HTML-документа, задается при помощи парных тэгов <HTML> и </HTML>. Данный тэг позволяет браузеру распознать, что им обрабатывается HTML-код. В нем можно указать версию языка программирования HTML:

<HTML version = “4/01”>,

а также можно задавать используемый язык (атрибут lang) и направление текста (атрибут dir):

<HTML lang = “ru” dir = LTR > – задан русский язык, направление текста слева направо.

Заголовок документа

В заголовке могут использоваться элементы HEAD, TITLE, BASE, META.

Элемент HEAD является парным тэгом и служит для описания заголовка документа. Этот тэг не является обязательным и может быть опущен.

Элемент TITLE является парным тэгом и служит для описания заголовка документа:

<TITLE> Знакомство с языком программирования HTML </TITLE>

Элемент BASE – непарный тэг. Служит для указания базового индикатора ресурса URI, который содержит универсальный адрес ресурса и полное имя файла ресурса. Для задания URI в данном тэге используется атрибут href:

<BASE href = “D:\test\test.html”>

В заголовке HEAD может указываться также одиночный элемент ***META*** – метаданные, который может содержать атрибуты: *name* – имя переменной, *content* – значение переменной, *lang* – код языка, *http-equiv* – указание браузеру дополнительных параметров по обработке документа. Например, с помощью элемента META можно описать ключевые слова, используемые в документе:

<META name = “Keywords” lang = “ru” content = “HTML, веб-дизайн, гипертекст, сайт, сайты, …”>

Тело HTML-документа

Начало и конец документа обозначаются парным тэгом <BODY> и </BODY>. Эти тэги не являются обязательными, но они лучше отражают структуру документа. В элементе BODY используют следующие атрибуты:

background – URI, указывается изображение для фона, обычно берется небольшое изображение, которое размножается для заполнения фона всего документа;

bgcolor – цвет фона документа;

text – цвет шрифта документа;

link – цвет непосещенных гиперссылок;

vlink – цвет посещенных гиперссылок;

alink – цвет гиперссылок при выборе их пользователем.

Цвет можно задавать текстовыми константами или шестнадцатеричными константами (табл. 7.6).

Таблица 7.6

Идентификаторы и значения цветов, используемых в HTML-документах

Название	Значение	Название	Значение
черный	*black*	каштановый	*maroon*
зеленый	*green*	красный	*red*
серебряный	*silver*	синий	*blue*
лимонный	*lime*	пурпурный	*purple*
серый	*gray*	бирюзовый	*teal*
оливковый	*olive*	фуксиновый	*fuchsia*
белый	*white*	голубой	*aqua*
желтый	*yellow*	темно-синий	*navy*

Пример 7.2. Пример заголовка документа и задания параметров цветов.

```
<HTML> ‘ начало веб-документа
<HEAD> ‘ заголовочная часть документа
<TITLE> Пример задания цветов в теле HTML-документа
</TITLE> ‘ заголовок
</HEAD>
<BODY
    background = “D:\WallpapersMania_val45-1222.jpg”
    bgcolor = “yellow”
    text = “black”
    link = “#0080FF”
    vlink = “#00F000”
    alink = “#000080”>
    Обычный неформатированный текст должен отображаться черным цветом, непосещенные гиперссылки – голубым цветом, посещенные гиперссылки – коричневым, а выделенные гиперссылки – темно-синим.
    <P> <A HREF = “ref1”> Непосещенная гиперссылка (голубой) </A>
    <P> <A HREF = “ref2”> Посещенная гиперссылка (коричневый)
    </A>
    <P> <A HREF = “ref3”> Выделенная гиперссылка (темно-синий)
    </A>
</BODY>
</HTML>
```

Пример выполнения фрагмента программы приведен на рисунке 7.8. При этом текст заголовка отображается на вкладке браузера.

Для оформления текста документа, вставки таблиц, гиперссылок, рисунков и т. п. в языке программирования предусмотрены специальные тэги. Рамки учебного пособия не позволяют подробно рассматривать все тэги, поэтому ограничимся лишь некоторыми наиболее часто используемыми.

Совет: для просмотра нужных Вам тэгов оформите документ в текстовом процессоре MS Word, сохраните его как веб-страницу и просмотрите код в Блокноте.

Тэги оформления заголовков. Имеется шесть уровней заголовков, которым соответствуют элементы H1, H2, H3, H4, H5, H6. Заголовок H1 самый крупный. Заголовки задаются с помощью парных тэгов, например: *<H1> Заголовок первого уровня </H1>*.

Тэги форматирования текста приведены в таблице 7.7.

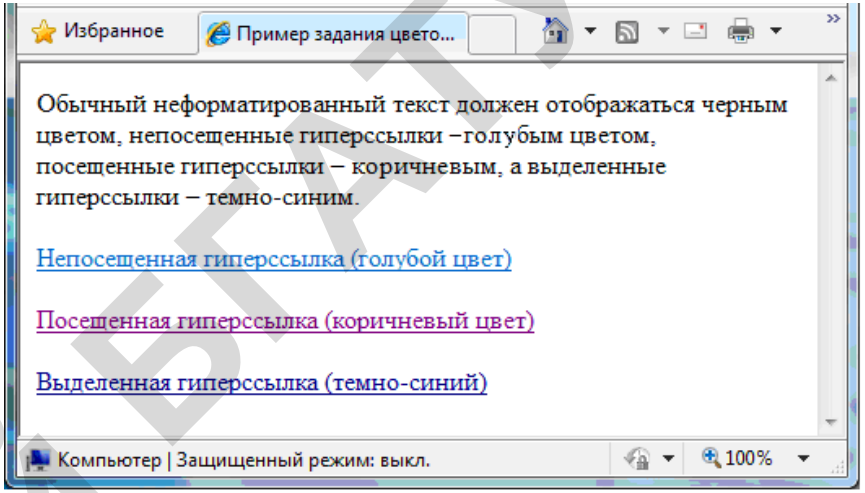


Рис. 7.8. Пример задания названия и параметров цветов

Таблица 7.7

Тэги форматирования текста

Элемент	Описание	Элемент	Описание
B	Полужирный текст	BIG	Текст с увеличенным размером шрифта
I	Наклонный текст, курсив	SMALL	Текст с уменьшенным размером шрифта
U	Подчеркивание текста	SUB	Верхний индекс
S, STRIKE	Перечеркнутый текст	SUP	Нижний индекс

Для перевода курсора на новую строку используется непарный тэг «возврат каретки» –
. Теги форматирования могут быть вложенными:

```
<B> Полужирный текст </B> <BR>
<B> <I> Полужирный курсив <I> <B> <BR>
```

Параметры шрифта: тип, размер, цвет – задаются с помощью элемента FONT с соответствующими атрибутами:

fase – задает тип шрифта: Arial, Times New Roman, Curier и т. д.;

size – задает размер шрифта. Для атрибута size имеется только семь значений, которые задаются цифрами от 1 до 7, причем размер 1 – самый мелкий шрифт;

color – задает цвет шрифта.

<FONT fase = "arial" size = "3" color "black"> Текст документа </FONT>

Специальные тэги:

<CENTER> и *</CENTER>* – выравнивание текста по центру;

<PRE> и *</PRE>* – сохранение исходного форматирования текста: *<PRE> Текст документа </PRE>*

<NOBR> и *</NOBR>* – запрет разрывов строк;

&shy – мягкий перенос. Вставляется в текст документа в нужном месте. Например: для переноса слова информатика в тексте HTML-документа можно указать: инфор&shyмати&shyка. Мягкий перенос игнорируется, если текст указан в тэгах запрета разрыва строк NOBR;

<! - - текст - -> – вставка комментариев в текст HTML-документа. Комментарий может занимать несколько строк.

Форматирование абзацев. Для выделения абзацев используется парный тэг <P> и </P>. Для элемента P можно задать несколько атрибутов:

align – выравнивание текста. Этот атрибут может принимать три значения: *left* – выравнивание по левому краю (принимается по умолчанию), *right* – выравнивание по правому краю, *center* – выравнивание по центру и *justify* – выравнивание по ширине;

title – задает текст подсказки.

Пример 7.3. Использование абзацев (рис. 7.9).

```
<TITLE> Пример управления абзацами и текстом </TITLE>
<BODY>
    Неформатированный текст <BR>
    <P> <FONT size = 5> Увеличенный текст </FONT> </P>
    <P align = right> Выравнивание текста по правому краю </P>
    <P align = center title "Выравнивание по центру">
    Выравнивание текста по центру </P>
    <P align = justify title "Выравнивание по ширине">
        <FONT size = 2> Текст 1 <B> Текст выделенный
            <I> Текст полужирный курсив </I> Текст полужирный </B>
        </FONT>
    </P>
</BODY>
```

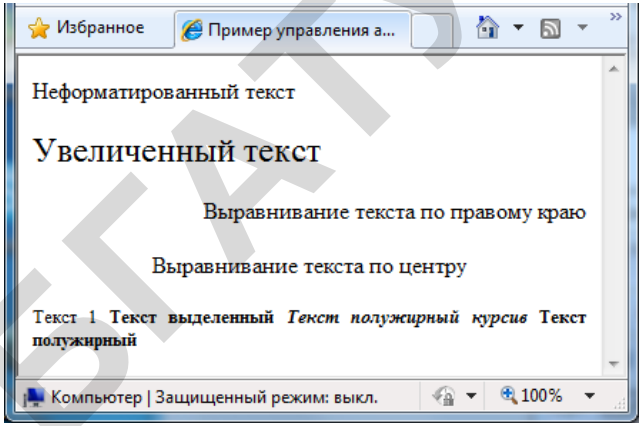



Рис. 7.9. Пример управления абзацами и шрифтами

Вставка изображения

Для вставки изображения из файла используется одиночный тэг <IMG src = «спецификация файла»>, например:

<IMG src = «filename.gif»> – вставить изображение из файла filename.gif.

Поддерживаются форматы gif и jpg. Для создания ссылки на другой сайт применяется тэг <A HREF = "http://www.website.by"> ссылка </A> – сделать слово «ссылка» гипертекстовой ссылкой на сервер www.website.by.

Пример 7.4. Вставка рисунка из файла (рис. 7.10).

```
<HTML>
<HEAD>
<BASE> href = "D:\пример4.html"
<TITLE> Вставка рисунка из файла </TITLE>
</HEAD>
<BODY>
<CENTER>
<BR>
<FONT color = "blue" size = 5> <H2> Братья наши меньшие HTML.
</H2> </FONT>
<FONT color = "green"> Сохраняйте природу! </FONT>
</CENTER>
</BR>
<P align = center> <B> Пример домашней странички </B> </P>
```

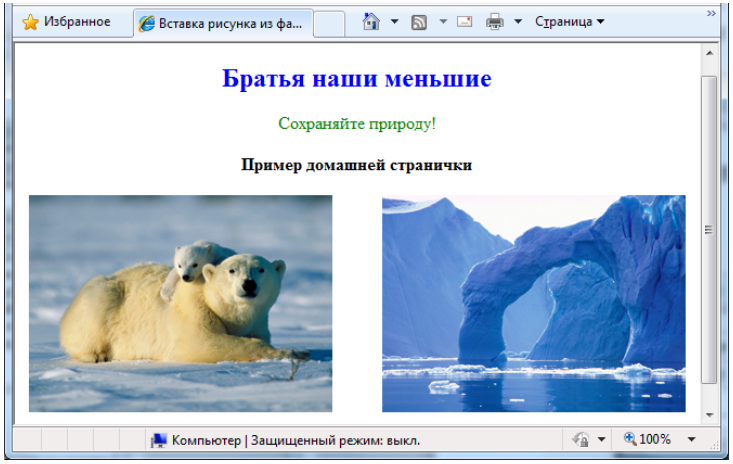


Рис. 7.10. Пример вставки рисунка из файла

```
<IMG src = “D:zoo20.jpg” height = 200 width = 280 alt = “Здесь должно быть изображение” align = left>
<IMG src = “D:09.jpg" height = 200 width = 280 alt = "Здесь должно быть изображение” align = right>
</BODY>
</HTML>
```

Вставка таблиц

Таблица состоит из рядов, каждый из которых включает набор элементов. Элемент таблицы может содержать текст, рисунок, другую таблицу и т. п., то есть все то, что может содержать сам документ. В простейшем случае таблица – это традиционно понимаемая таблица из текстовых ячеек. Рассмотрим тэги, используемые для создания таблицы.

<TABLE WIDTH = 100 % BORDER = 0> </TABLE> – начало и конец таблицы. Параметр WIDTH определяет ширину таблицы в процентах от ширины окна браузера либо в пикселах, если не указан знак %. Параметр BORDER определяет толщину линий обрамления ячеек таблицы в пикселах. Нулевое значение – нет обрамления. Любой из параметров может отсутствовать.

<TR> </TR> – начало и конец строки таблицы, должен находиться внутри тэга <TABLE> </TABLE>.

<TD WIDTH = 20 % ALIGN = left VALIGN = middle BGCOLOR = “blue”> </TD> – определяет ячейку таблицы, которая должна находиться внутри тэга <TR> </TR>.

ALIGN и VALIGN определяют выравнивание содержимого внутри ячейки по горизонтали и вертикали соответственно, значения ALIGN: left, center, right (выравнивание по левому краю, по центру, по правому краю), значения VALIGN: top, middle, bottom (выравнивание по верхнему краю, по центру, по нижнему краю).

BGCOLOR определяет цвет фона ячейки. Любой из параметров может отсутствовать. Рассмотрим пример. Данные приведены в таблице 7.8.

Таблица.7.8

Исходные данные для построения таблицы

Мониторы	Цена	Количество
Samsung	200 $	10
ASUS	250 $	20

Пример 7.5. Создание таблицы (рис. 7.11).

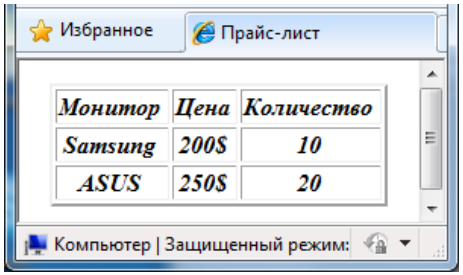


Рис. 7.11. Пример разработки таблиц

```
<HTML>
<HEAD>
<TITLE> Прайс-лист </TITLE>
</HEAD>
<BODY>
<CENTER>
<TABLE BORDER = 2 width = 60 %>
<TR align=center>
<TD width = 33% ><B> <I>Монитор</B> </I> </TD>
<TD width = 34 % > <B> <I>Цена</B> </I> </TD>
<TD width = 33 % > <B> <I>Количество</B> </I> </TD>
</TR>
```

```
<TR align = center>
<TD width = 33 %> <B> <I>Samsung</B> </I> </TD>
<TD width = 34 %> <B> <I>200 $</B> </I> </TD>
<TD width = 33 %> <B> <I>10</B> </I> </TD>
</TR>
<TR align = center>
<TD width = 33 %> <B> <I>ASUS</B> </I> </TD>
<TD width = 34 %> <B> <I>250 $</B> </I> </TD>
<TD width = 33 %> <B> <I>20</B> </I> </TD>
</TR>
</TABLE>
</CENTER>
</BODY>
</HTML>
```

Вставка гиперссылки

Для быстрого перемещения в пределах текущего документа или между веб-страницами используются гиперссылки.

Для создания простейшей гиперссылки, обеспечивающей переход к нужному HTML-элементу, достаточно задать в качестве значения атрибута href элемента A адрес нужного ресурса:

<A href = "1.html"> Первая страница сайта </A> – переход между страницами сайта;

<A href = "zoo20.jpg"> Природа </A> – загрузка файла.

Для перемещения в пределах документа необходимо ознакомиться с понятием «*якорь*». Различают два якоря: начальный якорь и якорь назначения. Переход по гиперссылке осуществляется от начального якоря к якорю назначения. Части документов, на которые могут устанавливаться гиперссылки, выделяются специальным образом, то есть на них необходимо установить какие-то уникальные метки. Такими объектами могут быть, например, заголовки разделов. Установка меток осуществляется с помощью аргумента name элемента A:

<A name = "par1"> <H1> Раздел 1 </H1> – установлена метка;

<A href = "#par1"> Раздел 1 </A> – создана гиперссылка для перехода к созданному якорю из другого места текущего документа.

Для перехода к данному якорю из другого документа в адресе перехода необходимо дополнительно указать имя документа, содержащего якорь. Пусть имя файла, содержащего якорь, – par1, file1.html. Тогда гиперссылка будет иметь вид:

<A href = "file1.html#par1"> Раздел 1 </A>

Для создания якоря можно использовать идентификаторы элементов HTML-документов. Тогда команда создания якоря будет иметь следующий вид:

<H1 id = "par1"> Раздел 1 </H>

Регистрация веб-страницы

После создания веб-страницы ее необходимо разместить на каком-либо сервере, чтобы она была доступна через Интернет. Перед тем как размещать на сервере страницу, необходимо ознакомиться с правилами размещения, которые приводятся прямо на выбранном сервере. Существует ряд бесплатных серверов, на которых можно разместить веб-страницы. Среди них можно выделить русскоязычные: http://www.by.ru/; http://www.tut.by/; http://www.chat.ru/; http://www.halyva.ru/; англоязычные: http://www.geocites.com/; http://www.hypermart.net/; http://www.angelfire.com/.

Контрольные вопросы

1. Что представляет собой файл HTML?
2. Какие расширения может иметь файл, содержащий HTML-документ?
3. Как называются команды HTML?
4. Каким символом закрываются парные тэги?
5. Какие тэги позволяют установить полужирный шрифт и курсив?
6. Как установить цвет и размер букв текста?
7. Как вставить изображение из файла в веб-страницу?
8. Какие тэги используются для формирования таблицы?
9. С помощью какого тэга можно регулировать толщину линий обрамления ячеек таблицы?
10. Какие русскоязычные сайты можно использовать для регистрации веб-странички?

Заключение

Интернет в настоящее время является наиболее доступным средством обмена информацией. Объем информации, содержащейся в Интернет, практически неограничен, а постоянное пополнение и обновление этой информации делает его универсальным источником данных во всех областях человеческой деятельности. В данном разделе рассмотрены вопросы организации работы в Интернет: поиск информации, работа с электронной почтой. В заключение даны основы HTML, что позволит будущему инженеру создавать веб-странички для активной рекламы результатов собственных исследований.

ЛИТЕРАТУРА

К РАЗДЕЛУ 1

1. Алексеев, А. П. Информатика / А. П. Алексеев. – Москва : Солон-Р, 2002. – 400 с. : ил.

2. Бройдо, В. Л. Архитектура ЭВМ и систем / В. Л. Бройдо, О. П. Ильина. – Санкт-Петербург : Питер, 2009. – 720 с. : ил.

3. Быков, В. Л. Основы информатики : конспект лекций / В. Л. Быков. – Брест : БГТУ, 2002. – 253 с.

4. Быков, В. Л. Основы информатики : пособие / В. Л. Быков, Ю. П. Ашаев. – Брест : БГТУ, 2006. – 430 с. : ил.

5. Информатика / под ред. Н. В. Макаровой. – Москва : Финансы и статистика, 2000. – 768 с. : ил.

6. Информатика. Базовый курс / С. В. Симонович [и др.]. – Санкт-Петербург : Питер, 2000. – 640 с. : ил.

7. Использование Microsoft Windows XP Home Edition. – Москва : Вильямс, 2002. – 896 с. : ил.

8. Коуров, Л. В. Словарь-справочник по информатике / Л. В. Коуров. – Минск : Амалфея, 2000. – 176 с.

9. Леонтьев, В. П. Новейший и полный самоучитель. Компьютер + Офис + Интернет + Развлечения / В. П. Леонтьев. – Москва : ОЛМА Медиа Групп, 2010. – 800 с. : ил.

10. Ляхович, В. Ф. Основы информатики / В. Ф. Ляхович. – Ростов-на-Дону : Феникс, 2006. – 640 с.

11. Новейший самоучитель работы на компьютере. – Москва : ДЕСС КОМ, 2000. – 654 с.

12. Олифер, Н. А. Сетевые операционные системы / Н. А. Олифер, В. Г. Олифер. – Санкт-Петерубрг : Питер, 2009. – 672 с.

13. Основы информатики : учебное пособие / под ред. А. Н. Морозевича. – Минск : Новое знание, 2007. – 544 с. : ил.

14. Острейковский, В. А. Информатика / В. А. Острейковский. – Москва : Высшая школа, 2000. – 511 с. : ил.

15. Романова, Ю. Д. Информатика и информационные технологии : учебное пособие / Ю. Д. Романова. – Москва : Эксмо, 2010. – 688 с.

16. Фигурнов, В. Ф. IBM PC для пользователя / В. Ф. Фигурнов. – Москва : Финансы и статистика, 2003. – 688 с.

17. Экономическая информатика / под ред. : П. В. Конюховского, Д. Н. Колесова. – Санкт-Петербург : Питер, 2001. – 560 с. : ил.

К РАЗДЕЛУ 2

18. Гурский, Д. А. Вычисления в MathCad / Д. А. Гурский. – Минск : Новое знание, 2003. – 814 с.

19. Ильин, В. П. Численные методы решения задач строительной механики : справ. пособие / В. П. Ильин, В. В. Карпов, А. М. Масленников; под общ. ред. В. П. Ильина. – Минск : Выш. шк., 1990. – 349 с.

20. Кирьянов, Д. В. Самоучитель MathCad 11 / Д. В. Кирьянов. – Санкт-Петербург : БХВ-Петербург, 2003. – 560 с. : ил.

21. Кулаичев, А. П. Методы и средства анализа данных в среде Windows. Stadia 6.0 / А. П. Кулачев. –2-е изд., перераб. и доп. – Москва : Информатика и компьютеры, 1998. – 270 с.

22. Математический энциклопедический словарь / гл. ред. Ю. В. Прохоров; ред. кол. : С. И. Адян [и др.]. – Москва : Сов. Энциклопедия, 1988. – 847 с. : ил.

23. Рыжиков, Ю. И. Решение научно-технических задач на персональном компьютере / Ю. И. Рыжиков. – Санкт-Петербург : КОРОНА принт, 2000. – 272 с.

24. Mathcad 6.0 PLUS. Финансовые, инженерные и научные расчеты в среде Windows 95 : пер. с англ. – Москва: Филинь, 1996. – 712 с.

К РАЗДЕЛУ 3

См. раздел 1 п. 10.

25. Дьяконов, В. П. Справочник по алгоритмам и программам на языке БЕЙСИК для персональных ЭВМ / В. П. Дьяконов. – Москва : Наука. Гл. ред. физ.-мат. лит., 1987. – 240 с.

26. Турчак, Л. И. Основы численных методов / Л. И. Турчак. – Москва : Наука. Гл. ред. физ.-мат. лит., 1987. – 320 с.

К РАЗДЕЛУ 4

Для заметок

См. раздел 1 пп. 4, 5, 7, 9,15.

К РАЗДЕЛУ 5

См. раздел 1 пп. 9, 15.

К РАЗДЕЛУ 6

См. раздел 1 пп. 4, 9, 13, 14, 15.

27. Вадзинский, Р. Статистические вычисления в среде Excel / Р. Вадзинский. – Санкт-Петербург : Питер, 2008. – 608 с. : ил.

28. Гарнаев, А. Ю. Excel, VBA, Internet в экономике и финансах / А. Ю. Гарнаев. – Санкт-Петербург : БХВ, 2005. – 816 с. : ил.

29. Гельман В. Я. Решение математических задач средствами Excel : практикум / В. Я. Гельман. – Санкт-Петербург : Питер, 2003. – 240 с. : ил.

30. Демидова, Л. А., Пылькин А. Н. Программирование в среде Visual Basic for Applications : практикум / Л. А. Демидова, А. Н. Пылькин. – Москва : Горячая линия-Телеком, 2004. – 175 с. : ил.

К РАЗДЕЛУ 7

См. раздел 1 пп. 9, 10, 15.

31. Истабрук, Н. Internet. Освой самостоятельно за 24 часа : пер. с англ. / Н. Истабрук. – Москва : БИНОМ, 1998. – 320 с. : ил.

32. Стратегия развития информационного общества в Республике Беларусь на период до 2015 года : постановление Совета Министров Респ. Беларусь, 9 авг. 2010 г., № 1174 // Нац. реестр правовых актов Респ. Беларусь. – 2010. – № 197. – 5/32317.

33. Холмогоров, В. Основы Web-мастерства : учебный курс / В. Холмогоров. – Санкт-Петербург : Питер, 2002. – 352 с. : ил.

34. Чиртик, А. А. HTML : популярный самоучитель / А. А. Чиртик. – Санкт-Петербург : Питер, 2008. – 256 с. : ил.

Для заметок

Учебное издание

Быков Вячеслав Леонидович
Серебрякова Наталья Григорьевна

ИНФОРМАТИКА

Учебно-методическое пособие

Ответственный за выпуск Н. Г. Серебрякова
Редактор А. И. Третьякова
Компьютерная верстка А. И. Третьяковой

Подписано в печать 4.03.2013. Формат $60\times84^{1}/_{16}$.
Бумага офсетная. Ризография.
Усл. печ. л. 38,13. Уч.-изд. л. 29,8. Тираж 100 экз. Заказ 201.

Издатель и полиграфическое исполнение: учреждение образования
«Белорусский государственный аграрный технический университет».
ЛИ № 02330/0552984 от 14.04.2010.
ЛП № 02330/0552743 от 02.02.2010.
Пр. Независимости, 99-2, 220023, Минск.