Министерство сельского хозяйства и продовольствия

Белорусский государственный
аграрный технический университет

Кафедра информационных процессов и технологий

Программирование на языке Object Pascal в визуальной среде Delphi

Методические указания к лабораторным занятиям
по дисциплине
“Основы информатики и вычислительной техники”

Часть 1

Для студентов факультета предпринимательства и управления

Минск - 2001

Репозиторий БГАТУ

СИСТЕМЫ СЧИСЛЕНИЯ

Цель: изучить различные системы счисления и научиться производить в этих системах простейшие арифметические операции.

Краткие теоретические сведения

Под системой счисления понимают способ записи чисел с помощью цифр и символов (букв).

Число R в p-ичной системе счисления можно представить в развёрнутом виде:

$$R = a_n p^n + a_{n-1} p^{n-1} + ... + a_1 p^1 + a_0 p^0 + a_{-1} p^{-1} + ... + a_{-k} p^{-k} = \sum_{i=-k}^{n} a_i p^i,$$

где p – основание системы счисления;

a_i – разрядные коэффициенты.

Основанием системы счисления называется количество цифр и символов, используемых в данной системе счисления. Например, в десятичной системе счисления (p=10) используются 10 цифр: 0,1,2,…,9.

Обычно число R записывают в сокращённом виде с помощью разрядных коэффициентов:

$$R_p = a_n a_{n-1} ... a_1 a_0 a_{-1} ... a_{-k}.$$

Например, в десятичной системе счисления (p=10)

$$R_{10} = 504{,}38_{10} = 5 \cdot 10^2 + 0 \cdot 10^1 + 4 \cdot 10^0 + 3 \cdot 10^{-1} + 8 \cdot 10^{-2}.$$

Двоичная система счисления

В двоичной системе счисления для изображения любых чисел используются две цифры: 0 и 1.

Число R в двоичной системе счисления (p=2) запишется следующим образом:

$$R_2 = 110{,}101_2 = 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3}.$$

Приведём примеры выполнения операций сложения и вычитания в двоичной системе счисления:

$$\begin{array}{r} 0{,}1101 \\ +\ 1{,}0111 \\ \hline 10{,}0100 \end{array} \qquad \begin{array}{r} 101{,}011 \\ -\ 10{,}001 \\ \hline 11{,}010 \end{array}$$

Репозиторий БГАТУ

Шестнадцатеричная система счисления

В шестнадцатеричной системе счисления для изображения чисел используются 16 символов: 10 цифр (0,1,2,…,9) и 6 букв (A,B,C,D,E,F).

Пример записи числа в шестнадцатеричной системе счисления (p=16):

$$R_{16} = 3A5,D7_{16} = 3\cdot16^2 + A\cdot16^1 + 5\cdot16^0 + D\cdot16^{-1} + 7\cdot16^{-2}.$$

Между символами шестнадцатеричной и десятичной систем счисления существует соответствие, показанное в таблице:

Шестнадца-теричная система	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10
Десятичная система	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Этой таблицей удобно пользоваться при выполнении арифметических действий в шестнадцатеричной системе счисления, например:

$$\begin{array}{r} 5B9,A3 \\ +\ E3C,45 \\ \hline 13F5,E8 \end{array} \qquad \begin{array}{r} A3E,2A \\ -\ 7B4,1C \\ \hline 28A,0E \end{array}$$

Перевод чисел из одной системы счисления в другую

Для перевода числа из шестнадцатеричной системы в двоичную необходимо каждую шестнадцатеричную цифру записать в виде двоичной тетрады (группы из четырёх двоичных цифр) согласно таблице:

Шестнадца-теричная система	0	1	2	3	4	5	6
Десятичная система	0000	0001	0010	0011	0100	0101	0110

7	8	9	A	B	C	D	E	F
0111	1000	1001	1010	1011	1100	1101	1110	1111

Например: $E5A,7B_{16}$ = 1110 0101 1010, 0111 $1011_{2.}$

Для перевода числа из двоичной системы в шестнадцатеричную необходимо разбить его на тетрады влево и вправо от запятой и каждую тетраду записать в виде шестнадцатеричной цифры, например:

$11010101011,1100111_2$ = 0110 1010 1011, 1100 1110_2 = $6AB,CE_{16}$

Репозиторий БГАТУ

Для перевода числа из двоичной или шестнадцатеричной системы счисления в десятичную необходимо записать это число в развёрнутом виде, а затем выполнить действия в результирующей (т.е. десятичной) системе счисления, например:

$$110111{,}01101_2 = 1\cdot 2^5 + 1\cdot 2^4 + 0\cdot 2^3 + 1\cdot 2^2 + 1\cdot 2^1 + 1\cdot 2^0 + 0\cdot 2^{-1} + 1\cdot 2^{-2} +$$

$$+ 1\cdot 2^{-3} + 0\cdot 2^{-4} + 1\cdot 2^{-5} = 55{,}40625_{10}$$

$$AC4, B7_{16} = A\cdot 16^2 + C\cdot 16^1 + 4\cdot 16^0 + B\cdot 16^{-1} + 7\cdot 16^{-2} =$$

$$= 10\cdot 16^2 + 12\cdot 16^1 + 4\cdot 16^0 + 11\cdot 16^{-1} + 7\cdot 16^{-2} \approx 2756{,}7148_{10}.$$

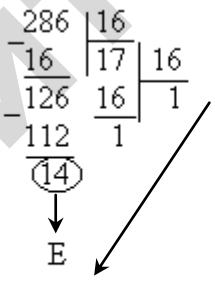
При переводе чисел из десятичной системы в двоичную или шестнадцатеричную целая и дробная части десятичного числа переводятся отдельно различными способами.

Для перевода целой части десятичного числа применяется способ последовательного деления на основание системы счисления. В качестве результата выписываются последнее частное от деления и все остатки в *обратном порядке*.

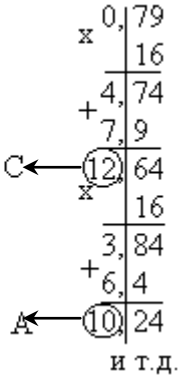
Для перевода дробной части десятичного числа применяется способ последовательного умножения на основание системы счисления. Умножается только дробная часть результата. Последовательное умножение выполняется до получения требуемой точности. В качестве результата выписываются все целые части отдельных произведений в прямом порядке начиная с нуля.

Пусть, например, требуется перевести число $286{,}79_{10}$ из десятичной системы счисления в шестнадцатеричную.

Целая часть



Дробная Часть



$286_{10}=11E_{16}$ $0{,}79_{10}\approx CA_{16}$

Окончательно имеем: $286{,}79_{10}\approx 11E{,}CA_{16}$

Индивидуальные задания

Репозиторий БГАТУ

Вариант	Вычислить значение выражения	Ответ
1	**2**	**3**
1	$(54,76_{16}+10,1_2)+(5B,E8_{16}-79,23_{10})+(79,23_{10}-54,76_{16})-5B,E8_{16}$	$10,1_2$
2	$(79,23_{10}-10,1_2)+54,76_{16}+(5B,E8_{16}+10,1_2)-(79,23_{10}+5B,E8_{16})$	$54,76_{16}$
3	$(5B,E8_{16}-10,1_2)+(54,76_{16}+79,23_{10}-5B,E8_{16})+(10,1_2-54,76_{16})$	$79,23_{10}$
4	$(79,23_{10}+54,76_{16}-10,1_2)+(5B,E8_{16}-79,23_{10}+10,1_2-54,76_{16})$	$5B,E8_{16}$
5	$C9,7B_{16}-(35,72_{16}-1,111_2)+(83,64_{10}-C9,7B_{16})+(35,72_{16}-83,64_{10})$	$1,111_2$
6	$(1,111_2-C9,7B_{16}+35,72_{16})+(83,64_{10}-1,111_2)+(C9,7B_{16}-83,64_{10})$	$35,72_{16}$
7	$(1,111_2-C9,7B_{16})+(C9,7B_{16}-35,72_{16}+83,64_{10})+(35,72_{16}-1,111_2)$	$83,64_{10}$
8	$(83,64_{10}-1,111_2+35,72_{16})+(C9,7B_{16}-83,64_{10})+(1,111_2-35,72_{16})$	$C9,7B_{16}$
9	$(73,45_{16}+11,01_2)+(A4,6C_{16}-37,29_{10})+(37,29_{10}-73,45_{16})-A4,6C_{16}$	$11,01_2$
10	$(37,29_{10}-11,01_2)+73,45_{16}+(A4,6C_{16}+11,01_2)-(37,29_{10}+A4,6C_{16})$	$73,45_{16}$
11	$(A4,6C_{16}-11,01_2)+(73,45_{16}+37,29_{10}-A4,6C_{16})+(11,01_2-73,45_{16})$	$37,29_{10}$
12	$(37,29_{10}+73,45_{16}-11,01_2)+(A4,6C_{16}-73,45_{16})+(11,01_2-37,29_{10})$	$A4,6C_{16}$
13	$(A9,B5_{16}-67,54_{16}+(98,23_{10}+101,011_2)-A9,B5_{16})+(67,54_{16}-98,23_{10})$	$101,011_2$
14	$(101,011_2-A9,B5_{16}+67,54_{16})+(98,23_{10}-101,011_2)+(A9,B5_{16}-98,23_{10})$	$67,54_{16}$
15	$(101,011_2-A9,B5_{16})+(A9,B5_{16}-67,54_{16}+98,23_{10})+(67,54_{16}-101,011_2)$	$98,23_{10}$
1	**2**	**3**

Репозиторий БГАТУ

16	$(98,23_{10}-101,011_2+67,54_{16})+(A9,B5_{16}-98,23_{10})+(101,011_2-67,54_{16})$	$A9,B5_{16}$
17	$(53,26_{16}+110,11_2)+(8C,A7_{16}-71,28_{10})+(71,28_{10}-53,26_{16})-8C,A7_{16}$	$110,11_2$
18	$(71,28_{10}-110,11_2)+53,26_{10}+(8C,A7_{16}+110,11_2)-(71,28_{10}+8C,A7_{16})$	$53,26_{16}$
19	$(8C,A7_{16}-110,11_2)+(53,26_{16}+71,28_{10}-8C,A7_{16})+(110,11_2-53,26_{16})$	$71,28_{10}$
20	$(71,28_{10}+53,26_{16}-110,11_2)+(8C,A7_{16}-53,26_{16})+(110,11_2-71,28_{10})$	$8C,A7_{16}$
21	$(BC,AE_{16}-72,15_{16})+(111,101_2+24,58_{10}-BC,AE_{16})+(72,15_{16}-24,58_{10})$	$111,101_2$
22	$(111,101_2-BC,AE_{16}+72,15_{16})+(24,58_{10}-111,101_2)+(BC,AE_{16}-24,58_{10})$	$72,15_{16}$
23	$(111,101_2-BC,AE_{16})+(BC,AE_{16}-72,15_{16}+24,58_{10})+(72,15_{16}-111,101_2)$	$24,58_{10}$
24	$(24,58_{10}-111,101_2+72,15_{16})+(BC,AE_{16}-24,58_{10})+(111,101_2-72,15_{16})$	BC,AE_{16}
25	$(72,15_{16}+100,001_2)+(BC,AE_{16}-24,58_{10})+(24,58_{10}-72,15_{16}-BC,AE_{16})$	$100,001_2$
26	$(89,46_{10}-100,01_2)+54,72_{16}+(CE,AB_{16}+100,01_2)-(89,46_{10}+CE,AB_{16})$	$54,72_{16}$
27	$(CE,AB_{16}-100,001_2)+(54,72_{16}+89,46_{10}-CE,AB_{16})+(100,001_2-54,72_{16})$	$89,46_{10}$
28	$(89,46_{10}+54,72_{16}-100,001_2)+(CE,AB_{16}-54,72_{16})+(100,001_2-89,46_{10})$	CE,AB_{16}
29	$(E7,6A_{16}-54,72_{16})+(100,001_2+63,51_{10}-E7,6A_{16})+(54,72_{16}-100,001_2)$	$63,51_{10}$
30	$(89,46_{10}-100,001_2+54,72_{16})+(E7,6A_{16}-89,46_{10})+(100,001_2-54,72_{16})$	$E7,6A_{16}$

Репозиторий БГАТУ

Лабораторная работа №1

ПРОГРАММИРОВАНИЕ ЛИНЕЙНЫХ АЛГОРИТМОВ

Цель лабораторной работы: изучить основные элементы визуальной среды, освоить использование простейших компонентов DELPHI для ввода/вывода данных, и создать приложение, которое использует линейный алгоритм.

1.1.Визуальная среда DELPHI

При запуске DELPHI на экране появляется панель интерфейса, показанная на рис. 1.1. Среда DELPHI визуально реализуется в виде нескольких одновременно раскрытых на экране монитора окон. Количество, расположение, размер и вид окон может изменяться программистом.

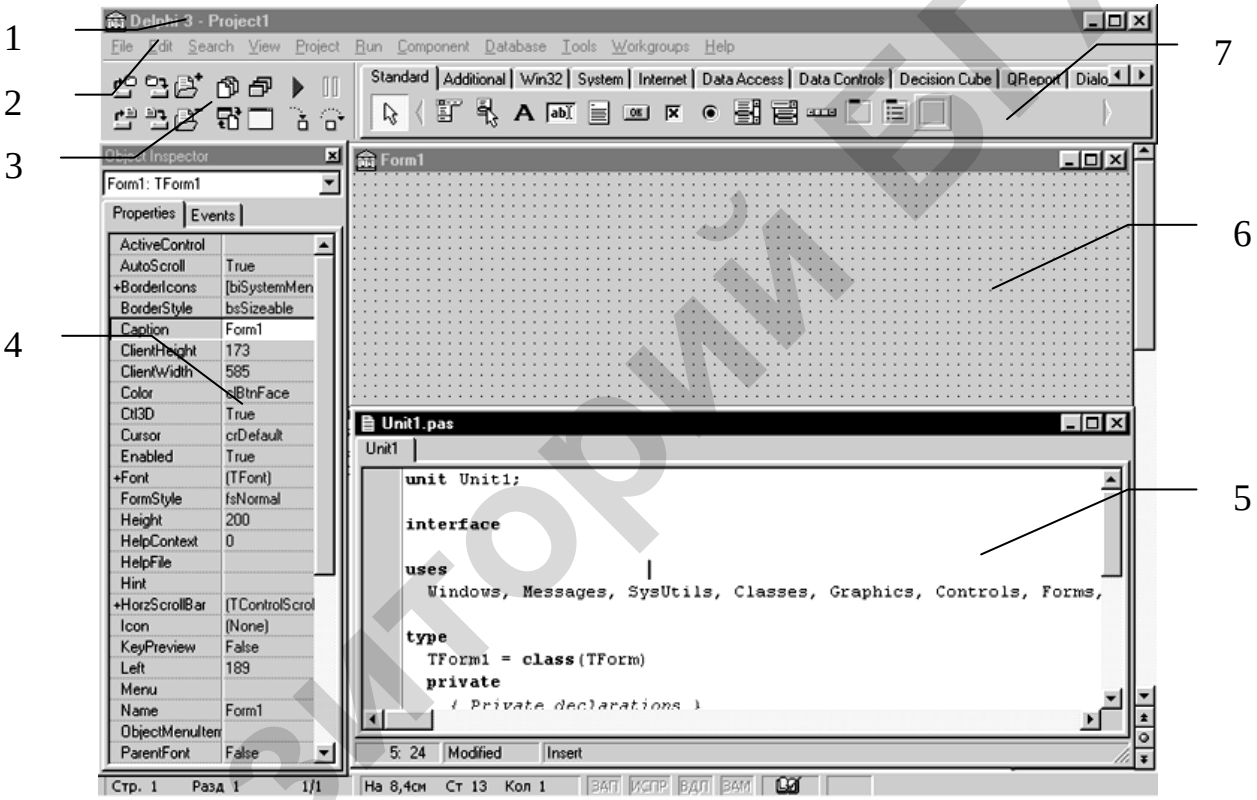


Рис.2.1

1- *главное окно*; 2 – *главное меню*, 3 – *пиктограммы главного меню*,
4 - окно *Инспектора Объектов*; 5 – окно *Редактора Кода*,
6- окно пустой *Формы*; 7 – *Палитра Компонентов*.

Главное окно всегда присутствует на экране и предназначено для управления процессом создания приложения.

Главное меню содержит все необходимые средства для управления проектом.

Пиктограммы главного меню облегчают доступ к наиболее часто применяемым командам. Команды главного меню приведены в приложении 1.

Палитра Компонентов обеспечивает доступ к набору библиотечных программ среды DELPHI, которые описывают некоторый элемент (компонент), помещенный программистом в окно Формы. Каждый компонент имеет

Репозиторий БГАТУ

определенный набор свойств, которые программист может выбирать и изменять по своему усмотрению. Например, заголовок окна, надпись на кнопке, размер, цвет и тип шрифта и др. Свойства компонентов приведены в приложении 2.

Окно Инспектора Объектов предназначено для изменения свойств выбранных компонентов и состоит из двух страниц. Страница *Properties* (Свойства) предназначена для изменения необходимых свойств компонента. Страница *Events* (События) – для определения реакции компонента на то или иное событие (например, щелчок кнопки “мыши ”).

Окно Формы представляет собой интерфейс проектируемого Windows-приложения. В это окно на этапе проектирования приложения помещаются необходимые компоненты, которые разработчик берет из Палитры Компонентов. Каждой Форме проекта соответствует модуль (**Unit**), текст которого на языке Object Pascal размещается в окне Редактора Кода.

Окно Редактора Кода предназначено для просмотра, создания и редактирования текстов модулей проекта. При первоначальной загрузке в окне Редактора Кода находится текст модуля, содержащий минимальный набор операторов для нормального функционирования пустой Формы в качестве Windows-приложения. При размещении некоторого компонента в окне Формы, текст модуля автоматически дополняется необходимыми операторами.

Обо всех происходящих в системе событиях, таких как создание Формы, нажатие кнопки мыши или клавиатуры и т.д., ядро Windows информирует окна путем посылки соответствующих сообщений. Среда DELPHI принимает и обрабатывает сообщения с помощью обработчиков событий (например, щелчок кнопки “мыши” – событие OnClick, создание Формы – OnCreate). Наиболее часто применяемые события представлены в табл. 1.1.

Таблица 1.1

Событие	**Описание события**
OnActivate	Возникает при активизации Формы
OnCreate	Возникает при создании Формы. В обработчике данного события следует задавать действия, которые должны происходить в момент создания Формы, например установка начальных значений.
OnClick	Возникает при нажатии кнопки мыши в области компонента.
OnDblClick	Возникает при двойном нажатии кнопки мыши в области компонента
OnKeyPress	Возникает при нажатии клавиши на клавиатуре. Параметр Key имеет тип Char и содержит ASCII-код нажатой клавиши (клавиша Enter клавиатуры имеет код #13, клавиша Esc - #27 и т.д.). Обычно это событие используется в том случае, когда необходима реакция на нажатие одной из клавиш.
OnKeyDown	Возникает при нажатии клавиши на клавиатуре. Обработчик этого события получает информацию о нажатой клавише и состоянии клавиш Shift, Alt и Ctlr, а также о нажатой кнопке мыши.

Репозиторий БГАТУ

Для создания обработчика события программисту необходимо раскрыть список компонентов в верхней части окна Инспектора Объектов и выбрать необходимый компонент. Затем, на странице Events Инспектора Объектов, нажатием левой клавиши мыши выбрать название обработчика и дважды щелкнуть по его правой (белой) части. В ответ DELPHI активизирует окно Редактора Кода модуля и покажет заготовку процедуры обработки выбранного события. Для каждого обрабатываемого события в тексте модуля организуется процедура (***procedure***), между ключевыми словами ***begin*** и ***end*** которой программист на языке Object Pascal записывает требуемый алгоритм обработки события.

Переключение между окном Формы и окном Редактора Кода осуществляется кнопкой главного меню с пиктограммой  или клавишей F12.

1.2. Пример создания приложения

Задание: создать Windows-приложение для вычисления выражения

$$u = tg^5(\sqrt{x} - y^3) + e^{y/z} \cdot \sin z^2 \quad .$$

Численные значения данных *x*, *y* и *z* занести с клавиатуры в соответствующие поля панели интерфейса. Один из возможных вариантов панели интерфейса создаваемого приложения показан на рис. 1.2.

1.2.1. Сохранение проекта

В процессе проектирования приложения DELPHI создает несколько файлов - *проект.* Каждый проект целесообразно хранить в отдельной, заранее созданной папке. С помощью подходящего приложения Windows создадим папку и назовем ее, например, LAB1.

Для сохранения проекта откройте в главном меню пункт File и щелкните “мышью” на опции Save Project As…(Сохранить проект как…). Сначала DELPHI откроет панель диалога Save Unit1 As (Сохранить модуль как) для сохранения модуля проекта. В этой панели найдем созданную папку LAB1 и сохраним в ней модуль под именем, например, UnLinAlg. Обратите внимание на то, что DELPHI по умолчанию присвоит этому файлу тип Delphi unit с расширением *.pas. Затем откроется панель диалога Save Project1 As . Назовем наш проект, например, PrLinAlg и сохраним его в этой же папке. Здесь DELPHI даст файлу тип Delphi project и расширение *.dpr. Убедитесь в том, что главное окно DELPHI теперь называется PrLinAlg, окно главного файла проекта- PrLinAlg.dpr, а окно модуля проекта- UnLinAlg.pas.

Старайтесь давать файлам осмысленные имена вместо однообразных Unit1 и Project1, предлагаемых DELPHI.

Чтобы избежать потери файлов проекта в аварийных ситуациях связанных, например, с выключениием питания, зависании системы и т.д., рекомендуется периодически сохранять проект, используя пиктограмму  главного меню или опцию Save All в меню File.

Репозиторий БГАТУ

1.2.2. Настройка окон

Чтобы работать с окном, необходимо сделать его активным, щелкнув “мышью” в любом месте окна. У активного окна заголовок становится выделенным, например, на рис. 1.1 активным является окно Редактора Кода.

Окна Формы и Редактора Кода модуля в правом верхнем углу имеют кнопки управления, которые предназначены: -для свертывания окна в пиктограмму, -для разворачивания окна на весь экран и возвращения к исходному размеру , -для закрытия окна.

С помощью “мыши”, захватывая одну из кромок окна или выделенную строку заголовка, отрегулируете нужные размеры окон Формы, Редактора Кода, Инспектора Объектов и их положение на экране.

1.2.3. Изменение заголовка Формы

Новая Форма имеет одинаковые имя (Name) и заголовок (Caption) - FORM1. Начинающим программистам имя Формы менять не рекомендуется, т.к. оно используется в тексте модуля.

Для изменения заголовка активизируйте окно Инспектора Объектов и на странице Properties в свойстве Caption замените заголовок **Form1** на **Лаб. работа №1-Линейный алгоритм**. Убедитесь, что одновременно изменился заголовок окна Формы.

1.2.4. Размещение компонентов на Форме

Будем размещать компоненты на Форме так, чтобы они соответствовали панели, показанной на рис 1.2.

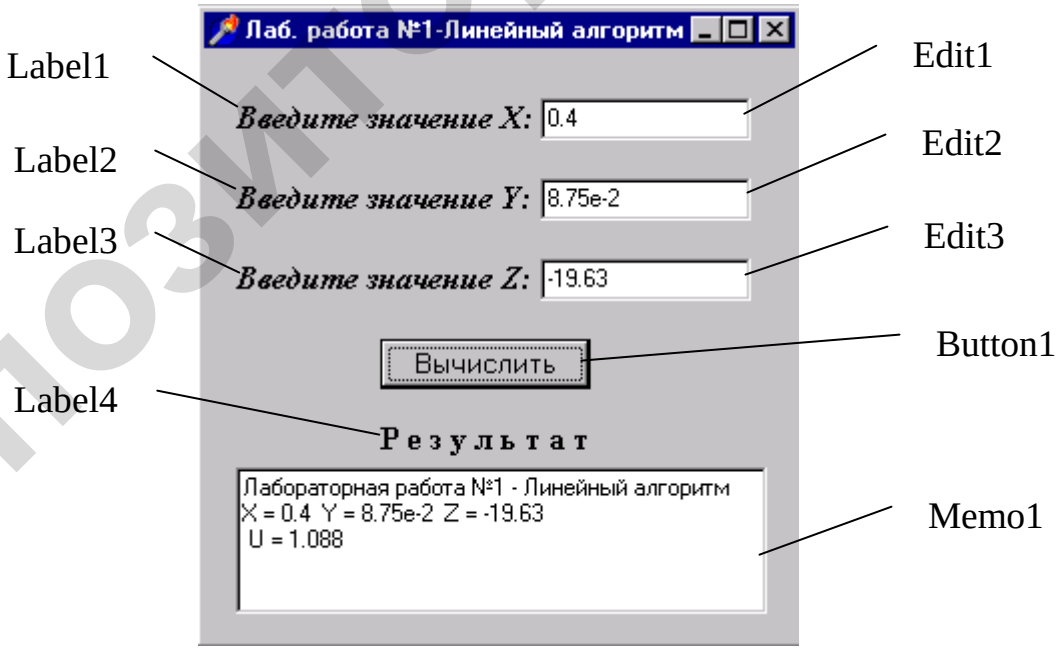


Рис.1.2

Репозиторий БГАТУ

Для нанесения надписей на Форму используется компонент Label. Выберите в Палитре Компонентов на странице Standard пиктограмму  компонента Label и щелкните на ней “мышью”. Затем в нужном месте Формы щелкните “мышью” - появится надпись Label1. В свойстве Caption Инспектора Объектов замените надпись **Label1** на ***Введите значение X:***. В свойстве Font подберите шрифт. Аналогично нанесите на Форму остальные надписи. Щелкнув “мышью” на любом из размещенных компонентов, отрегулируйте его местоположение на Форме и размер.

Для ввода/вывода данных в простейших случаях используются компоненты Edit и Memo. Компонент Edit применяется в тех случаях, когда данные представляются одной строкой. Если данные представляют собой несколько строк, то используется компонент Memo.

Для создания полей ввода численных значений переменных **x, y** и **z** используем компонент Edit. Выберите в Палитре Компонентов на странице Standard пиктограмму  и разместите компонент Edit в нужных местах Формы так же, как Вы это делали с компонентом Label.

Для вывода результатов используем компонент Memo. Выберите в Палитре Компонентов на странице Standard пиктограмму  , поместите компонент Memo на Форму и откорректируйте его местоположение и размеры.

1.2.5. Написание процедуры обработки события создания Формы (FormCreate)

Если программист желает, чтобы при появлении панели интерфейса на экране в соответствующих полях находились начальные значения данных, он должен учесть, что при запуске приложения возникает событие - создание Формы (OnCreate). Создадим процедуру обработки этого события, которая занесет начальные значения переменных x, y, z в поля Edit1, Edit2 и Edit3 соответственно, а в поле Memo1 поместит строку **Лабораторная работа №1 – Линейный алгоритм**. Для этого дважды щелкните мышью на любом свободном месте Формы. На экране появится текст модуля UnLinAlg, в котором DELPHI автоматически создает заготовку процедуры-обработчика события создания Формы: **Procedure TForm1.FormCreate(Sender:TObject)**. Между операторами **begin** и **end** этой процедуры вставьте операторы, которые выполняют необходимые действия (текст модуля приведен в п.1.2.7).

1.2.6. Написание процедуры обработки события нажатия кнопки Button1 (Button1Click)

Поместим на Форму кнопку, нажатие которой приведет к вычислению выражения. Выберите в Палитре Компонентов на странице Standart пиктограмму  компонента Button. В свойстве Caption Инспектора Объектов замените надпись на кнопке **Button1** на **Вычислить**. В свойстве Font подберите шрифт. Отрегулируйте положение и размер кнопки. Затем дважды щелкните

Репозиторий БГАТУ

“мышью” на кнопке, после чего курсор установится в тексте процедуры-обработчика события нажатия кнопки Button1 : **Procedure TForm1.Button1Click(Sender:TObject**). Внимательно наберите операторы этой процедуры, используя текст модуля UnLinAlg.

1.2.7. Текст модуля UnLinAlg

```
Unit UnLinAlg;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls;

type
  TForm1 = class(TForm)
    Label1: TLabel;
    Edit1: TEdit;
    Label2: TLabel;
    Edit2: TEdit;
    Label3: TLabel;
    Edit3: TEdit;
    Label4: TLabel;
    Memo1: TMemo;
    Button1: TButton;
    procedure FormCreate(Sender: TObject);
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

{$R *.DFM}
// Процедура обработки события создания Формы:
procedure TForm1.FormCreate(Sender: TObject);
begin
 Edit1.Text:='0.4';          // начальное значение X
 Edit2.Text:='8.75e-2';    // начальное значение Y
```

Репозиторий БГАТУ

```
 Edit3.Text:='-19.63';     // начальное значение Z
 Memo1.Clear;          // очистка  Memo1
// Вывод строки в  Memo1:
 Memo1.Lines.Add('Лабораторная работа №1 - Линейный алгоритм');
end;
// Процедура обработки события нажатия кнопки Button1:
procedure TForm1.Button1Click(Sender: TObject);
 var
  x,y,z,a,b,c,u : extended;       // объявление локальных переменных
begin
 x:=StrToFloat(Edit1.Text);  // X присваивается содержимое Edit1
 y:=StrToFloat(Edit2.Text);  // Y присваивается содержимое Edit2
 z:=StrToFloat(Edit3.Text);  // Z присваивается содержимое Edit3
// Вычисляется  выражение
    a:=sqrt(x)-y*y*y;
    b:=sin(a)/cos(a);
    c:=Exp(5*Ln(b));
    u:=c+exp(y/z)*sin(z*z);
 Memo1.Lines.Add('X = '+Edit1.Text+'   Y = '+Edit2.Text+
  '   Z = '+Edit3.Text);         // контрольный вывод X, Y, Z в Memo1
 // Вывод результата в Memo1:
 Memo1.Lines.Add(' U = '+FloatToStrF(u,ffFixed,8,3));
end;

end.
```

Данные, с которыми работают компоненты Edit и Memo, имеют тип String. Поэтому в процедуре TForm1.Button1Click при присваивании содержимого полей Edit1,Edit2,Edit3 переменным X,Y,Z с помощью функции StrToFloat осуществляется преобразование данных типа String в действительные значения с плавающей точкой типа Extended. Если необходимо работать с данными целого типа, используется функция StrToInt.

При выводе данных в Memo1 используется метод Add свойства Lines, причем для преобразования данных из действительного значения в строковое и управления формой представления выводимого результата используется функция FloatToStrF.

Подробный перечень процедур и функций для работы со строками приведен в приложении 3.

1.2.8. Работа с приложением

Для запуска созданного приложения нажмите пиктограмму  главного меню или клавишу F9. При этом происходит компиляция модулей и, если нет ошибок, компоновка проекта и создание выполняемого файла PrLinAlg.exe. На экране появляется панель интерфейса приложения (рис.1.2).

Щелкните “мышью” на кнопке “**Вычислить**” и в поле Memo1 появляется результат. Измените исходные значения x, y, z в полях Edit и снова нажмите

Репозиторий БГАТУ

кнопку ”**Вычислить**”. Убедитесь, что в поле Memo1 отображаются новые результаты. Завершить работу приложения можно нажатием кнопки  в правом верхнем углу панели интерфейса.

В случае нештатного функционирования приложения восстановить первоначальный режим работы с проектом можно путем выбора в меню Run опции ProgramReset или нажать клавиши Ctrl+F2.

1.3. Выполнение индивидуального задания

По указанию преподавателя выберите свое индивидуальное задание. Уточните условие задания, количество и типы исходных данных. В соответствии с этим оформите дизайн панели интерфейса проектируемого приложения, установите необходимое количество полей Edit, тексты заголовков на Форме, размеры шрифтов, а также типы переменных и функции преобразования при вводе и выводе результатов.

Изучите в приложении 1 описание опций главного меню File, Edit, Run, а в приложении 2 описание компонентов Label, Edit, Memo, Button.

Индивидуальные задания

1. $t = \dfrac{2\cos\left(x - \dfrac{\pi}{6}\right)}{0.5 + \sin^2 y}\left(1 + \dfrac{z^2}{3 - z^2/5}\right).$

   При x=14.26, y=-1.22, z=3.5×10^{-2} t=0.564849.

2. $u = \dfrac{\sqrt[3]{8 + |x-y|^2 + 1}}{x^2 + y^2 + 2} - e^{|x-y|}\left(tg^2 z + 1\right)^x.$

   При x=-4.5, y=0.75×10^{-4},z=0.845×10^{2} u=-55.6848.

3. $v = \dfrac{1 + \sin^2(x+y)}{\left|x - \dfrac{2y}{1 + x^2 y^2}\right|} x^{|y|} + \cos^2\left(arctg\,\dfrac{1}{z}\right).$

   При x=3.74×10^{-2}, y=-0.825, z=0.16×10^{2} v=1.0553.

4. $w = \left|\cos x - \cos y\right|^{\left(1+2\sin^2 y\right)}\left(1 + z + \dfrac{z^2}{2} + \dfrac{z^3}{3} + \dfrac{z^4}{4}\right).$

   При x=0.4×10^{4} ,y=-0.875, z=-0.475×10^{-3} w=1.9873.

5. $\alpha = \ln\left(y^{-\sqrt{|x|}}\right)\left(x - \dfrac{y}{2}\right) + \sin^2 arctg(z).$

   При x=-15.246, y=4.642×10^{-2}, z=20.001×10^{2} α=-182.036.

Репозиторий БГАТУ

6. $\beta = \sqrt{10\left(\sqrt[3]{x} + x^{y+2}\right)}\left(\arcsin^2 z - |x-y|\right)$.

При x=16.55×10^{-3}, y=-2.75, z=0.15 β=-40.631.

7. $\gamma = 5arctg(x) - \frac{1}{4}\arccos(x)\frac{x+3|x-y|+x^2}{|x-y|z+x^2}$.

При x=-0.1722, y=6.33, z=3.25×10^{-4} γ=-9.0467.

8. $\varphi = \frac{e^{|x-y|}|x-y|^{x+y}}{arctg(x)+arctg(z)} + \sqrt[3]{x^6 + \ln^2 y}$.

При x=-2.235×10^{-2}, y=2.23, z=15.221 φ=39.374.

9. $\psi = \left|x^{\frac{y}{x}} - \sqrt[3]{\frac{y}{x}}\right| + (y-x)\frac{\cos y - z/(y-x)}{1+(y-x)^2}$.

При x=1.825×10^{2}, y=18.225, z=-3.298×10^{-2} ψ=1.2131.

10. $a = 2^{-x}\sqrt{x + \sqrt[4]{|y|}}\sqrt[3]{e^{x-1/\sin z}}$.

При x=3.981×10^{-2}, y=-1.625×10^{3}, z=0.512 a=1.26185.

11. $b = y^{\sqrt[3]{|x|}} + \cos^3(y)\frac{|x-y|\left(1+\frac{\sin^2 z}{\sqrt{x+y}}\right)}{e^{|x-y|}+\frac{x}{2}}$.

При x=6.251, y=0.827, z=25.001 b=0.7121.

12. $c = 2^{(y^x)} + \left(3^x\right)^y - \frac{y\left(arctgz - \frac{\pi}{6}\right)}{|x| + \frac{1}{y^2+1}}$.

При x=3.251, y=0.325, z=0.466×10^{-4} c=4.2514.

13. $f = \frac{\sqrt[4]{y+\sqrt[3]{x-1}}}{|x-y|\left(\sin^2 z + tgz\right)}$.

При x=17.421, y=10.365×10^{-3}, z=0.828×10^{5} f=0.33056.

14. $g = \frac{y^{x+1}}{\sqrt[3]{|y-2|}+3} + \frac{x+y/2}{2|x+y|}(x+1)^{-1/\sin z}$.

При x=12.3×10^{-1}, y=15.4, z=0.252×10^{3} g=82.8257.

Репозиторий БГАТУ

15. $h = \dfrac{x^{y+1} + e^{y-1}}{1 + x|y - tgz|}\left(1 + |y - x|\right) + \dfrac{|y-x|^2}{2} - \dfrac{|y-x|^3}{3}$.

При x=2.444, y=0.869×10^{-2}, z=-0.13×10^{3} h = -0.49871.

16. Вывести на экран *1* или *0* в зависимости от того, имеют ли три заданных целых числа одинаковую четность или нет.
17. Найти сумму цифр заданного четырехзначного числа.
18. Определить число, полученное выписыванием в обратном порядке цифр заданного трехзначного числа.
19. Вывести на экран *1* или *0* в зависимости от того, равна ли сумма двух первых цифр заданного четырехзначного числа сумме двух его последних цифр.
20. Вывести на экран *1* или *0* в зависимости от того, равен ли квадрат заданного трехзначного числа кубу суммы цифр этого числа.
21. Вывести на экран *1* или *0* в зависимости от того, есть ли среди первых трех цифр дробной части заданного положительного вещественного числа цифра ноль.
22. Вывести на экран *1* или *0* в зависимости от того, есть ли среди цифр заданного трехзначного числа одинаковые.
23. Присвоить целой переменной *k* третью от конца цифру в записи положительного целого числа *n*.
24. Присвоить целой переменной *k* первую цифру из дробной части положительного вещественного числа.
25. Целой переменной S присвоить сумму цифр трехзначного целого числа *k*.
26. Идет *k*-я секунда суток. Определить, сколько полных часов (*h*) и полных минут (*m*) прошло к этому моменту.
27. Определить *f* – угол (в градусах) между положением часовой стрелки в начале суток и ее положением в *h* часов, *m* минут и *s* секунд (*0<=h<=11, 0<=m, s<=59*).
28. Определить *h* – полное количество часов и *m* – полное количество минут, прошедших от начала суток до того момента (в первой половине дня), когда часовая стрелка повернулась на *f* градусов(*0<=f<360, f – вещественное число*).
29. Пусть *k* – целое от 1 до 365. Присвоить целой переменной *n* значение 1,2,…6 или 7 в зависимости от того, на какой день недели (понедельник, вторник,…, субботу или воскресенье) приходится *k*- й день невисокосного года, в котором 1 января - понедельник.
30. Поменять местами значения целых переменных *x* и *y*, не используя дополнительные переменные.

Репозиторий БГАТУ

Лабораторная работа №2

ПРОГРАММИРОВАНИЕ РАЗВЕТВЛЯЮЩИХСЯ АЛГОРИТМОВ

Цель лабораторной работы: освоить использование простейших компонентов-переключателей и создать приложение, которое использует разветвляющийся алгоритм.

2.1. Пример создания приложения

Задание: создать Windows-приложение для вычисления выражения

$$Z=\begin{cases} f(x), x < y \\ y, иначе \end{cases}, \quad \text{где} \quad f(x)=\begin{cases} \sin(x) \\ \cos(x) \end{cases}$$ по желанию пользователя. В панели интерфейса предусмотреть возможность управления контрольным выводом исходных данных.

Один из возможных вариантов панели интерфейса создаваемого приложения показан на рис. 2.1.

2.1.1. Размещение компонентов на Форме

Будем размещать компоненты на Форме так, чтобы они соответствовали панели, показанной на рис 2.1.

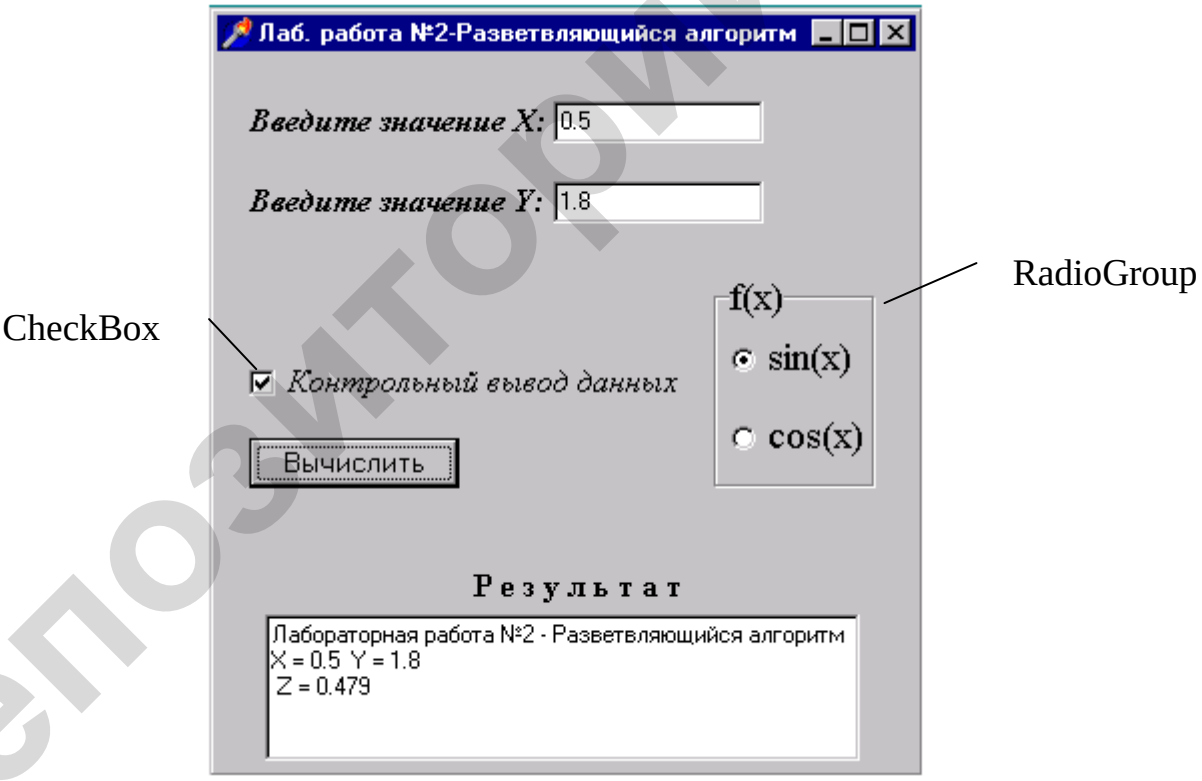


Рис. 2.1

При создании приложений в DELPHI часто используются компоненты в виде кнопок-переключателей. Состояние такой кнопки (включено - выключено) визуально отражается на Форме. На панели (рис. 2.1) представлены кнопки-переключатели двух типов: CheckBox и RadioGroup .

Репозиторий БГАТУ

Компонент **CheckBox** организует кнопку независимого переключателя, с помощью которой пользователь может указать свое решение типа “да/нет”.

Компонент **RadioGroup** организует *группу кнопок* - зависимых переключателей. При нажатии одной из кнопок группы все остальные кнопки выключаются.

Поместите на Форму компоненты Label, Edit и Memo в соответствии с рис.2.1. Выберите в Палитре Компонентов на странице Standard пиктограмму  компонента CheckBox и разместите ее в нужном месте Формы. . В свойстве Caption Инспектора Объектов замените надпись **CheckBox1** на ***Контрольный вывод данных***. . Чтобы при запуске приложения кнопка CheckBox оказалась включена, свойство Checked установите равным True.

Выберите в Палитре Компонентов Standard пиктограмму  компонента RadioGroup и поместите ее в нужное место Формы. В свойстве Caption измените заголовок **RadioGroup1** на **f(x)**. Для размещения кнопок в один столбец, свойство Columns установите равным 1. Дважды щелкните “мышью” по правой части свойства Items - появится строчный редактор списка наименований кнопок. Наберите 2 строки с именами: в первой строке - sin(x), во второй - cos(x) и нажмите ОК. После этого на Форме появится группа из двух кнопок - переключателей с соответствующими надписями. Чтобы при запуске приложения первая кнопка RadioGroup оказалась включена, свойство ItemIndex установите равным 0.

2.1.2. Создание процедур обработки событий FormCreate и Button1Click

Технология создания процедур обработки событий FormCreate и Button1Click ничем не отличается от предыдущей работы. Внимательно наберите операторы этих процедур, используя текст модуля UnRazvAlg.

2.1.3. Текст модуля UnRazvAlg

```
Unit UnRazvAlg;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, ExtCtrls;

type
  TForm1 = class(TForm)
    Label1: TLabel;
    Edit1: TEdit;
    Label2: TLabel;
    Edit2: TEdit;
    Label4: TLabel;
    Memo1: TMemo;
```

Репозиторий БГАТУ

```
    Button1: TButton;
    RadioGroup1: TRadioGroup;
    CheckBox1: TCheckBox;
    procedure FormCreate(Sender: TObject);
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation
{$R *.DFM}
// Процедура обработки события создания Формы:
procedure TForm1.FormCreate(Sender: TObject);
begin
 Edit1.Text:='0.5';     // начальное значение X
 Edit2.Text:='1.8';     // начальное значение Y
 Memo1.Clear;           // очистка  Memo1
// Вывод строки в  Memo1:
 Memo1.Lines.Add('Лабораторная работа №2 - Разветвляющийся алгоритм');
end;
// Процедура обработки события нажатия кнопки Button1:
procedure TForm1.Button1Click(Sender: TObject);
 var
  x,y,z,fx : extended; // объявление локальных переменных
begin
 x:=StrToFloat(Edit1.Text);  // X присваивается содержимое Edit1
 y:=StrToFloat(Edit2.Text);  // Y присваивается содержимое Edit2
 fx:=sin(x);                 // fx присваивается начальное значение
// Выбор функции, соответствующей нажатой кнопке:
 case RadioGroup1.ItemIndex of
  0:fx:=sin(x);
  1:fx:=cos(x);
 end;
// Вычисление выражения:
 if x<y then
        z:=fx
        else
         z:=y;
 // Проверка состояния кнопки CheckBox1:
```

Репозиторий БГАТУ

```
 if CheckBox1.Checked then
                Memo1.Lines.Add('X = '+Edit1.Text+
  '  Y = '+Edit2.Text);        // контрольный вывод X, Y, Z в Memo1
 // Вывод результата в Memo1:
 Memo1.Lines.Add(' Z = '+FloatToStrF(z,ffFixed,8,3));
end;
end.
```

Если нажата первая кнопка RadioGroup1, в переменную целого типа RadioGroup1.ItemIndex заносится нуль, если вторая – единица. Если кнопка CheckBox1 нажата, логическая переменная CheckBox1.Checked имеет значение True, если нет – False.

2.1.4. Работа с приложением

Запустите созданное приложение. Используя все управляющие компоненты панели интерфейса, убедитесь в правильном функционировании приложения во всех предусмотренных режимах работы.

2.2. Выполнение индивидуального задания

По указанию преподавателя выберите свое индивидуальное задание. Изучите в приложении 2 описание компонентов CheckBox и RadioGroup. Создайте приложение и протестируйте его работу.

Индивидуальные задания

Для заданий №1-№15 на панели интерфейса предусмотреть возможность выбора одной из трех функций f(x): sh(x), x^2, e^X.

1. $$a=\begin{cases}(f(x)+y)^2-\sqrt{f(x)y}, & x\le 0\\ (f(x)+y)^3+\sqrt{|f(x)y|}, & 0<x\le 1\\ (f(x)+y)^4+1, & иначе.\end{cases}$$

2. $$b=\begin{cases}\ln(f(x))+(f(x)^2+y)^3, & x/y\rangle 0\\ \ln|f(x)/y|+(f(x)+y)^2, & x/y\langle 0\\ (f(x)^3+y)^2, & -2\le x<2\\ 0, & иначе.\end{cases}$$

3. $$c=\begin{cases}f(x)^2+y^2+\sin(y), & x+y>0\\ (f(x)-y^3)^2+\cos(y), & x\rangle 0, y<0\\ (y-f(x))^2+tg(y), & x-y\langle 0.\end{cases}$$

4. $$d=\begin{cases}f^2(x)+arctg(f(x)), & 1\le x<5\\ (y-f(x))^2+arctg(f(x)), & y\rangle x\\ (y+f(x))^3+0.5, & иначе.\end{cases}$$

5. $$e=\begin{cases}i\sqrt{f(x)}, & i-нечетное, x\rangle 0\\ i/2\sqrt{|f(x)|}, & i-четное, x\langle 0\\ \sqrt{|if(x)|}, & иначе.\end{cases}$$

6. $$g=\begin{cases}e^{f(x)-|b|}, & 0.5\langle xb\langle 10\\ \sqrt{|f(x)+b|}, & 0.1\langle xb\langle 0.5\\ 2f(x)^2, & иначе.\end{cases}$$

Репозиторий БГАТУ

7. $s=\begin{cases} e^{f(x)}, & 1\langle xb\langle 10 \\ \sqrt{|f(x)+4*b|}, & 12\langle xb\langle 40 \\ bf(x)^2, & \textit{иначе.} \end{cases}$

8. $j=\begin{cases} \sin\left(5f(x)+3m|f(x)|\right), & -1\langle m\langle x \\ \cos\left(3f(x)+5m|f(x)|\right), & x\le m \\ \left(f(x)+m\right)^2, & \textit{иначе.} \end{cases}$

9. $l=\begin{cases} 2f(x)^3+3p^2, & x\rangle|p| \\ |f(x)-p|, & 3\langle x\langle|p| \\ \left(f(x)-p\right)^2, & \textit{иначе.} \end{cases}$

10. $k=\begin{cases} \ln\left(|f(x)|+|q|\right), & |xq|\rangle 10 \\ e^{f(x)+q}, & |xq|\langle 10 \\ f(x)+q, & \textit{иначе.} \end{cases}$

11. $m=\dfrac{\max\left(f(x),y,z\right)}{\min\left(f(x),y\right)}+5.$

12. $n=\dfrac{\min\left(f(x)+y,y-z\right)}{\max\left(f(x),y\right)}.$

13. $p=\dfrac{|\min\left(f(x),y\right)-\max\left(y,z\right)|}{2}.$

14. $q=\dfrac{\max\left(f(x)+y+z,xyz\right)}{\min\left(f(x)+y+z,xyz\right)}.$

15. $r=\max\left(\min\left(f(x),y\right),z\right).$

16. Известно, что из четырех чисел a_1, a_2, a_3 и a_4 одно отлично от трех других, равных между собой. Присвоить номер этого числа переменной *n*.
17. По номеру *n* (*n>0*) некоторого года определить *c* – номер его столетия (учесть, что, к примеру, началом XX столетия был 1901, а не 1900 год!).
18. Значения переменных *a, b* и *c* поменять местами так, чтобы оказалось a<=b<=c.
19. Дано целое *k* от 1 до 180. Определить, какая цифра находится в *k*-й позиции последовательности 10111213…9899 , в которой выписаны подряд все двузначные числа.
20. Дано натуральное *k*. Определить *k*–ю цифру в последовательности 110100100010000100000…, в которой выписаны подряд степени 10.
21. В старояпонском календаре был принят 60-летний цикл, состоявший из пяти 12-летних подциклов. Подциклы обозначались названиями цвета: green (зеленый), red (красный), yellow (желтый), white (белый) и black (черный). Внутри каждого подцикла годы носили названия животных: крысы, коровы, тигра, зайца, дракона, змеи, лошади, овцы, обезьяны, курицы, собаки и свиньи. (1984 год – *год зеленой крысы* –был началом очередного цикла). Разработать программу, которая вводит номер некоторого года нашей эры и выводит его название по старояпонскому календарю.
22. Если сумма трех попарно различных действительных чисел *x, y, z* меньше единицы, то наименьшее из этих трех чисел заменить полусуммой двух других; в противном случае заменить меньшее из *x* и *y* полусуммой двух оставшихся значений.
23. Для целого числа *k* от 1 до 99 вывести фразу “мне *k* лет”, учитывая при этом, что при некоторых значениях *k* слово “лет” надо заменить на слово “год” или “года”.
24. Даны три действительных числа. Вывести из них те, которые принадлежат интервалу (1,3).

Репозиторий БГАТУ

25. **Type**
курс=(С,В,Ю,З); *// север, восток, юг, запад*
Приказ=(вперед, вправо, назад, влево);
Var
K1,K2:курс;
ПР:приказ;
Корабль сначала шел по курсу **K1**, а затем его курс был изменен согласно приказу **ПР**. Определить **K2** - новый курс корабля.

26. **Type**
месяц=(январь,февраль,март,апрель,май,июнь,июль,август,
сентябрь,октябрь,ноябрь,декабрь);
день=1..31;
var
d1,d2:день;
m1,m2:месяц;
t:integer;
Переменной **t** присвоить значение **1** если дата **d1**, **m1** предшествует (в рамках года) дате **d2**, **m2**, и значение **0** в других случаях.

27. **Type**
нота=(до,ре,ми,фа,соль,ля,си);
интервал=(секунда,терция,кварта,квинта,секста,септима);
var
n1,n2:нота;
i:интервал;
Определить **i**-й интервал, образованный нотами **n1** и **n2(n1≠n2)**: секунда - это интервал из двух соседних (по кругу) нот (например, ре и ми, си и до), терция-интервал через ноту (например, фа и ля, си и ре) и т.д.

28. **Type**
единица=(дециметр,километр,метр,миллиметр,сантиметр);
длина=real;
Var
x : длина;
P : единица;
Значение переменной **x**, означающее некоторую длину в единицах **p**,заменить на величину этой же длины в метрах.

29. **Type**
сезон=(зима,весна,лето,осень);
Var
m:месяц; // определение «месяц» см. в 26
S:сезон;
Определить **s**-сезон, на который приходится месяц **m**.

30. **Var**
k:1..9;
Вывести значение переменной **k** римскими цифрами.

Репозиторий БГАТУ

Лабораторная работа №3

ПРОГРАММИРОВАНИЕ ЦИКЛИЧЕСКИХ АЛГОРИТМОВ

Цель лабораторной работы: освоить простейшие средства отладки модулей проекта и создать приложение, которое использует циклический алгоритм.

3.1. Отладка модулей проекта

Отладка представляет собой процесс обнаружения, локализации и устранения ошибок в проекте. Она занимает значительную часть рабочего времени программиста, нередко большую, чем разработка проекта.

Практически любой нетривиальный проект перед началом отладки содержит хотя бы одну синтаксическую или логическую ошибку.

3.1.1. Отладка синтаксических ошибок

Синтаксические ошибки состоят в нарушении формальных правил использования операторов. Эти ошибки появляются в результате недостаточного знания разработчиком языка программирования и невнимательности при наборе операторов на экране дисплея.

Поиск синтаксических ошибок в модулях проекта осуществляется компилятором. Чтобы дать программисту как можно больше информации об ошибках, допущенных в модуле, компилятор отмечает ошибки и продолжает работу до тех пор, пока не будут обработаны все операторы модуля. Следует иметь в виду, что:

1) компилятор распознает *не все ошибки*;
2) некоторые ошибки могут повлечь за собой то, что правильные операторы будут восприниматься компилятором как ошибочные, и наоборот – ошибочные операторы компилятор воспримет как верные;
3) ошибка в одном месте модуля может повлечь за собой серию диагностических сообщений компилятора в других местах модуля;
4) из-за некоторых ошибок компиляция модуля может вообще прекращаться и проверка последующих операторов не производится.

Информация обо всех ошибках, найденных в модуле, выводится в специальное окно, которое появляется в нижней части экрана. Каждая строка этого окна содержит имя файла, номер строки, в которой обнаружена ошибка и характер ошибки. Если дважды щелкнуть “мышью” на строке с описанием ошибки, курсор установится в той строке модуля, где обнаружена ошибка. Следует исправлять ошибки последовательно, сверху вниз и после исправления каждой ошибки компилировать программу заново. С целью сокращения времени компиляции рекомендуется осуществлять проверку наличия ошибок в режимах Syntax Check и Compile меню Project. Для получения более полной информации о характере ошибки можно обратится к HELP нажатием клавиши F1.

Отладка синтаксиса считается завершенной, когда после очередной компиляции в режиме Build All меню Project отсутствуют диагностические сообщения об ошибках.

Репозиторий БГАТУ

3.1.2. Отладка логических ошибок

Логические ошибки условно можно разделить на ошибки алгоритма и семантические ошибки. Причинами таких ошибок могут быть несоответствие алгоритма поставленной задаче, неправильное понимание программистом смысла (семантики) операторов языка программирования, нарушение допустимых пределов и правил представления данных, невнимательность при технической подготовке проекта к обработке на компьютере.

Для выявления ошибок служат *тесты*. Тест – это такой набор исходных данных, который дает результат, не вызывающий сомнений. Промежуточные и конечные результаты теста используются для контроля правильности выполнения приложения.

Составление тестов – непростая задача. Тесты должны быть с одной стороны, достаточно простыми, чтобы результат легко проверялся, с другой стороны – достаточно сложными, чтобы комплексно проверить алгоритм.

Тесты составляются по схеме алгоритма *до программирования*, так как составление тестов помогает выявить многие ошибки в алгоритмизации.

Количество тестов и их сложность зависят от алгоритма. Комплекс тестов должен быть таким, чтобы все ветви схемы алгоритма были пройдены, по крайней мере, по одному разу. Несовпадение результатов, выдаваемых приложением с результатами тестов – признак наличия ошибок. Эти ошибки проявляются в том, что результат расчета оказывается неверным либо происходит переполнение, деление на 0 и др.

Для локализации места ошибки рекомендуется поступать следующим образом. В окне Редактора Кода установите курсор в строке перед подозрительным участком и нажмите клавишу F4 (выполнить до курсора). Выполнение приложения будет остановлено на той строке модуля, в которой был установлен курсор. Текущее значение любой переменной можно увидеть, если накрыть курсором идентификатор переменной на 1-2 сек. Нажимая клавишу F8 (пошаговое выполнение), можно построчно выполнять программу, контролируя содержимое переменных и правильность вычислений.

3.2. Пример создания приложения

Задание: создать Windows-приложение, которое выводит таблицу значений функции $Y(x) = (1 - \frac{x^2}{2})\cos x - \frac{x}{2}\sin x$ и ее разложения в ряд в виде суммы

$S(x) = \sum_{n=0}^{n} (-1)^n \frac{2n^2 + 1}{(2n)!} x^{2n}$ для значений x от x_n до x_k с шагом $h=(x_k - x_n)/10$.

В панели интерфейса предусмотреть возможность управления выводом исходных данных и погрешности вычислений.

Один из возможных вариантов панели интерфейса создаваемого приложения показан на рис.3.1.

Репозиторий БГАТУ

3.2.1. Размещение компонентов на Форме

Вместо компонента Edit используем компонент SpinEdit, который обеспечивает отображение и редактирование целого числа с возможностью его изменения посредством двойной кнопки.

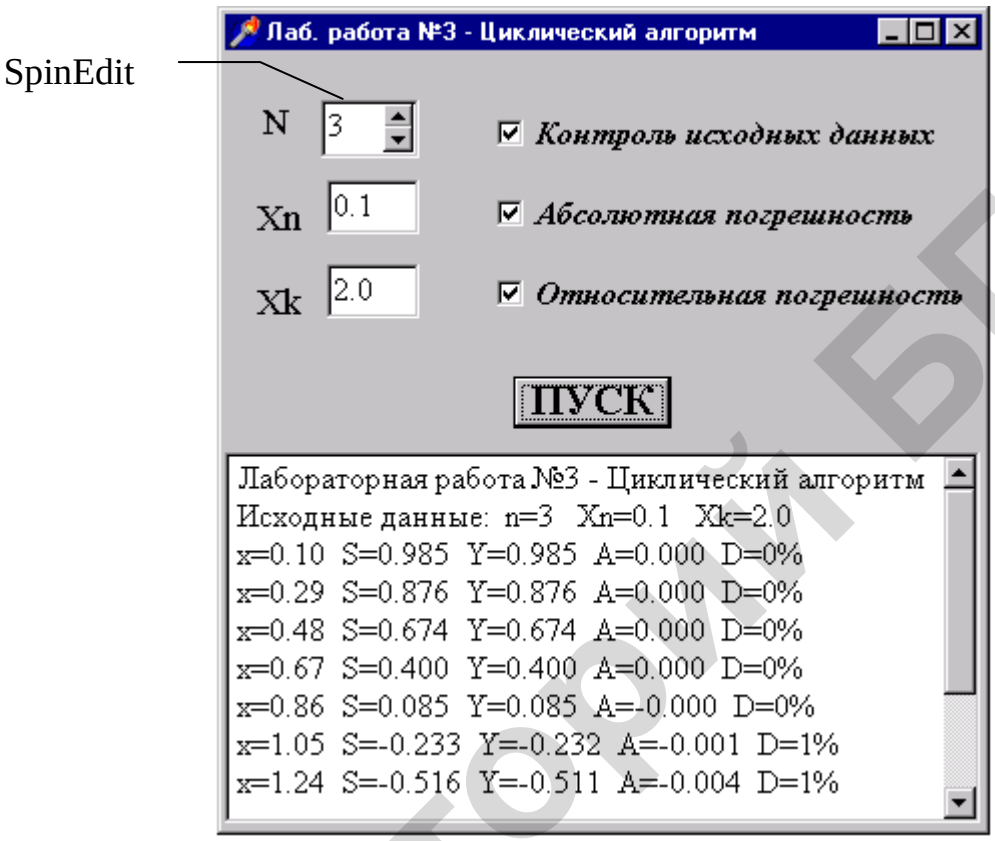


Рис. 3.1

Компонент SpinEdit находится на странице Samples Палитры Компонентов. В тех случаях, когда объем выводимой информации превышает размер поля компонента Memo, целесообразно снабдить его линейками прокрутки. В свойстве ScrollBars компонента Memo1 установим значение ssVertical – появится вертикальная линейка прокрутки. Присвоим модулю имя UnCiklAlg.

3.2.2. Текст модуля UnCiklAlg

```
Unit UnCiklAlg;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, ExtCtrls, Spin;
type
  TForm1 = class(TForm)
```

Репозиторий БГАТУ

```
    Memo1: TMemo;
    Button1: TButton;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Edit1: TEdit;
    Edit2: TEdit;
    SpinEdit1: TSpinEdit;
    CheckBox1: TCheckBox;
    CheckBox2: TCheckBox;
    CheckBox3: TCheckBox;
    procedure FormCreate(Sender: TObject);
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;
implementation
{$R *.DFM}
procedure TForm1.FormCreate(Sender: TObject);
begin
  SpinEdit1.text:='3'; // начальное значение N
  Edit1.text:='0.1';   // начальное значение Xn
  Edit2.text:='2.0';   // начальное значение Xk
  Memo1.Clear;
  Memo1.Lines.Add('Лабораторная работа №3 - Циклический алгоритм');
end;
procedure TForm1.Button1Click(Sender: TObject);
 var xn,xk,x,h,c,s,y,al,del:extended;
     n,k:integer;
begin
 n:=StrToInt(SpinEdit1.Text);
 xn:=StrToFloat(Edit1.Text);
 xk:=StrToFloat(Edit2.Text);
 if CheckBox1.Checked then
   Memo1.Lines.Add('Исходные данные:  n='+IntToStr(n)+
     '  Xn='+FloatToStrF(xn,ffFixed,6,1)+
     '  Xk='+FloatToStrF(xk,ffFixed,6,1));
 h:=(xk-xn)*0.1;        // шаг h
 x:=xn;
 repeat                 // цикл по x
  c:=-x*x*0.5;
```

Репозиторий БГАТУ

```
 S:=1;
 for k:=1 to n do
  begin
   s:=s+c*(2*k*k+1);
   c:=-c*x*x/((2*k+1)*(2*k+2));
  end;
 y:=(1-x*x*0.5)*cos(x)-0.5*x*sin(x);
 if CheckBox2.Checked then
   if CheckBox3.Checked then
    begin
     al:=s-y;                    // абсолютная погрешность
     del:=abs((s-y)/y)*100; // относительная погрешность
     Memo1.Lines.Add('x='+FloatToStrF(x,ffFixed,6,2)+
                  '  S='+ FloatToStrF(s,ffFixed,6,3)+
                  '  Y='+ FloatToStrF(y,ffFixed,6,3)+
                  '  A='+ FloatToStrF(al,ffFixed,6,3)+
                  '  D='+ FloatToStrF(del,ffFixed,6,0)+'%');
    end
                              else
    begin
     al:=s-y;
     Memo1.Lines.Add('x='+FloatToStrF(x,ffFixed,6,2)+
                  '  S='+ FloatToStrF(s,ffFixed,6,3)+
                  '  Y='+ FloatToStrF(y,ffFixed,6,3)+
                  '  A='+ FloatToStrF(al,ffFixed,6,3));
    end
                             else
   if CheckBox3.Checked then
    begin
     del:=abs((s-y)/y)*100;
     Memo1.Lines.Add('x='+FloatToStrF(x,ffFixed,6,2)+
                  '  S='+ FloatToStrF(s,ffFixed,6,3)+
                  '  Y='+ FloatToStrF(y,ffFixed,6,3)+
                  '  D='+ FloatToStrF(del,ffFixed,6,0)+'%');
    end
                               else
     Memo1.Lines.Add('x='+FloatToStrF(x,ffFixed,6,2)+
                  '  S='+ FloatToStrF(s,ffFixed,6,3)+
                  '  Y='+ FloatToStrF(y,ffFixed,6,3));
  x:=x+h;
 until x>xk;
end;
end.
```

Репозиторий БГАТУ

3.3. Выполнение индивидуального задания

По указанию преподавателя выберите свое индивидуальное задание. Создайте приложение и протестируйте его работу.

Индивидуальные задания

В заданиях с №1 по №15 необходимо вывести на экран таблицу значений функции Y(x) и ее разложения в ряд S(x) для значений x от x_n до x_k с шагом $h = (x_k - x_n)/10$. Близость значений S(x) и Y(x) во всем диапазоне значений x указывает на правильность вычисления S(x) и Y(x).

№	x_n	x_k	S(x)	n	Y(x)
1	0.1	1	$x - \frac{x^3}{3!} + \ldots + (-1)^n \frac{x^{2n+1}}{(2n+1)!}$	16	$\sin x$
2	0.1	1	$1 + \frac{x^2}{2!} + \ldots + \frac{x^{2n}}{(2n)!}$	10	$\frac{e^x + e^{-x}}{2}$
3	0.1	1	$1 + \frac{\cos\frac{\pi}{4}}{1!}x + \ldots + \frac{\cos n\frac{\pi}{4}}{n!}x^n$	12	$e^{x\cos\frac{\pi}{4}}\cos(x\sin\frac{\pi}{4})$
4	0.1	1	$1 - \frac{x^2}{2!} + \ldots + (-1)^n \frac{x^{2n}}{(2n)!}$	8	$\cos x$
5	0.1	1	$1 - \frac{x^2}{1!} + \frac{x^4}{2!} - \frac{x^6}{3!} + \ldots + (-1)^n \frac{x^{2n}}{n!}$	14	e^{-x^2}
6	0.1	1	$x + \frac{x^3}{3!} + \ldots + \frac{x^{2n+1}}{(2n+1)!}$	8	$\frac{e^x - e^{-x}}{2}$
7	0.1	1	$\frac{x^3}{3} - \frac{x^5}{15} + \ldots + (-1)^{n+1} \frac{x^{2n+1}}{4n^2 - 1}$	12	$\frac{1+x^2}{2} arctgx - \frac{x}{2}$
8	0.1	1	$1 + \frac{2x}{1!} + \ldots + \frac{(2x)^n}{n!}$	10	e^{2x}
9	0.1	1	$1 + 2\frac{x}{2} + \ldots + \frac{n^2+1}{n!}\left(\frac{x}{2}\right)^n$	14	$\left(\frac{x^2}{4} + \frac{x}{2} + 1\right)e^{\frac{x}{2}}$
10	0.1	0.5	$x - \frac{x^3}{3} + \ldots + (-1)^n \frac{x^{2n+1}}{2n+1}$	15	$arctgx$
11	0.1	0.8	$\frac{x^2}{2} - \frac{x^4}{12} + \ldots + (-1)^{n+1} \frac{x^{2n}}{2n(2n-1)}$	10	$xarctgx - \ln\sqrt{1+x^2}$
12	0.1	1	$-\frac{(2x)^2}{2} + \frac{(2x)^4}{24} - \ldots + (-1)^n \frac{(2x)^{2n}}{(2n)!}$	8	$2(\cos^2 x - 1)$

Репозиторий БГАТУ

13	-2	-0.1	$-(1+x)^2+\frac{(1+x)^4}{2}+\ldots+(-1)^n\frac{(1+x)^{2n}}{n}$	16	$\ln\frac{1}{2+2x+x^2}$
14	0.2	0.8	$\frac{x}{3!}+\frac{4x^2}{5!}+\ldots+\frac{n^2}{(2n+1)!}x^n$	12	$\frac{1}{4}\left(\frac{x+1}{\sqrt{x}}sh\sqrt{x}-ch\sqrt{x}\right)$
15	-1	1	$x-\frac{x^2}{2}+\frac{x^3}{3}-\frac{x^4}{4}+\ldots+(-1)^{n-1}\frac{x^n}{n}$	10	$\ln(1+x)$

16. Подсчитать k - количество цифр в десятичной записи целого неотрицательного числа n.
17. Переменной t присвоить значение 1 или 0 в зависимости от того, является ли натуральное число k степенью 3.
18. Дано n вещественных чисел. Вычислить разность между максимальным и минимальным из них.
19. Дана непустая последовательность различных натуральных чисел, за которой следует 0. Определить порядковый номер наименьшего из них.
20. Даны целое n>0 и последовательность из n вещественных чисел, среди которых есть хотя бы одно отрицательное число. Найти величину наибольшего среди отрицательных чисел этой последовательности.
21. Дано n вещественных чисел. Определить, образуют ли они возрастающую последовательность.
22. Дана последовательность из n целых чисел. Определить со скольких отрицательных чисел она начинается.
23. Определить k – количество трехзначных натуральных чисел, сумма цифр которых равна n(1<=n<=27). Операции деления (/, div и mod) не использовать.
24. Вывести на экран в возрастающем порядке все трехзначные числа, в десятичной записи которых нет одинаковых цифр (операции деления не использовать).
25. Переменной t присвоить значение 1 или 0 в зависимости от того, можно или нет натуральное число n представить в виде трех полных квадратов.
26. Дано натуральное число n. Выяснить, входит ли цифра 3 в запись числа n^2.
27. Дано натуральное число n. Найти сумму его цифр.
28. Дано целое n>0, за которым следует n вещественных чисел. Определить, сколько среди них отрицательных.
29. Дано натуральное число n. Переставить местами первую и последнюю цифры числа n.
30. Дано натуральное число n. Изменить порядок следования цифр числа n на обратный.

Репозиторий БГАТУ

Лабораторная работа №4

ПРОГРАММИРОВАНИЕ АЛГОРИТМОВ С ИСПОЛЬЗОВАНИЕМ МАССИВОВ

Цель лабораторной работы: освоить применение компонента StringGrid и создать приложение, в котором используются массивы.

4.1. Пример создания приложения

Задание: создать Windows-приложение для вычисления вектора $x=\{x_1,x_2,\ldots,x_m\}$,равного p-й строке матрицы $A=\{a_{ij}\}(x_j=a_{pj},\ j=1,2,\ldots,m)$ и вектора $y=\{y_1,y_2,\ldots,y_n\}$, равного q-му столбцу матрицы $A=\{a_{ij}\}(y_i=a_{iq},\ i=1,2,\ldots,n)$ $(n\le 6,m\le 8)$. В панели интерфейса предусмотреть возможность управления размерностью массивов.

Один из возможных вариантов панели интерфейса создаваемого приложения показан на рис.4.1.

4.1.1. Размещение компонентов на Форме

При работе с массивами ввод и вывод информации на экран удобно организовывать с помощью компонента StringGrid.

Рис. 4.1

Компонент **StringGrid** используется для отображения информации в виде таблицы. Таблица содержит две зоны – фиксированную и рабочую. *Фиксированная зона* служит для вывода наименований строк и столбцов рабочей зоны и управления их размерами с помощью “мыши”. Фиксированная зона выделена другим цветом и в нее запрещен ввод информации с клавиатуры. Количество строк и столбцов фиксированной зоны устанавливается в свойствах FixedRows и FixedCols, соответственно.

Репозиторий БГАТУ

Рабочая зона содержит RowCount строк и ColCount столбцов информации, которую можно изменять как программно, так и с помощью “мыши” или клавиатуры.

Доступ к информации в программе осуществляется с помощью свойства Cells[ACol, ARow: integer]: string, где ACol-номер столбца, а ARow – номер строки таблицы, причем *нумерация начинается с нуля.*

Пиктограмма компонента StringGrid находится на странице Additional Палитры Компонентов. Так как в нашем задании для всех компонентов StringGrid фиксированная зона не используется, в Инспекторе Объектов значения свойств FixedCols и FixedRows установите равными 0. В соответствии с заданием установите предельные значения количества строк n и столбцов m для компонента StringGrid1: ColCount=8, а RowCount=6 (восемь столбцов и шесть строк). Для компонента StringGrid2 ColCount=1, RowCount=8, а для компонента StringGrid3 ColCount=1, RowCount=6.

По умолчанию в компонент StringGrid запрещен ввод информации с клавиатуры, поэтому для компонента StringGrid1 необходимо в Инспекторе Объектов дважды щелкнуть “мышью” на символе + свойства +Options и в открывшемся списке опций установить значение goEditing в True.

Для удобства работы с компонентами SpinEdit установите для компонента SpinEdit1 значения свойств: MinValue=1, MaxValue=6, а для компонента SpinEdit2: MinValue=1, MaxValue=8.

4.1.2. Создание процедур обработки событий SpinEdit1Change и SpinEdit2Change

События SpinEdit1Change и SpinEdit2Change возникают при любом изменении значения в поле редактора SpinEdit1 и SpinEdit2 соответственно. Создадим процедуры обработки этих событий, в которых присвоим значения n и m, полученные из полей редакторов SpinEdit, свойствам ColCount и RowCount компонентов StringGrid. Это позволит управлять размерами таблиц StringGrid с помощью компонентов SpinEdit без дополнительных кнопок, так как изменение значений в поле редактора SpinEdit сразу приведет к изменению размера таблиц StringGrid. Дважды щелкните “мышью” на компоненте SpinEdit1 – курсор установится в тексте процедуры-обработчика события SpinEdit1Change: **procedure TForm1.SpinEdit1Change(Sender: TObject)**. Внимательно наберите операторы этой процедуры, используя текст модуля UnMas(см. п.4.1.3)**.** Аналогичным образом создайте процедуру-обработчик события SpinEdit2Change: **procedure TForm1.SpinEdit2Change(Sender: TObject)**.

4.1.3. Текст модуля UnMas

```
Unit UnMas;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, Spin, Grids;
```

Репозиторий БГАТУ

```
type
  TForm1 = class(TForm)
    Label1: TLabel;
    SpinEdit1: TSpinEdit;
    SpinEdit2: TSpinEdit;
    Label8: TLabel;
    StringGrid1: TStringGrid;
    StringGrid2: TStringGrid;
    StringGrid3: TStringGrid;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    SpinEdit3: TSpinEdit;
    SpinEdit4: TSpinEdit;
    Label6: TLabel;
    Label7: TLabel;
    Button1: TButton;
    procedure FormCreate(Sender: TObject);
    procedure SpinEdit1Change(Sender: TObject);
    procedure SpinEdit2Change(Sender: TObject);
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;
implementation
{$R *.DFM}
var
 A:array[1..6,1..8] of extended;// объявление двумерного массива A
 X:array[1..8] of extended;     // объявление одномерного массива X
 Y:array[1..6] of extended;     // объявление одномерного массива Y
 n,m,p,q:integer;               // объявление глобальных переменных
procedure TForm1.FormCreate(Sender: TObject);
begin
 SpinEdit1.Text:='4';           // начальное значение n
 SpinEdit2.Text:='6';           // начальное значение m
 SpinEdit3.Text:='2';           // начальное значение p
 SpinEdit4.Text:='3';           // начальное значение q
 StringGrid1.RowCount:=4;    // количество строк массива A
 StringGrid1.ColCount:=6;    // количество столбцов массива A
```

Репозиторий БГАТУ

```
 StringGrid2.RowCount:=6;    // количество строк массива X
 StringGrid3.RowCount:=4;    // количество строк массива Y
end;
procedure TForm1.SpinEdit1Change(Sender: TObject);
begin
 n:=StrToInt(SpinEdit1.Text);// n присваивается содержимое поля редактора
 StringGrid1.RowCount:=n;    // устанавливается количество строк массива A
 StringGrid3.RowCount:=n;    // устанавливается количество строк массива Y
end;
procedure TForm1.SpinEdit2Change(Sender: TObject);
begin
 m:=StrToInt(SpinEdit2.Text);// m присваивается содержимое поля редактора
 StringGrid1.ColCount:=m; // устанавливается количество столбцов массива A
 StringGrid2.RowCount:=m;   // устанавливается количество строк массива X
end;
procedure TForm1.Button1Click(Sender: TObject);
var
 i,j:integer;              // объявление локальных переменных
begin
 n:=StrToInt(SpinEdit1.Text);
 StringGrid1.RowCount:=n;
 StringGrid3.RowCount:=n;
 m:=StrToInt(SpinEdit2.Text);
 StringGrid1.ColCount:=m;
 StringGrid2.RowCount:=m;
 p:=StrToInt(SpinEdit3.Text);
 q:=StrToInt(SpinEdit4.Text);
// Ввод значений из таблицы в массив A
 for i:=1 to n do
  for j:=1 to m do
   A[i,j]:=StrToFloat(StringGrid1.Cells[j-1,i-1]);
for j:=1 to m do   // формирование массива X и вывод его значений в таблицу
  begin
  X[j]:=A[p,j];
  StringGrid2.Cells[0,j-1]:=FloatToStrF(X[j],ffFixed,3,1);
  end;
for i:=1 to n do   // формирование массива Y и вывод его значений в таблицу
  begin
  Y[i]:=A[i,q];
  StringGrid3.Cells[0,i-1]:=FloatToStrF(Y[i],ffFixed,3,1);
  end;
end;
end.
```

Репозиторий БГАТУ

4.1.4. Работа с приложением

Запустите созданное приложение. Занесите числовые значения в элементы матрицы A и убедитесь в том, что приложение функционирует в соответствии с заданием.

4.2. Выполнение индивидуального задания

Изучите в приложении 2 описание компонентов StringGrid и DrawGrid. По указанию преподавателя выберите свое индивидуальное задание. Создайте приложение и протестируйте его работу.

Индивидуальные задания

1. Задана целочисленная матрица *A* размером *NxM*. Получить массив *B*, присвоив его *k*-му элементу значение *0*, если все элементы *k*-го столбца матрицы нулевые, и значение *1* в противном случае(*k=1,2,…,M*).
2. Задана целочисленная матрица *A* размером *NxM*. Получить массив *B*, присвоив его *k*-му элементу значение *1*, если элементы *k*–й строки матрицы упорядочены по убыванию, и значение *0* в противном случае(*k=1,2,…,N*).
3. Задана целочисленная матрица *A* размером *NxM*. Получить массив *B*, присвоив его *k*-му элементу значение *1*, если *k*-я строка матрицы симметрична, и значение *0* в противном случае(*k=1,2,…,N*)..
4. Задана целочисленная матрица размером *NxM*. Определить *k*–количество “особых” элементов матрицы, считая элемент “особым”, если он больше суммы остальных элементов своего столбца.
5. Задана целочисленная матрица размером *NxM*. Определить *k*–количество “особых” элементов матрицы, считая элемент “особым”, если в его строке слева от него находятся элементы, меньшие его, а справа – б*о*льшие.
6. Задана символьная матрица размером *NxM*. Определить *k*-количество различных элементов матрицы (т.е. повторяющиеся элементы считать один раз).
7. Дана вещественная матрица размером *NxM*. Упорядочить ее строки по неубыванию их первых элементов.
8. Дана вещественная матрица размером *NxM*. Упорядочить ее строки по неубыванию суммы их элементов.
9. Дана вещественная матрица размером *NxM*. Упорядочить ее строки по неубыванию их наибольших элементов.
10. Определить является ли заданная квадратная матрица *n*-го порядка симметричной относительно побочной диагонали.
11. Для заданной целой матрицы размером *NxM* вывести на экран все ее седловые точки. Элемент матрицы называется седловой точкой, если он является наименьшим в своей строке и одновременно наибольшим в своем столбце или, наоборот, является наибольшим в своей строке и наименьшим в своем столбце.
12. В матрице *n*-го порядка переставить строки так, чтобы на главной диагонали матрицы были расположены элементы, наибольшие по абсолютной величине.

Репозиторий БГАТУ

13. В заданной действительной матрице *n*-го порядка найти максимальный среди элементов, лежащих ниже побочной диагонали и минимальный среди элементов, лежащих выше главной диагонали.
14. В заданной действительной матрице размером *NxM* поменять местами строку, содержащую элемент с наибольшим значением со строкой, содержащей элемент с наименьшим значением.
15. Из заданной целочисленной матрицы *n*-го порядка получить матрицу порядка *n-1* путем удаления из исходной матрицы строки и столбца, на пересечении которых расположен элемент с наибольшим по модулю значением (предполагается что такой элемент единственный).
16. Дан массив из *k* символов. Вывести на экран сначала все цифры, входящие в него, а затем все остальные символы, сохраняя при этом взаимное расположение символов в каждой из этих двух групп.
17. Дан массив, содержащий от 1 до *k* символов, за которым следует точка. Вывести этот текст в обратном порядке.
18. Дан непустой массив из цифр. Вывести на экран цифру, наиболее часто встречающуюся в этом массиве.
19. Элементы массива *X* отсортировать по возрастанию.
20. Элементы массива *X* расположить в обратном порядке.
21. Элементы массива *X* циклически сдвинуть на *k* позиций влево.
22. Элементы массива *X* циклически сдвинуть на *n* позиций вправо.
23. Преобразовать массив *X* по следующему правилу: все отрицательные элементы массива перенести в начало, а все остальные – в конец, сохраняя исходное взаимное расположение как среди отрицательных, так и среди остальных элементов.
24. Элементы каждого из массивов *X* и *Y* упорядочены по неубыванию. Объединить элементы этих двух массивов в один массив *Z* так, чтобы они снова оказались упорядоченными по неубыванию.
25. Дан массив из *k* символов. Определить, симметричен ли он, т. е. читается ли он одинаково слева направо и справа налево.
26. Даны два массива. Найти наименьшее среди тех элементов первого массива, которые не входят во второй массив
27. Дан массив *X* из *k* целых чисел. Определить количество инверсий в этом массиве (т.е. таких пар элементов, в которых большее число находится слева от меньшего: $x_i > x_j$ при $i<j$).
28. Дан массив из строчных латинских букв. Вывести на экран в алфавитном порядке все буквы, которые входят в этот текст по одному разу.
29. Вывести на экран заданный массив из *k* символов, удалив из него повторные вхождения каждого символа.
30. Определить сколько различных символов входит в заданный массив, содержащий не более *k* символов и оканчивающийся точкой (в сам текст точка не входит).

Репозиторий БГАТУ

Лабораторная работа №5

ПРОГРАММИРОВАНИЕ АЛГОРИТМОВ С ИСПОЛЬЗОВАНИЕМ СТРОК

Цель лабораторной работы: освоить применение компонентов ListBox и ComboBox и создать приложение, в котором используются строки.

5.1. Пример создания приложения

Задание: создать Windows-приложение для подсчета количества слов в произвольной строке. Слова в строке разделяются любым количеством пробелов. Ввод строки заканчивать нажатием клавиши Enter. Работа приложения должна завершаться нажатием кнопки Close.

Один из возможных вариантов панели интерфейса создаваемого приложения показан на рис.5.1.

5.1.1. Размещение компонентов на Форме

При работе со строками ввод и вывод информации на экран удобно организовывать с помощью компонентов ListBox и ComboBox.

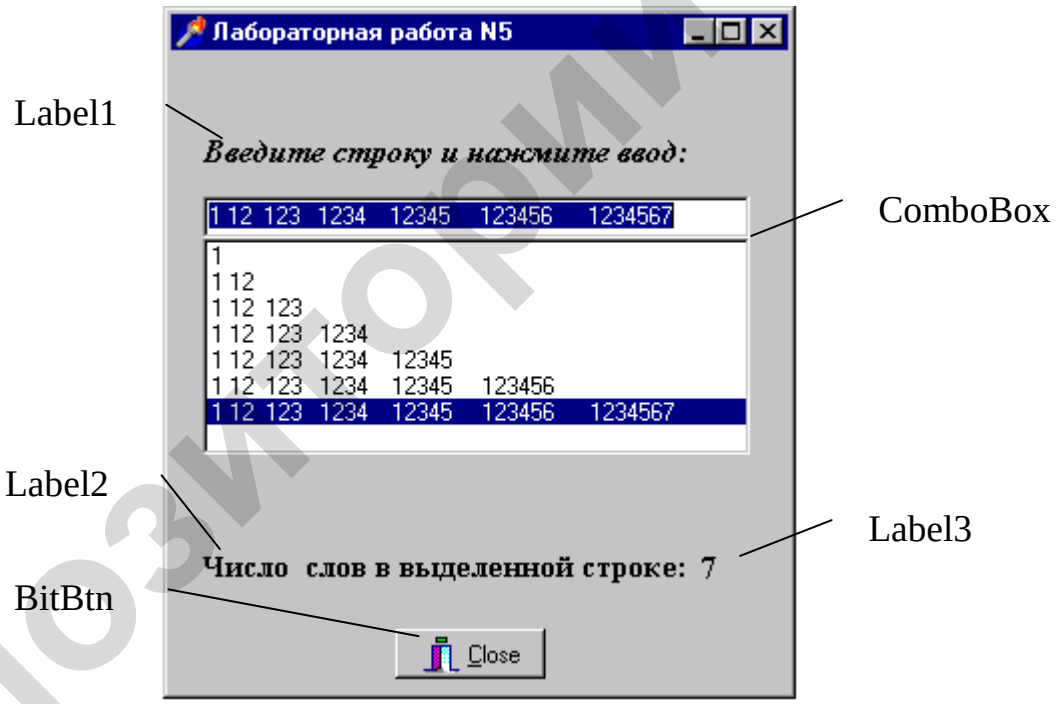


Рис. 5.1

Компонент **ListBox** представляет собой список, элементы которого выбираются при помощи клавиатуры или “мыши”. Список элементов задается свойством Items, методы Add, Delete и Insert которого используются для добавления, удаления и вставки строк, соответственно. Для определения номера выделенного элемента используется свойство ItemIndex.

Компонент **ComboBox** представляет собой комбинацию списка ListBox и редактора Edit, поэтому практически все свойства заимствованы у этих компонентов. Для работы с окном редактирования используется свойство Text

Репозиторий БГАТУ

как в Edit, а для работы со списком выбора используется свойство Items как в ListBox. Существует 5 модификаций компонента, определяемых его свойством Style. В модификации csSimple список всегда раскрыт, в остальных он раскрывается после нажатия кнопки справа от редактора.
Компоненты ListBox и ComboBox находятся на странице Standard Палитры Компонентов.

Компонент **BitBtn** расположен на странице Additional Палитры Компонентов и представляет собой разновидность стандартной кнопки Button. Его отличительная особенность – наличие растрового изображения на поверхности кнопки, которое определяется свойством Glyph. Кроме того, имеется свойство Kind, которое задает одну из 11 стандартных разновидностей кнопок. Нажатие любой из них, кроме bkCustom и bkHelp закрывает модальное окно. Кнопка bkClose закрывает главное окно и завершает работу программы.

5.1.2. Создание процедур обработки событий

В момент запуска приложения, когда панель интерфейса появляется на экране, для пользователя удобно чтобы курсор уже находился в поле редактора компонента ComboBox. При активизации Формы возникает событие OnActivate, которое можно использовать для передачи фокуса ввода компоненту ComboBox. Для создания процедуры-обработчика этого события необходимо в Инспекторе Объектов выбрать компонент Form1, на странице Events найти событие OnActivate и дважды щелкнуть “мышью” по его правой (белой) части. Курсор установится в тексте процедуры-обработчика события активизации Формы: **procedure TForm1.FormActivate(Sender: TObject)**. В этом месте процедуры наберите оператор передачи фокуса ввода компоненту ComboBox1 (см. текст модуля UnStr, который приведен в п. 5.1.3).

В соответствии с заданием необходимо, чтобы при нажатии клавиши Enter строка символов, которую пользователь набрал в поле редактирования, переносилась в список выбора компонента ComboBox. Для создания процедуры-обработчика этого события необходимо в Инспекторе Объектов выбрать компонент ComboBox1, на странице Events найти событие OnKeyPress и дважды щелкнуть “мышью” по его правой части. Курсор установится в тексте процедуры-обработчика события нажатия клавиши на клавиатуре: **procedure TForm1.ComboBox1KeyPress(Sender:TObject;var Key:Char)**. В этом месте процедуры, пользуясь текстом модуля UnStr, наберите операторы, которые при нажатии клавиши Enter переносят строку из поля редактирования в список выбора и очищают поле редактирования.

Процесс создания процедуры-обработчика события нажатия клавиши “мыши” в списке выбора **procedure TForm1.ComboBox1Click(Sender: TObject)**
выполняется аналогично для события OnClick компонента ComboBox1. Операторы, которые осуществляют основной алгоритм обработки символов выбранной строки, наберите пользуясь текстом модуля UnStr.

Репозиторий БГАТУ

5.1.3. Текст модуля UnStr

```
Unit UnStr;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, Buttons;
type
  TForm1 = class(TForm)
    Label2: TLabel;
    Label3: TLabel;
    BitBtn1: TBitBtn;
    ComboBox1: TComboBox;
    Label1: TLabel;
    procedure ComboBox1KeyPress(Sender: TObject; var Key: Char);
    procedure ComboBox1Click(Sender: TObject);
    procedure FormActivate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;

Implementation
{$R *.DFM}

// Обработка события активизации Формы
procedure TForm1.FormActivate(Sender: TObject);
begin
ComboBox1.SetFocus;              // передача фокуса ввода ComboBox1
end;

// Обработка события ввода символа и нажатия клавиши Enter
procedure TForm1.ComboBox1KeyPress(Sender: TObject; var Key: Char);
begin
 if key=#13 then                          // если нажата клавиша Enter то
  begin                                              // строка из поля редактирования
заносится
ComboBox1.Items.Add(ComboBox1.Text); // в список выбора
ComboBox1.Text:='';                  // очистка окна редактирования
  end;
end;
```

Репозиторий БГАТУ

```
// Обработка события нажатия клавиши "мыши" в списке выбора
procedure TForm1.ComboBox1Click(Sender: TObject);
var
 st       : string;
 n,i,nst,ind: integer;
begin
   n:=0;                              // n содержит количество слов
 ind:=0;
 nst:=ComboBox1.ItemIndex;   // определение номера выбранной строки
  st:=ComboBox1.Items[nst];   // st присваивается выбранная строка
  for i:=1 to Length(st) do       // просмотр всех символов строки
    case ind of
    0 : if st[i]<>' ' then            // если встретился символ
                        begin
                         ind:=1;
                          n:=n+1;    // количество слов увеличивается на единицу
                        end;
    1 : if st[i]=' ' then             // если встретился пробел
                      ind:=0;
end;
 Label3.Caption:=IntToStr(n); // вывод количества слов в Label3
end;
end.
```

5.2. Выполнение индивидуального задания

Во всех заданиях исходные данные вводить с помощью компонента Edit в компонент ListBox, либо с помощью свойства Text в свойство Items компонента ComboBox. Результат выводить с помощью компонента Label. Ввод строки заканчивать нажатием клавиши Enter. Работа приложения должна завершаться нажатием кнопки Close.

Для проверки функционирования приложения подготовить несколько тестов.

Индивидуальные задания

1. Дана строка, состоящая из групп нулей и единиц. Каждая группа отделяется друг от друга одним или несколькими пробелами. Найти количество групп с пятью символами.
2. Дана строка, состоящая из групп нулей и единиц. Каждая группа отделяется друг от друга одним или несколькими пробелами. Найти и вывести на экран самую короткую группу.
3. Дана строка, состоящая из групп нулей и единиц. Каждая группа отделяется друг от друга одним или несколькими пробелами. Подсчитать количество символов в самой длинной группе.
4. Дана строка, состоящая из групп нулей и единиц. Каждая группа отделяется друг от друга одним или несколькими пробелами. Найти и вывести на экран группы с четным количеством символов.

Репозиторий БГАТУ

5. Дана строка, состоящая из групп нулей и единиц. Каждая группа отделяется друг от друга одним или несколькими пробелами. Подсчитать количество единиц в группах с нечетным количеством символов.
6. Дана строка, состоящая из букв, цифр, запятых, точек, знаков “+” и “-“. Выделить подстроку, которая соответствует записи целого числа (т.е. начинается со знака “+” или “-“ и внутри подстроки нет букв, запятых и точек).
7. Дана строка символов, состоящая из букв, цифр, запятых, точек, знаков “+” и “-“. Выделить подстроку, которая соответствует записи вещественного числа с фиксированной точкой
8. Дана строка символов, состоящая из букв, цифр, запятых, точек, знаков “+” и “-“. Выделить подстроку, которая соответствует записи вещественного числа с плавающей точкой
9. Дана строка символов, состоящая из произвольных десятичных цифр, разделенных пробелами. Вывести на экран числа этой строки в порядке возрастания их значений.
10. Дана строка символов, состоящая из произвольных десятичных цифр, разделенных пробелами. Вывести четные числа этой строки.
11. Дана строка символов, состоящая из произвольного текста на английском языке, слова отделены пробелами. Вывести на экран слова этого текста в порядке, соответствующем латинскому алфавиту.
12. Дана строка символов, состоящая из произвольного текста на английском языке, слова отделены пробелами. Вывести на экран порядковый номер слова, накрывающего *k*-ю позицию (если на *k*-ю позицию попадает пробел, то номер предыдущего слова).
13. Дана строка символов, состоящая из произвольного текста на английском языке, слова отделены пробелами. Разбить исходную строку на две подстроки, причем первая длиной *k*-символов (если на *k*-ю позицию попадает слово, то его следует отнести ко второй строке, дополнив первую пробелами до *k*-позиций).
14. Дана строка символов, состоящая из произвольного текста на английском языке, слова отделены пробелами. Вывести на экран порядковый номер слова максимальной длины и номер позиции строки, с которой оно начинается.
15. Дана строка символов, состоящая из произвольного текста на английском языке, слова отделены пробелами. Вывести на экран порядковый номер слова минимальной длины и количество символов в этом слове.
16. Дана строка символов, состоящая из произвольного текста на английском языке, слова отделены пробелами. В каждом слове заменить первую букву прописной (большой).
17. Дана строка символов, состоящая из произвольного текста на английском языке, слова отделены пробелами. Удалить первых *k* слов из строки, сдвинув на их место последующие слова строки.
18. Дана строка символов, состоящая из произвольного текста на английском языке, слова отделены пробелами. Поменять местами *i*-е и *j*-е слова.

Репозиторий БГАТУ

19. Дана строка символов, состоящая из произвольного текста на английском языке, слова отделены пробелами. Поменять местами первую и последнюю буквы каждого слова.
20. Дана строка символов, состоящая из произвольного текста на английском языке, слова отделены пробелами. Замените буквы латинского алфавита соответствующими буквами русского алфавита.
21. Дана строка символов $S_1 S_2 \ldots S_m$, в которой могут встречаться цифры, пробелы, буква “E” и знаки “+”, “-“. Из данной строки выделить, подстроки разделенные пробелами. Определить, является ли первая подстрока числом. Если да, то выяснить: целое или вещественное число, положительное или отрицательное.
22. Дана строка символов, содержащая некоторый текст на русском языке. Разработать программу форматирования этого текста, т.е. его разбиения на отдельные строки (по *k* символов в каждой строке) и выравнивания по правой границе путем вставки между отдельными словами необходимого количества пробелов.
23. Дана строка символов, содержащая некоторый текст на русском языке. Замените буквы русского алфавита соответствующими буквами латинского алфавита.
24. Дана строка символов, содержащая некоторый текст. Разработать программу, которая определяет, является ли данный текст палиндромом, т.е. читается ли он слева направо так же, как и справа налево (например, «А роза упала на лапу Азора»).
25. Составить программу, которая читает построчно текст другой программы (ввести с клавиатуры) на языке *Pascal,* обнаруживает комментарии и выводит их на экран.
26. Составить программу, которая читает построчно текст другой программы (ввести с клавиатуры) на языке *Pascal*, подсчитывает количество ключевых слов «*begin*» и «*end*», и выводит на экран соответствующее сообщение.
27. Дан текст из *k* символов. Вывести на экран только строчные русские буквы, входящие в этот текст.
28. Дан текст из *k* символов. Вывести на экран в алфавитном порядке все различные прописные (большие) русские буквы, входящие в этот текст.
29. Разработать программу, которая заданное целое число от 1 до 999 выводит на экран римскими цифрами.
30. Дан текст из заглавных латинских букв, за которым следует пробел. Определить является ли этот текст правильной записью римскими цифрами целого числа от 1 до 999, и, если является, вывести на экран это число арабскими цифрами (в десятичной системе).

Репозиторий БГАТУ

Приложение 1

П1. КОМАНДЫ ОСНОВНОГО МЕНЮ

В меню **File** находятся команды для выполнения операций с проектами, модулями и файлами:

Команда	**Описание**
New	Позволяет выбрать тип элемента из репозитория (архива, в котором хранятся заготовки для новых программ) и создать его
New Application	Создает новый проект, состоящий из формы, модуля и файла проекта
New Form	Создает новую форму и подключает ее к проекту
New Data Module	Создает новый модуль данных и подключает его к проекту
Open	Открывает ранее созданный проект, модуль, форму или текстовый файл
Reopen	Вызывает список ранее загружавшихся проектов и форм для выбора и повторной загрузки
Save	Сохраняет текущую форму или модуль или файл
Save As	Сохраняет текущую форму с новым именем
Save Project As	Сохраняет текущий проект с новым именем
Save All	Сохраняет все открытые файлы, проект и используемые им модули
Close	Закрывает текущую форму
Close All	Закрывает все открытые файлы
Use Unit	Добавляет имя указанного модуля в список используемых модулей (USES) текущего активного модуля
Add to Project	Добавляет файл к проекту
Remove From Project	Удаляет файл из проекта
Print	Выводит содержимое активного файла на печать
Exit	Завершает работу Delphi

В меню **Edit** расположены команды, осуществляющие операции редактирования, работы с областью обмена данными, отмены действий и управления отображением компонентов:

Команда	**Описание**
Undo	Отменяет ранее выполненные действия
Redo	Восстанавливает отмененные действия
Cut	Вырезает выделенный объект и помещает его в буфер обмена данными
Copy	Копирует выделенный объект и или фрагмент текста программы и помещает его в буфер обмена данными
Paste	Копирует содержимое буфера обмена данными в редактор или форму
Delete	Удаляет выбранный объект или фрагмент программы
Select All	Выделяет все компоненты формы или весь текст программы
Align to Grid	Выравнивает выбранный компонент по сетке

Репозиторий БГАТУ

Bring to Front	Перемещает выбранный компонент поверх других компонентов
Send to Back	Перемещает выбранный компонент под другие компоненты
Align	Выравнивает компоненты
Size	Изменяет размер выделенных компонентов
Scale	Изменяет размер всех компонентов в форме
Tab Order	Изменяет порядок табуляции компонентов в активной форме
Creation Order	Задает порядок создания не визуальных компонентов
Lock Controls	Запрещает перемещение компонентов внутри формы
Add To Interface	Позволяет определить новую процедуру, функцию или свойство компонента ActiveX

Меню **Search** предоставляет команды для поиска и замены, а также команды для поиска указанных символов и строк, содержащих ошибки, найденные компилятором:

Команда	**Описание**
Find	Поиск указанного фрагмента текста
Find in files	Поиск указанного текста в нескольких файлах, задаваемых в диалоговой панели
Replace	Поиск указанного фрагмента текста и замена его на новый текст
Search Again	Повторный поиск или повторная замена
Incremental Search	Поиск текста по мере его ввода
Go to Line Number	Перемещает курсор на строку с указанным номером
Show Last Compile Error	Перемещение курсора на строку, содержащую ошибку, найденную компилятором
Find Error	Поиск ошибки времени исполнения (run-time error)
Browse Symbol	Показывает характеристики указанного символа программы по его имени

В меню **View** содержатся команды для отображения различной информации и вызова Менеджера Проектов, Инспектора Объектов, Броузера Объектов и других информационных утилит:

Команда	**Описание**
Project Manager	Менеджер Проектов (Project Manager)
Project Source	Отображает исходный текст файла проекта
Object Inspector	Инспектор Объектов (Object Inspector)
Alignment Palette	Палитра выравнивания компонентов
Browser	Броузер Объектов (Object Browser)
Breakpoints	Список точек останова (Breakpoints List)
Call Stack	Стек вызовов (Call Stack)
Watches	Список точек слежения за переменными (Watch List)
Threads	Список потоков команд и их статус
Modules	Список модулей, загружаемых при выполнении данного проекта
Component List	Список компонентов

Репозиторий БГАТУ

Window List	Список открытых окон
Toggle Form/Unit	Переключает активность из окна формы в окно текста программы и обратно
Unit	Показывает окно текста программы
Forms	Показывает окно формы
Type library	Отображает содержимое библиотеки типов для компонентов ActiveX, серверов ActiveX и других COM-объектов
New Edit Window	Открывает новое окно с текстом текущей программы
SpeedBar	Отображает (прячет) панель быстрого доступа
Component Palette	Отображает (прячет) Палитру Компонентов

В меню **Project** содержатся команды для компиляции и сборки проектов, а также для установки опций текущего проекта:

Команда	**Описание**
Add to Project	Добавляет файл к проекту
Remove from Project	Удаляет файл из проекта
Import Type Library	Импортирует в проект библиотеку типов элементов ActiveX.
Add To Repository	Добавляет проект в репозиторий объектов
Compile	Компилирует модули, исходный текст которых изменился после последней компиляции
Build All	Компилирует все модули и создает исполняемую программу
Syntax Check	Проверяет синтаксическую правильность программы
Information	Отображает информацию о проекте
Web Deployment Options	Позволяет задать опции для внедрения компонента ActiveX или активной фирмы на Web-узел
Web Deploy	Внедряет компонент ActiveX или активную фирму на Web-узел
Options	Задает опции компилятора и компоновщика, управляет рабочими каталогами

В меню **Run** расположены команды для отладки программ. Эти команды позволяют управлять различными функциями встроенного отладчика:

Команда	**Описание**
Run	Компилирует и выполняет программу
Parameters	Задает параметры командной строки
Register ActiveX Server	Регистрирует сервер ActiveX в реестре Windows
Unregister ActiveX Server	Удаляет информацию о ранее зарегистрированном сервере ActiveX в реестре Windows
Step Over	Пошагово выполняет программу
Trace Into	Пошагово выполняет программу с заходом в подпрограммы
Trace To Next Source Line	Пошагово выполняет программу до следующей строки исходного текста
Run To Cursor	Выполняет программу до строки в окне редактора, на которой находится курсор

Репозиторий БГАТУ

Show Execution Point	Отображает оператор, на котором было прервано выполнение программы
Program Pause	Приостанавливает выполнение программы
Program Reset	Завершает выполнение программы
Add Watch	Добавляет точку слежения за переменными
Add Breakpoint	Добавляет точку останова
Evaluate/Modify	Позволяет узнать или изменить значение переменной

В меню **Component** содержатся команды для создания компонентов, установки новых компонентов, импорта компонентов ActiveX, создания нового компонента на базе существующего и установки пакетов:

Команда	**Описание**
New Component	Вызывает окно Эксперта Компонентов
Install Component	Помещает компонент в существующий или новый проект
Import ActiveX Control	Импортирует компонент ActiveX
Create Component Template	Сохраняет компонент как шаблон для создания других компонентов
Install Package	Устанавливает пакеты, необходимые для прогона программы
Configure Palette	Вызывает диалоговую панель конфигурации Палитры Компонентов

Меню **Database** содержит средства для работы с базами данных:

Команда	**Описание**
Explore	Вызывает инструмент исследования баз данных - Database Explorer или SQL Database (в зависимости от версии DELPHI)
SQL Monitor	Вызывает инструмент запросов к БД – SQL Monitor
Form Wizard	Вызывает окно эксперта форм для создания формы, отображающей наборы данных из удаленных или локальных БД.

Из меню **Tools** доступны средства настройки среды, дополнительные утилиты, входящие в состав Delphi, а также репозиторий объектов.

Команда	**Описание**
Environment Options	Вызывает диалоговую панель настройки среды
Repository	Вызывает репозиторий
Configure Tools	Вызывает диалоговую панель редактирования опции Tools
Package Collection Editor	Вызывает окно редактора пакетов
Image Editor	Вызывает окно редактора графики
Database Desktop	Вызывает инструмент обслуживания БД – Database Desktop

Репозиторий БГАТУ

Меню **Workgroups** содержит средства для работы с коллективными проектами:

Команда	**Описание**
Browse PVCS Projects	Показывает окно коллективной работы нескольких программистов над одним проектом программы
Mange Archive Directories	Показывает диалоговое окно управления архивом коллективного проекта программы
Add Project to Version Control	Сохраняет текущую версию коллективного проекта
Set Data Directories	Показывает диалоговое окно выбора каталогов для размещения версий коллективного проекта

В меню **Help** содержатся команды для вызова различных разделов справочной системы и отображения диалоговой панели «О программе»:

Команда	**Описание**
Contents	Отображает содержание справочной системы
Keyword Search	Выполняет поиск справки по ключевому слову
What's New	Отображает справку по новым возможностям продукта
Getting Started	Выводит онлайновый вариант книги «Getting Started»
Using Object Pascal	Выводит онлайновый вариант книги «Using Object Pascal»
Developing Applications	Выводит онлайновый вариант книги «Developing Applications»
Object and Component Reference	Выводит онлайновый вариант книги «Object and Component Reference»
Borland Home Page	Соединяет с главной страницей Web-узла фирмы Borland
Delphi Home Page	Соединяет со страницей Web-узла фирмы Borland, посвященной Delphi
Borland Programs and Services	Соединяет со страницей Web-узла фирмы Borland, посвященной программам и сервисам
About	Отображает диалоговую панель «О программе»

Репозиторий БГАТУ

Приложение 2

П2. СВОЙСТВА КОМПОНЕНТОВ

П2.1. Общие свойства компонентов

Многие стандартные визуальные компоненты имеют одинаковые свойства.

Свойство Align

Задает способ выравнивания компонента внутри формы. Имеет одно из следующих значений:

Значение	**Описание**
alNone	Выравнивание не используется. Компонент располагается на том месте, куда был помещен во время создания программы. Принимается по умолчанию
alTop	Компонент перемещается в верхнюю часть формы, и его ширина становится равной ширине формы. Высота компонента не изменяется
alBottom	Компонент перемещается в нижнюю часть формы, и его ширина становится равной ширине формы. Высота компонента не изменяется
alLeft	Компонент перемещается в левую часть формы, и его высота становится равной высоте формы. Ширина компонента не изменяется
alRight	Компонент перемещается в правую часть формы, и его высота становится равной высоте формы. Ширина компонента не изменяется
alClient	Компонент занимает всю рабочую область формы

Свойство Color

Задает цвет фона формы, цвет компонента или графического объекта. Может иметь одно из следующих значений:

Значение	**Цвет**
clBlack	Черный (Black)
clMaroon	Темно-красный (Maroon)
clGreen	Зеленый (Green)
clOlive	Оливковый (Olive green)
clNavy	Темно-синий (Navy blue)
clPurple	Фиолетовый (Purple)
clTeal	Сине-зеленый (Teal)
clGray	Серый (Gray)
clSilver	Серебряный (Silver)
clRed	Красный (Red)
clLime	Ярко-зеленый (Lime green)
clBlue	Голубой (Blue)
clFuchsia	Сиреневый (Fuchsia)
clAqua	Ярко-голубой (Aqua)
dWhite	Белый (White)

Репозиторий БГАТУ

Цвета, приведенные в следующей таблице, являются системными цветами Windows и зависят от используемой цветовой схемы.

Значение	**Цвет**
cl Background	Текущий цвет фона окна
clActiveCaption	Текущий цвет заголовка активного окна
cllnactiveCaption	Текущий цвет заголовка неактивного окна
cIMenu	Текущий цвет фона меню
clWindow	Текущий цвет фона Windows
clWlndowFrame	Текущий цвет рамки окна
cIMenuText	Текущий цвет текста элемента меню
clWindowText	Текущий цвет текста внутри окна
cICaptionText	Текущий цвет заголовка активного окна
ciActiveBorder	Текущий цвет рамки активного окна
cllnactiveBorder	Текущий цвет рамки неактивного окна
clAppWorkSpace	Текущий цвет рабочей области окна
clHighlight	Текущий цвет фона выделенного текста
clHightlightText	Текущий цвет выделенного текста
clBtnFace	Текущий цвет кнопки
clBtnShadow	Текущий цвет фона кнопки
clGrayText	Текущий цвет недоступного элемента меню
clBtnText	Текущий цвет текста кнопки

Помимо перечисленных в таблице цветов значение свойства Color может задаваться шестнадцатеричными значениями.

Свойство Ctl3D

Позволяет задать вид компонента. Если значение этого свойства равно False, компонент имеет двухмерный вид, если True — трехмерный (значение по умолчанию).

Свойство Cursor

Позволяет определить вид курсора, который он будет иметь, находясь в активной области компонента. В Delphi предопределено большое количество стандартных курсоров. Кроме того, пользователь может создавать свои собственные курсоры или использовать созданные другими.

Свойство DragCursor

Позволяет определить вид курсора, который будет отображаться, когда в компонент «перетаскивается» другой компонент. Значения этого свойства те же, что и у свойства Cursor.

Свойство DragMode

Позволяет определить режим поддержки протокола drag-and-drop. Возможны следующие значения:

Значение	**Описание**
dmAutomatic	Компонент можно «перетаскивать», «зацепив» мышью

Репозиторий БГАТУ

dmManual	Компонент не может быть «перетащен» без вызова метода BeginDrag

Свойство Enabled

Если это свойство имеет значение True, компонент реагирует на сообщения от мыши, клавиатуры и таймера. В противном случае (значение False), эти сообщения игнорируются.

Свойство Font

Многие визуальные компоненты используют шрифт по умолчанию. При создании компонента изначальное значение свойства Font (класс TFont) имеет следующие параметры:

Свойство	**Значение**
Color	clWindowText
Height	- MulDiv(10, GetDeviceCaps(DC, LOGPIXELSY), 72)
Name	System
Pitch	FpDefault
Size	10
Style	[]

Свойство Height

Это свойство задает вертикальный размер компонента или формы.

Свойство HelpContext

Задает номер контекста справочной системы. Этот номер должен быть уникальным для каждого компонента. Если компонент активен (находится в фокусе), нажатие клавиши F1 приводит к отображению экрана справочной системы (если такой существует для данного компонента).

Свойство Hint

Задает текст, который будет отображаться при обработке события OnHint, происходящего, если курсор находится в области компонента.

Свойство Left

Задает горизонтальную координату левого угла компонента относительно формы в пикселах. Для форм это значение указывается относительно экрана.

Свойство ParentColor

Это свойство позволяет указать, каким цветом будет отображаться компонент. Если значение этого свойства равно True, компонент использует цвет (значение свойства Color) родительского компонента. Если же значение свойства ParentColor равно False, компонент использует значение собственного свойства Color.

Свойство ParentCtl3D

Это свойство позволяет указать, каким образом компонент будет определять, является он трехмерным, или нет. Если значение этого свойства равно True, то вид компонента задается значением свойства Ctl3D его

Репозиторий БГАТУ

владельца, если же значение этого свойства равно False — то значением его собственного свойства Ctl3D.

Свойство ParentFont

Это свойство позволяет указать, каким образом компонент будет определять используемый им шрифт. Если значение этого свойства равно True, используется шрифт, заданный у владельца компонента, если же это значение равно False, то шрифт задается значением его собственного свойства Font.

Свойство PopupMenu

Это свойство задает название локального меню, которое будет отображаться при нажатии правой кнопки мыши. Локальное меню отображается только в случае, когда свойство AutoPopup имеет значение True или когда вызывается метод Popup.

Свойство TabOrder

Задает порядок получения компонентами фокуса при нажатии клавиши Tab. По умолчанию этот порядок определяется размещением компонентов в форме: первый компонент имеет значение этого свойства, равное 0, второй — 1 и так далее. Для изменения этого порядка необходимо изменить значение свойства TabOrder определенного компонента. TabOrder может использоваться только совместно со свойством TabStop.

Свойство TabStop

Это свойство позволяет указать, может компонент получать фокус или нет. Компонент получает фокус, если значение его свойства TabStop равно True.

Свойство Tag

С помощью этого свойства можно «привязать» к любому компоненту значение типа LongInt.

Свойство Top

Это свойство задает вертикальную координату левого верхнего угла интерфейсного элемента относительно формы в пикселах. Для формы это значение указывается относительно экрана.

Свойство Visible

Это свойство позволяет определить, видим ли компонент на экране. Значением этого свойства управляют методы Show и Hide.

Свойство Width

Это свойство задает горизонтальный размер интерфейсного элемента или формы в пикселах.

Репозиторий БГАТУ

П2.2. Компоненты страницы STANDARD

П2.2.1. MainMenu

Компонент служит для создания главного меню формы. После установки компонента на форму необходимо создать его опции. Для этого следует путем двойного нажатия на левую клавишу “мыши” вызвать конструктор меню. Создание опций меню достаточно простой процесс. Надо выбрать опцию, перейти в окно Инспектора Объектов, в строке Caption набрать необходимое и нажать клавишу Enter. Для создания новых опций необходимо выбирать строку справа, для создания подопций – снизу. Для определения символа быстрого доступа к опции перед ним ставится символ “&”. Для вставки разделительной черты очередной элемент называется “-“. Для создания разветвленных меню, т.е. таких, у которых подопции вызывают новые списки подопций нажмите Ctrl-*Вправо*, где *Вправо* – клавиша смещения курсора вправо.

Каждый элемент меню является объектом класса TMenuItem и обладает следующими свойствами:

Property Break: TMenuBreak;	Позволяет создать многоколончатый список подменю
Property Checked: Boolean;	Если True, рядом с опцией появляется галочка
Property Command: Word;	Используется при разработке приложений, обращающихся непосредственно к API-функциям Windows
Property Count: Integer;	Содержит количество опций в подчиненном меню, связанном с данным элементом (только для чтения)
Property Default: Boolean;	Определяет, является ли данная опция подменю умалчиваемой (умалчиваемая опция выделяется цветом и выбирается двойным щелчком мыши на родительской опции)
Property GroupIndex: Byte;	Определяет групповой индекс для зависимых опций
Property Items[Index: [nteger]: TMenuItem;	Позволяет обратиться к любой опции подчиненного меню по ее индексу
Property MenuIndex: Integer;	Определяет индекс опции в списке Items родительской опции
Property RadioItem: Boolean;	Определяет, зависит ли данная опция от выбора других опций в той же группе GroupIndex. Только одна опция группы может иметь True в свойстве Checked. Рядом с такой опцией вместо галочки изображается круг
Property ShortCut : TShortCut	Задает клавиши быстрого выбора данной опции

П2.2.2. PopupMenu

Данный компонент является локальным меню, которое становится доступным, когда пользователь нажимает правую кнопку мыши в рабочей области формы или компонента. Обычно локальное меню используется для динамического изменения свойств того или иного интерфейсного элемента.
Редактируется локальное меню также как и главное с помощью Конструктора Меню.

Репозиторий БГАТУ

Чтобы связать нажатие правой кнопки мыши с раскрытием вспомогательного меню, в свойство PoupMenu необходимо поместить имя компонента-меню.

Свойство Alignment задает местонахождение локального меню.

П2.2.3. Label

Компоненты предназначены для размещения на форме различного рода текстовых надписей.

Property AutoSize: Boolean;	Указывает, будет ли метка изменять свои размеры в зависимости от помещенного в ее свойство Caption текста (True - будет)
Property FocusControl: TWinControl;	Содержит имя оконного компонента, который связан с меткой (выбор компонента Label приводит к перемещению фокуса на связанный с ним компонент)
TtextLayout = (tlTop, tlCenter, tlBottom) ; **Property** Layout: TTextLayout;	Определяет выравнивание текста по вертикали относительно границ метки: tlTop - текст располагается вверху; tlCenter - текст центрируется по вертикали; tlBottom -текст располагается внизу
Property ShowAccelChar: Boolean;	Если содержит True, символ & в тексте метки предшествует символу-акселератору
Property Transparent: Boolean;	Определяет прозрачность фона метки. Если False, фон закрашивается собственным цветом Color, в противном случае используется фон родительского компонента
Property WordWrap: Boolean;	Разрешает/запрещает разрыв строки на границе слова. Для вывода многострочных надписей задайте AuroSize=False, WordWrap=True и установите подходящие размеры метки

П2.2.4. Edit

Компонент представляет собой однострочный редактор текста. С его помощью можно вводить и/или отображать достаточно длинные текстовые строки. Следует помнить, что этот компонент не распознает символы конца строки (#13#10).

Property AutoSelect: Boolean;	Указывает, будет ли выделяться весь текст в момент получения компонентом фокуса ввода
Property AutoSize: Boolean;	Если True и BorderStyle = bsSingle, высота компонента автоматически меняется при изменении свойства Font - Size.
Property BorderStyle: TBorderStyle;TBorderStyle = bsNone..bsSingle;	Определяет стиль обрамления компонента: bsNone - нет обрамления; bsSingle - компонент обрамляется одной линией
Property CharCase: TEditCharCase; TEditCharCase = (ecNormal, ecUpperCase, ecLowerCase);	Определяет автоматическое преобразование высоты букв: ecNormal - нет преобразования; ее UpperCase - все буквы заглавные; ecLowerCase -все буквы строчные. Правильно работает с кириллицей
Property HideSelection: Boolean;	Если False, выделение текста сохраняется при потере фокуса ввода

Репозиторий БГАТУ

Property MaxLength: Integer;	Определяет максимальную длину текстовой строки. Если имеет значение 0, длина строки не ограничена
Property Modified: Boolean;	Содержит True, если текст был изменен
Property OnChange: TNotifyEvent;	Определяет обработчик события OnChange, которое возникает после любого изменения текста
Property OEMConvert: Boolean;	Содержит True, если необходимо перекодировать текст из кодировки MS-DOS в кодировку Windows и обратно
Property PasswordChar: Char;	Если символ PasswordChar определен, он заменяет собой любой символ текста при отображении в окне. Используется для ввода паролей
Property ReadOnly: Boolean;	Если содержит True, текст не может изменяться
Property SelLength: Integer;	Содержит длину выделенной части текста
Property SelStart: Integer;	Содержит номер первого символа выделенной части текста
Property SelText: String;	Содержит выделенный текст

Методы компонента:

procedure Clear;	Удаляет весь текст
procedure ClearSelection;	Удаляет выделенный текст
procedure CopyToClipboard;	Копирует выделенный текст в Clipboard
procedure CutToClipboard;	Копирует выделенный текст в Clipboard, после чего удаляет выделенный текст из компонента
function GetSelTextBuf(Buffer: PChar; BufSize: Integer) : Integer;	Копирует не более BufSize символов выделенного текста в буфер Buffer
procedure PasteFromClipboard;	Заменяет выделенный текст содержимым Clipboard, а если нет выделенного текста, копирует содержимое Clipboard в позицию текстового курсора
procedure SelectAll;	Выделяет весь текст
procedure SetSelTextBuf(Buffer: PChar);	Заменяет выделенный текст содержимым Buffer, а если нет выделенного текста, копирует содержимое Buffer в позицию текстового курсора

П2.2.5. Memo

Компоненты предназначены для ввода, редактирования и/или отображения достаточно длинного текста, содержащего большое количество строк.

Большинство свойств этого компонента аналогичны свойствам компонента Edit. Свойство Wordwrap аналогично свойству LabelWordwrap.

Property Lines: TStrings;	Содержит редактируемый текст. Используется для построчного доступа. Методы Add, Delete, Insert используются для добавления, удаления и вставки строк

Репозиторий БГАТУ

Property ScrollBars: TscrollStyle;TscrollStyle = (ssNone, ssHorizontal, ssVertical, ssBoth);	Определяет наличие в окне редактора полос прокрутки: ssNone – нет полос; ssHorizontal - есть горизонтальная полоса; ssVertical- есть вертикальная полоса; ssBoth – есть обе полосы
Property WantReturns: Boolean;	Если содержит True, нажатие Enter вызывает переход на новую строку, в противном случае – обрабатывается системой. Для перехода на новую строку в этом случае следует нажать Ctrl+Enter
Property WantTabs: Boolean;	Если содержит True, нажатие Tab вызывает ввод в текст символа табуляции, в противном случае – обрабатывается системой. Для ввода символа табуляции в этом случае следует нажать Ctrl-Tab

П2.2.6. Button

Компонент представляет собой стандартную кнопку, которая может содержать текст, описывающий выполняемое действие.

Property Cancel: Boolean;	Если имеет значение True, событие OnClick кнопки возникает при нажатии клавиши Esc
Property Default: Boolean;	Если имеет значение True, событие OnClick кнопки возникает при нажатии клавиши Enter
Property Enabled: Boolean;	Если имеет значение False, то кнопка недоступна для нажатия
Property TModalResult; TModalResult = Low(Integer)..High (Integer);	Определяет результат, с которым было закрыто модальное окно

В терминологии Windows модальными окнами называются такие специальные окна, которые, раз появившись на экране, блокируют работу пользователя с другими окнами вплоть до своего закрытия. Если у кнопки определено свойство ModalResult, нажатие на нее приводит к закрытию модального окна и возвращает в программу значение ModalResult как результат диалога с пользователем. В DELPHI определены следующие стандартные значения ModalResult:

mrNone	Модальное окно не закрывается
mrOk	Была нажата кнопка Ok
mrCancel	Была нажата кнопка Cancel
mrAbort	Была нажата кнопка Abort
mrRetry	Была нажата кнопка Retry
mrIgnore	Была нажата кнопка Ignore
mrYes	Была нажата кнопка Yes
mrNo	Была нажата кнопка No
mrAll	Была нажата кнопка All

П2.2.7. CheckBox

Кнопка с независимой фиксацией позволяет выбрать или отменить определенную функцию. Свойство State позволяет установить значение кнопки.

Репозиторий БГАТУ

Кнопка может находиться во включенном, выключенном и неактивном состоянии.

Property Alignment: TLeftRight;TLeftRight = (taLeftJustify, taRightJustify);	Определяет положение текста: taLeftJustify -с левой стороны компонента; taRightJustify -с правой стороны
Property AllowGrayed: Boolean;	Разрешает/запрещает использование неактивного состояния cbGrayed
Property Checked: Boolean;	Содержит выбор пользователя типа Да/Нет. Состояния cbUnchecked и cbGrayed отражаются как False
Property State: CheckBoxState; TCheckBoxState = (cbUnchecked, cbChecked, cbGrayed) ;	Содержит состояние компонента: cbUnchecked – нет; cbChecked - да; cbGrayed – неактивен

П2.2.8. RadioButton

Кнопки с зависимой фиксацией предназначены для выбора одной опции из нескольких взаимоисключающих, поэтому таких кнопок должно быть как минимум две. Для группировки кнопок с зависимой фиксацией внутри формы, их необходимо разместить внутри компонента Panel, GroupBox или ScrollBox. Состояние кнопки содержится в свойстве Checked.

П2.2.9. ListBox

Интерфейсный элемент этого типа содержит список элементов, которые могут быть выбраны при помощи клавиатуры или мыши. В компоненте предусмотрена возможность программной прорисовки элементов, поэтому список может содержать не только строки, но и произвольные изображения.

Property Canvas: TCanvas;	Канва для программной прорисовки элементов
Property Columns: Longint;	Определяет количество колонок элементов в списке
Property ExtendedSelect: Boolean;	Если ExtendedSelect=True и MultiSelect=True, выбор элемента без одновременного нажатия Crti или Alt отменяет предыдущий выбор
Property IntegralHeight: Boolean;	Если IntegralHeight=True и Style<>lbOwnerDraw-Variable, в списке показывается целое число элементов
Property ItemIndex: Integer;	Содержит индекс сфокусированного элемента. Если MultiSelect=False, совпадает с индексом выделенного элемента
Property ItemHeight: Integer;	Определяет высоту элемента в пикселях для Style=lbOwnerDrawFixed
Property Items: TStrings;	Содержит набор строк, показываемых в компоненте
Property MultiSelect: Boolean;	Разрешает/отменяет выбор нескольких элементов
Property SelCount: Integer;	Содержит количество выбранных элементов
Property Selected[X: Integer]: Boolean;	Содержит признак выбора для элемента с индексом X (первый элемент имеет индекс 0)
Property Sorted: Boolean;	Разрешает/отменяет сортировку строк в алфавитном порядке

Репозиторий БГАТУ

TlistBoxStyle = (IbStandard, IbOwnerDrawFixed, IbOwnerDrawVariable); **Property** Style: TListBoxStyle;	Определяет способ прорисовки элементов: IbStandard - элементы рисует Windows, IbOwnerDrawFixed - рисует программа, все элементы имеют одинаковую высоту, определяемую свойством ItemHeight, IbOwnerDrawVariable -рисует программа, элементы имеют разную высоту
Property Toplndex: Integer;	Индекс первого видимого в окне элемента

П2.2.10. ComboBox

Комбинированный список представляет собой комбинацию списка ListBox и редактора Edit, поэтому большинство его свойств и методов заимствованы у этих компонентов.

Существуют пять модификаций компонента, определяемые его свойством Style: csSimple, csDropDown, csDropDownList, csOwnerDrawFixed и csOwnerDrawVariable. В первом случае список всегда раскрыт, в остальных он раскрывается после нажатия кнопки справа от редактора. В модификации csDropDownList редактор работает в режиме отображения выбора и его нельзя использовать для ввода новой строки. Модификации сыOwnerDrawFixed и csOwnerDrawVariable используются для программной прорисовки модификации csDropDown..

Свойство DropDownCount определяет количество элементов списка, появление которых еще не приводит к необходимости прокрутки списка. Свойство DroppedDown определяет, раскрыт ли список в данный момент.

П2.2.11. ScrollBar

Компонент ScrollBar является полосой прокрутки и обычно используется для визуального управления значением какой либо величины.

Property Kind: ScrollBarKind; TSrollBarKind = (sbHorizontal, sbVertical);	Определяет ориентацию компонента: sbHorizontal - бегунок перемещается по горизонтали; sbVertical - бегунок перемещается по вертикали
Property LargeChange: TScrollBarInc;	«Большой» сдвиг бегунка (при щелчке мышью рядом с концевой кнопкой)
Property Max: Integer;	Максимальное значение диапазона изменения числовой величины
Property Min: Integer;	Минимальное значение диапазона изменения числовой величины
Property Position: Integer;	Текущее значение числовой величины
Property SmallChange: TScrollBarInc;	«Малый» сдвиг бегунка (при щелчке мышью по концевой кнопке)

П2.2.12. GroupBox

Этот компонент служит контейнером для размещения дочерних компонентов и представляет собой прямоугольное окно с рамкой и текстом в разрыве рамки. Обычно с его помощью выделяется группа управляющих элементов, объединенных по функциональному назначению. После того как компоненты помещены в группу, она становится их родительским классом.

Репозиторий БГАТУ

П2.2.13. RadioGroup

Компонент представляет собой специальный контейнер, предназначенный для размещения зависимых переключателей. Каждый размещаемый в нем переключатель помещается в специальный список Items и доступен по индексу, что упрощает обслуживание группы.

Property Columns: Integer;	Определяет количество столбцов переключателей
Property ItemIndex: Integer;	Содержит индекс выбранного переключателя
Property Items: TStrings;	Содержит список строк с заголовками элементов. Добавление/удаление элементов достигается добавлением/удалением строк списка Items

П2.2.14. Panel

Панель используется в качестве контейнера для расположения других интерфейсных элементов.

Property BevelInner:TPanelBevel;	Определяет стиль внутренней кромки
Property BevelOuter: PanelBevel;	Определяет стиль внешней кромки
Property BevelWidth: TBevelWidth;TBevelWidth = 1..MaxInt;	Задает ширину кромок в пикселях
TBorderStyle = bsNone..bsSingle; **Property** BorderStyle: TBorderStyle;	Определяет стиль рамки: bsNone - нет рамки; bsSingle - компонент по периметру обводится линией толщиной в 1 пиксел
Property BorderWidth: TBorderWidth;TborderWidth: 0..Maxint;	Определяет расстояние в пикселях от внешней кромки до внутренней
Property FullRepaint: Boolean;	Разрешает/запрещает перерисовку панели и всех ее дочерних элементов при изменении ее размеров

П2.3. Компоненты страницы ADDITIONAL

П2.3.1. BitBtn

Пиктографическая кнопка представляет собой разновидность стандартной кнопки Button, которая помимо текста может содержать графическое изображение. Растровое изображение определяется с помощью свойства Glyph. В комплект поставки DELPHI (поддиректория Images/Buttons) входит около 160 различных вариантов растровых изображений для кнопок. Кроме того, пользователь может самостоятельно создать растровое изображение с помощью встроенного в DELPHI графического редактора.

Свойство Kind позволяет выбрать одну из 11 стандартных разновидностей кнопки (рис.П2.1.)

Репозиторий БГАТУ

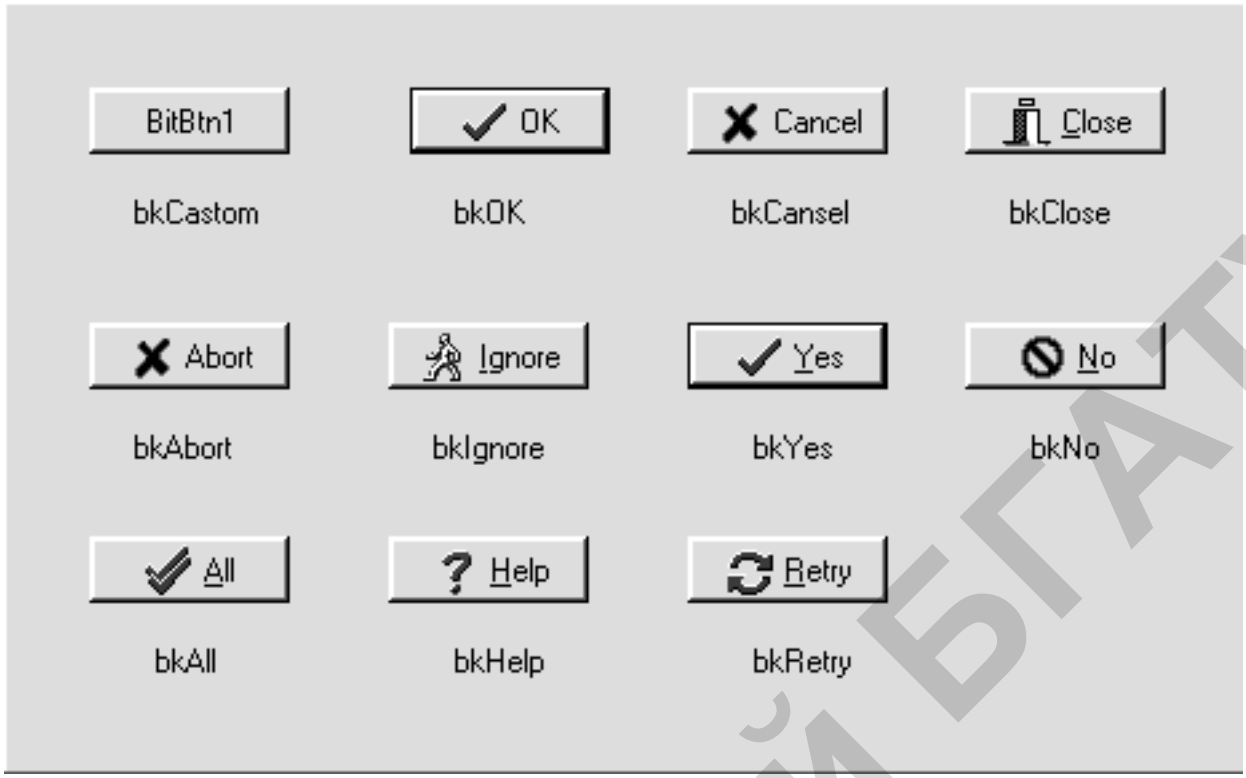


Рис. П2. 1.

Нажатие любой из кнопок, кроме bkCustom и bkHelp, закрывает модальное окно и возвращает в программу результат mrXXX: bkOk -mrOk, bkCancel - mrCancel и т.д. Кнопка bkClose для модального окна возвращает mrCancel, а для главного окна программы - закрывает его и завершает работу программы. Кнопка bkHelp автоматически вызывает раздел справочной службы, связанный с HelpContext формы, на которую она помещена.

Property Glyph: TBitmap;	Определяет связанные с кнопкой растровые изображения (до 4)
Property Kind: TBitBtnKind; TBitBtnKind = (bkCustom, bkOK, bkCancel, bkHelp, bkYes, bkNo, bkClose, bkAbort, bkRetry, bkIgnore, bkAll);	Определяет разновидность кнопки
Property Layout: TButtonLayout; TButtonLayout = (blGlyphLeft, blGlyphRight, blGlyphTop, blGlyphBottom) ;	Определяет край кнопки, к которому прижимается пиктограмма
Property Margin: Integer;	Определяет расстояние в пикселях от края кнопки до пиктограммы
Property NumGlyphs: TNumGlyphs;TNumGlyphs: 1..4 ;	Определяет количество растровых изображений. Таких состояний может быть четыре: нормальное, запрещенное, нажатое, и утопленное
Property Spacing: Integer;	Определяет расстояние в пикселях от пиктограммы до надписи на кнопке

Репозиторий БГАТУ

Property Style: TButtonStyle.; TButtonStyle = (bsAutoDetect, bsWin31, bsNew);	Определяет стиль оформления кнопки, зависящий от операционной системы

П2.3.2. SpeedButton

Еще один вариант кнопки, который отличается от BitBtn тремя обстоятельствами: во-первых, не предусмотрен вывод надписи, во-вторых, имеется возможность фиксации в утопленном состоянии и, в-третьих, она не может закрыть модальное окно.

П2.3.3. MaskEdit

Специализированный редактор предназначен для ввода текста, соответствующего некоторому шаблону, задаваемому свойством EditMask:String. Если это свойство не задано, MaskEdit работает как обычный редактор Edit.

Шаблон состоит из трех частей, отделенных друг от друга символами «;». Первая часть задает маску ввода, вторая - это символ «0» или «1», определяющий, записывается ли в Text результат наложения маски или исходный текст («0» - исходный текст). В третьей части указывается символ, который в окне редактора будет стоять в полях, предназначенных для ввода символов.

Описатели полей ввода представлены в следующей таблице:

Символ	Поле
L	должно содержать букву
1	может содержать букву
A	должно содержать букву или цифру
a	может содержать букву или цифру
C	должно содержать любой символ
c	может содержать любой символ
O	должно содержать цифру
9	может содержать цифру
#	может содержать цифру, «+», «-»

Специальные символы:

Символ	Значение
****	Следующий символ - литерал. Позволяет вставить в маску литералы из символов описателей полей ввода и специальных символов
:	На это место вставляется символ-разделитель Windows для часов, минут, секунд
/	На это место вставляется символ-разделитель Windows для полей даты
/	Разделитель частей шаблона
!	Подавляет все ведущие пробелы
>	Все следующие за ним поля ввода преобразуют буквы к заглавным
<	Все следующие за ним поля ввода преобразуют буквы к строчным
o	Отменяет преобразование букв

Репозиторий БГАТУ

П2.3.4. DrawGrid

Компонент используется для отображения информации в виде таблицы. Таблица делится на две части - фиксированную и рабочую. Фиксированная часть служит для показа заголовков столбцов/рядов и для ручного управления их размерами. Рабочая часть - содержит произвольное количество столбцов и рядов, содержащих как текстовую, так и графическую информацию, и может изменяться программно.

Property BorderStyle: TBorderStyle;	Определяет наличие или отсутствие внешней рамки таблицы
Property Col: Longint;	Содержит номер столбца сфокусированной ячейки
Property ColCount: Longint;	Содержит количество столбцов таблицы
Property ColWidths[Index: Longint]: Integer;	Содержит ширину столбца с индексом Index
Property DefaultColWidth: Integer;	Содержит умалчиваемое значение ширины столбца
Property DefaultDrawing: Boolean;	Разрешает/запрещает автоматическую прорисовку служебных элементов таблицы - фиксированной зоны, фона и прямоугольника сфокусированной ячейки и т.п.
Property DefaultRowHeight: Integer;	Содержит умалчиваемую высоту рядов
Property EditorMode: Boolean;	Разрешает/запрещает редактирование ячеек. Игнорируется, если свойство Options включает goAlwayseShowEditor или не включает soEditing
Property FixedColor: TColor;	Определяет цвет фиксированной зоны
Property FixedCols: Integer;	Определяет количество столбцов фиксированной зоны
Property FixedRows: Integer;	Определяет количество рядов фиксированной зоны
Property GridHeight: Integer;	Содержит высоту таблицы
Property GridLineWidth: Integer;	Определяет толщину линий, расчерчивающих таблицу
Property GridWidth: Integer;	Содержит ширину таблицы
Property LeftCol: Longint;	Содержит номер самого левого столбца, видимого в зоне прокрутки
Property Options: TGridOptions;	Содержит параметры таблицы (см. ниже)
Property Row: Longint;	Содержит номер ряда сфокусированной ячейки
Property RowCount: Longint;	Содержит количество рядов таблицы
Property RowHeights[Index: Longint]: Integer;	Содержит высоту ряда с индексом Index
Property TabStops[Index: Longint]: Boolean;	Разрешает/запрещает выбирать столбец с индексом Index при обходе ячеек клавишей Tab. Игнорируется, если Options не содержит goTabs

Репозиторий БГАТУ

Property Selection: TGridRect; TGridRect = record case Integer of 0: (Left, Top, Right/ Bottom: Longint); 1: (TopLeft, BottomRight: TGridCoord); end;	Определяет группу выделенных ячеек в координатах левая верхняя и правая нижняя ячейки(нумерация столбцов и рядов идет от нуля, включая столбцы и ряды фиксированной зоны). После выделения сфокусированной окажется правая нижняя ячейка
Property TopRow: Longint;	Содержит номер самого верхнего ряда, видимого в прокручиваемой зоне ячеек
Property VisibleColCount: Integer;	Содержит количество столбцов, полностью видимых в зоне прокрутки
Property VisibleRowCount: , Integer;	Содержит количество рядов, полностью видимых в зоне прокрутки

Элементы множества TGridOptions имеют следующий смысл:

goFixedVertLine	Столбцы фиксированной зоны разделяются вертикальными линиями
goFixedHorzLine	Ряды фиксированной зоны разделяются горизонтальными линиями
goVertLine	Столбцы рабочей зоны разделяются вертикальными линиями
goHorzLine	Ряды рабочей зоны разделяются горизонтальными линиями
goRangeSelect	Разрешено выделение нескольких ячеек. Игнорируется, если включен элемент goEdit
GoDrawFocus-Selected	Разрешено выделять сфокусированную ячейку так же, как и выделенные
GoRowSizing	Разрешено ручное (мышью) изменение высоты строк
GoColSizing	Разрешено ручное изменение ширины рядов
GoRowMoving	Разрешено ручное перемещение рядов
goColMoving	Разрешено ручное перемещение столбца
goEditing	Разрешено редактирование ячейки. Игнорируется, если включен элемент goRowSelect. Редактирование начинается после щелчка мыши или нажатия клавиши F2 и завершается при щелчке подругой ячейке или нажатии Enter
goTabs	Разрешено выбирать ячейки клавишей Tab (Shift-Tab)
goRowSelect	Обязывает выделять сразу все ячейки ряда
GoAlwaysShowEditor	Разрешено редактировать сфокусированную ячейку. Игнорируется, если не включен элемент goEditing
GoThumbTracking	Разрешено обновление при прокрутке. Если этот элемент отсутствует, обновление ячеек произойдет только после окончания прокрутки

П2.3.5. *StringGrid*

В отличии от компонента DrawGrid, компонент StringGrid может отображать только текстовую информацию.

Репозиторий БГАТУ

Property Cells[ACol, ARow: Integer]: string;	Определяет содержимое ячейки с табличными координатами (ACol.ARow)
Property Cols[Index: Integer]: TStrings;	Содержит все строки колонки с индексом Index
Property Objects [ACol, ARow: Integer]: TObject;	Обеспечивает доступ к объекту, связанному с ячейкой (ACol,ARow)
Property Rows[Index: Integer]: TStrings;	Содержит все строки ряда с индексом Index

П2.3.6. Image

Этот компонент служит для размещения на форме одного из трех поддерживаемых DELPHI типов изображений: растровой картинки, пиктограммы или метафайла.

П2.3.7. Shape

Компонент рисует одну из простейших геометрических фигур (прямоугольник, квадрат, скругленный прямоугольник, скругленный квадрат, эллипс, окружность).

П2.3.8. Bevel

Предназначен для выделения группы элементов или отделения их друг для друга, и носит чисто оформительский характер.

П2.3.9. ScrollBox

Компонент является контейнером для размещения других компонентов и имеет возможность прокрутки.

П2.3.10. CheckListBox

Группирует независимые переключатели, позволяя обратиться к любому из них по индексу.

Property AllowQrayed: Boolean;	Разрешает/запрещает использовать в переключателях третье состояние cbGrayed
Property Checked[Index: Integer]: Boolean;	Содержит выбор пользователя типа Да/Нет для переключателя с индексом Index, Состояния cbUnchecked и cbGrayed отражаются как False
Property Sorted: Boolean;	Сортирует по алфавиту надписи на переключателях
Property State[Index: Integer]: TCheckBoxState;	Содержит состояние переключателя с индексом Index: cbUncheeked; cbChecked; cbGrayed

П2.3.11. Splitter

Предназначен для ручного (с помощью мыши) управления размерами контейнеров TPanel, TGroupBox или подобных им во время выполнения программы.

Property Beveled: Boolean;	Управляет трехмерным изображением компонента. Если False, компонент виден как узкая полоска фона между разделяемыми им компонентами.

Репозиторий БГАТУ

Property MinSize: NaturalNumber; NaturalNumber = 1..High(Integer) ;	Содержит минимальный размер любого из компонентов, которых разделяет TSplitter. Если выравнивание alLefi или alRight - минимальная ширина компонента слева и справа от TSplitter, если alTop или alBottom - минимальная высота компонента выше или ниже от него.

П2.3.12. StaticText

Подобен компоненту Label за исключением того, что во-первых, он имеет Windows-окно и во-вторых, в его свойстве BorderStyle добавлено значение bsSunken, которое создает иллюзию "вдавленности" компонента.

П2.3.13. Chart

Облегчает создание специальных полей для графического представления данных.

П2.4. Компоненты страницы DIALOGS

П2.4.1. Правила использования диалоговых панелей

Работа со стандартными диалоговыми окнами осуществляется в три этапа:

1. На форму помещается соответствующий компонент и осуществляется настройка его свойств. Следует обратить внимание на то, что компонент-диалог не виден в момент работы программы, видно лишь создаваемое им стандартное окно.

2. Осуществляется вызов стандартного для диалогов метода Execute, который создает и показывает настроенное окно на экране. Вызов этого метода обычно располагается внутри обработчика какого-либо события. После обращения к Execute на экране появляется соответствующее диалоговое окно. Окно диалога является модальным окном, поэтому сразу после обращения к нему дальнейшее выполнение программы приостанавливается до тех пор, пока пользователь не закроет окно.

3. Использование введенных из диалогового окна данных (имя файла, настройки принтера и т.д.) для продолжения работы программы.

П2.4.2. OpenDialog и SaveDialog

Эти компоненты имеют идентичные свойства и отличаются только внешним видом. Свойство FileName: (тип String) содержит маршрут поиска и имя выбранного файла при успешном завершении диалога программы. Для проверки наличия файла на диске используется глобальная функция FileExists. Свойство Filter: String используется для фильтрации (отбора) файлов, показываемых в диалоговом окне. Это свойство можно устанавливать с помощью специального редактора или программно. Для доступа к редактору достаточно щелкнуть по кнопке в строке Filter окна Инспектора Объектов. При программном вводе фильтры задаются одной длинной строкой, в которой символы «|» служат для разделения фильтров друг от друга, а также для разделения описания фильтруемых файлов от соответствующей маски выбора. С помощью свойства DefaultExt: String[3] формируется полное имя файла, если при ручном вводе пользователь не указал расширение. В этом случае к имени файла прибавляется разделительная точка и содержимое этого свойства.

Репозиторий БГАТУ

Настройка диалога может варьироваться с помощью свойства
property Options: TOpenOptions;
TOpenOptions = **set of** TOpenOption;
TOpenOption = (of Readonly, ofOverwritePrompt, ofHideReadOnly,
ofNoChangeDir, ofShowHelp, ofNoValidate, ofAllowMuItiSelect,
ofExtensionDifferent, ofPathMustExist, ofFileMustExist,
ofCreatePrompt, ofShareAware, ofNoReadOnlyReturn,
ofNoTestFileCreate, ofNoNetworkButton, ofNoLongNames,
ofOldStyleDialog, ofNoDereferenceLinks);

Значения этого свойства имеют следующий смысл:

ofReadOnly	Устанавливает переключатель “Только для чтения”
ofOverwritePrompt	Требует согласия пользователя при записи в существующий файл
ofHideReadOnly	Прячет переключатель “Только для чтения“
ofNoChangeDir	Запрещает смену каталога
ofShowHelp	Включает в окно кнопку Help
ofNoValidate	Запрещает автоматическую проверку правильности набираемых в имени файла символов
ofAllowMultiSelec	Разрешает множественный выбор файлов
ofExtensionDiffer	При завершении диалога наличие этого значения в свойстве Options говорит о том, что пользователь ввел расширение, отличающееся от умалчиваемого
ofPathMustExist	Разрешает указывать файлы только из существующих каталогов
ofFileMustExist	Разрешает указывать только существующие файлы
ofCreatePrompt	Требует подтверждения для создания несуществующего файла
ofShareAware	Разрешает выбирать файлы, используемые другими параллельно выполняемыми программами
ofNoReadOnlyRetur	Запрещает выбор файлов, имеющих атрибут “Только для чтения”
ofNoTestFileCreat	Запрещает проверку доступности сетевого или локального диска
ofNoNetworkButton	Запрещает вставку кнопки для создания сетевого диска
ofNoLongNames	Запрещает использование длинных имен файлов
ofOldStyleDialog	Создает диалог в стиле Windows 3.x.

П2.4.3. OpenPictureDialog и SavePictureDialog

Специализированные диалоги для открытия и сохранения графических файлов являются расширенными вариантами компонентов OpenDialog и SaveDialog, в которых предусмотрено наличие стандартного фильтра для выбора графических файлов и панель предварительного просмотра.

Репозиторий БГАТУ

П2.4.4. FontDialog

Компонент используется для вызова стандартной диалоговой панели выбора щрифтов и их характеристик. Свойство Device определяет тип устройства, для которого выбирается fdScreen - экран; fdPrinter - принтер; fdBoth - шрифты, поддерживаемые и экраном, и принтером. Диапазон возможных значений размеров шрифтов определяется свойствами MinFontSize и MaxFontSize. Значения этих свойств задаются в пунктах (1 пункт равен приблизительно 0,36 мм). Если свойства содержат 0, ограничения на размер шрифта отсутствуют.

Свойство Options используется для настройки диалога. Значения этого свойства имеют следующий смысл:

fdAnsiOnly	Показывает только шрифты с набором символов Windows
fdTrueTypeOnly	Показывает только TrueType-шрифты
fdEffects	Включает в окно переключатели Подчеркнутый и Зачеркнутый, а также список выбора цвета шрифта
fdFixedPitchOnly	Включает только моноширинные шрифты
fdForceFontExist	Предупреждает о выборе несуществующего шрифта
fdNoFaceSel	Запрещает выделение имени шрифта в момент открытия окна
fdNoOEMFonts	Запрещает выбор MS-DOS-шрифтов
fdNoSimulations	Исключает шрифты, которые синтезируются графическим интерфейсом Windows
fdNoSizeSel	Запрещает выделение размера шрифта в момент открытия окна
fdNoStyleSel	Запрещает выделение стиля шрифта в момент открытия окна
fdNoVectorFonts	Исключает векторные шрифты
fdShowHelp	Включает в диалоговое окно кнопку Help
fdWysiwyg	Включает шрифты, которые поддерживаются и экраном, и принтером
fdLimitSize	Включает ограничения на размер шрифта, заданные свойствами MaxFontSize и MinFontSize
fdScalableOnly	Включает только масштабируемые шрифты (векторные и TrueType)
fdApplyButton	Включает в окно кнопку “Применить”

П2.4.5. ColorDialog

Компонент используется для вызова и обслуживания стандартного диалогового окно выбора цвета

Репозиторий БГАТУ

П2.4.6. PrintDialog

Компонент служит для создания стандартного диалогового окна для выбора параметров печати.

Property Collate: Boolean;	Если имеет значение True, то окно показывается с выбранным переключателем “Разобрать” (Collate). Если этот переключатель выбран, печать нескольких копий документа будет идти по копиям: сначала первая копия, затем вторая и т.д., в противном случае - по страницам: сначала все копии первой страницы, затем второй и т.д.
Property Copies: Integer;	Определяет количество копий (0 - одна копия)
Property FromPage: Integer;	Определяет начальную страницу печати
Property MaxPage: Integer;	Определяет верхнюю границу диапазона страниц для свойств FromPage, ToPage
Property MinPage: Integer;	Определяет нижнюю границу диапазона страниц для свойств FromPage, ToPage
Property Options: TPrintDialogOptions;	Определяет настройку окна: po PrintToFile -печатать в файл; poPageNums - разрешает выбор диапазона страниц; poSelection -разрешает печать выбранного текста; poWarning - предупреждать пользователя о неустановленном принтере; poHelp – вставить в окно кнопку Help; poDisablePrintToFile – запрещает печать в файл
Property PrintRange: TPrintRange;	Определяет диапазон печатаемых страниц: prAll Pages - все страницы; prSelection -выделенный фрагмент текста; prPageNums -страницы по номерам
Property PrintToFile: Boolean;	Содержит True, если пользователь выбрал печать в файл
Property ToPage: Integer;	Определяет конечную страницу печати

П2.4.7. PrinterSetupDialog

Компонент создает окно настройки параметров принтера, вид которого зависит от типа принтера. Этот диалог взаимодействует с драйвером принтера и не возвращает в программу никакой информации, поэтому его метод Execute - процедура, а не функция.

П2.4.8. FindDialog

Стандартное диалоговое окно компонента используется для поиска фрагмента текста.

Property FindText: string;	Указывает образец для поиска
Property Left: Integer;	Содержит горизонтальную позицию левого верхнего угла места появления окна
Property Options:TFindOptions;	Определяет настройку диалога
Property Position: TPoint;	Содержит горизонтальную и вертикальную позицию левого верхнего угла места появления окна

Репозиторий БГАТУ

Property Top: Integer;	Содержит вертикальную позицию левого верхнего угла места появления окна

Для компонента определен следующий тип, использующийся в свойстве Options: TFindOptions. Его значения имеют такой смысл:

FrDown	Устанавливает поиск вперед по тексту
FrDown frFindNext	Сообщает программе, что пользователь нажал кнопку “Найти далее”
FrHideMatchCase	Убирает выбор в переключателе “С учетом регистра”
FrHideWholeWord	Убирает выбор в переключателе “Только слово целиком”
FrHideUpDown	Прячет кнопки выбора направления поиска
FrMatchCase	Устанавливает выбор в переключателе “С учетом регистра”
frDisableMatchCase	Запрещает выбор “С учетом регистра”
frDisableUpDown	Запрещает выбор направления поиска
frDisableWholeWord	Запрещает выбор “Только слово целиком”
frReplace	Используется в компоненте ReplaceDialog и указывает на необходимость замены текущего выбора
frReplaceAll	Используется в компоненте Replace Dialog и указывает на необходимость замены всех вхождений образца поиска
frWholeWord	Устанавливает выбор в переключателе “Только слово целиком”
frShowHelp	Включает в окно кнопку Help

П2.4.9. ReplaceDialog

Компонент создает и обслуживает окно поиска и замены текстового фрагмента. Класс TReplaceDialog наследует большинство свойств классаTFind-Dialog. Дополнительно в компоненте определено свойство ReplaceText (тип String), в котором содержится текст замены, и событие OnReplace, которое возникает при нажатии кнопки “Заменить” или “Заменить все”.

Репозиторий БГАТУ

Приложение 3

П3. ПРОЦЕДУРЫ И ФУНКЦИИ ДЛЯ РАБОТЫ СО СТРОКАМИ

Для работы со строками применяются следующие процедуры и функции (в квадратных скобках указываются необязательные параметры).

Процедуры и функции для работы со строками	
Function Concat(S1 [, S2, ..., SN]: String): String;	Возвращает строку, представляющую собой сцепление строк-параметров S1, S2, … , SN
Function Copy(St: String; Index, Count: Integer): String;	Копирует из строки St Count символов, начиная с символа с номером Index
Procedure Delete(St: String; Index, Count: Integer);	Удаляет Count символов из строки St, начиная с символа с номером Index
Procedure Insert(SubSt ,St: String; Index: Integer) ;	Вставляет подстроку SubSt в строку St, начиная с символа с номером Index
Function Length(St: String): Integer;	Возвращает текущую длину строки St
Function Pos(SubSt, St: String): Integer;	Отыскивает в строке St первое вхождение подстроки SubSt и возвращает номер позиции, с которой она начинается. Если подстрока не найдена, возвращается ноль
Procedure SetLength(St: String; NewLength: Integer);	Устанавливает новую (меньшую) длину NewLength строки St. Если NewLength больше текущей длины строки, обращение к SetLength игнорируется
Процедуры и функции преобразования строк в другие типы	
Function StrToCurr(St: String): Currency;	Преобразует символы строки St в целое число типа Currency. Строка не должна содержать ведущих или ведомых пробелов
Function StrToDate(St: String): TDateTime;	Преобразует символы строки St в дату. Строка должна содержать два или три числа, разделенных правильным для Windows разделителем даты; (в русифицированной версии таким разделителей, является «.») Первое число - день, второе – месяц, если указано третье число, оно задает год
Function StrToDateTime(St: String): TDateTime;	Преобразует символы строки St в дату и время. Строка должна содержать дату и время, разделенные пробелом
Function StrToFloat(St: String): Extended;	Преобразует символы строки St в вещественное число. Строка не должна содержать ведущих или ведомых пробелов
Function StrToInt(St: String): Integer;	Преобразует символы строки St в целое число. Строка не должна содержать ведущих или ведомых пробелов

Репозиторий БГАТУ

Function StrToIntDef(St: String; Default: Integer): Integer;	Преобразует символы строки St в целое число. Если строка не содержит правильного представления целого числа, возвращается значение Default
Function StrToIntRange(St: String; Min, Max: Longint) : Longint;	Преобразует символы строки St в целое число и возбуждает исключение ERangeError, если число выходит из заданного диапазона Min.. Max
Function StrToTime(St: String): TDateTime;	Преобразует символы строки St во время
Procedure Val(St: String; var X; Code: Integer);	Преобразует строку символов St во внутреннее представление целой или вещественной переменной X, которое определяется типом этой переменной. Параметр Code содержит ноль, если преобразование прошло успешно, и тогда в X помещается результат преобразования, в противнем случае он содержит номер позиции в строке St, где обнаружен ошибочный символ, и в этом случае содержимое X не меняется. В строке St могут быть ведущие и/или ведомые пробелы
Процедуры и функции обратного преобразования	
Function DateToStr(Value: TDateTime): String;	Преобразует дату из параметра Value в строку символов
Function DateTimeToStr(Value: TDateTime): String;	Преобразует дату и время из параметра Value в строку символов
Procedure DateTimeToString (var St: String; Format: String; Value: TDataTime) ;	Преобразует дату и время из параметра Value в строку St
Function FormatDateTime (Format: String; Value: TDateTime): String;	Преобразует дату и время из параметра Value в строку символов
Function FloatToStr(Value: Extended): String;	Преобразует вещественное значение Value в строку символов
Function FloatToStrF(Value: Extended; Format: TFloatFor-mat; Precision, Digits: Integer) : String;	Преобразует вещественное значение Value в строку символов с учетом параметров Precision и Digits (см. пояснения ниже)
Function FormatFloat(Format: String; Value: Extended): String;	Преобразует вещественное значение Value в строку
Function IntToStr(Value: Integer) : String;	Преобразует целое значение Value в строку символов
Function TimeToStr(Value: TDateTime): String;	Преобразует время из параметра Value в строку символов

Репозиторий ВГАТУ

Procedure Str(X [:width [:Decimals]]; var St: String);	Преобразует число X любого вещественного или целого типа в строку символов St; параметры Width и Decimals, если они присутствуют, задают формат преобразования: Width определяет общую ширину поля, выделенного под соответствующее символьное представление вещественного или целого числа X, а Decimals – количество символов в дробной части (этот параметр имеет смысл только в том случае, когда X - вещественное число).

Правила использования параметров функции FloatToStrF показаны ниже:

Значение Format	**Описание**
ffExponent	Научная форма представления с множителем eXX («умножить на 10 в степени XX»). Precision задает общее количество десятичных цифр мантиссы. Digits - количество цифр в десятичном порядке XX. Число округляется с учетом первой отбрасываемой цифры: 3.1416E+00.
ffFixed	Формат с фиксированным положением разделителя целой и дробной частей. Precision задает общее количество десятичных цифр в представлении числа. Digits - количество цифр в дробной части. Число округляется с учетом первой отбрасываемой цифры: 3,14.
ffGeneral	Универсальный формат, использующий наиболее удобную для чтения форму представления вещественного числа. Соответствует формату ffFixed, если количество цифр в целой части меньше или равно Precision, а само число - больше или равно 0,00001, в противном случае соответствует формату ffExponent: 3,1416.
ffNumber	Отличается от ffFixed использованием символа-разделителя тысяч при выводе больших чисел (для русифицированной версии Windows таким разделителем является пробел). Для Value = π* 1000 получим 3 141,60.
ffCurrency	Денежный формат. Соответствует ffNumber, но в конце строки ставится символ денежной единицы (для русифицированной версии Windows - символы «р.»). Для Value = π*1000 получим: 3 141,60р.

Репозиторий БГАТУ

Приложение 4

П4. МАТЕМАТИЧЕСКИЕ ФОРМУЛЫ

Язык Object Pascal имеет ограниченное количество встроенных математических функций. Поэтому при необходимости использовать другие функции следует применять известные соотношения. В таблице приведены выражения наиболее часто встречающихся функций через встроенные функции языка Object Pascal.

Функция	**Соотношение**	**Соотношение на языке Object Pascal**
$Log_a(x)$	$\frac{Ln(x)}{Ln(a)}$	Ln(x)/Ln(a)
x^a	$e^{a*Ln(x)}$	Exp(a*Ln(x))
$Tg(x)$	$\frac{Sin(x)}{Cos(x)}$	Sin(x)/Cos(x)
$Ctg(x)$	$\frac{Cos(x)}{Sin(x)}$	Cos(x)/Sin(x)
$ArcSin(x)$	$ArcTg\left(\sqrt{\frac{x}{1-x^2}}\right)$	ArcTan(Sqrt(x/(1-sqr(x))))
$ArcCos(x)$	$\frac{\pi}{2} - ArcSin(x)$	Pi/2- ArcTan(Sqrt(x/(1-sqr(x))))
$ArcCtg(x)$	$\frac{\pi}{2} - ArcTg(x)$	Pi/2-ArcTan(x)
$Sh(x)$	$\frac{e^x - e^{-x}}{2}$	(Exp(x)-Exp(-x))/2
$Ch(x)$	$\frac{e^x + e^{-x}}{2}$	(Exp(x)+Exp(-x))/2
$Csc(x)$	$\frac{1}{\sin(x)}$	1/Sin(x)
$Sc(x)$	$\frac{1}{\cos(x)}$	1/Cos(x)

Репозиторий БГАТУ

Л И Т Е Р А Т У Р А

1. Информатика. Базовый курс / Симонович С.В. и др. -СПб.:Издательство “Питер”,1999.
2. Симонович С.,Евсеев Г.,Алексеев А. Специальная информатика: универсальный курс. М.: АСТ-ПРЕСС; Информком-пресс,1999.
3. Симонович С.,Евсеев Г.,Алексеев А. Практическая информатика: универсальный курс. М.: АСТ-ПРЕСС; Информком-пресс,1999.
4. Симонович С.,Евсеев Г.,Алексеев А. Общая информатика. М.: АСТ-ПРЕСС; Информком-пресс,1998.
5. Гук М. Аппаратные средства PC: Энциклопедия. СПб.: Питер, 1999.
6. Новейший самоучитель работы на компьютере. Под ред. Симоновича С.- М.:Десс; Информком-Пресс, 1999.
7. Фаронов В.В. DELPHI 3: Учебный курс. – М.: Нолидж, 1998.
8. Федоров А.Г. Delphi 3.0 для всех.. - М.: КомпьютерПресс, 1998.
9. Гофман В., Хомоненко А. Delphi 5. СПб.: БХВ- Санкт-Петербург, 2000/
10. Программирование в среде Delphi: Лабораторный практикум для студентов всех специальностей / А.Б.Закалюкин, С.В.Колосов, А.А.Навроцкий, А.К.Синицын, А.И.Шакирин; Под общей ред. А.К.Синицина. - Мн.: БГУИР, 1998.

Репозиторий БГАТУ

Содержание Стр.

Репозиторий БГАТУ