



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 2

Для студентов  
факультета предпринимательства и управления

Минск - 2001

## ПРОГРАММИРОВАНИЕ АЛГОРИТМОВ С ИСПОЛЬЗОВАНИЕМ ЗАПИСЕЙ

**Цель лабораторной работы:** создать приложение, в котором используются данные типа запись.

### 6.1. Пример создания приложения

**Задание:** создать Windows-приложение для обработки ведомости об успеваемости учащихся в количестве 9 человек. Каждая запись должна содержать фамилию, инициалы, а также оценки по физике, математике и сочинению. Вывести список учащихся, отсортированный в порядке уменьшения их среднего балла.

Один из возможных вариантов панели интерфейса создаваемого приложения показан на рис.6.1.

#### 6.1.1. Размещение компонентов на Форме

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



Рис. 6.1

В этом задании для нанесения соответствующих надписей в колонках и строках используется фиксированная зона компонента StringGrid, поэтому в Инспекторе Объектов значения свойств FixedCols и FixedRows установите равными 1. В соответствии с заданием установите значение свойства ColCount=6, а значение свойства RowCount=10. Для возможности просмотра всего списка учащихся в компоненте StringGrid удобно использовать вертикальную линейку прокрутки, поэтому установите свойство ScrollBars в состояние ssVertical. Откройте список опций свойства +Options и установите значение goEditing в True – это даст возможность редактировать информацию в компоненте StringGrid с помощью клавиатуры и “мыши”.

### 6.1.2 Создание процедур обработки событий *FormCreate* и *Button1Click*

Двойным нажатием клавиши “мыши” на Форме и кнопке *Button1* создайте соответствующие процедуры обработки событий. Используя текст модуля *UnZap*, внимательно наберите операторы этих процедур.

### 6.1.3 Текст модуля *UnZap*

**Unit** *UnZap*;

**interface**

**uses**

*Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls, Buttons, Grids*;

**type**

*TForm1 = class(TForm)*

*StringGrid1: TStringGrid*;

*Button1: TButton*;

**procedure** *FormCreate*(Sender: TObject);

**procedure** *Button1Click*(Sender: TObject);

*private*

*{ Private declarations }*

*public*

*{ Public declarations }*

**end**;

**var**

*Form1: TForm1*;

**implementation**

*{ \$R \*.DFM }*

**type**

*zap=record* // объявление записи

*fio :string[20]*;

*mat,fiz,soch:integer*;

*srbal :extended*

*end*;

**var**

*MZap:array[1..9] of zap*; // объявление массива записей

**procedure** *TForm1.FormCreate*(Sender: TObject);

**var**

*i:integer*;

**begin**

**with** *StringGrid1 do*

**begin** // занесение информации в ячейку *StringGrid1*

*Cells[0,0]:='№пп'*;

*Cells[1,0]:='Фамилия,инициалы'*;

*Cells[2,0]:='Математика'*;

*Cells[3,0]:='Физика'*;

*Cells[4,0]:='Сочинение'*;

```

Cells[5,0]:='Ср.балл';
for i:=1 to 9 do
  Cells[0,i]:=IntToStr(i);
  Cells[1,1]:='Первый П.П.'; Cells[2,1]:='3'; Cells[3,1]:='3'; Cells[4,1]:='3';
  Cells[1,2]:='Второй В.В.'; Cells[2,2]:='3'; Cells[3,2]:='3'; Cells[4,2]:='4';
  Cells[1,3]:='Третий Т.Т.'; Cells[2,3]:='3'; Cells[3,3]:='4'; Cells[4,3]:='4';
  Cells[1,4]:='Четвертый Ч.Ч.'; Cells[2,4]:='4'; Cells[3,4]:='4'; Cells[4,4]:='4';
  Cells[1,5]:='Пятый П.П.'; Cells[2,5]:='3'; Cells[3,5]:='4'; Cells[4,5]:='5';
  Cells[1,6]:='Шестой Ш.Ш.'; Cells[2,6]:='5'; Cells[3,6]:='4'; Cells[4,6]:='4';
  Cells[1,7]:='Седьмой С.С.'; Cells[2,7]:='5'; Cells[3,7]:='5'; Cells[4,7]:='4';
  Cells[1,8]:='Восьмой В.В.'; Cells[2,8]:='5'; Cells[3,8]:='5'; Cells[4,8]:='5';
  Cells[1,9]:='Девятый Д.Д.'; Cells[2,9]:='3'; Cells[3,9]:='5'; Cells[4,9]:='5';
for i:=1 to 9 do
  with MZap[i] do
    begin // формирование полей массива записей
    fio:=Cells[1,i];
    mat:=StrToInt(Cells[2,i]);
    fiz:=StrToInt(Cells[3,i]);
    soch:=StrToInt(Cells[4,i]);
    srbal:=(mat+fiz+soch)/3; // вычисление значения ср. балла
    Cells[5,i]:=FloatToStrF(srbal,ffFixed,5,2); // вывод значения ср.балла
    end; // в последнюю колонку StringGrid1
  end;
end;
procedure TForm1.Button1Click(Sender: TObject);
var
  i,j :integer;
  vper:zap;
begin
  for i:=1 to 9 do
    with StringGrid1,MZap[i] do
      begin
        fio:=Cells[1,i];
        mat:=StrToInt(Cells[2,i]);
        fiz:=StrToInt(Cells[3,i]);
        soch:=StrToInt(Cells[4,i]);
        srbal:=(mat+fiz+soch)/3;
        Cells[5,i]:=FloatToStrF(srbal,ffFixed,5,2);
        end;
      for i:=2 to 9 do // сортировка методом "пузырька"
        for j:=9 downto i do
          if MZap[j-1].srbal<MZap[j].srbal then

            begin
              vper:=MZap[j-1];

```

```

MZap[j-1]:=MZap[j];
MZap[j]:=vper;
end;
for i:=1 to 9 do // заполнение ячеек StringGrid1 полями массива записей
with StringGrid1,MZap[i] do
begin
Cells[1,i]:=fio;
Cells[2,i]:=IntToStr(mat);
Cells[3,i]:=IntToStr(fiz);
Cells[4,i]:=IntToStr(soch);
Cells[5,i]:=FloatToStrF(srbal,ffFixed,5,2);
end;
end;
end.

```

### 6.2 Выполнение индивидуального задания

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

#### Индивидуальные задания

1. Поля шахматной доски характеризуются записью

Type

Pole=record

Ver:(a,b,c,d,e,f,g,h); {вертикальные координаты}

Hor:1..8; {горизонтальные координаты}

end;

Вывести шахматную доску, пометив крестиками все поля, которые «бьет» ферзь, стоящий на поле с координатами  $Ver_i$  и  $Hor_i$ , и ноликами все остальные поля.

2. Поля шахматной доски характеризуются записью (см. задание 1)

Var Figura:Pole;

Вывести сообщение может ли конь за один ход перейти с поля  $Figura_i$  на поле  $Figura_j$ .

3. Type

Karta=record

m: (piki,trefi,bubni,cherni); {масть}

d:(shest,sem,vosem,devyat,desyat,valet,dama,korol,tuz); {достоинство}

end;

Var k1,k2:Karta;

Вывести сообщение «бьет» ли карта  $k1$ , карту  $k2$ , с учетом того, что масть  $m_i$  является козырной.

4. В магазине формируется список лиц, записавшихся на покупку товара повышенного спроса. Каждая запись этого списка содержит: порядковый номер, Ф.И.О., домашний адрес покупателя и дату постановки на учет. Удалить из списка все повторные записи, проверяя Ф.И.О. и домашний адрес.

5. Список товаров, имеющих на складе, включает в себя наименование товара, количество единиц товара, цену единицы и дату поступления товара на склад. Вывести в алфавитном порядке список товаров, хранящихся больше месяца, стоимость которых превышает 1000000 руб., а также общую стоимость всех этих товаров.
6. Для получения места в общежитии формируется список студентов, который включает Ф.И.О. студента, группу, средний балл, доход на члена семьи. Общежитие в первую очередь предоставляется тем, у кого доход на члена семьи меньше двух минимальных зарплат, затем остальным в порядке уменьшения среднего балла. Вывести список очередности предоставления мест в общежитии.
7. В справочной автовокзала хранится расписание движения автобусов. Для каждого рейса указаны его номер, тип автобуса, пункт назначения, время отправления и прибытия. Вывести информацию о рейсах, которыми можно воспользоваться для прибытия в пункт назначения раньше заданного времени.
8. На междугородней АТС информация о разговорах содержит дату разговора, код и название города, время разговора, тариф, номер телефона в этом городе и номер телефона абонента. Вывести по каждому городу общее время разговоров с ним и сумму.
9. Информация о сотрудниках фирмы включает: Ф.И.О., табельный номер, количество проработанных часов за месяц, почасовой тариф. Рабочее время свыше 144 часов считается сверхурочным и оплачивается в двойном размере. Вывести размер заработной платы каждого сотрудника фирмы за вычетом подоходного налога, который составляет 12% от суммы заработка.
10. Информация об участниках спортивных соревнований содержит: наименование страны, название команды, Ф.И.О. игрока, игровой номер, возраст, рост, вес. Вывести информацию о самой молодой, рослой и легкой команде.
11. Для книг, хранящихся в библиотеке, задаются: регистрационный номер книги, автор, название, год издания, издательство, количество страниц. Вывести список книг с фамилиями авторов в алфавитном порядке, изданных после заданного года.
12. Различные цеха завода выпускают продукцию нескольких наименований. Сведения о выпущенной продукции включают: наименование, количество, номер цеха. Для заданного цеха необходимо вывести количество выпущенных изделий по каждому наименованию в порядке убывания количества.
13. Информация о сотрудниках предприятия содержит: Ф.И.О., номер отдела, должность, дату начала работы. Вывести списки сотрудников по отделам в порядке убывания стажа.
14. Ведомость абитуриентов, сдавших вступительные экзамены в университет содержит: Ф.И.О., адрес, оценки. Определить количество абитуриентов, проживающих в г.Минске и сдавших экзамены со средним баллом не ниже 4.5, вывести их фамилии в алфавитном порядке.

15. В справочной аэропорта хранится расписание вылета самолетов на следующие сутки. Для каждого рейса указаны: номер рейса, тип самолета, пункт назначения, время вылета. Вывести все номера рейсов, типы самолетов и времена вылета для заданного пункта назначения в порядке возрастания времени вылета.
16. У администратора железнодорожных касс хранится информация о свободных местах в поездах дальнего следования на ближайшую неделю в следующем виде: дата выезда, пункт назначения, время отправления, число свободных мест. Оргкомитет международной конференции обращается к администратору с просьбой зарезервировать  $m$  мест до города  $N$  на  $k$ -й день недели с временем отправления поезда не позднее  $t$  часов вечера. Вывести время отправления или сообщение о невозможности выполнить заказ в полном объеме.
17. Ведомость абитуриентов, сдавших вступительные экзамены в университет содержит: Ф.И.О. абитуриента, оценки. Определить средний балл по университету и вывести список абитуриентов, средний балл которых выше среднего балла по университету. Первыми в списке должны идти студенты, сдавшие все экзамены на 5.
18. В радиоателье хранятся квитанции о сданной в ремонт радиоаппаратуре. Каждая квитанция содержит следующую информацию: наименование группы изделий(телевизор, радиоприемник и т. п.), марка изделия, дата приемки в ремонт, состояние готовности заказа (выполнен, не выполнен). Вывести информацию о состоянии заказов на текущие сутки по группам изделий.
19. Разработать программу формирования ведомости об успеваемости студентов. Каждая запись этой ведомости должна содержать: номер группы, Ф.И.О. студента, оценки за последнюю сессию. Вывести списки студентов по группам. В каждой группе Ф.И.О. студентов должны быть расположены в порядке убывания среднего балла.
20. В исполкоме формируется список учета нуждающихся в улучшении жилищных условий. Каждая запись этого списка содержит: порядковый номер, Ф.И.О., величину жилплощади на одного члена семьи и дату постановки на учет. По заданному количеству  $k$ -квартир, выделяемых по данному списку в течение года, вывести весь список с указанием ожидаемого года получения квартиры.
21. Имеется список женихов и список невест. Каждая запись списка содержит пол, имя, возраст, рост, вес, а также требования к партнеру: наименьший и наибольший возраст, наименьший и наибольший вес, наименьший и наибольший рост. Объединить эти списки в список пар с учетом требований к партнерам без повторений женихов и невест.
22. В библиотеке имеется список книг. Каждая запись этого списка содержит: фамилии авторов, название книги, год издания. Вывести информацию о книгах, в названии которых встречается некоторое ключевое слово (ввести с клавиатуры).

23. В магазине имеется список поступивших в продажу автомобилей. Каждая запись этого списка содержит: марку автомобиля, стоимость, расход топлива на 100 км, надежность (число лет безотказной работы), комфортность (отличная, хорошая, удовлетворительная). Вывести перечень автомобилей, удовлетворяющих требованиям покупателя, которые вводятся с клавиатуры в виде некоторого интервала допустимых значений.
24. Каждая запись списка вакантных рабочих мест содержит: наименование организации, расстояние в км от центра города, должность, квалификация(разряд или образование), стаж работы по специальности, заработная плата, наличие социального страхования(да/нет), продолжительность ежегодного оплачиваемого отпуска. Вывести список рабочих мест в соответствии с требованиями клиента.
25. В технической службе аэропорта имеется справочник, содержащий записи следующей структуры: тип самолета, год выпуска, расход горючего на 1000 км. Для определения потребности в горючем техническая служба запрашивает расписание полетов. Каждая запись расписания содержит следующую информацию: номер рейса, пункт назначения, дальность полета. Вывести суммарное количество горючего, необходимое для обеспечения полетов на следующие сутки.
26. Для участия в конкурсе на замещение вакантной должности сотрудника фирмы желающие подают следующую информацию: Ф.И.О., год рождения, образование(среднее, специальное, высшее), знание иностранных языков(английский, немецкий, французский, владею свободно, читаю и перевожу со словарем), владение компьютером (MSDOS,Windows), стаж работы , наличие рекомендаций. Вывести список претендентов в соответствии с требованиями руководства фирмы.
27. При постановке на учет в ГАИ автолюбители указывают следующие данные: марка автомобиля, год выпуска, номер двигателя, номер кузова, цвет, номерной знак, Ф.И.О и адрес. владельца. Вывести список автомобилей, проходящих техосмотр в текущем году, сгруппированных по маркам автомобилей. Учесть, что если текущий год четный, техосмотр проходят автомобили с четными номерами двигателей, иначе – с нечетными номерами.
28. Для участия в конкурсе исполнителей необходимо заполнить следующую анкету: Ф.И.О., год рождения, название страны, класс музыкального инструмента(гитара, фортепиано, скрипка, виолончель). Вывести список самых молодых лауреатов конкурса по классам инструментов в порядке занятых мест.
29. Список группы студентов содержит следующую информацию: Ф.И.О., рост и вес. Вывести Ф.И.О. студентов, рост и вес которых чаще всего встречаются в списке.
30. Список группы студентов содержит следующую информацию: Ф.И.О., рост и вес. Вывести Ф.И.О. студентов, рост и вес которых являются в списке уникальными.

## ПРОГРАММИРОВАНИЕ АЛГОРИТМОВ С ИСПОЛЬЗОВАНИЕМ ФАЙЛОВ

**Цель лабораторной работы:** освоить применение компонентов OpenFileDialog и SaveDialog и создать приложение, в котором используются файлы.

### 7.1. Пример создания приложения

**Задание:** создать Windows-приложение для записи в файл и чтения из файла ведомости об успеваемости учащихся в количестве 10 человек. Каждая запись файла должна содержать фамилию, инициалы, а также оценки по физике, математике и сочинению. Вывести список учащихся, отсортированный в алфавитном порядке и записать эту информацию в текстовый файл.

Один из возможных вариантов панели интерфейса создаваемого приложения показан на рис.7.1.

#### 7.1.1. Размещение компонентов на Форме

При работе с файлами чтение и запись информации удобно организовывать с помощью компонентов OpenFileDialog и SaveDialog.

Компоненты OpenFileDialog и SaveDialog находятся на странице Dialogs (см. приложение 2). Все компоненты этой страницы являются невизуальными, т.е. не видны в момент выполнения приложения. Поэтому их можно разместить в любом удобном месте Формы. Оба рассматриваемых компонента имеют идентичные свойства и отличаются только внешним видом.



Рис. 7.1

Для установки компонентов OpenFileDialog и SaveDialog на Форму необходимо на странице Dialogs Палитры Компонентов щелкнуть “мышью”

соответственно по пиктограмме  или  и разместить ее в любом свободном месте Формы. При выполнении приложения в момент вызова компонента появляется диалоговое окно, с помощью которого пользователь выбирает имя файла и маршрут к нему. В случае успешного завершения диалога имя выбранного файла и маршрут поиска содержится в свойстве FileName.

Пользователь имеет возможность настроить параметры окна диалога по своему усмотрению. В частности, изменить заголовок окна можно с помощью свойства Title. В свойстве DefaultExt можно указать расширение файла, если оно не задано пользователем. Свойство Filter используется для поиска (фильтрации) файлов, отображаемых в окне. Установка фильтра производится следующим образом. Выделив соответствующий компонент, необходимо дважды щелкнуть по правой(белой) части свойства Filter Инспектора Объектов. В появившемся окне редактора фильтра-Filter Editor необходимо в колонке Filter Name набрать текст, характеризующий соответствующий фильтр, а в колонке Filter – маску. Для компонента OpenFileDialog1 установим значения масок как показано на рис. 7.2.

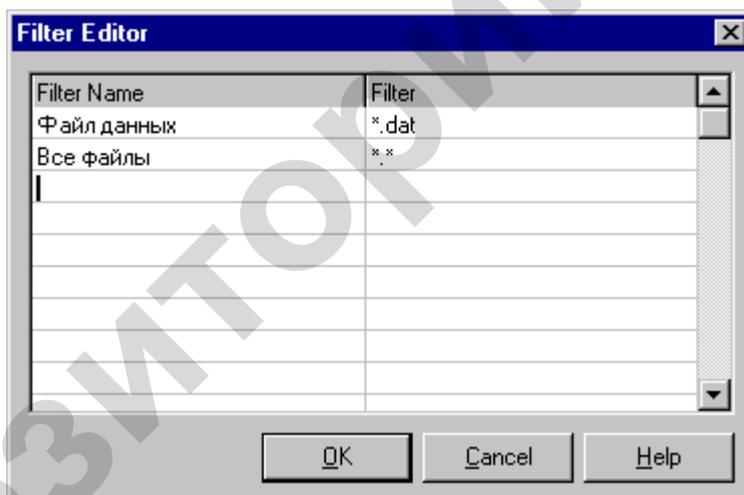


Рис.7.2

Маска \*.dat означает что, будут видны файлы данных с любым именем и с расширением dat, а маска \*.\* - что будут видны все файлы (с любым именем и с любым расширением).

Для того, чтобы файл автоматически записывался с расширением dat в свойстве DefaultExt запишем требуемое расширение - dat.

Аналогичным образом настроим компонент SaveDialog1 для текстового файла (расширение txt).

### 7.1.2 Создание процедур обработки событий

Для удобства работы с несколькими различными процедурами обработки событий в свойстве Name каждого компонента BitBtn замените программные имена кнопок: BitBtn1 – на BitBtnNew, BitBtn2 – на BitBtnOpen, BitBtn3 – на BitBtnSort, BitBtn4 – на BitBtnSave. Двойным нажатием клавиши “мыши” на кнопках BitBtn создайте соответствующие процедуры обработки событий. Пользуясь текстом модуля UnFile, внимательно наберите операторы этих процедур.

### 7.1.3 Текст модуля UnFile

**Unit** UnFile;

**Interface**

**uses**

Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls, Buttons, Grids, Spin;

**type**

```
TForm1 = class(TForm)
  StringGrid1: TStringGrid;
  BitBtnNew: TBitBtn;
  BitBtnOpen: TBitBtn;
  BitBtnSort: TBitBtn;
  BitBtnSave: TBitBtn;
  SaveDialog1: TSaveDialog;
  SpinEdit1: TSpinEdit;
  Label1: TLabel;
  OpenFileDialog1: TOpenDialog;
  procedure TForm1.FormCreate(Sender: TObject);
  procedure BitBtnSortClick(Sender: TObject);
  procedure BitBtnNewClick(Sender: TObject);
  procedure SpinEdit1Change(Sender: TObject);
  procedure BitBtnOpenClick(Sender: TObject);
  procedure BitBtnSaveClick(Sender: TObject);
```

private

```
{ Private declarations }
```

public

```
{ Public declarations }
```

**end;**

**var**

```
Form1: TForm1;
```

**Implementation**

```
{ $R *.DFM }
```

```
type
```

```
zap=record           // объявление записи
```

```
  fio      :string[20];
```

```
  mat,fiz,soch:integer;
```

```
end;
```

```
var
```

```
MZap:array[1..25] of zap; // объявление массива записей
```

```
FileZap:file of zap; // объявление файла записей
```

```
FileText:TextFile; // объявление текстового файла
```

```
FileNameZap,FileNameText:string; // имена файла записей и текстового файла
```

```
n:integer; // текущее кол-во элементов массива записей
```

```
{ Обработчик события создания Формы }
```

```
procedure TForm1.FormCreate(Sender: TObject);
```

```
begin
```

```
  with StringGrid1 do
```

```
    begin // занесение информации в ячейки StringGrid1
```

```
      Cells[0,0]:=’№пп’;
```

```
      Cells[1,0]:=’Фамилия,инициалы’;
```

```
      Cells[2,0]:=’Математика’;
```

```
      Cells[3,0]:=’Физика’;
```

```
      Cells[4,0]:=’Сочинение’;
```

```
    end;
```

```
    BitBtnSort.Hide; // спрятать кнопку ”Sort”
```

```
    BitBtnSave.Hide; // спрятать кнопку ”Save”
```

```
end;
```

```
{ Обработчик нажатия кнопки Sort }
```

```
procedure TForm1.BitBtnSortClick(Sender: TObject);
```

```
var
```

```
  i,j :integer;
```

```
  vper:zap;
```

```
begin
```

```
  for i:=1 to n do
```

```
    with StringGrid1,MZap[i] do
```

```
      begin
```

```
        fio:=Cells[1,i];
```

```
        mat:=StrToInt(Cells[2,i]);
```

```
        fiz:=StrToInt(Cells[3,i]);
```

```
        soch:=StrToInt(Cells[4,i]);
```

```
      end;
```

```
{ сортировка методом ”пузырька” }
```

```
  for i:=2 to n do
```

```

for j:=n downto i do
  if MZap[j-1].fio>MZap[j].fio then
    begin
      vper:=MZap[j-1];
      MZap[j-1]:=MZap[j];
      MZap[j]:=vper;
    end;
for i:=1 to n do // заполнение ячеек StringGrid1 полями массива записей
  with StringGrid1,MZap[i] do
    begin
      Cells[0,i]:=IntToStr(i);
      Cells[1,i]:=fio;
      Cells[2,i]:=IntToStr(mat);
      Cells[3,i]:=IntToStr(fiz);
      Cells[4,i]:=IntToStr(soch);
    end;
end;

  { Создание нового файла записей }
procedure TForm1.BitBtnNewClick(Sender: TObject);
var
  i:integer;
begin // вывод на экран окна с предупреждающим сообщением
  if MessageDlg('Содержимое существующего файла будет уничтожено. Вы уверены?',
  mtConfirmation, mbYesNoCancel, 0)=mrYes then
    begin
      for i:=1 to n do
        with StringGrid1,MZap[i] do
          begin // формирование полей массива записей
            fio:=Cells[1,i];
            mat:=StrToInt(Cells[2,i]);
            fiz:=StrToInt(Cells[3,i]);
            soch:=StrToInt(Cells[4,i]);
          end;
        with OpenFileDialog1 do
          begin
            Title:='Создание файла'; // заголовок окна диалога
            if Execute then // выполнение стандартного диалога выбора имени файла
              begin
                FileNameZap:=FileName; // присваивание имени файла
                AssignFile(FileZap,FileNameZap); // назначить файлу FileZap имя FileNameZap
                Rewrite(FileZap); // открыть файл на запись
                for i:=1 to n do
                  write(FileZap,MZap[i]); // запись в файл массива записей
              end
            end
          end
        end
      end
    end
  end

```

```

CloseFile(FileZap);      // закрытие файла записей
end;
end;
end;
end;

```

*{Обработчик кнопки изменения размера}*

```

procedure TForm1.SpinEdit1Change(Sender: TObject);
var
  i,m:integer;
begin
  m:=StrToInt(SpinEdit1.Text); // присвоить новое значение размера
with StringGrid1 do
  begin
    RowCount:=m+1; // пересчитать количество строк
    if m>n then // если строки добавлены то
    for i:=n+1 to m do // инициализировать новые ячейки
    begin
      Cells[0,i]:=IntToStr(i);
      Cells[1,i]:"";
      Cells[2,i]:"";
      Cells[3,i]:"";
      Cells[4,i]:"";
    end;
  end;
  n:=m; // запомнить новое значение размера
end;

```

*{ обработчик нажатия кнопки Open }*

```

procedure TForm1.BitBtnOpenClick(Sender: TObject);
var
  i:integer;
begin
  with OpenFileDialog1 do
  begin
    Title:='Открытие файла'; // заголовок окна диалога
    if Execute then // выполнение стандартного диалога выбора имени файла
    begin
      FileNameZap:=FileName; // присваивание имени файла
      AssignFile(FileZap,FileNameZap); // назначить файлу FileZap имя FileNameZap
      ReSet(FileZap); // открыть файл на чтение
      n:=0; // инициализация счетчика кол-ва прочитанных из файла элементов
      while not EoF(FileZap) do
      begin
        n:=n+1; // увеличение счетчика

```

```

    read(FileZap,MZap[n]); // чтение из файла n-го элемента массива записей
end;
SpinEdit1.Text:=IntToStr(n); // занести кол-во прочитанных записей в SpinEdit
StringGrid1.RowCount:=n+1; // присвоить кол-во строк
for i:=1 to n do
    with StringGrid1,MZap[i] do
        begin // заполнение ячеек StringGrid1 полями массива записей
            Cells[0,i]:=IntToStr(i);
            Cells[1,i]:=fio;
            Cells[2,i]:=IntToStr(mat);
            Cells[3,i]:=IntToStr(fiz);
            Cells[4,i]:=IntToStr(soch);
        end;
    CloseFile(FileZap); // закрытие файла записей
end;
end;
BitBtnSort.Show; // показать кнопку "Sort"
BitBtnSave.Show; // показать кнопку "Save"
end;

{ Сохранение данных в текстовом файле }
procedure TForm1.BitBtnSaveClick(Sender: TObject);
var
    i:integer;
begin
    with SaveDialog1 do
    if Execute then // выполнение стандартного диалога выбора имени файла
        begin
            FileNameText:=FileName; // присваивание имени файла
            AssignFile(FileText,FileNameText); // назначить файлу FileText имя
            FileNameText
            ReWrite(FileText); // открыть текстовый файл на запись
            for i:=1 to n do
                with MZap[i] do // запись в текстовый файл
                    writeln(FileText,i:3,fio:20,mat:5,fiz:5,soch:5);
                CloseFile(FileText); // закрытие текстового файла по окончании записи
            end;
            BitBtnSort.Hide; // спрятать кнопку "Sort"
        end;
    end.

```

#### 7.1.4 Работа с приложением

Запустите созданное приложение. Занесите в соответствующие поля панели интерфейса информацию об успеваемости учащихся. Кнопкой "New" сохраните данные в файле. Завершите выполнение приложения.

Вновь запустите приложение и кнопкой “Open” откройте только что созданный файл. Убедитесь, что информация не содержит ошибок. При необходимости обнаруженные ошибки можно исправить, а также дополнить ведомость новой информацией. Для сортировки ведомости в алфавитном порядке воспользуйтесь кнопкой “Sort” и сохраните отсортированную информацию кнопкой “New”.

Еще раз завершите и вновь запустите приложение.

Кнопкой “Open” откройте файл и убедитесь, что в нем теперь содержится ведомость, отсортированная в алфавитном порядке. Кнопкой “Save” сохраните информацию в текстовом файле. Для просмотра содержимого текстового файла воспользуйтесь, например, приложением “Microsoft Word”.

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

### **7.2 Индивидуальные задания**

Во всех заданиях предусмотреть сохранение вводимых данных в файле и возможность чтения из ранее созданного файла. Результаты выводить в панель интерфейса и в текстовый файл.

1. Ведомость абитуриентов, сдавших вступительные экзамены в университет содержит: Ф.И.О. абитуриента, оценки. Определить средний балл по университету и вывести список абитуриентов, средний балл которых выше среднего балла по университету. Первыми в списке должны идти студенты, сдавшие все экзамены на 5.

2. В радиоателье хранятся квитанции о сданной в ремонт радиоаппаратуре. Каждая квитанция содержит следующую информацию: наименование группы изделий (телевизор, радиоприемник и т. п.), марка изделия, дата приемки в ремонт, состояние готовности заказа (выполнен, не выполнен). Вывести информацию о состоянии заказов на текущие сутки по группам изделий.

3. У администратора железнодорожных касс хранится информация о свободных местах в поездах дальнего следования на ближайшую неделю в следующем виде: дата выезда, пункт назначения, время отправления, число свободных мест. Оргкомитет международной конференции обращается к администратору с просьбой зарезервировать  $m$  мест до города  $N$  на  $k$ -й день недели с временем отправления поезда не позднее  $t$  часов вечера. Вывести время отправления или сообщение о невозможности выполнить заказ в полном объеме.

4. Разработать программу формирования ведомости об успеваемости студентов. Каждая запись этой ведомости должна содержать: номер группы, Ф.И.О. студента, оценки за последнюю сессию. Вывести списки студентов по группам. В каждой группе Ф.И.О. студентов должны быть расположены в порядке убывания среднего балла.

5. В справочной аэропорта хранится расписание вылета самолетов на следующие сутки. Для каждого рейса указаны: номер рейса, тип самолета, пункт назначения, время вылета. Вывести все номера рейсов, типы самолетов и времена вылета для заданного пункта назначения в порядке возрастания времени вылета.

6. В исполкоме формируется список учета нуждающихся в улучшении жилищных условий. Каждая запись этого списка содержит: порядковый номер, Ф.И.О., величину жилплощади на одного члена семьи и дату постановки на учет. По заданному количеству  $k$ -квартир, выделяемых по данному списку в течение года, вывести весь список с указанием ожидаемого года получения квартиры.

7. Ведомость абитуриентов, сдавших вступительные экзамены в университет содержит: Ф.И.О., адрес, оценки. Определить количество абитуриентов, проживающих в г.Минске и сдавших экзамены со средним баллом не ниже 4.5, вывести их фамилии в алфавитном порядке.

8. Имеется список женихов и список невест. Каждая запись списка содержит пол, имя, возраст, рост, вес, а также требования к партнеру: наименьший и наибольший возраст, наименьший и наибольший вес, наименьший и наибольший рост. Объединить эти списки в список пар с учетом требований к партнерам без повторений женихов и невест.

9. Информация о сотрудниках предприятия содержит: Ф.И.О., номер отдела, должность, дату начала работы. Вывести списки сотрудников по отделам в порядке убывания стажа.

10. В библиотеке имеется список книг. Каждая запись этого списка содержит: фамилии авторов, название книги, год издания. Вывести информацию о книгах, в названии которых встречается некоторое ключевое слово (вести с клавиатуры).

11. Различные цеха завода выпускают продукцию нескольких наименований. Сведения о выпущенной продукции включают: наименование, количество, номер цеха. Для заданного цеха необходимо вывести количество выпущенных изделий по каждому наименованию в порядке убывания количества.

12. В магазине имеется список поступивших в продажу автомобилей. Каждая запись этого списка содержит: марку автомобиля, стоимость, расход топлива на 100 км, надежность (число лет безотказной работы), комфортность (отличная, хорошая, удовлетворительная). Вывести перечень автомобилей, удовлетворяющих требованиям покупателя, которые вводятся с клавиатуры в виде некоторого интервала допустимых значений.

13. Для книг, хранящихся в библиотеке, задаются: регистрационный номер книги, автор, название, год издания, издательство, количество страниц. Вывести список книг с фамилиями авторов в алфавитном порядке, изданных после заданного года.

14. Каждая запись списка вакантных рабочих мест содержит: наименование организации, расстояние в км от центра города, должность, квалификация(разряд или образование), стаж работы по специальности,

заработная плата, наличие социального страхования(да/нет), продолжительность ежегодного оплачиваемого отпуска. Вывести список рабочих мест в соответствии с требованиями клиента.

15. Информация об участниках спортивных соревнований содержит: наименование страны, название команды, Ф.И.О. игрока, игровой номер, возраст, рост, вес. Вывести информацию о самой молодой, рослой и легкой команде.

16. В технической службе аэропорта имеется справочник, содержащий записи следующей структуры: тип самолета, год выпуска, расход горючего на 1000 км. Для определения потребности в горючем техническая служба запрашивает расписание полетов. Каждая запись расписания содержит следующую информацию: номер рейса, пункт назначения, дальность полета. Вывести суммарное количество горючего, необходимое для обеспечения полетов на следующие сутки.

17. Информация о сотрудниках фирмы включает: Ф.И.О., табельный номер, количество проработанных часов за месяц, почасовой тариф. Рабочее время свыше 144 часов считается сверхурочным и оплачивается в двойном размере. Вывести размер заработной платы каждого сотрудника фирмы за вычетом подоходного налога, который составляет 12% от суммы заработка.

18. Для участия в конкурсе на замещение вакантной должности сотрудника фирмы желающие подают следующую информацию: Ф.И.О., год рождения, образование(среднее, специальное, высшее), знание иностранных языков(английский, немецкий, французский, владею свободно, читаю и перевожу со словарем), владение компьютером (MSDOS,Windows), стаж работы, наличие рекомендаций. Вывести список претендентов в соответствии с требованиями руководства фирмы.

19. На междугородней АТС информация о разговорах содержит дату разговора, код и название города, время разговора, тариф, номер телефона в этом городе и номер телефона абонента. Вывести по каждому городу общее время разговоров с ним и сумму.

20. При постановке на учет в ГАИ автолюбители указывают следующие данные: марка автомобиля, год выпуска, номер двигателя, номер кузова, цвет, номерной знак, Ф.И.О и адрес. владельца. Вывести список автомобилей, проходящих техосмотр в текущем году, сгруппированных по маркам автомобилей. Учесть, что если текущий год четный, техосмотр проходят автомобили с четными номерами двигателей, иначе – с нечетными номерами.

21. В справочной автовокзала хранится расписание движения автобусов. Для каждого рейса указаны его номер, тип автобуса, пункт назначения, время отправления и прибытия. Вывести информацию о рейсах, которыми можно воспользоваться для прибытия в пункт назначения раньше заданного времени.

22. Для участия в конкурсе исполнителей необходимо заполнить следующую анкету: Ф.И.О., год рождения, название страны, класс музыкального инструмента(гитара, фортепиано, скрипка, виолончель). Вывести список самых молодых лауреатов конкурса по классам инструментов в порядке занятых мест.

23. Для получения места в общежитии формируется список студентов, который включает Ф.И.О. студента, группу, средний балл, доход на одного члена семьи. Общежитие в первую очередь предоставляется тем, у кого доход на одного члена семьи меньше двух минимальных зарплат, затем остальным в порядке уменьшения среднего балла. Вывести список очередности предоставления мест в общежитии.

24. Список группы студентов содержит следующую информацию: Ф.И.О., рост и вес. Вывести Ф.И.О. студентов, рост и вес которых чаще всего встречаются в списке.

25. Список товаров, имеющихся на складе, включает в себя наименование товара, количество единиц товара, цену единицы и дату поступления товара на склад. Вывести в алфавитном порядке список товаров, хранящихся больше месяца, стоимость которых превышает 1000000 руб., а также общую стоимость всех этих товаров.

26. Список группы студентов содержит следующую информацию: Ф.И.О., рост и вес. Вывести Ф.И.О. студентов, рост и вес которых являются в списке уникальными.

27. В магазине формируется список лиц, записавшихся на покупку товара повышенного спроса. Каждая запись этого списка содержит: порядковый номер, Ф.И.О., домашний адрес покупателя и дату постановки на учет. Удалить из списка все повторные записи, проверяя Ф.И.О. и домашний адрес.

28. Разработать приложение, которое формирует файл F1, содержащий целые числа, и переписывает этот файл в другой файл – F2, помещая в него из F1 только положительные числа.

29. Разработать приложение, переписывающее в текстовый файл T2 содержимое текстового файла T1, но без строк, содержащих цифры.

30. Разработать приложение для слияния двух отсортированных по убыванию значений элементов файлов F1 и F2. Результатом слияния должен быть файл F3, элементы которого упорядочены по возрастанию.

## ПРОГРАММИРОВАНИЕ АЛГОРИТМОВ С ИСПОЛЬЗОВАНИЕМ ФУНКЦИЙ И ПРОЦЕДУР. СОЗДАНИЕ МОДУЛЕЙ

**Цель лабораторной работы:** освоить методику создания модулей, содержащих процедуры и функции, и использования их в проекте.

### 8.1. 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  $h=(x_k - x_n)/10$ . Создать модуль, в котором вычисление значений  $Y(x)$  оформить в виде функции, а вычисление  $S(x)$  - в виде процедуры. Подключить модуль к проекту и выполнить созданное приложение.

Один из возможных вариантов панели интерфейса создаваемого приложения показан на рис.8.1.

#### 8.1.1. Размещение компонентов на Форме

Разместим на Форме компоненты Label, Edit, SpinEdit, BitBtn и Memo.

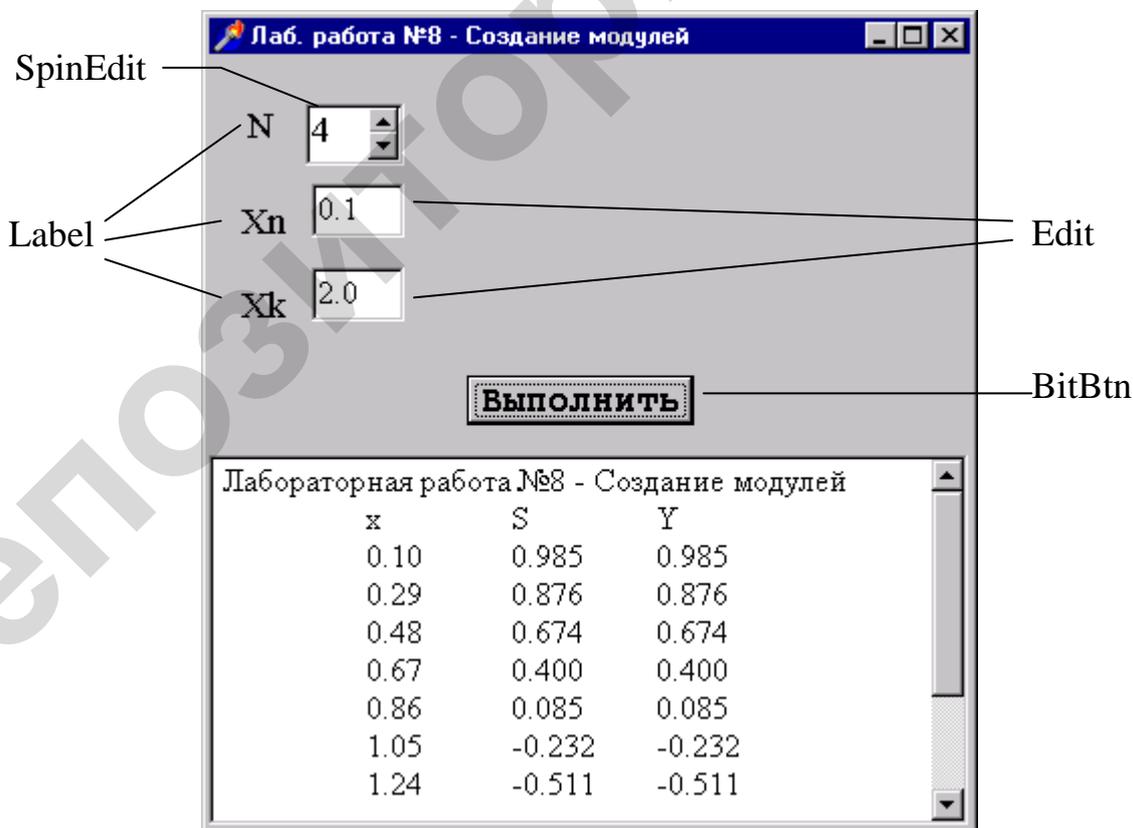


Рис. 8.1

Сохраним модуль под именем UnModul (текст модуля приведен в п.8.1.4).

### 8.1.2. Создание модуля и подключение его к проекту

В соответствии с заданием создадим модуль, в котором вычисление значений  $Y(x)$  оформим в виде функции, а вычисление  $S(x)$  - в виде процедуры. Для создания модуля откроем в главном меню пункт **F**ile и щелкнем “мышью” на опции **N**ew...(Новый...). Delphi откроет панель New Items(Репозиторий), в которой сделаем активной пиктограмму **U**nit(Модуль) и нажмем кнопку ОК. Откроется окно с “пустым” модулем Unit2. С помощью опции **S**ave **A**s... меню **F**ile сохраним модуль в папке с файлами проекта, присвоив ему имя, например, UnFuncProc.

В этом модуле операторы вычисления  $Y(x)$  в виде подпрограммы-функции **F** и операторы вычисления  $S(x)$  в виде подпрограммы-процедуры **S**um оформим по правилам создания модулей (текст модуля UnFuncProc приведен в п.8.1.3).

Для подключения модуля UnFuncProc к проекту необходимо сделать активным окно с текстом модуля UnModul, затем в меню **F**ile выбрать опцию **U**se Unit...и в открывшемся окне Use Unit указать имя используемого модуля – UnFuncProc. Убедитесь в том, что в разделе Implementation модуля UnModul появился оператор **U**ses **U**nFuncProc; , который Delphi вставила в текст модуля UnModul.

Откройте главный файл проекта и убедитесь в том, что проект не содержит посторонних модулей и файлов.

### 8.1.3. Текст модуля UnFuncProc

```
unit UnFuncProc;
```

```
interface
```

```
var
```

```
n:integer; // количество слагаемых в сумме S
```

```
Function F(x:extended):extended;
```

```
Procedure Sum(x:extended;Var s:extended);
```

```
Implementation
```

```
Function F(x:extended):extended;
```

```
begin
```

```
result:=(1-x*x*0.5)*cos(x)-0.5*x*sin(x);
```

```
end;
```

```
Procedure Sum(x:extended;Var s:extended);
```

```
var
```

```
c:extended;
```

```
k:integer;
```

```
begin
```

```
c:=-x*x*0.5;
```

```
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;
end;
end.

```

#### 8.1.4. Текст модуля UnModul

```

Unit UnModul;

```

```

interface

```

```

uses

```

```

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

```

```

type

```

```

  TForm1 = class(TForm)
    Memo1: TMemo;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Edit1: TEdit;
    Edit2: TEdit;
    SpinEdit1: TSpinEdit;
    BitBtn1: TBitBtn;
    procedure FormCreate(Sender: TObject);
    procedure BitBtn1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

```

```

Type

```

```

  func=function(x:extended):extended;           // функциональный тип
  proc=procedure(x:extended;Var s:extended); // процедурный тип
var
  Form1: TForm1;

```

```

implementation

```

```

uses UnFuncProc; // Delphi подключает модуль UnFuncProc

```

```

{$R *.DFM}

```

```
procedure TForm1.FormCreate(Sender: TObject);
```

```
begin
```

```
SpinEdit1.text:='4'; // начальное значение  $N$ 
```

```
Edit1.text:='0.1'; // начальное значение  $X_n$ 
```

```
Edit2.text:='2.0'; // начальное значение  $X_k$ 
```

```
Memo1.Clear;
```

```
Memo1.Lines.Add('Лабораторная работа №8 - Создание модулей');
```

```
end;
```

```
{В процедуре Tablica вычисляется и выводится таблица значений  $x$ ,  $S(x)$  и  $Y(x)$ }
```

```
procedure Tablica(Sum:proc;F:func;n:integer;xn,xk,h:extended);
```

```
var
```

```
x,y,s:extended;
```

```
begin
```

```
Form1.Memo1.Lines.Add(#9+'x'+#9+'S'+#9+'Y');// заголовок таблицы
```

```
x:=xn;
```

```
repeat // цикл по  $x$ 
```

```
Sum(x,s); // вызов процедуры Sum для вычисления  $S(x)$ 
```

```
y:=F(x); // обращение к функции F для вычисления  $Y(x)$ 
```

```
Form1.Memo1.Lines.Add(#9+FloatToStrF(x,ffFixed,5,2) // вывод  $x$ 
```

```
+#9+FloatToStrF(s,ffFixed,6,3) // вывод  $S$ 
```

```
+#9+FloatToStrF(y,ffFixed,6,3)); // вывод  $Y$ 
```

```
x:=x+h;
```

```
until x>xk;
```

```
end;
```

```
procedure TForm1.BitBtn1Click(Sender: TObject);
```

```
var
```

```
xn,xk,h:extended;
```

```
begin
```

```
n:=StrToInt(SpinEdit1.Text);
```

```
xn:=StrToFloat(Edit1.Text);
```

```
xk:=StrToFloat(Edit2.Text);
```

```
h:=(xk-xn)*0.1; // шаг  $h$ 
```

```
Tablica(Sum,F,n,xn,xk,h); // вызов процедуры Tablica для вычисления таблицы
```

```
end;
```

```
end.
```

## 8.2. Выполнение индивидуального задания

По указанию преподавателя выберите из таблицы два варианта индивидуальных заданий. Создайте модуль, в котором вычисление значений  $S(x)$  реализуйте в виде подпрограммы-процедуры, а вычисление значений  $Y(x)$  - в виде подпрограммы-функции. На панели интерфейса установите компонент, с помощью которого реализуйте возможность выбора соответствующего варианта задания и вывод таблицы значений  $x, S_i(x), Y_i(x)$ , где  $i$ -номер варианта. Созданный модуль подключите к проекту и выполните приложение.

**Индивидуальные задания**

В заданиях 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	$1 + \frac{\ln 3}{1!}x + \frac{\ln^2 3}{2!}x^2 + \dots + \frac{\ln^n 3}{n!}x^n$	8	$3^x$
2	$\frac{\pi}{5}$	$\frac{9\pi}{5}$	$\cos x + \frac{\cos 2x}{2} + \dots + \frac{\cos nx}{n}$	18	$-\ln \left  2 \sin \frac{x}{2} \right $
3	$\frac{\pi}{5}$	$\frac{4\pi}{5}$	$\sin x - \frac{\sin 2x}{2} + \dots + (-1)^{n+1} \frac{\sin nx}{n}$	6	$\frac{x}{2}$
4	0.1	0.8	$x \sin \frac{\pi}{4} + x^2 \sin 2 \frac{\pi}{4} + \dots + x^n \sin n \frac{\pi}{4}$	12	$\frac{x \sin \frac{\pi}{4}}{1 - 2x \cos \frac{\pi}{4} + x^2}$
5	0.1	0.8	$x + \frac{x^5}{5} + \dots + \frac{x^{4n+1}}{4n+1}$	16	$\frac{1}{4} \ln \frac{1+x}{1-x} + \frac{1}{2} \operatorname{arctg} x$
6	0.1	1	$1 + \frac{\cos x}{1!} + \dots + \frac{\cos nx}{n!}$	14	$e^{\cos x} \cos(\sin x)$

7	0.1	1	$1 + \frac{3}{1!}x^2 + \dots + \frac{2n+1}{n!}x^{2n}$	8	$(1 + 2x^2)e^{x^2}$
8	0.1	0.8	$\frac{x \cos \frac{\pi}{3}}{1} + \frac{x^2 \cos 2 \frac{\pi}{3}}{2} + \dots + \frac{x^n \cos n \frac{\pi}{3}}{n}$	10	$-\frac{1}{2} \ln(1 - 2x \cos \frac{\pi}{3} + x^2)$
9	0.2	1	$\frac{x-1}{x+1} + \frac{1}{3} \left( \frac{x-1}{x+1} \right)^3 + \dots + \frac{1}{2n+1} \left( \frac{x-1}{x+1} \right)^{2n+1}$	6	$\frac{1}{2} \ln x$
10	$\frac{\pi}{5}$	$\pi$	$-\cos x + \frac{\cos 2x}{2^2} + \dots + (-1)^n \frac{\cos nx}{n^2}$	16	$\frac{1}{4} \left( x^2 - \frac{\pi^2}{3} \right)$
11	0.1	0.8	$\frac{\cos 2x}{3} + \frac{\cos 4x}{15} + \dots + \frac{\cos 2nx}{4n^2 - 1}$	20	$\frac{1}{2} - \frac{\pi}{4}  \sin x $
12	0.1	1	$1 - \frac{3}{2}x^2 + \dots + (-1)^n \frac{2n^2 + 1}{(2n)!} x^{2n}$	18	$\left(1 - \frac{x^2}{2}\right) \cos x - \frac{x}{2} \sin x$
13	0.1	0.8	$x \cos \frac{\pi}{4} + x^2 \cos 2 \frac{\pi}{4} + \dots + x^n \cos n \frac{\pi}{4}$	14	$\frac{x \cos \frac{\pi}{4} - x^2}{1 - 2x \cos \frac{\pi}{4} + x^2}$
14	0.1	0.8	$3x + 8x^2 + \dots + n(n+2)x^n$	12	$\frac{x(3-x)}{(1-x)^3}$
15	$\frac{\pi}{5}$	$\pi$	$\cos x + \frac{\cos 3x}{3^2} + \dots + \frac{\cos(2n-1)x}{(2n-1)^2}$	18	$\frac{\pi^2}{8} - \frac{\pi}{4}  x $
16	-1	1	$x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \dots + (-1)^{n-1} \frac{x^n}{n}$	10	$\ln(1+x)$

17	0.2	0.8	$\frac{x}{3!} + \frac{4x^2}{5!} + \dots + \frac{n^2}{(2n+1)!} x^n$	12	$\frac{1}{4} \left( \frac{x+1}{\sqrt{x}} sh\sqrt{x} - ch\sqrt{x} \right)$
18	-2	-0.1	$-(1+x)^2 + \frac{(1+x)^4}{2} + \dots + (-1)^n \frac{(1+x)^{2n}}{n}$	16	$\ln \frac{1}{2+2x+x^2}$
19	0.1	1	$-\frac{(2x)^2}{2} + \frac{(2x)^4}{24} - \dots + (-1)^n \frac{(2x)^{2n}}{(2n)!}$	8	$2(\cos^2 x - 1)$
20	0.1	0.8	$\frac{x^2}{2} - \frac{x^4}{12} + \dots + (-1)^{n+1} \frac{x^{2n}}{2n(2n-1)}$	10	$x \operatorname{arctg} x - \ln \sqrt{1+x^2}$
21	0.1	0.5	$x - \frac{x^3}{3} + \dots + (-1)^n \frac{x^{2n+1}}{2n+1}$	15	$\operatorname{arctg} x$
22	0.1	1	$1 + 2\frac{x}{2} + \dots + \frac{n^2+1}{n!} \left(\frac{x}{2}\right)^n$	14	$\left(\frac{x^2}{4} + \frac{x}{2} + 1\right) e^{\frac{x}{2}}$
23	0.1	1	$1 + \frac{2x}{1!} + \dots + \frac{(2x)^n}{n!}$	10	$e^{2x}$
24	0.1	1	$\frac{x^3}{3} - \frac{x^5}{15} + \dots + (-1)^{n+1} \frac{x^{2n+1}}{4n^2-1}$	12	$\frac{1+x^2}{2} \operatorname{arctg} x - \frac{x}{2}$
25	0.1	1	$x + \frac{x^3}{3!} + \dots + \frac{x^{2n+1}}{(2n+1)!}$	8	$\frac{e^x - e^{-x}}{2}$
26	0.1	1	$1 - \frac{x^2}{1!} + \frac{x^4}{2!} - \frac{x^6}{3!} + \dots + (-1)^n \frac{x^{2n}}{n!}$	14	$e^{-x^2}$
27	0.1	1	$1 - \frac{x^2}{2!} + \dots + (-1)^n \frac{x^{2n}}{(2n)!}$	8	$\cos x$
28	0.1	1	$1 + \frac{\cos \frac{\pi}{4}}{1!} x + \dots + \frac{\cos n \frac{\pi}{4}}{n!} x^n$	12	$e^{x \cos \frac{\pi}{4}} \cos(x \sin \frac{\pi}{4})$
29	0.1	1	$1 + \frac{x^2}{2!} + \dots + \frac{x^{2n}}{(2n)!}$	10	$\frac{e^x + e^{-x}}{2}$
30	0.1	1	$x - \frac{x^3}{3!} + \dots + (-1)^n \frac{x^{2n+1}}{(2n+1)!}$	16	$\sin x$

## ПРОГРАММИРОВАНИЕ АЛГОРИТМОВ С ИСПОЛЬЗОВАНИЕМ ДИНАМИЧЕСКИХ СТРУКТУР ДАННЫХ

**Цель лабораторной работы:** освоить методику создания приложений, в которых используются динамические структуры данных.

### Примеры создания приложений

#### 9.1 Использование динамических массивов

**Задание:** создать приложение для вычисления наименьшего и наибольшего из всех значений элементов целочисленной матрицы  $A = \{a_{ij}\}$ , где  $i=1,2,\dots, m$ ;  $j=1,2,\dots, n$ . Значения  $m$  и  $n$  задаются пользователем на панели интерфейса, а элементы матрицы  $A$  генерируются с помощью датчика случайных чисел и размещаются в памяти динамически.

Один из возможных вариантов панели интерфейса создаваемого приложения показан на рис.9.1.

##### 9.1.1. Размещение компонентов на Форме

Разместим на Форме компоненты Label, SpinEdit, Button и StringGrid.



Рис. 9.1

Сохраним модуль под именем UnDinMas (текст модуля приведен в п.9.1.3).

#### 9.1.2 Создание процедур обработки событий FormCreate и Button1Click

Двойным нажатием клавиши “мыши” на Форме и кнопке Button1 создайте соответствующие процедуры обработки событий. Пользуясь текстом модуля UnDinMas, внимательно наберите операторы этих процедур.

При желании можно создать процедуру, которая будет выделять заданным цветом границы ячеек с наименьшим и наибольшим значениями в

компоненте StringGrid. Для создания такой процедуры сделайте активным компонент StringGrid и на странице Events(события) Инспектора Объектов дважды щелкните “мышью” в правой части события OnDrawCell. В ответ Delphi создаст обработчик этого события – процедуру **procedure TForm1.StringGrid1DrawCell** и установит курсор между операторами **begin** и **end** этой процедуры. Используя текст модуля UnDinMas, внимательно наберите операторы процедуры TForm1.StringGrid1DrawCell.

### 9.1.3 Текст модуля UnDinMas

**Unit** UnDinMas;

**interface**

**uses**

Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls, Spin, Grids, Buttons;

**type**

TForm1 = class(TForm)

Label1: TLabel;

SpinEdit1: TSpinEdit;

SpinEdit2: TSpinEdit;

Label8: TLabel;

StringGrid1: TStringGrid;

Label2: TLabel;

Label5: TLabel;

Label3: TLabel;

Button1: TButton;

Label4: TLabel;

Label6: TLabel;

Label7: TLabel;

Label9: TLabel;

**procedure** FormCreate(Sender: TObject);

**procedure** SpinEdit1Change(Sender: TObject);

**procedure** SpinEdit2Change(Sender: TObject);

**procedure** StringGrid1DrawCell(Sender: TObject; Col, Row: Integer;  
Rect: TRect; State: TGridDrawState);

**procedure** Button1Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

**end;**

**var**

Form1: TForm1;

**implementation**

{ \$R \*.DFM }

**Type**

Mas=array[1..1] of integer; // массив целочисленных значений

pMas=array[1..1] of ^mas; // массив указателей

**var** // объявление глобальных переменных

pA:^pMas; // указатель на массив указателей

m,n,max,min:integer;

**procedure** TForm1.FormCreate(Sender: TObject);

**begin**

m:=6; // начальное значение m

n:=8; // начальное значение n

SpinEdit1.Text:='6';

SpinEdit2.Text:='8';

StringGrid1.RowCount:=m; // количество строк

StringGrid1.ColCount:=n; // количество столбцов

**end;**

**procedure** TForm1.SpinEdit1Change(Sender: TObject);

**begin**

m:=StrToInt(SpinEdit1.Text); // m присваивается содержимое поля редактора

StringGrid1.RowCount:=m;

**end;**

**procedure** TForm1.SpinEdit2Change(Sender: TObject);

**begin**

n:=StrToInt(SpinEdit2.Text); // n присваивается содержимое поля редактора

StringGrid1.ColCount:=n;

**end;**

**procedure** TForm1.Button1Click(Sender: TObject);

**label 1;**

**var**

i,j,k,l,r:integer;

**begin**

Randomize; // инициализация датчика случайных чисел

GetMem(pA,4\*m); // выделение памяти для массива из m указателей

**for** i :=1 to m **do**

**begin** // формирование i-й строки массива

{ Выделение памяти для n элементов i-й строки }

GetMem(pA^[i],SizeOf(integer)\*n);

pA^[1]^1:=Random(1000); // случайное целое число занести в массив

**for** j:=1 to n **do**

**begin** // формирование j-го элемента строки

l:r:=Random(1000); // генерирование случайного числа

```

for k:=1 to i do
  for l:=1 to j do
    if r=pA^[k]^l then // если такое число уже есть в массиве тогда...
      goto 1;
    pA^[i]^j:=r; // случайное число занести в массив
  end;
end;
for i:=1 to m do // элементы массива занести в ячейки
  for j:=1 to n do // компонента StringGrid1
    StringGrid1.Cells[j-1,i-1]:=IntToStr(pA^[i]^j);
  { Поиск min и max значений среди элементов массива}
  max:=pA^[1]^1;
  min:=max;
  for i:=1 to m do
    for j:=1 to n do
      if pA^[i]^j<min then
        min:=pA^[i]^j
      else
        if pA^[i]^j>max then
          max:=pA^[i]^j;
        Label7.Caption:=IntToStr(min); // вывод min значения
        Label9.Caption:=IntToStr(max); // вывод max значения
      for i:=1 to m do
        { Освобождение памяти, занимаемой n элементами i-й строки}
        FreeMem(pA^[i],SizeOf(integer)*n);
        { Освобождение памяти, занимаемой массивом из m указателей}
        FreeMem(pA,4*m);
      end;
    procedure TForm1.StringGrid1DrawCell(Sender: TObject; Col, Row: Integer;
      Rect: TRect; State: TGridDrawState);
    begin
      with StringGrid1.Canvas do
        if StringGrid1.Cells[Col,Row]=IntToStr(min) then // если элемент ячейки
          begin // равен min тогда...
            Brush.Color:=clGreen; // установить цвет кисти зеленый
            FrameRect(Rect); // выделить границы ячейки заданным цветом
          end
          else
            if StringGrid1.Cells[Col,Row]=IntToStr(max) then // если элемент ячейки
              begin // равен max тогда...
                Brush.Color:=clRed; // установить цвет кисти красный
                FrameRect(Rect); // выделить границы ячейки заданным цветом
              end
            end;
          end.

```

## 9.2 Использование динамических списков

**Задание2:** создать приложение для формирования стека, который заполняется путем ввода целых положительных чисел с клавиатуры. Как только будет введено первое отрицательное число, содержимое стека выводится на панель интерфейса, а память занимаемая его элементами освобождается.

Один из возможных вариантов панели интерфейса создаваемого приложения показан на рис.9.2.

### 9.2.1. Размещение компонентов на Форме

Разместим на Форме компоненты Label, Edit, Button и Мемо.

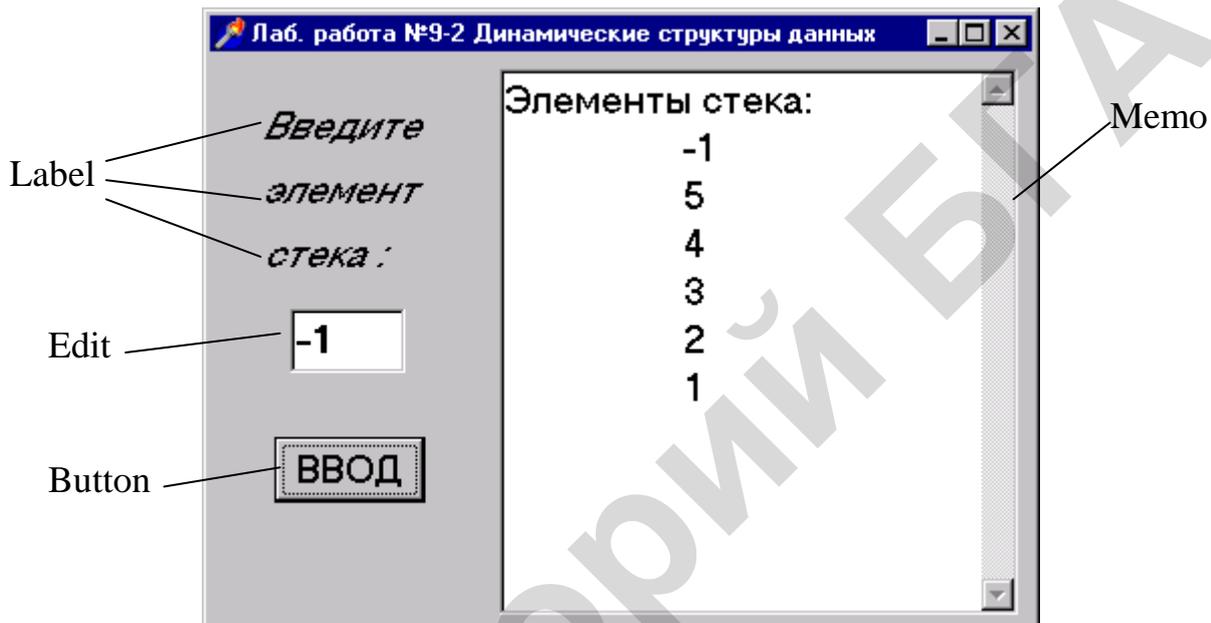


Рис. 9.2

Сохраним модуль под именем UnStek (текст модуля приведен в п.9.2.3).

### 9.2.2 Создание процедур обработки событий FormCreate и Button1Click

Двойным нажатием клавиши “мыши” на Форме и кнопке Button1 создайте соответствующие процедуры обработки событий. Используя текст модуля UnStek, внимательно наберите операторы этих процедур.

### 9.2.3 Текст модуля UnStek

```
Unit UnStek;
```

```
interface
```

```
uses
```

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

```
type
```

```
TForm1 = class(TForm)
```

```
Label1: TLabel;
```

```

Edit1: TEdit;
Button1: TButton;
Label2: TLabel;
Label3: TLabel;
Memo1: TMemo;
procedure Button1Click(Sender: TObject);
procedure FormCreate(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
var
  Form1: TForm1;
implementation
{$R *.DFM}
Type
PSt=^Zap;
Zap=record
  inf:integer;
  adr:PSt
end;
Var      // объявление глобальных переменных:
PVer,     // указатель вершины стека
PTek:PSt; // текущий указатель
ElSt:integer; // элемент стека
procedure TForm1.Button1Click(Sender: TObject);
begin
New(PTek);           // выделить память
ElSt:=StrToInt(Edit1.Text); // в ElSt занести значение из Edit1
PTek^.inf:=ElSt;    // в информационную часть стека занести ElSt
PTek^.adr:=PVer;    // в адресную часть занести указатель на вершину
PVer:=PTek; // указатель вершины должен указывать на последний элемент
if ElSt>=0 then    // если элемент стека неотрицательный тогда...
  begin
    Edit1.Text:=""; // очистить окно редактора Edit1
    Edit1.SetFocus; // передать фокус ввода редактору Edit1
  end
  else
    begin
      Memo1.Lines.Add('Элементы стека:'); // вывести заголовок
    repeat
      Memo1.Lines.Add(#9+IntToStr(PTek^.inf)); // вывод элементов
      PVer:=PTek^.adr;
      Dispose(PTek); // освободить память
    
```

```

    PTek:=PVer
    until PTek=nil;
end;
end;
procedure TForm1.FormCreate(Sender: TObject);
begin
    PVer:=nil; // инициализировать указатель вершины
    ElSt:=0;   // инициализировать элемент стека
end;
end.

```

### 9.3. Выполнение индивидуального задания

По указанию преподавателя выберите два варианта индивидуальных заданий. В заданиях №1-№15 необходимо использовать динамические массивы, а в заданиях №16-№30 – динамические списки. Во всех заданиях необходимо предусмотреть контрольный вывод исходных данных.

1. Создать приложение, которое осуществляет ввод  $k$  значений элементов одномерного массива с клавиатуры, меняет местами элементы с наибольшим и наименьшим значениями среди четных и выводит полученный массив.
2. Создать приложение, которое осуществляет ввод  $m$  строк и  $n$  столбцов двумерного массива с клавиатуры и выводит номер строки и номер столбца наименьшего из всех значений его элементов.
3. Создать приложение, которое осуществляет ввод  $k$  значений элементов одномерного массива с клавиатуры и выводит порядковый номер элемента с наименьшим значением среди нечетных.
4. Создать приложение, которое осуществляет ввод значений элементов двумерного массива  $n$ -го порядка с клавиатуры и выводит значение наибольшего из элементов главной диагонали.
5. Создать приложение, которое осуществляет ввод  $k$  значений элементов одномерного массива с клавиатуры, изменяет порядок следования элементов на противоположный и выводит полученный массив.
6. Создать приложение, которое осуществляет ввод  $k$  значений элементов одномерного массива с клавиатуры и выводит порядковый номер элемента с наибольшим значением среди четных.
7. Создать приложение, которое осуществляет ввод значений элементов двумерного массива  $n$ -го порядка с клавиатуры и выводит значение суммы элементов главной диагонали.
8. Создать приложение, которое осуществляет ввод  $k$  значений элементов одномерного массива с клавиатуры, меняет местами элементы с минимальным и максимальным значениями и выводит полученный массив.

9. Создать приложение, которое осуществляет ввод  $k$  значений элементов одномерного массива с клавиатуры и выводит порядковый номер элемента с наименьшим значением среди положительных.

10. Создать приложение, которое осуществляет ввод значений элементов двумерного массива  $n$ -го порядка с клавиатуры, изменяет порядок следования элементов главной диагонали на противоположный и выводит преобразованный массив.

11. Создать приложение, которое осуществляет ввод  $k$  значений элементов одномерного массива с клавиатуры, меняет местами элементы с минимальным и максимальным значениями среди положительных и выводит полученный массив.

12. Создать приложение, которое осуществляет ввод  $k$  значений элементов одномерного массива с клавиатуры и выводит порядковый номер элемента с наибольшим значением среди отрицательных.

13. Создать приложение, которое осуществляет ввод  $k$  значений элементов одномерного массива с клавиатуры, меняет местами элементы с наибольшим значением среди отрицательных и наименьшим среди положительных и выводит полученный массив.

14. Создать приложение, которое осуществляет ввод  $k$  значений элементов одномерного массива с клавиатуры и выводит среднее арифметическое значение элементов массива.

15. Создать приложение, которое осуществляет ввод  $k$  значений элементов одномерного массива с клавиатуры, меняет местами элементы с наименьшим значением среди четных и наибольшим среди нечетных и выводит полученный массив.

16. Создать приложение, которое заносит в стек целые положительные числа с клавиатуры, выводит содержимое стека и среднее арифметическое значение его элементов.

17. Создать приложение, которое заносит в стек символы с клавиатуры, выводит содержимое стека и сообщение о том, содержится или нет в стеке заданный символ.

18. Создать приложение, которое заносит в каждый элемент стека английское слово с клавиатуры и, как только будет введено слово "end", выводит содержимое стека.

19. Создать приложение, которое заносит в стек произвольные целые числа с клавиатуры, выводит содержимое стека и сообщение о том, содержится или нет в стеке заданное число.

20. Создать приложение, которое заносит в стек символы с клавиатуры, выводит содержимое стека и сообщение о том, упорядочены ли элементы стека по алфавиту или нет.

21. Создать приложение, которое заносит в стек положительные целые числа с клавиатуры и, как только будет введено число, равное сумме введенных чисел, выводит содержимое стека.

22. Создать приложение, которое заносит в стек произвольные числа с клавиатуры, выводит содержимое стека и сообщение о том, упорядочены ли элементы стека по убыванию.

23. Создать приложение, которое заносит в каждый элемент стека русское слово с клавиатуры и, как только будет введено слово “конец”, выводит содержимое стека и сообщение о том, содержится или нет в стеке заданное слово.

24. Создать приложение, которое заносит в стек произвольные числа с клавиатуры, выводит содержимое стека и удаляет из стека наибольший элемент.

25. Создать приложение, которое заносит в стек произвольные целые числа с клавиатуры. Как только сумма введенных чисел станет равна нулю, приложение должно вывести содержимое стека.

26. Создать приложение, которое заносит в каждый элемент стека русское слово с клавиатуры и, как только будет введено слово “конец”, выводит содержимое стека и затем удаляет из него любое слово, содержащееся в стеке (ввести с клавиатуры).

27. Создать приложение, которое заносит в стек произвольные целые числа с клавиатуры. Как только будет введено число 0(ноль), приложение должно вывести содержимое стека, а затем сформировать из него два новых стека: в первом должны находиться положительные, а во втором – отрицательные числа.

28. Создать приложение, которое символы, упорядоченные по алфавиту, вводит с клавиатуры и заносит в стек. Приложение должно вывести содержимое стека, ввести с клавиатуры заданный символ и вставить его в стек так, чтобы элементы стека опять оказались упорядоченными по алфавиту.

29. Создать приложение, которое заносит в стек буквы русского и латинского алфавитов в произвольном порядке с клавиатуры. Как только будет введен символ “.”(точка), приложение должно вывести содержимое стека, а затем сформировать из него два новых стека: в первом должны находиться буквы русского, а во втором – буквы латинского алфавита.

30. Создать приложение, которое заносит в стек целые числа с клавиатуры. Как только будет введено число 9999, приложение выводит содержимое стека и реверсирует стек(направления ссылок в стеке изменяются так, что вершина и дно стека меняются местами).

## СОЗДАНИЕ WINDOWS-ПРИЛОЖЕНИЙ

**Цель лабораторной работы:** совершенствовать навыки использования возможностей визуальной среды *Delphi* для создания *Windows*-приложений.

### **Выполнение индивидуального задания**

1. Используя визуальные компоненты Edit, Label и Button, создать приложение, реализующее простейший калькулятор с четырьмя арифметическими действиями (сложение, вычитание, умножение и деление).

2. Используя визуальные компоненты Edit, Label и Button, создать приложение, реализующее простейший калькулятор с двумя арифметическими действиями (сложение, вычитание) и памятью.

3. Используя визуальные компоненты Edit, Label и Button, создать приложение, реализующее простейший калькулятор с двумя арифметическими действиями (сложение, вычитание) и функциями (sin, cos).

4. Используя визуальные компоненты Edit, Label и Button, создать приложение, реализующее простейший калькулятор с двумя арифметическими действиями (умножение, деление) и функциями (Exp, Ln).

5. Используя визуальные компоненты Edit, Label, Button и ListBox, создать приложение, реализующее простейший калькулятор с двумя арифметическими действиями (сложение, вычитание) и памятью в виде ListBox.

6. Используя визуальные компоненты Edit, Label, Button и ListBox, создать приложение, реализующее простейший калькулятор с двумя арифметическими действиями (умножение, деление) и памятью в виде ListBox.

7. Используя визуальные компоненты Edit, Label, Button и ListBox, создать приложение, реализующее простейший калькулятор с двумя арифметическими действиями (сложение, вычитание), функциями (sin, cos) и памятью в виде ListBox.

8. Используя визуальные компоненты Edit, Label, Button и ListBox, создать приложение, реализующее простейший калькулятор с двумя арифметическими действиями (умножение, деление), функциями (Exp, Ln) и памятью в виде ListBox.

9. Используя визуальные компоненты Edit, Label и Button, создать приложение, реализующее ввод строки из нескольких слов и подсчет количества слов в строке.

10. Используя визуальные компоненты Edit, Label и Button, создать приложение, реализующее ввод строки из нескольких слов и вывод слова наибольшей длины.

11. Используя визуальные компоненты Edit, Label, Button и ListBox, создать приложение, реализующее ввод строки из нескольких слов и сортировку в ListBox слов в алфавитном порядке.

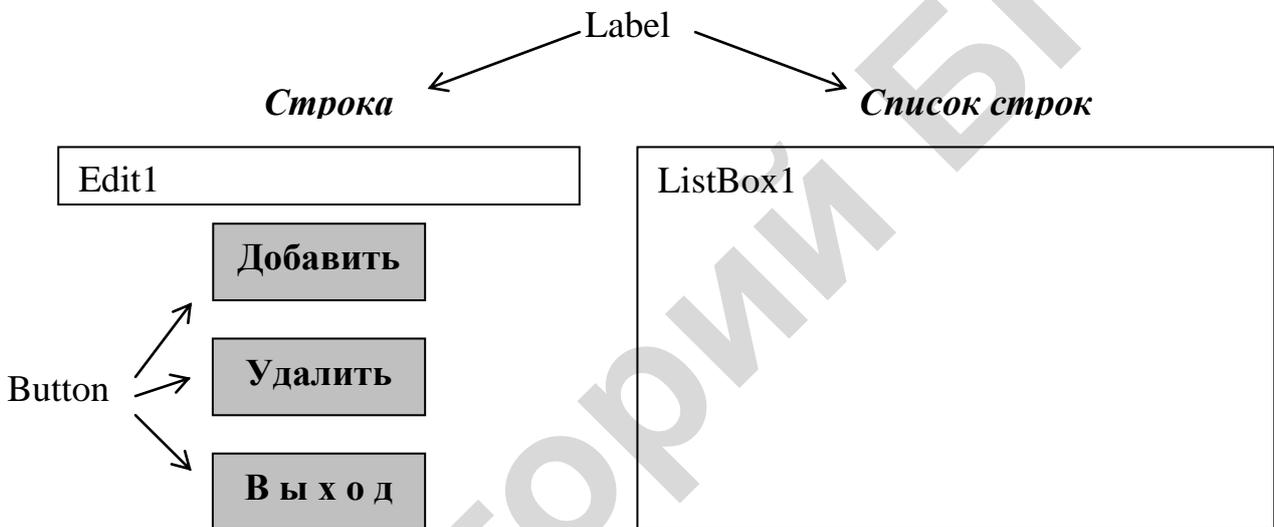
12. Используя визуальные компоненты Edit, Label, Button и ListBox, создать приложение, реализующее ввод строки из нескольких слов и перестановку в ListBox слов в порядке, обратном введенному.

13. Используя визуальные компоненты Edit, Label, Button и ListBox, создать приложение, реализующее ввод отдельных строк, добавление их в ListBox и удаление из списка ListBox.

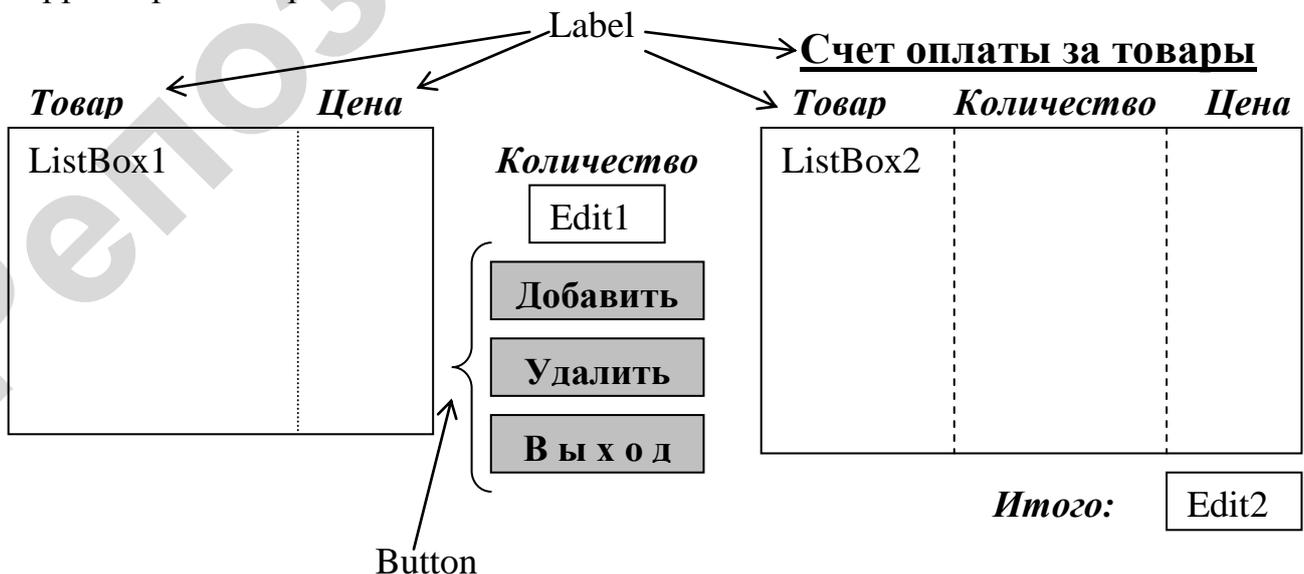
14. Используя визуальные компоненты Edit, Label, Button и ListBox, создать приложение, реализующее ввод отдельных строк, добавление их в ListBox и удаление из списка ListBox.

15. Создать приложение, реализующее вывод текстового файла в компонент Мемо.

16. Создать приложение, на панели интерфейса которого с помощью кнопок “Добавить” и “Удалить” можно обновлять список строк.



17. Создать приложение, на панели интерфейса которого из заданного в программе ценника (ListBox1) формировать счет оплаты за товары (ListBox2). С помощью кнопок “Добавить” и “Удалить” реализовать возможность корректировки строк счета.



18. Создать приложение, на панели интерфейса которого вводится имя текстового файла и нажатием кнопки “Читать” поле Memo1 заполняется строками этого файла. С помощью кнопки “Очистить” реализовать возможность очистки Edit1 и Memo1.

**Имя текстового файла:**

**Текст файла:**

Edit1	Memo1
<b>Читать</b>	
<b>Очистить</b>	
<b>В ы х о д</b>	

19. Создать приложение, на панели интерфейса которого вводится строка, а кнопкой “Добавить” формируется поле “Список строк1”. Кнопка “Перенести” служит для переноса сразу нескольких выбранных строк из поля “Список строк1” в поле “Список строк2”. Кнопка “Очистить” должна очищать все поля.

**Введите строку:**

**Список строк1**

**Список строк2**

Edit1	ListBox1	ListBox2
<b>Добавить</b>		
<b>Перенести</b>		
<b>Очистить</b>		
<b>В ы х о д</b>		

20. Создать приложение для формирования экзаменационной ведомости с панелью интерфейса следующего вида

**Экзаменационная ведомость**

	<i>Группа</i>	<i>Ф.И.О.</i>	<i>Оценка</i>
<i>Группа:</i>	ComboBox1   ▼		
<i>Ф.И.О.:</i>	Edit1		
<i>Оценка:</i>	ComboBox2   ▼		
<b>Добавить</b>	<b>Очистить</b>	<b>В ы х о д</b>	
	ListBox1		

21. Создать приложение для перевода вещественных чисел из десятичной системы счисления в шестнадцатеричную систему и обратно с панелью интерфейса следующего вида

<i>Число в 10 с/с:</i>	<i>Число в 16 с/с:</i>
<input type="text" value="Edit1"/>	<input type="text" value="Edit2"/>
<input type="button" value="10→16"/>	<input type="button" value="10←16"/>
<input type="button" value="В ы х о д"/>	

22. Создать приложение для учета оплаты за телефон с панелью интерфейса следующего вида

<i>Телефон:</i>	<input type="button" value="Добавить"/>	<i>Телефон</i>	<i>Время</i>	<i>Сумма</i>						
<input type="text" value="Edit1"/>		<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%;"><i>Телефон</i></th> <th style="width: 25%;"><i>Время</i></th> <th style="width: 25%;"><i>Сумма</i></th> </tr> </thead> <tbody> <tr> <td colspan="3" style="text-align: center;">ListBox1</td> </tr> </tbody> </table>	<i>Телефон</i>	<i>Время</i>	<i>Сумма</i>	ListBox1				
<i>Телефон</i>	<i>Время</i>		<i>Сумма</i>							
ListBox1										
<i>Время:</i>	<input type="button" value="Очистить"/>									
<input type="text" value="Edit2"/>										
<i>Тариф:</i>	<input type="button" value="В ы х о д"/>									
<input type="text" value="Edit3"/>				<i>Итого:</i> <input type="text" value="Edit4"/>						

23. Создать приложение для перезаписи файлов, выбранных из *каталога 1*, в *каталог 2* с панелью интерфейса следующего вида

<i>Каталог 1</i>	<i>Каталог 2</i>	
<input type="text" value="Edit1"/>	<input type="text" value="Edit2"/>	
<i>Имена файлов:</i>	<i>Имена файлов:</i>	
ListBox1	ListBox2	
<input type="button" value="Показать"/>	<input type="button" value="Переписать"/>	<input type="button" value="В ы х о д"/>

24. Создать приложение для вывода содержимого выбранного текстового файла с панелью интерфейса следующего вида

<i>Каталог</i>		<i>Содержимое выбранного файла:</i>	
Edit1	Показать	ListBox1	
<b>Файлы:</b>			
ComboBox1	Вывести		
	В ы х о д		

25. Создать приложение для организации списка адресов с панелью интерфейса следующего вида

<i>Фамилия</i>		<i>Список адресов</i>						
Edit1	Добавить	<i>Фамилия</i>	<i>Областной Улица Дом Кв иентр</i>					
<i>Обл. центр</i>		<table border="1"> <tr> <td>ListBox1</td> <td></td> <td></td> <td></td> <td></td> </tr> </table>		ListBox1				
ListBox1								
ComboBox1 ▼	Убрать							
<i>Улица</i>	Сортировка							
Edit2								
<i>Дом</i>	<i>Кв.</i>							
		В ы х о д						

26. Создать приложение для организации ведомости сдачи экзаменов с панелью интерфейса следующего вида

<i>Дисциплина</i>		<i>Веломость сдачи экзаменов</i>				
Edit1	Добавить	<i>Фамилия</i>	<i>Дисциплина</i>			
		<table border="1"> <tr> <td>ListBox1</td> <td></td> <td></td> </tr> </table>		ListBox1		
ListBox1						
<i>Оценка</i>	Очистить					
Edit2	Сортировка					
<i>Фамилия, инициалы</i>	В ы х о д					
Edit3		Средний балл: Label1				

27. Создать приложение для организации списка сотрудников фирмы с панелью интерфейса следующего вида

<b>Фамилия</b> Edit1	<b>Добавить</b>	<b>Фамилия Должность Оклад</b> ListBox1
<b>Должность</b> ComboBox1 ▼	<b>Очистить</b>	
<b>Оклад</b> Edit2	<b>Сортировать</b>	
	<b>В ы х о д</b>	

28. Создать приложение для организации списка приема пациентов с панелью интерфейса следующего вида

<b>Врач</b> ComboBox1 ▼	<b>Добавить</b>	<b>Список приема пациентов</b> <b>Врач      Время      Пациент</b> ListBox1
<b>Время приема</b> ComboBox2 ▼	<b>Очистить</b>	
<b>Пациент</b> Edit1	<b>В ы х о д</b>	

29. Создать приложение для организации расписания вылета самолетов с панелью интерфейса следующего вида

<b>Номер рейса</b> ComboBox1 ▼	<b>Добавить</b>	<b>Расписание вылета самолетов</b> <b>Рейс      Самолет      Город      Время</b> ListBox1
<b>Тип самолета</b> ComboBox2 ▼	<b>Сортировать</b>	
<b>Пункт назначения</b> ComboBox3 ▼	<b>Очистить</b>	
<b>Время вылета</b> Edit1	<b>В ы х о д</b>	

## ЛИТЕРАТУРА

1. Фаронов В.В. DELPHI 3: Учебный курс. – М.: Нолидж, 1998.
2. Федоров А.Г. Delphi 3.0 для всех.. - М.: КомпьютерПресс, 1998.
3. Гофман В., Хомоненко А. Delphi 5. СПб.: БХВ- Санкт-Петербург, 2000.
4. Программирование на языке *Borland Pascal* в среде *Windows*. Лабораторный практикум по курсу “Программирование”. В 2-х частях для студентов радиотехнических и экономических специальностей / А.В.Епихин, А.Б.Закалюкин, С.В.Колосов, А.А.Навроцкий, А.И.Шакирин; Под общ. ред. С.В.Колосова. – Мн.:БГУИР, 1997.
5. 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.Синицына. – Мн.: БГУИР, 1998.
6. Лабораторный практикум по программированию. Ч.2./ А.А.Бурцев, С.В.Колосов, Д.В.Офицеров, В.И.Убийконь, А.И.Шакирин; Под общ. ред. С.В.Колосова. -Мн.:БГУИР, 1994.
7. Н.Вирт. Алгоритмы + структуры данных = программы. \_М.:Мир, 1985.
8. Д.Кнут. Искусство программирования для ЭВМ. т.3 – М.: Мир, 1978.
9. Прищепов М.А., Степанцов В.П., Севернева Е.В. Основы алгоритмизации и программирования. Мн.: Элайда, 2000.

Программирование алгоритмов с использованием записей.....	3
Программирование алгоритмов с использованием файлов.....	10
Программирование алгоритмов с использованием функций и процедур. Создание модулей.....	21
Программирование алгоритмов с использованием динамических структур данных.....	28
Создание Windows-приложений.....	37
Литература.....	43

Репозиторий БГАТУ