

## 8 Работа со строками и записями

В этой главе мы познакомимся с двумя типами данных – строками и записями. На примерах будет показана работа основных функций обработки строк. Далее читатель может познакомиться с записями (*record*). *Запись* – это сложная структура данных, состоящая из фиксированного количества объектов, называемых *полями*. Поля записи могут быть разного типа.

### 8.1 Обработка текста

Часто в задачах программирования приходится обрабатывать текст. Обработка текста в Free Pascal состоит из обработки символов и обработки строк. Понятие строки уже было введено в главе 2 и там же рассмотрены основные функции работы со строками. Напомним основные моменты.

*Символ* - это буква, цифра или какой-либо знак. Кодовая таблица символов состоит из 256 позиций, т.е. каждый символ имеет свой уникальный код от 0 до 255. Так как код символа представляет собой число не более 255, то очевидно, что в памяти компьютера один символ занимает 1 байт. Для работы с символами существует тип данных `char` (1 байт).

*Строка* – это набор символов. Текстовая строка имеет определённую длину. Длина строки – это количество символов, которые она содержит. Если один символ занимает 1 байт, то строка из  $N$  символов занимает в памяти соответственно  $N$  байт. Для работы со строками существует тип данных `string`.

Для задания в программе символьной или строковой переменной используются одинарные кавычки, например:

```
s:='Q'; s1:='Privet';
```

В программе можно ограничить длину текста, хранимого в строковой переменной. Для этого строковую переменную нужно объявить следующим образом (в скобках указывается максимальная длина строки):

```
var str: string[20];
```

Рассмотрим основные операции со строками.

Одной из основных операций является объединение строк. Во Free Pascal для этого используют операцию «+». Например:

```

var s: string;
begin
  s:='my ' + ' ' + 'text';
  ...

```

Результатом работы оператора будет строка `my text`.

*Строка* - это массив символов. Для обращения к элементу строки достаточно указать его номер в квадратных скобках рядом с именем строки. Например:

```

var s: string; c: char;
begin
  s:='Privet';
  c:=s[4];
  ...

```

Результатом работы будет символ `v`.

Основные функции и процедуры обработки строк, приведены в табл. 2.7. Некоторые из них мы неоднократно применяли при решении задач. В основном это были функции преобразования строк в численные типы данных и наоборот. Рассмотрим другие функции работы со строками из табл. 2.7 на примере:

```

var Str1, Str2, Str3 , Str4, Str5: String;
    K, L : Integer;
begin
  Str1:='Иванов';    //Первая строка.
  Str2:='Сергей';    //Вторая строка.
  Writeln('Первая строка -', Str1);
  Writeln('Вторая строка -', Str2);
  //-----
  //Результат работы операторов:
  //Первая строка – Иванов
  //Вторая строка – Сергей
  //-----
  //Объединение 1-й и 2-й строк.
  Str3:=Str1+' '+Str2; //Третья строка.
  Writeln('Третья строка -', Str3);
  //-----
  //Результат работы операторов:
  //Третья строка – Иванов Сергей
  //-----

```

```
//Определение длины строки.
L:= Length(Str3);
Writeln('Длина третьей строки-',L,' символов');
//-----
//Результат работы операторов:
//Длина третьей строки - 13 символов
//-----
Str4:='в'; //Четвертая строка.
//Поиск буквы «в» в третьей строке.
K:=Pos(Str4, Str3);
Writeln('Позиция «в» в третьей строке -',K);
//-----
//Результат работы операторов:
//Позиция «в» в третьей строке - 2
//-----
//Копирование в переменную Str4 6
//символов, начиная с 8-го, из строки Str3.
Str4:=copy(Str3,8,6);
Writeln('Четвертая строка -',Str4);
//-----
//Результат работы операторов:
//Четвертая строка - Сергей
//-----
//Выделение первого слова из третьей строки.
//Первое слово - набор букв от первого
//символа до пробела.
Str5:=Copy(Str3,1,Pos(' ',Str3)-1);
Writeln('Первое слово в 3-й строке - ',Str5);
//-----
//Результат работы операторов:
//Первое слово в третьей строке - Иванов
//-----
//Удаление из третьей строки
//двух символов, начиная с 5-й позиции.
delete(Str3,5,2);
Writeln('3-я строка после удаления -',Str3);
//-----
//Результат работы операторов:
```

```
//Третья строка после удаления — Иван Сергей  
//-----  
end.
```

### ЗАДАЧА 8.1. Найти сумму элементов числового массива.

Задача очень простая. Ее алгоритм подробно рассмотрен в п. 5.5. Но так как в данной главе речь идет об обработке строк, предположим, что массив вводится в поле Edit формы, то есть представляет собой строку символов. Значит, для решения задачи нужно строку преобразовать в числовой массив. Это можно сделать так. Из исходной строки выделить группу символов (подстроку), расположенных до первого пробела. Полученную подстроку преобразовать в число. Если преобразование прошло успешно, получен первый элемент числового массива. Далее нужно удалить подстроку и следующий за ней пробел из исходной строки. Исходная строка стала короче, и второй элемент массива передвинулся в начало. Выделение подстроки до первого пробела и преобразование ее в число даст второй элемент массива и так далее.

Итак, создадим новый проект. На форме (рис. 8.1) разместим поле Edit1 для ввода исходных данных, компонент ListBox1 для вывода результатов и кнопку Button1 для запуска проекта.

Предположим, что свойство Text поля Edit1 имеет вид:

Text =1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2.0

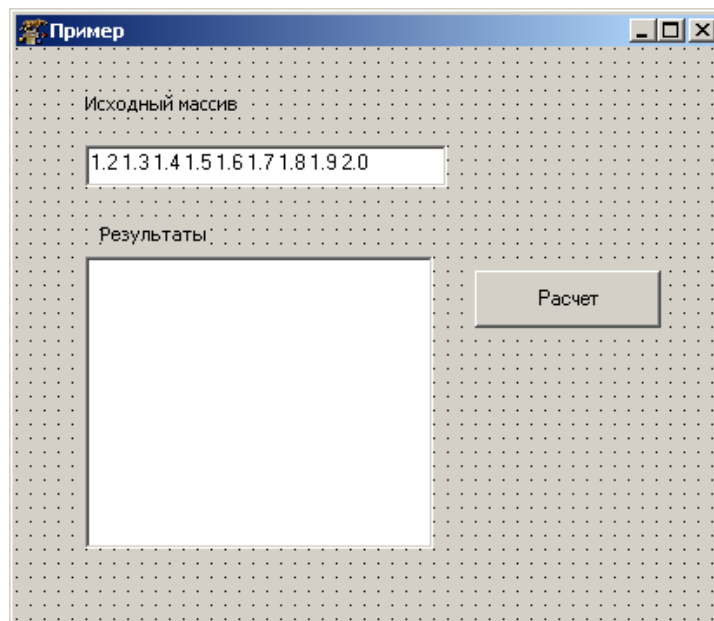


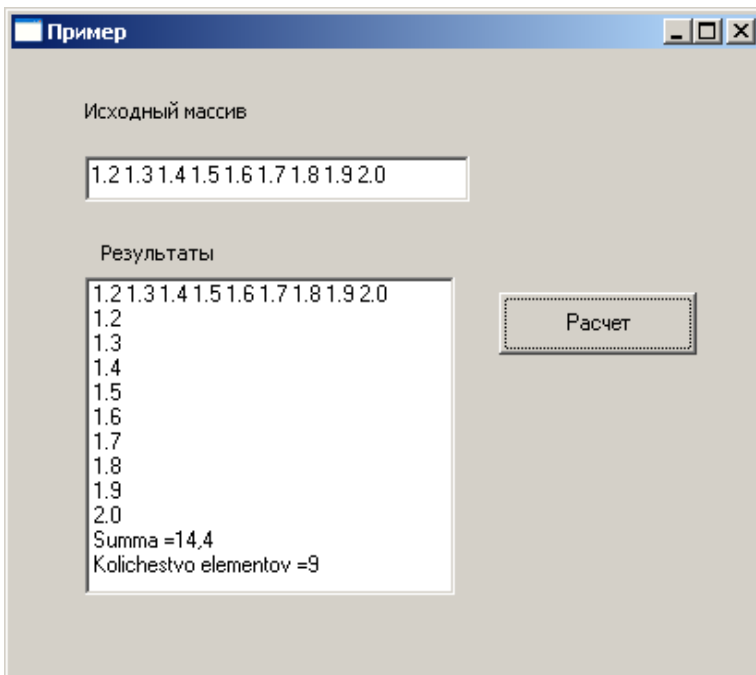
Рисунок 8.1: Окно формы для считывания массива

Ниже приведен листинг программы с комментариями. Результаты работы приложения показаны на рис. 8.2.

```
unit Unit1;
{$mode objfpc}{$H+}
interface
uses
  Classes, SysUtils, LResources, Forms,
    Controls, Graphics, Dialogs, StdCtrls;
type
  { TForm1 }
  TForm1 = class(TForm)
    Button1: TButton;
    Edit1: TEdit;
    Label1: TLabel;
    Label2: TLabel;
    ListBox1: TListBox;
    Memo1: TMemo;
  procedure Button1Click(Sender: TObject);
  private
    { private declarations }
  public
    { public declarations }
end;
var
  Form1: TForm1;
implementation
  { TForm1 }
  procedure TForm1.Button1Click(Sender: TObject);
  var Str1, Str2, Str3: String;
      i: integer;
      word: String;
      x: array [0..100] of real;
      Sum: real;
  begin
    //Чтение строки из поля Edit1 диалогового окна.
    Str1:=Edit1.text;
    //Вывод строки в поле ListBox1.
    ListBox1.Items.Add(Str1);
```

```
//Если первый символ - пробел, то удаляем его
if Str1[1] = ' ' then Delete(Str1,1,1);
//Если в строке есть двойные пробелы,
//то заменяем их одинарным.
while Pos(' ',Str1) > 0 do
    Delete(Str1,Pos(' ',Str1),1);
//Пусть последний символ в строке - пробел.
if Str1[Length(Str1)] <> ' ' then
    Str1:=Str1+' ';
i:=0; Sum:=0;
//Организуем цикл для работы
//с элементами строки.
repeat
//Выделяем группу символов до пробела.
    Word:=Copy(Str1,1,Pos(' ',Str1)-1);
//Преобразовываем строку в вещественное число,
//записываем его в i-ый элемент массива
    x[i]:=StrToFloat(Word);
//Вычисляем сумму элементов массива.
    sum:=sum+x[i];
    inc(i);
//Удаляем из исходной строки подстроку
//и следующий за ней пробел.
    Delete(Str1,1,Length(Word)+1);
//Выводим в поле ListBox1 подстроку.
    ListBox1.Items.Add(Word);
//Цикл выполняется до тех пор, пока
//длина строки не будет равна 0.
until Length(Str1) = 0;
//Преобразуем вещественное значение суммы
//в строковое и формируем строку для
//вывода в поле ListBox1.
Str2:='Summa =' + FloatToStr(sum);
//Преобразуем целочисленное значение
//количества элементов массива в строковое
//и формируем строку для вывода
//в поле ListBox1.
Str3:='Kolichestvo elementov =' + IntToStr(i);
```

```
//Выводим сумму элементов массива.
ListBox1.Items.Add(str2);
//Выводим количество элементов массива.
ListBox1.Items.Add(str3);
end;
initialization
    {$I unit1.lrs}
end.
```



*Рисунок 8.2: Результаты работы программы к задаче 8.1.*

## 8.2 Работа с записями

Иногда возникает необходимость объединить в одном типе несколько разных типов данных. В Free Pascal для этого применяется структурный тип данных — запись. Запись состоит из фиксированного числа компонентов, называемого полями записи. В общем случае описание записи выглядит так:

```
type
    имя_записи = record
        поле1: тип;
        поле2 : тип;
        ...
        полеN : тип;
```

```
end;
```

В программе переменная типа записи объявляется следующим образом:

```
var имя_переменной : имя_записи;
```

К каждому из полей записи можно получить доступ, используя составное имя. Для этого вначале пишется имя переменной, затем точка, затем имя поля.

**ЗАДАЧА 8.2.** Известны длины сторон треугольника  $a$ ,  $b$  и  $c$ . Вычислить площадь треугольника  $S$ .

Подобная задача была рассмотрена во второй главе. В задаче 2.1 подробно описана математическая постановка и алгоритм решения. Рассмотрим поставленную задачу в контексте данной главы. Создадим запись — Triangle (треугольник) с тремя полями —  $a$ ,  $b$  и  $c$  (стороны треугольника) и вычислим площадь треугольника по формуле Герона.

```
program Project1;
type
    //Запись — треугольник,
    //поля - стороны треугольника
    Triangle= record
        a, b, c: real;
    end;
//Объявление переменной типа Triangle.
var x: Triangle;
    p, s:real;
begin
    write('a=');
    readln(x.a);    //Ввод значения в поле a.
    write('b=');
    readln(x.b);    //Ввод значения в поле b.
    write('c=');
    readln(x.c);    //Ввод значения в поле c.
    //Вычисление полупериметра
    p:=(x.a+x.b+x.c)/2;
//Проверка существования треугольника -
//подкоренное выражение для формулы Герона
//должно быть положительным.
    if p*(p-x.a)*(p-x.b)*(p-x.c)>0 then
```



```

begin
    //Вычисление и вывод площади.
    s:=sqrt(p*(p-x.a)*(p-x.b)*(p-x.c));
    writeln('Площадь S=', s:7:2);
end
else
    writeln('Треугольник с заданными
            сторонами не существует.');
```

end.

Элементами записей могут быть как простые, так и структурированные типы. Никаких ограничений на вложение одной структуры в другую не существует.

Например, создадим запись *Student*, которая будет состоять из полей: фамилия, имя, группа, оценки по пяти дисциплинам и адрес. В свою очередь поле «Адрес» также сделаем записью, состоящей из полей: город, улица, дом, квартира.

```

type
    adress = record          //Запись «Адрес».
                                //поля:
        city,                //город,
        street: string;      //улица,
        house,               //дом,
        apartment: integer;  //квартира.
    end;
    student = record         //Запись «Студент».
                                //поля:
        surname,             //фамилия,
        name: string;        //имя,
        group: string;       //группа,
                                //оценки,
        estimation: array [1..5] of integer;
        residence: adress;    //адрес.
    end;
```

После объявления такой записи обращение к полям осуществляется так:

```

var Ivanov: student;
    x: array [1..100] of student;
begin
```

```
Ivanov.group:='Ф08';  
Ivanov.residence.city:='Киев';  
//у первого студента 5-я оценка =3  
x[1].estimation[5]:=3;
```

...

С использованием ключевого слова `With` к полям записи можно обращаться без указания префикса каждый раз с названием поля:

```
with <переменная> do <оператор>
```

Например:

```
with stud do  
begin  
  with residence do  
  begin  
    city:='Донецк';  
    street :='Артема';  
    house:=145;  
    apartment:=31;  
  end;  
  surname:='Иванов';  
  name:='Андрей';  
  birthday :='01.11.1990';  
  group :='Ф07';  
  estimation[1]:=3; estimation[2]:=5;  
  estimation[3]:=4;  
  estimation[4]:=3; estimation[5]:=5;  
end;
```

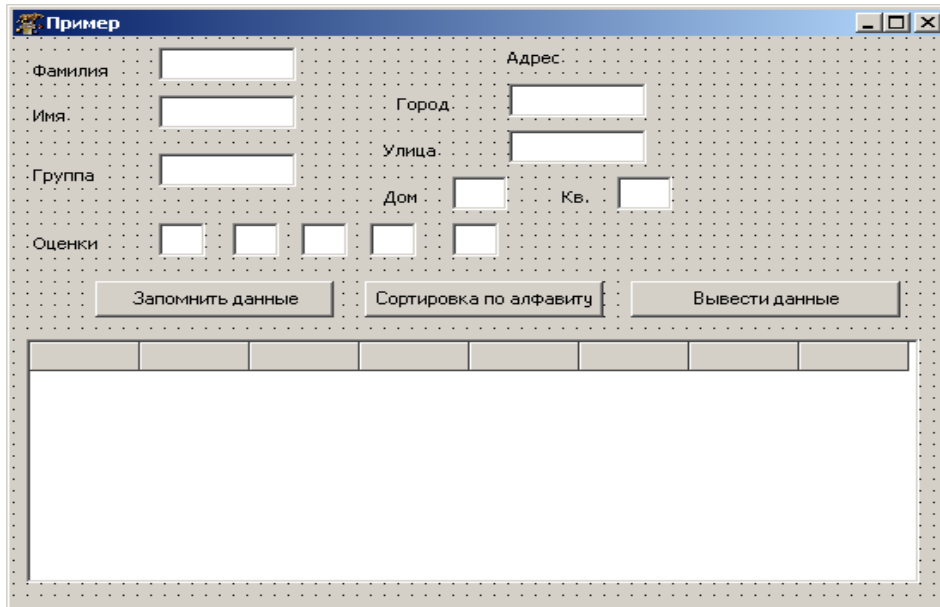
**ЗАДАЧА 8.3.** Создать базу данных «Сведения о студентах». В программе предусмотреть расчет среднего балла студента, сортировку по алфавиту, вывод результатов в диалоговое окно и в текстовый файл.

Создадим новый проект. На форму (рис. 8.3) поместим двенадцать объектов типа `TEdit` для ввода исходных данных, объект типа `StringGrid` (`ColCount=8`, `RowCount=1`) для вывода результатов и три кнопки типа `Tbutton`:

- `Button1` — для чтения данных из полей ввода и их последующей очистки;

- Button2 – для сортировки данных по алфавиту;
- Button3 – для вывода результатов в таблицу StringGrid1 и записи их в текстовый файл.

Текст программы с комментариями приведен далее.



*Рисунок 8.3: Пример формы для задачи 8.3*

```

unit Unit1;
{$mode objfpc}{$H+}
interface
uses
  Classes, SysUtils, LResources, Forms,
  Controls, Graphics, Dialogs, StdCtrls, Grids;
type
  { TForm1 }
  TForm1 = class(TForm)
    Button1: TButton;
    Button2: TButton;
    Button3: TButton;
    Edit1: TEdit;
    Edit10: TEdit;
    Edit11: TEdit;
    Edit12: TEdit;
    Edit2: TEdit;
    Edit3: TEdit;
    Edit4: TEdit;
    Edit5: TEdit;
  end;

```

```
    Edit6: TEdit;
    Edit7: TEdit;
    Edit8: TEdit;
    Edit9: TEdit;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Label8: TLabel;
    Label9: TLabel;
    StringGrid1: TStringGrid;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
private
    { private declarations }
public
    { public declarations }
end;
//Описание записи «Адрес студента».
address = record
    city, street : string;    //город, улица,
    house, apartment: integer;//дом, квартира.
end;
//Описание записи «Сведения о студенте».
student = record
    surname , name : string; //фамилия, имя,
    group : string;         //группа,
                             //оценки,
    estimation: array [1..5] of integer;
    residence : address;    //адрес,
    s_ball: real;           //средний балл.
end;
var
```

```
Form1: TForm1;
//Массив переменных типа student.
x: array [0..100] of student;
//Локальная переменная для подсчета
//количества студентов.
i: integer;
implementation
{ TForm1 }
//Процедура инициализации формы.
procedure TForm1.FormCreate(Sender: TObject);
begin
    i:=0;//количество студентов равно 0.
end;
//Процедура работы кнопки «Запомнить данные».
procedure TForm1.Button1Click(Sender: TObject);
var sum, j:integer;
begin
    //Чтение данных из полей ввода.
    x[i].surname:= Edit1.Text;
    x[i].name:= Edit2.Text;
    x[i].group:= Edit3.Text;
    x[i].residence.city:=Edit5.Text ;
    x[i].residence.street:= Edit6.Text;
    x[i].residence.house:=
        StrToInt(Edit7.Text);
    x[i].residence.apartment:=
        StrToInt(Edit8.Text);
    x[i].estimation[1]:=strToInt(Edit4.Text);
    x[i].estimation[2]:=strToInt(Edit9.Text);
    x[i].estimation[3]:=strToInt(Edit10.Text);
    x[i].estimation[4]:=strToInt(Edit11.Text);
    x[i].estimation[5]:=
        strToInt(Edit12.Text) ;
    //Вычисление среднего балла студента.
    sum:=0;
    for j:=1 to 5 do
        sum:=sum+x[i].estimation[j];
    x[i].s_ball:=sum/5; inc(i);
```

```
//Очищение полей ввода для следующих данных.
  Edit1.Text:= ''; Edit2.Text:= '';
  Edit3.Text:= ''; Edit4.Text:= '';
  Edit5.Text:= ''; Edit6.Text:= '';
  Edit7.Text:= ''; Edit8.Text:= '';
  Edit9.Text:= ''; Edit10.Text:= '';
  Edit11.Text:= ''; Edit12.Text:= '';
end;
//Процедура работы кнопки «Вывести данные».
procedure TForm1.Button2Click(Sender: TObject);
var f:textfile; j: integer; s: string;
begin
  //Количество строк в таблице StringGrid
  //на один больше, чем количество студентов,
  //одна строка для шапки таблицы.
  StringGrid1.RowCount:= i+1;
  //Вывод шапки таблицы.
  StringGrid1.Cells[1,0]:='Фамилия' ;
  StringGrid1.Cells[2,0]:='Имя' ;
  StringGrid1.Cells[3,0]:='Группа' ;
  StringGrid1.Cells[4,0]:='Город' ;
  StringGrid1.Cells[5,0]:='Улица' ;
  StringGrid1.Cells[6,0]:='Дом/кв.' ;
  StringGrid1.Cells[7,0]:='Средний балл' ;
  //Вывод сведений о студентах в j-ю строку.
  for j:=1 to i do
  begin
    StringGrid1.Cells[1,j]:=
      x[j-1].surname ;
    StringGrid1.Cells[2,j]:=x[j-1].name;
    StringGrid1.Cells[3,j]:=x[j-1].group;
    StringGrid1.Cells[4,j]:=
      x[j-1].residence.city;
    StringGrid1.Cells[5,j]:=
      x[j-1].residence.street;
    s:=inttostr(x[j-1].residence.house)+
      '/' +inttostr(x[j-1].residence.apartment);
    StringGrid1.Cells[6,j]:=s ;
```

```
        StringGrid1.Cells[7,j]:=
            floattostr(x[j-1].s_ball);
    end
    //Вывод результатов в текстовый файл.
    assignfile(f,'g:\student.txt');
    rewrite(f);
    for j:=1 to i do
    begin
        writeln(f,x[j-1].surname:20,
                x[j-1].name:15,
                x[j-1].residence.city:15,', ',
                x[j-1].residence.street:15,
                x[j-1].residence.house:4,'/',
                x[j-1].residence.apartment,
                ' Sr_ball=',x[j-1].s_ball:4:1);
    end;
    closefile(f);
end;
//Процедура работы кнопки
//«Сортировать по алфавиту».
procedure TForm1.Button3Click(Sender: TObject);
var j, k :integer;
    temp: student; //Переменная для сортировки.
    f:textfile;
Begin
//Сортировка по полю surname методом пузырька.
    for j:= 0 to i-1 do
        for k:=j+1 to i-1 do
            if x[j].surname > x[k].surname then
            begin
                //Строки меняются местами.
                temp:=x[j]; x[j]:=x[k];x[k]:=temp;
            end;
        end;
    end;
initialization
    {$I unit1.lrs}
end.
```

Результаты работы программы представлены на рис. 8.5 — 8.7.

The screenshot shows a window titled "Пример" with a form for entering student data. The fields are as follows:

- Фамилия: Иванов
- Имя: Иван
- Группа: Ф056
- Оценки: 5, 4, 4, 4, 4
- Адрес:
  - Город: Донецк
  - Улица: Артема
  - Дом: 132
  - Кв.: 145

Below the form are three buttons: "Запомнить данные", "Сортировка по алфавиту", and "Вывести данные". At the bottom, there is a table with 8 columns and 1 row, which is currently empty.

*Рисунок 8.5: Окно формы ввода сведений о студенте*

The screenshot shows the same window "Пример" but now displaying a list of student data in a table. The input fields are empty. The table has 8 columns: "Фамилия", "Имя", "Группа", "Город", "Улица", "Дом/кв.", and "Средний балл". The "Вывести данные" button is highlighted with a dotted border.

	Фамилия	Имя	Группа	Город	Улица	Дом/кв.	Средний балл
	Иванов	Иван	Ф056	Донецк	Артема	132/145	4,2
	Петрова	Ольга	Ф056	Донецк	Ткаченко	147/2	3,8
	Громов	Илья	МП05	Горловка	Ленина	17/8	3,6
	Смирнова	Ольга	МП06	Горловка	Петрова	1/12	4,2
	Киреев	Андрей	МП05	Донецк	Свободы	21/67	5
	Чернова	Светлана	УА05а	Донецк	Щетинина	1/123	4,4

*Рисунок 8.6: Окно формы вывода сведений о студентах*



При запуске программы и вводе сведений окно формы выглядит так, как показано на рис 8.5. Когда запись введена, следует щелкнуть по кнопке «Запомнить данные», при этом поля ввода очищаются для ввода следующей записи. После щелчка по кнопке «Вывести данные» таблица заполняется введенными сведениями (рис.8.6). После щелчка по кнопке «Сортировать по алфавиту» нужно повторно щелкнуть по кнопке «Вывести данные», чтобы увидеть отсортированный список (рис.8.7).

The screenshot shows a window titled "Пример" with a form for data entry and a table of sorted data. The form has fields for "Фамилия", "Имя", "Группа", "Оценки", "Адрес", "Город", "Улица", "Дом", and "Кв.". Below the form are three buttons: "Запомнить данные", "Сортировка по алфавиту", and "Вывести данные". The table below the buttons contains the following data:

	Фамилия	Имя	Группа	Город	Улица	Дом/кв.	Средний
	Антонова	Ирина	МП05	Донецк	Светлая	2/21	4,6
	Борисова	Татьяна	Ф056	Макеевка	Транспортн	12/6	5
	Громов	Илья	МП05	Горловка	Ленина	17/8	3,6
	Иванов	Иван	Ф056	Донецк	Артема	132/145	4,2
	Киреев	Андрей	МП05	Донецк	Свободы	21/67	5
	Петрова	Ольга	Ф056	Донецк	Ткаченко	147/2	3,8

Рисунок 8.7: Окно формы после сортировки записей по полю «Фамилия»

### 8.3 Задачи для самостоятельного решения по теме «Строки»

Дана строка текста. Выполнить с ней следующие действия:

1. Посчитать количество запятых в строке.
2. Заменить в строке все цифры на пробел. Вывести количество замен.
3. Посчитать в строке количество цифр.
4. Удалить из строки все запятые.
5. Посчитать в строке количество слов.
6. Удалить из строки все слова, начинающиеся на букву «о».
7. После каждого пробела вставить символ \*.

8. Найти в строке самое длинное слово.
9. Перед каждым пробелом вставить пробел и символ +.
10. Посчитать сумму всех чисел, которые встречаются в строке.
11. Посчитать в строке количество слов, начинающихся на «Ав».
12. Заменить в строке двойные пробелы на одинарный пробел. Вывести количество замен.
13. Вставить после каждого слова запятую.
14. После каждого слова вставить символ «;».
15. Посчитать в строке количество символов «:» и «;».
16. Удалить из строки все цифры.
17. Посчитать в строке количество слов, заканчивающихся символами «ая».
18. Найти в строке самое короткое слово.
19. Вставить после каждого слова, заканчивающегося на букву «о», слово «Ого».
20. Удалить из строки все слова, состоящие из пяти букв.
21. Найти в строке количество слов, начинающихся на букву «а» и заканчивающихся буквой «т».
22. Удалить из строки второе, третье и пятое слова.
23. Перед каждой цифрой вставить символ №.
24. Посчитать в строке количество гласных букв.
25. Удалить из строки все слова, начинающиеся и заканчивающиеся на «о».

#### **8.4 Задачи для самостоятельного решения по теме «Записи»**

1. Создать структуру с данными по таблице 8.1:

Таблица 8.1. Прирост населения в городах

Город	Прирост населения, тыс. чел.				
	1999	2000	2001	2002	2003
Макеевка	2,5	1,3	-0,2	-0,1	0,6
...					

Добавить и вычислить в структуре поле «Средний прирост». Определить количество городов с отрицательным приростом в 2003 году. Упорядочить записи по возрастанию среднего прироста.

2. Создать структуру с данными по таблице 8.1. Добавить и вычислить в структуре поле «Минимальный прирост». Определить ко-

личество городов с приростом в 2003 году более 2 тыс. чел. Выполнить сортировку записей по полю «Город». Названия городов упорядочить по алфавиту.

3. Создать структуру с данными по таблице 8.2:

Таблица 8.2. Сведения о товаре

Название	Фабрика	Цена	Дата выпуска	Количество
Паровозик	Игрушка	125,00	01.02.2007	
...				

Добавить и вычислить в структуре поле «Цена со скидкой», вводя процент скидки с формы. Найти общее количество игрушек с фабрики «Игрушка». Упорядочить записи по убыванию поля «Цена».

4. Создать структуру с данными по таблице 8.2. Добавить и вычислить в структуре поле «Сумма продажи». Найти количество названий игрушек, у которых цена меньше общей средней цены всех игрушек. Упорядочить записи по названию игрушек.

5. Создать структуру с данными по таблице 8.5:

Таблица 8.5. Сведения о школьнике

Фамилия	Имя	Дата рождения	Школа	Класс
Сергеев	Сергей	05.05.1994	112	9-А
...				

Добавить и вычислить в структуре поле «Возраст», вводя текущую дату с формы. Определить количество школьников с именем Сергей. Упорядочить записи по номеру школы.

6. Создать структуру с данными по таблице 8.5. Добавить и вычислить в структуре поле «Год обучения», убрав из названия класса букву. Найти количество учеников 9-х классов. Выполнить сортировку записей по полю «Фамилия». Фамилии упорядочить по алфавиту.

7. Создать структуру с данными по таблице 8.6:

Таблица 8.6. Сведения о продажах

Принтер	Количество, шт.			Цена, \$
	Январь	Февраль	Март	
Samsung CLP-310	25	20	26	550
...				

Добавить и вычислить в структуре поле «Выручка». Найти среднюю цену принтеров. Упорядочить записи по возрастанию поля «Цена».

8. Создать структуру с данными по таблице 8.6. Добавить и вычислить в структуре поле «Среднее количество». Найти среднее количество принтеров в каждом месяце. Упорядочить записи по названию принтера.

9. Создать структуру с данными по таблице 8.6. Добавить и вычислить в структуре поле «Общее количество». Найти общее количество проданных принтеров в каждом месяце. Упорядочить записи по возрастанию поля «Общее количество».

10. Создать структуру с данными по таблице 8.6. Добавить и вычислить в структуре поле «Цена со скидкой», вводя процент скидки с формы. Найти количество принтеров с ценой более 500\$. Упорядочить записи по убыванию цены.

11. Создать структуру с данными по таблице 8.7:

Таблица 8.7. Сведения о сотруднике

ФИО	Дата рождения	Должность	Стаж	Оклад
Сергеев С. И.	12.03.1966	Менеджер	2	1250
...				

Добавить и вычислить в структуре поле «Премия», рассчитав ее по следующему принципу: 20% от оклада, если стаж более 10 лет, в противном случае 10%. Найти количество сотрудников со стажем более 10 лет. Упорядочить записи по должности.

12. Создать структуру с данными по таблице 8.8. Добавить и вычислить в структуре поле «Возраст», текущую дату вводить с формы. Найти средний оклад всех сотрудников. Упорядочить записи по ФИО.

13. Создать структуру с данными по таблице 8.8. Добавить и вычислить в структуре поле «Возраст», текущую дату вводить с формы. Определить количество молодых специалистов (моложе 25 лет). Упорядочить записи по возрастанию оклада.

14. Создать структуру с данными по таблице 8.8:

Таблица 8.8. Сведения о продажах путевок

Место отдыха	Количество, шт.			Цена, \$
	Июль	Август	Сентябрь	
Геленджик	255	203	198	510
...				

Добавить и вычислить в структуре поле «Среднее количество». Найти общее количество путевок в каждом месяце. Упорядочить за-

писи по месту отдыха.

15. Создать структуру с данными по таблице 8.8. Добавить и вычислить в структуре поле «Доход от путевок». Найти среднюю цену путевки. Упорядочить записи по возрастанию цены.

16. Создать структуру с данными по таблице 8.9:

Таблица 8.9. Сведения о сотруднике

<b>ФИО</b>	<b>Дата рождения</b>	<b>Должность</b>	<b>Пол</b>	<b>Оклад</b>
Сергеев С. И.	12.03.1966	Менеджер	Муж.	1250
...				

Добавить и вычислить в структуре поле «Зарплата», рассчитав ее по следующему принципу: к окладу добавить премию в размере 15% от оклада. Упорядочить записи по ФИО.

17. Создать структуру с данными по таблице 8.9. Добавить и вычислить в структуре поле «Возраст», текущую дату вводить с формы. Определить количество мужчин и женщин. Упорядочить записи по должности.

18. Создать структуру с данными по таблице 8.10:

Таблица 8.10. Сведения о научных сотрудниках

<b>Фамилия</b>	<b>Инициалы</b>	<b>Ученая степень</b>	<b>Год рождения</b>	<b>Количество статей</b>
Сергеев	С. А.	Доцент	1971	25
...				

Добавить и вычислить в структуре поле «Активность» по следующему принципу: если количество статей более 10, то в поле записать пробел, в противном случае – фразу «Работать лучше». Упорядочить записи по фамилии.

19. Создать структуру с данными по таблице 8.10. Удалить сотрудника с фамилией, которая вводится с формы. Определить количество доцентов. Упорядочить записи по должности.

20. Создать структуру с данными по таблице 8.11:

Таблица 8.11. Сведения о тираже книг

<b>Название</b>	<b>Автор</b>	<b>Издательство</b>	<b>Год издания</b>	<b>Цена, \$</b>	<b>Тираж</b>
Вий	Гоголь Н. В.	Правда	1971	6,5	15000
...					

Добавить и вычислить в структуре поле «Стоимость тиража». Найти общий тираж книг 2005 года. Упорядочить записи по автору.

21. Создать структуру с данными по таблице 8.11. Удалить все записи книг тиража 2000 года. Найти среднюю цену книг типографии «Правда». Упорядочить записи по году издания.

22. Создать структуру с данными по таблице 8.12:

Таблица 8.12. Сведения о телефонных звонках

ФИО абонента	Номер	Дата звонка	Город	Стоимость 1 минуты разговора	Количество минут
Моль Р. Ю.	956-25-78	12.05.2003	Казань	3,65	2
...					

Добавить и вычислить в структуре поле «Стоимость звонка». Найти общую стоимость звонков в город, вводимый по запросу. Упорядочить записи по ФИО абонента.

23. Создать структуру с данными по таблице 8.12. Удалить все записи звонков с номерами, начинающимися с цифры 3. Упорядочить записи по названию города.

24. Создать структуру с данными по таблице 8.14:

Таблица 8.14. Сведения о приборах

Название прибора	Шифр прибора	Дата выпуска	Количество	Гарантийный срок, мес.
Микроскоп	M12-08	12.06.2006	200	24
...				

Добавить и вычислить в структуре поле «Гарантийное обслуживание» по следующему принципу: записать фразу «1 год», если гарантийный срок более 3 лет, в противном случае – фразу «нет обслуживания». Найти общее количество всех приборов. Упорядочить записи по названию прибора.

25. Создать структуру с данными по таблице 8.14. Удалить все записи с гарантийным сроком менее 6 месяцев. Упорядочить записи по дате выпуска.