

Практическое занятие

Тема: РАЗРАБОТКА ПРИЛОЖЕНИЙ С ИСПОЛЬЗОВАНИЕМ ТИПИЗИРОВАННЫХ ФАЙЛОВ (продолжение)

Типизированные файлы не могут содержать в себе строки переменной длины. Для того, что бы записывать в файл строки переменной длины существуют текстовые файлы. Как и с любыми файлами, работа с текстовыми файлами осуществляется через файловую переменную. Для объявления файловой переменной текстового файла используется слово Text:

```
Var f:text;
```

Аналогично типизированным файлам, необходимо связать файловую переменную с файлом с помощью процедуры Assign(f). После работы с файлом его необходимо закрыть процедурой close(f).

В отличие от типизированных файлов чтение и запись в текстовый файл происходит целиком. Нельзя записать данные в какое-либо место файла или считать данные из какого либо места. Поэтому для текстовых файлов процедура rewrite(f) создаёт новый файл, стирая старый, только для записи в него. Процедура reset(f) открывает текстовый файл только для чтения.

Чтение и запись данных в текстовый файл осуществляется с помощью, до боли знакомых нам, процедур Write, Writeln, Read, Readln. Так же первым параметром идёт файловая переменная, затем список через запятую констант или переменных (параметров).

Прежде чем вести речь о том, как пользоваться этими процедурами, необходимо узнать структуру тестового файла. Текстовый файл состоит из строк переменной длины. В конце каждой строки находится специальное сочетание символов с порядковыми номерами 13 и 10. Например, если мы добавим данное сочетание в середину строки, то при выводе на экран в этом месте строка разорвётся и текст начнётся с новой строки:

```
var s:string;  
begin  
  s:='Привет'+chr(13)+chr(10)+'всем!';  
  write(s);
```

end.

Результат выполнения программы :

Привет
всем!

Наглядно структуру текстового файла можно продемонстрировать следующим образом:

```
.....#13#10
.....#13#10
.....#13#10
...#13#10
Eof
```

В данном примере: Eof – признак конца файла, который присутствует в любом файле. Символ «#» – говорит о том, что на этом месте находится символ с порядковым номером, который идёт далее.

Теперь можно говорить о процедурах чтения и записи.

Процедура Write делает запись в текстовый файл, но не вставляет комбинацию символов #13#10. Следовательно, при записи данных с помощью только этой процедуры в текстовом файле окажется только одна строка.

Процедура Writeln делает запись в текстовый файл и вставляет комбинацию символов #13#10. Следовательно, после вызова данной процедуры вставляется признак конца строки, и следующая запись будет происходить в новую строку.

Процедура Read считывает данные из файла, но не делает переход на следующую строку. Что бы считать данные и перейти на следующую строку необходимо использовать процедуру Readln.

Пример:

```
var f:text;
    s:string;
begin
  assign(f, 'c:\text.txt');
  rewrite(f);
  write (f, 'Привет'); //Записываем строку в файл
  Writeln(f, ' всем!'); //Записываем строку в файл
и переходим на следующую
```

```

        writeln(f, 'Ура, ура!'); //Записываем строку в
        файл и переходим на
                                   //следующую
        close(f);
        reset(f);
        read(f, s); //считываем строку
        writeln(s);
        readln(f); //переходим на следующую строку
        readln(f, s); //считываем строку и переходим на
        следующую
        writeln(s);
end.

```

Результат выполнения программы:

Привет всем!

Ура, ура!

Данными процедурам можно записывать в текстовый файл не только строки, но и символы, т.е. данные типа `char`. Ни каких тонкостей и нюансов при этом нет, происходит это так же как и со строками.

Так же вместо строк и символов можно записывать числа целые и вещественные, при этом они автоматически будут форматироваться в строку для записи и обратно для чтения. Только будьте внимательны и следите за соответствием данных:

```

var f:text;
      i:integer;
      r:real;
begin
    assign(f, 'c:\text.txt');
    rewrite(f);
    writeln (f, 12);
    Writeln(f, 3.14);
    close(f);
    reset(f);
    readln(f, i);
    writeln(i);
    readln(f, r);
    writeln(r);
end.

```

Результат выполнения программы:

12
3.14

Так же в текстовый файл можно вписать множество, однако считать его можно будет только как строку:

```
const m:set of integer = [3..4, 6, 55..57];
var f:text;
    s:string;
begin
    assign(f, 'c:\text.txt');
    rewrite(f);
    writeln (f,m);
    close(f);
    reset(f);
    readln(f,s);
    writeln(s);
    close(f);
end.
```

Результат выполнения программы:

[3, 4, 6, 55, 56, 57]

Для того, что бы целиком считать текстовый файл удобно пользоваться уже знакомой нам функцией Eof:

```
var f:text;
    i:integer;
    s:string;
begin
    assign(f, 'c:\text.txt');
    rewrite(f);
    //Создаем файл случайной длины со случайными
    значениями
    For i:=1 to random(50) do
        begin
            write (f,random(1000):4, ' ');
            if i mod 10 = 0 then writeln(f);
        end;
    close(f);
```

```
//Выводим содержимое файла на экран
reset(f);
while not eof(f) do
begin
  readln(f,s);
  writeln(s);
end;
close(f);
end.
```

Результат выполнения программы:

420	562	614	705	959	901	757	734	115	238
389	307	599	625	981	906	682	865	597	934
127	300	324	747	806	814	559	175	837	759
478	490	827	424	346	819	285	726		

И ещё одна специфическая процедура для работы с текстовыми файлами: **Append(f)** – открывает текстовый файл для дополнения, т.е. после открытия файла данной процедурой можно вписывать в него данные, при этом старое содержимое стёрто не будет, а запись будет происходить с того места, где файл был закончен. Поэтому если хотите начать запись с новой строки, то возможно понадобится добавить признак конца строки:

```
var f:text;
    i:integer;
    s:string;
begin
  assign(f, 'c:\text.txt');
  //Открываем файл для дополнения
  Append(f);
  writeln(f, chr(13), chr(10), 'Всё это случайные
числа');
  close(f);
  //Выводим содержимое файла на экран
  reset(f);
  while not eof(f) do
begin
  readln(f,s);
  writeln(s);
```

```
    end;  
    close(f);  
end.
```

Результат выполнения программы:

```
420  562  614  705  959  901  757  734  115  238  
389  307  599  625  981  906  682  865  597  934  
127  300  324  747  806  814  559  175  837  759  
478  490  827  424  346  819  285  726
```

Всё это случайные числа

В данном параграфе мы изучили типизированные и текстовые файлы и научились ими пользоваться.

Задачи.

1. Необходимо создать массив случайных целых чисел, записать его в файл, затем вывести его на экран из файла.

2. Создать с помощью программы текстовый файл (не менее 2-х строк). Добавить в него ещё одну строку с помощью процедуры Append. Вывести файл на экран.

3. Создать программу поиска слова в текстовом файле. Слово для поиска пользователь вводит с клавиатуры. Примечание: в предыдущем параграфе в 3-ей задаче необходимо было вывести в отдельную функцию процесс поиска слова в строке. При создании программы можно использовать данную функцию.

4. Написать функцию определения количества строк в текстовом файле. Написать функцию, считывающую определённую строку по номеру из текстового файла. Написать программу демонстрирующую работу данных подпрограмм.

5. Имеется текстовый файл со списком слов. Каждое слово находится в новой строчке. Необходимо отсортировать слова по алфавиту. Регистр букв не должен иметь значения. Примечание: для примера создать список не менее 20 слов.

РЕШЕНИЕ ЗАДАЧ

ПРИМЕР 1

```
Program Ispolzovanie_Massiva;  
//Программа демонстрирует работу с  
типизированными файлами
```

```

const imya_file = 'c:\massiv.int';
//Константа содержит название файла
    max = 30; // размер массива
var massiv: array [1..max] of integer;
    f: file of integer;
    i: integer;
begin
    //Заполняем массив
    for i:=1 to max do
        massiv[i]:=random(1,1000);
    //Записываем в файл
    assign(f,imya_file);
    rewrite(f);
    for i:=1 to max do
        write(f,massiv[i]);
    close(f);
    //Читаем массив из файла
    assign(f,imya_file);
    reset(f);
    for i:=1 to max do
        read(f,massiv[i]);
    close(f);
    //Выводим массив на экран
    for i:=1 to max do
        begin
            write(massiv[i]:5);
            if (i mod 10)=0 then writeln;
        //Создаем новую строку
        end;
    end.

```

Результат выполнения программы:

920	288	950	972	812	413	244	579	112	357
513	855	943	299	292	694	51	976	309	973
457	561	45	855	308	284	511	119	146	809

ПРИМЕР 2.

```

Program Sozdanie_textovogo_fayla;
//Программа создаёт текстовый файл, дополняет
//его и выводит его на экран
Var f: text;

```

```

        s:string;
Begin
    //Создаём текстовый файл
    assign(f, 'c:\Text.txt');
    Rewrite(f);
    writeln(f,
        'В классе 20 учеников. Все ученики учатся
по-разному. Одни ученики '+
        'являются отличниками или хорошистами,
другие ученики троечники, учеников'+
        ' двоечников нет. ');
    Writeln(f, 'В другом классе 22 ученика.
Двоечников учеников тоже нет. ');
    close(f);
    //Добавляем в текстовый файл строку
    Append(f);
    Writeln(f, 'Всего в школе есть один ученик
двоечник. ');
    close(f);
    //Выводим текстовый файл на экран
    reset(f);
    while not eof(f) do
        begin
            readln(f, s);
            writeln(s);
        end;
    close(f);
end.

```

Результат выполнения программы:

В классе 20 учеников. Все ученики учатся по-разному. Одни ученики являются отличниками или хорошистами, другие ученики троечники, учеников двоечников нет.

В другом классе 22 ученика. Двоечников учеников тоже нет.

Всего в школе есть один ученик двоечник.

ПРИМЕР 3.

Program Poisk_slova_v_fayle;


```

    //Программа ищет слово в файле и выводит,
    сколько раз это слово встречается
    Const imya_fayla = 'c:\Text.txt'; //файл в
    котором будем искать
    Var f:text;      //файловая переменная
        sTemp,      //Переменная для хранения текущей
строки
        s          //Переменная для хранения слова
для поиска
        :string;
        kolTek,    //количество найденных слов в
текущей строке
        kol        //количество найденных слов в
файле
        :integer;

    //Функция находит количество слов s2, входящих в
строку s1
    Function Poisc(s1,s2:string):word;
    var i,s:word;
    begin
    s:=0;
    i:=1;
    while i<length(s1)do
        begin
            if (PosEx(s2,s1,i)<>0) then
                begin
                    s:=s+1;
                    i:=PosEx(s2,s1,i);
                end;
                i:=i+1;
            end;
        Poisc:=s;
    end;

    begin
        Writeln('Введите слово для поиска');
        readln(s);
        assign(f,imya_fayla);
        reset(f);

```

```

//=====
//ищем слово в файле
kol:=0;
while not eof(f) do
    begin
        readln(f,sTemp);
        kolTek:=poisc(sTemp,s);
        if kolTek <> 0 then kol:=kol+kolTek;
    end;
//=====
close(f);
Writeln;
Write('В текстовом файле "',imya_fayla,'"
слово "',s,'" ');
if kol<>0 then write('встречается ',kol,'
раз.')
    else write('невстречается.');
```

end.

Результат выполнения программы:

Введите слово для поиска

ученик

В текстовом файле "с:\Text.txt" слово "ученик" встречается 8 раз.

ПРИМЕР 4.

Program Textovie_fayli;

//Программа содержит функцию выдающую количесвто строк в текстовом файле

//и функцию выдающую строку из файла по номеру

var kol_strok_v_fayle:integer; //содержит количество строк в файле

 nomer_stroki:integer; //Номер строки для вывода

 sn:string; //строка для вывода по номеру

//Функция определения количества строк в текстовом файле imya_fayla

function kol_strok(imya_fayla:string):integer;

```

var f:text;
    kol:integer;
begin
    kol:=0;
    assign(f, imya_fayla);
    reset(f);
    while not eof(f) do
        begin
            inc(kol);
            readln(f);
        end;
    kol_strok:=kol;
    close(f);
end;

//Функция выдаёт строку из файла imya_fayla по
её номеру n.
function stroka_po_nomeru(imya_fayla:string;n:in
teger):string;
var f:text;
    i:integer;
    sTemp:string;
begin
    assign(f, imya_fayla);
    reset(f);
    for i:=1 to n do readln(f, sTemp);
    stroka_po_nomeru:=sTemp;
    close(f);
end;

begin
    //Определяем количесвто строк в файле
    kol_strok_v_fayle:=kol_strok('c:\Text.txt');
    writeln('В текстовом файле "c:\Text.txt"
содержится ',
    kol_strok_v_fayle, ' строк. ');
    //Выводим строку по её номеру
    writeln('Введите номер строки для вывода. ');
    readln(nomer_stroki);

```

```

If kol_strok_v_fayle < nomer_stroki then
    writeln('В данном текстовом файле строки с
    таким номером нет.')
else
    begin
        sn:=stroka_po_nomeru('c:\Text.txt',nomer_
stroki);
        write('Строка с номером ',nomer_stroki,
            ' содержит в себе следующее:
            ',sn, '"');
        end;
    end.

```

Результат выполнения программы:

В текстовом файле "c:\Text.txt" содержится 3
 строк.

Введите номер строки для вывода.

2

Строка с номером 2 содержит в себе следующее: "В
 другом классе 22 ученика. Двоечников учеников тоже
 нет."

ПРИМЕР 5

```

Program Sortirovka_slov;
    //Программа сортирует список слов в текстовом
    файле по алфавиту
    var f,fTemp:text;
        sTemp, //Строка для временного хранения
        s1,s2, //Предыдущая и следующая строки
соответственно
        s11,s22//Предыдущая и следующая строки
соответственно в нижнем регистре
        :string;
        flg:boolean; //Если слова уже отсортированы
то равна false
    //Процедура для вывода списка на экран
    Procedure vivod_spiska_iz_fayla;
    begin
        reset(f);
        while not eof(f) do

```

```

    begin
        readln(f, sTemp);
        write(sTemp);
        if eof(f) then write('.') else write(',
');
    end;
    Writeln;
    close(f);
end;

begin
    assign(f, 'c:\Список.txt');
    assign(fTemp, 'c:\Temp.txt');
    write('Имеем следующий список: ');
    vivod_spiska_iz_fayla;
    //Сортируем список
    flg:=true; //Список ещё не отсортирован
    while flg do
        begin
            reset(f);
            readln(f, s1);
            rewrite(fTemp);
            flg:=false; //Если список уже отсортирован,
то далее в True не установится
            While not eof(f) do
                begin
                    readln(f, s2);
                    //Переводим строки в нижний регистр
                    s11:=LowerCase(s1);
                    s22:=LowerCase(s2);
                    //Если в нижнем регистре предыдущая строка
больше следующей строки, то
                    //меняем строки местами и устанавливаем flg в
true
                    if s11>s22 then
                        begin
                            flg:=true;
                            sTemp:=s1;
                            s1:=s2;

```

```

        s2:=Stemp;
    end;
    writeln(fTemp,s1); //Записываем во
временный файл
    s1:=s2;
    end;
    writeln(fTemp,s2); //Дописываем последнюю
строку
    close(f);
    close(fTemp);
    //Заменяем исходный файл временным
    erase(f);
    rename(fTemp,'с:\Список.txt');
    end;
    writeln;
    write('После сортировки: ');
    vivod_spiska_iz_fayla;
end.

```