



Файлы данных в языке Pascal

Файл – представляет собой последовательность компонент, расположенных на внешнем носителе информации.

Любой файл имеет три характерные особенности:

- 1) у него есть имя;
- 2) содержит компоненты одного типа;
- 3) длина создаваемого файла никак не оговаривается при его объявлении ограничивается только емкостью устройства внешней памяти.

В языке Pascal различают 3 типа файлов данных:

1) Типизированный файл — последовательность элементов одного типа. Все компоненты файла имеют общее имя, а каждый с еще одним индексом. Организация всех элементов файла осуществляется по принципу «записи», т.е. каждый элемент файла занимает определенное место. При формировании файла по знаку конца файла элемент автоматически записывается в место, соответствующее типу типизированного файла. Могут быть: **логические типы данных и записи.**

Типы файлов данных

2) **Текстовый файл** — это совокупность символов, разделенных на строки переменной длины.

В конце каждой строки стоит признак конца строки **EOLN** (end of line). Доступ к каждой строке возможен только последовательно, начиная с первой.

Компонентами текстового файла могут быть данные следующих типов: **символьные, строки, вещественные и целочисленные.**

Типы файлов данных

3) Нетипизированный файл – файл без типа представляется как последовательность элементов произвольного типа, но оговоренного размера. Это значит, что в файл можно записать значение любой переменной, имеющей заданный размер, а при чтении из такого файла допускается произвольная интерпретация содержимого очередного элемента.

Отсутствие типа компонентов делает эти файлы совместимыми с любыми другими файлами и позволяет организовать высокий скоростной обмен данными между внешним запоминающим устройством (ВЗУ) и оперативной памятью (ОП). Относятся к файлам прямого доступа.

Чтение данных из файла (т.е. ввод данных в оперативную память машины) и *запись* данных в файл (т.е. вывод данных из оперативной памяти на внешнее запоминающее устройство) осуществляется через *файловые переменные*.

Объявление файловых переменных

а) для типизированных файлов

Var имя файловой переменной : **file of** тип
КОМПОНЕНТ;

Type имя типа = **file of** тип КОМПОНЕНТ;

Var имя файловой переменной : имя типа;

где **file, of** - служебные слова (файл, из)

Объявление файловых переменных

а) для текстовых файлов

```
Var имя файловой переменной : text;
```

```
Type имя типа = text;
```

```
Var имя файловой переменной : имя типа;
```

где **text** - служебное слово (текстовый)

Объявление файловых переменных

а) для нетипизированных файлов

```
Var имя файловой переменной : file;
```

```
Type имя типа = file;
```

```
Var имя файловой переменной : имя типа;
```

где **file**- служебное слово (файл)

После описания файловой переменной в ОЗУ выделяется место под значение данной переменной.

Над файловыми переменными **нельзя** **выполнять** операции присваивания, сравнения и т.п.

Их можно использовать только для выполнения операций над файлами: *чтение, записи, удаления, открытия, переименования и т.д.*

С каждым файлом связана своя файловая переменная.

При работе с любым типом файла необходимо выполнить следующие

5 операций:

- 1) объявление файловой переменной;
- 2) связь файловой переменной с конкретным файлом на физическом уровне;
- 3) инициализация файла – означает указать направление передачи данных (т.е. открыть файл для чтения, записи или добавления);
- 4) обработка данных ;
- 5) закрытие файла.

Стандартные процедуры для работы с файлами

Имя и параметры	Типы параметров	Действие
Assign(f,name)	f – файловая переменная name - string	связывает файловую переменную f с файлом с именем name
Reset(f)	f - файловая переменная	Открывает файл, связанный с файловой переменной f . Текстовые - только на чтение, типизированные - на чтение и запись

Стандартные процедуры для работы с файлами

Имя и параметры	Типы параметров	Действие
Rewrite(f)	f – файловая переменная	создает и открывает новый файл, связанный с файловой переменной f . Если файл с указанным именем уже существует, то он удаляется, и вместо него создается новый файл. <i>Текстовые</i> -только на запись, <i>типизированные</i> - на чтение и запись

Стандартные процедуры для работы с файлами

Имя и параметры	Типы параметров	Действие
Append(f)	f - типа Text	Открывает <i>текстовый</i> файл на запись для добавления. Файловый указатель устанавливается в конец файла.
Close(f)	f - файловая переменная	Закрывает файл

Стандартные процедуры для работы с файлами

Имя и параметры	Типы параметров	Действие
Read(f,a,b,...)	f - файловая переменная, a,b - переменные простого типа, string или указатели	Считывает значения из файла f в переменные a, b ... Для <i>типизированного</i> файла переменные a, b ... - <u>базового типа</u> , а их значения считываются из файла в двоичном виде. Для <i>текстового</i> файла, переменные a, b ... могут иметь различные типы, а их значения храниться в текстовом виде

Стандартные процедуры для работы с файлами

Имя и параметры	Типы параметров	Действие
Write(f,a,b,...)	f - переменная файлового типа, a, b - выражения простого типа, string или указатели	записывает значения $a, b \dots$ в файл f . Для <i>типизированного</i> -типы значений $a, b \dots$ – базовые. Для <i>текстового</i> -значения $a, b \dots$ выводятся в текстовом виде, при этом могут быть использованы форматы вывода

Стандартные процедуры для работы с файлами

Имя и параметры	Типы параметров	Действие
Readln(f,a,b,..)	f - переменная типа Text, a,b - переменные простого типа, типа string или указатели	Считывает значения из <i>текстового</i> файла f в переменные a, b ... , после чего пропускает символы до конца строки. Вызов readln(f) просто пропускает символы до конца строки

Стандартные процедуры для работы с файлами

Имя и параметры	Типы параметров	Действие
<code>Writeln(f,a,b,..)</code>	f - переменная типа Text, a,b - выражения простого типа, типа string или указатели	Записывает значения a, b ... в текстовый файл f , после чего записывает в него символ конца строки. Вызов writeln(f) просто записывает в файл символ конца строки

Стандартные процедуры для работы с файлами

Имя и параметры	Типы параметров	Действие
Rename(<i>f</i> , <i>name</i>)	<i>f</i> - файловая переменная, <i>name</i> - string	Переименовывает файл, связанный с файловой переменной <i>f</i> . Файл должен быть закрыт
Erase(<i>f</i>)	<i>f</i> - переменная файлового типа	Удаляет файл, связанный с файловой переменной <i>f</i> . Файл должен быть закрыт

Стандартные функции для работы с файлами

Имя и параметры	Типы параметров	Действие
Seek(f,n)	f - переменная типа file	Перемещает файловый указатель в <i>типизированном</i> файле на n -ый элемент (нумерация начинается с нуля)
Truncate(f)	f - переменная типа file	Удаляет все элементы <i>типизированного</i> файла с текущей позиции файлового указателя до конца файла

Стандартные функции для работы с файлами

Имя и параметры	Типы параметров	Тип результата	Действие
FileExists (<i>name</i>)	<i>name</i> - string	boolean	Возвращает True , если на диске имеется файл с именем <i>name</i> , в противном случае возвращает False
CanCreateFile (<i>name</i>)	<i>name</i> - string	boolean	Возвращает True , если можно создать файл с именем <i>name</i> , в противном случае

Стандартные функции для работы с файлами

Имя и параметры	Типы параметров	Тип результата	Действие
Eof(f)	<i>f</i> - переменная файлового типа	boolean	возвращает True , если файловый указатель стоит на конце файла, и False в противном случае
Eoln(f)	<i>f</i> - переменная типа Text	boolean	Возвращает True , если файловый указатель стоит на конце строки, и False в противном случае

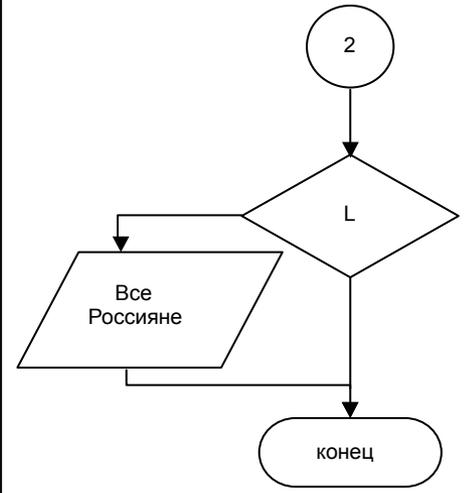
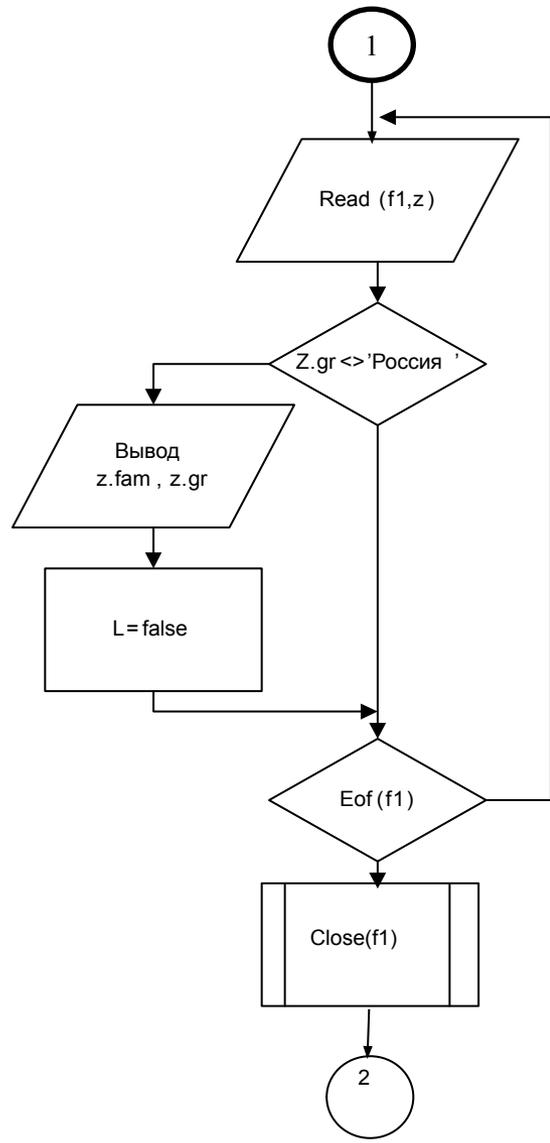
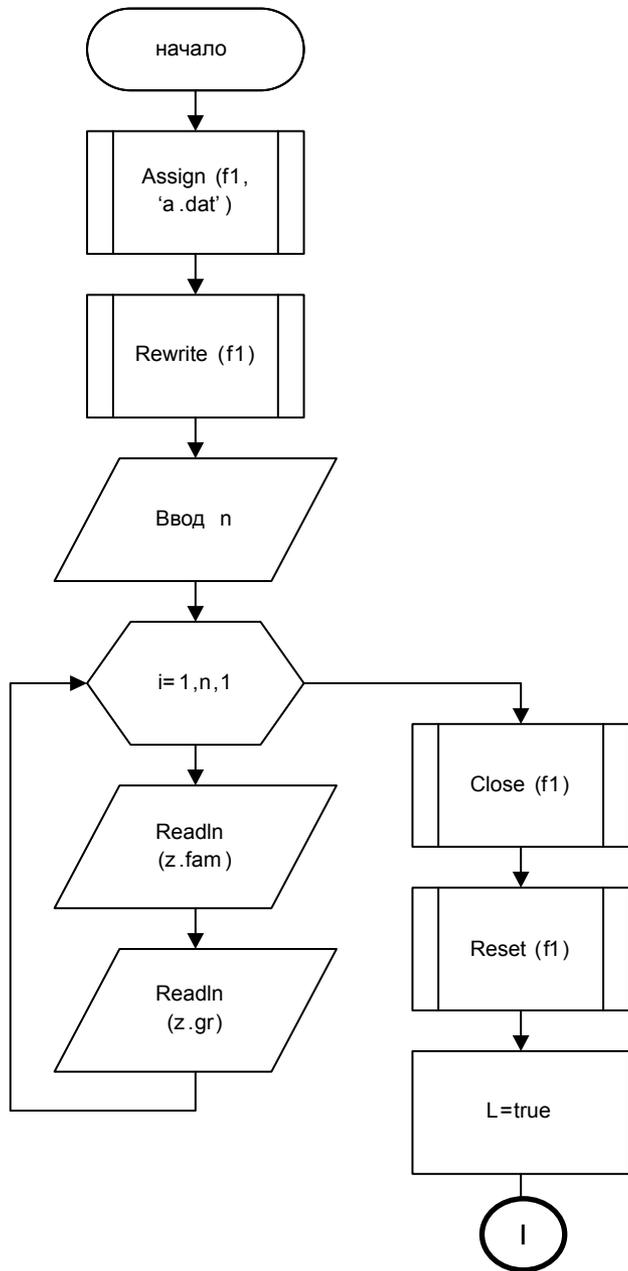
Стандартные функции для работы с файлами

Имя и параметры	Типы параметров	Тип результата	Действие
SeekEof(f)	<i>f</i> - переменная типа <i>Text</i>	boolean	Пропускает пробелы, символы табуляции и перехода на новую строку, после чего возвращает <i>True</i> , если файловый указатель стоит на конце файла, и <i>False</i> в противном случае
SeekEoln(f)	<i>f</i> - переменная типа <i>Text</i>	boolean	Пропускает пробелы, символы табуляции, после чего возвращает <i>True</i> , если файловый указатель стоит на конце строки, и <i>False</i> в противном случае

Стандартные функции для работы с файлами

Имя и параметры	Типы параметров	Тип результата	Действие
FileSize(f)	<i>f</i> - переменная типа <i>file</i>	integer	Возвращает количество элементов в <i>типизированном</i> файле
FilePos(f)	<i>f</i> - переменная типа <i>file</i>	integer	Возвращает позицию файлового указателя в <i>типизированном</i> файле

Пример 1: Создать файл, содержащий информацию о студентах: фамилия, гражданство. Вывести фамилии иностранных граждан.



```
Program file2;  
Type ved=record  
    fam:string[20];  
    gr:string[15]  
        end;  
Var f1:file of ved; // 1-ая  
    z:ved;  
    i,n:integer;  
    l:boolean;  
begin  
    assign(f1,'e:\a.dat'); // 2-ая  
    rewrite(f1); // 3-я  
    writeln('ВВЕДИТЕ КОЛИЧЕСТВО СТУДЕНТОВ');  
    readln(n);  
    for i:=1 to n do  
    begin  
        readln(z.fam);  
        readln(z.gr);  
        write(f1,z) // 4-ая  
    end ;  
    close(f1); // 5-ая
```

```
reset (f1) ;  
l:=true;  
repeat  
read (f1, z) ;  
if z.gr<>'Россия' then  
    begin  
        writeln (z.fam, '  
' , z.gr) ;  
        l:=false;  
    end  
Until eof (f1) ;  
close (f1) ;  
if l then write ('все граждане  
России')  
end
```

Результаты решения задачи:

введите количество студентов

4

Иванов А.Р.

Россия

Бортко П.О.

Украина

Смирнова К.Р.

Россия

Сванидзе А.О.

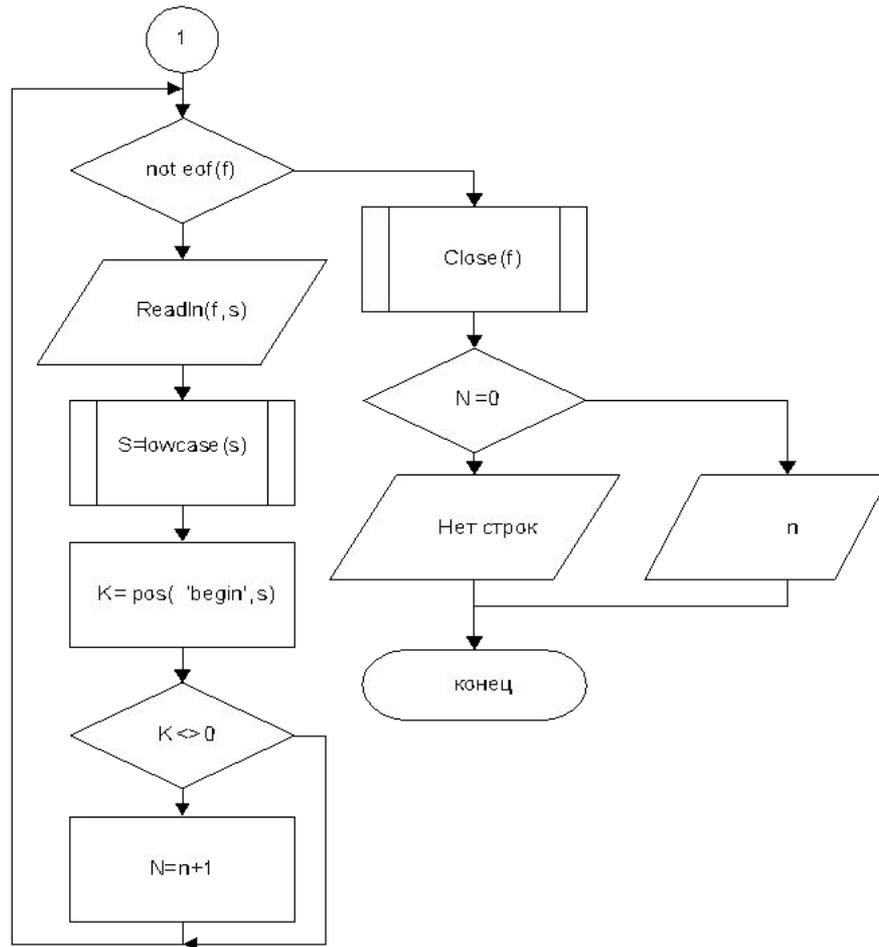
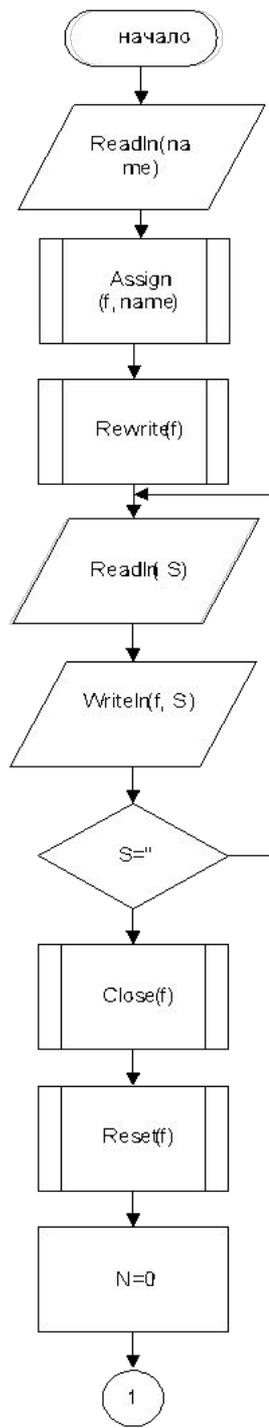
Грузия

Бортко П.О. - Украина

Сванидзе А.О. - Грузия



Пример 2: Текстовый файл представляет собой фрагмент программы на языке Pascal. Определить количество строк в которых есть служебное слово «begin».



```
Program file1;  
Var f:text;  
      S,name:string;  
      n,k:byte;  
Begin  
  Writeln('введите путь и имя файла');  
  Readln(name);  
  Assign(f, name);  
  Rewrite(f); {открыть файл для записи}  
  Repeat  
    Readln(s);  
    Writeln(f,s); {запись строки в файл}  
  Until s='end.'; {пока не end.}  
  Close(f); {закрывать файл}
```

```
Reset (f) ;    {открыть файл для чтения}
  n:=0;
  While not (eof (f)) do {пока не конец
файла}
  Begin
    Readln (f, s) ;    {читать строку из
файла}
    S:=LowerCase (s) ;    {преобразует строку к
нижнему регистру}
    K:=pos ( 'begin' , s) ;
    If k<>0 then n:=n+1;
  end;
  close (f) ;
  if n=0 then writeln ( 'нет строк со словом
begin' )
  else writeln ( 'кол-во строк=' , n)
end.
```

Результат решения задачи

введите путь и имя файла

```
e:\a.txt
```

```
Begin
```

```
end.
```

```
кол-во строк=1
```