

ФАЙЛЫ

ОПРЕДЕЛЕНИЕ ПОНЯТИЙ

- **Физический Файл** - это поименованная область на диске, содержащая какую-либо информацию.
- **Логический файл** - это одна из структур данных, используемых в программировании.

СТРУКТУРА ЛОГИЧЕСКОГО ФАЙЛА

Это способ восприятия файла в программе, т.е. «шаблон», через который мы смотрим на физическую структуру файла на диске. В ЯП таким шаблонам соответствуют типы данных, допустимые в качестве компонент файлов.

File of byte:

байт	байт	. . .	байт	Eof
------	------	-------	------	------------

File of char:

код символа	код символа	. . .	код символа	Eof
-------------	-------------	-------	-------------	------------

File of integer:

целое со знаком	целое со знаком	. . .	целое со знаком	Eof
-----------------	-----------------	-------	-----------------	------------

И другие

Логическая структура файла в принципе очень похожа на структуру массива.

Различия:

- У массива количество элементов фиксировано, а у файлов количество элементов может изменяться в процессе работы. (Количество в каждый момент времени неизвестно, но в конце файла стоит символ Eof)
- Массив целиком располагается в ОП, а файл находится на диске.
- Нумерация элементов массива выполняется соответственно значений нижней и верхней границ, указанных при его объявлении. Нумерация элементов файла выполняется слева направо, начиная с нуля

Классификация Файлов в ПАСКАЛЕ

ФАЙЛЫ

ПО ТИПУ

ПО МЕТОДУ
ДОСТУПА

Типизированные

Текстовые

Нетипизированн
ые

Последовательн
ого
доступа

Прямого доступа



ИСПОЛЬЗОВАНИЕ

Файлы используются для хранения данных. Из них можно считывать начальные данные, записывать результаты, изменять информацию в файле.

РАБОТА С ТЕКСТОВЫМИ ФАЙЛАМИ

var список имен файлов : **text**;

Текстовый файл может состоять из любых символов (в том числе и цифр)

Для работы с каким-либо физическим файлом (тем, который существует на диске) его необходимо связать с файловой переменной

Assign (имя файла, 'путь к файлу');

НАПРИМЕР:

На диске есть файл:

D:\MyFile.dat

...

```
Var f : text;
```

...

```
Begin
```

...

```
Assign (f; 'D:\MyFile.dat' );
```

Или:

```
Var f : text;
```

```
Name: string;
```

...

```
Begin
```

...

```
Name := 'D:\MyFile.dat' ;
```

```
Assign ( f, name);
```

ПРИНЦИПЫ РАБОТЫ С ФАЙЛАМИ

1. Открытие
2. Чтение из файла или запись в файл
3. Закрытие

RESET (название файла) – открытие файла для чтения из него информации в ОП

REWRITE (название файла) – открытие файла для записи данных в файл

APPEND (название файла) – открытие с целью дополнения данных

CLOSE (название файла) – закрытие файла

ИСПОЛЬЗОВАНИЕ ДАННЫХ ИЗ ФАЙЛА

- Для считывания данных из файла в ОП используют **read** и **readln**.

Read (название файла, список переменных)

Readln (название файла, список переменных)

Если в списке переменных есть переменная типа **char** или **string [10]**, то из строки в файле считывается 1 или 10 символов (вместе с пробелами) и присваиваются этой переменной.

- Запись в файл осуществляется процедурами **write** и **writeln**

Write (имя файла, список выражений);

Writeln (имя файла, список выражений).

- Если в списке переменных есть числовая переменная (`integer` или `real`), то считываются символы, которые трактуются как цифры до ближайшего пробела.
- Т. О. особенностью текстового файла является то, что происходит автоматическое преобразование числовых данных в цепочку символов при записи в файл и обратное преобразование символов в цифры при чтении из файла.

Функции и директивы для работы с файлами

- Функция **Eof** (имя файла) – true, если достигнут конец файла и False – иначе.
- **{ \$ I - }** , **{ \$ I + }** – директивы компилятору Pascal – отключают и включают контроль ошибок ввода-вывода. Если этого не сделать, то отсутствие файла приведет к аварийному завершению программы.
- Функция **IOResult** (имя файла) определяет наличие файла на диске (0 – есть, 1 – нет).

ПРИМЕРЫ ПРОГРАММ

Задача № 1

Пусть на диске (в текущем каталоге) есть файл **myfile.dat**, который состоит из некоторого числа целых чисел, разделенных пробелами. Написать программу, вычисляющую сумму этих элементов.

Program Files;

```
uses Crt;
var f :text;
    x: integer;
    Summa:longint;
begin
clrscr;
{$I-}
assign(f, 'myfile.txt');
reset(f);
{$I+}
if IOresult<>0 then
writeln('ошибка
открытия файла')
else
```

```
begin
    Summa:=0;
while not Eof(f) do
begin
    read(f,x);
    Summa:=Summa+x
end;
    Writeln('Summa= ',
Summa:8);
end;
    Close(f);
    readln;
end.
```

СОЗДАНИЕ ФАЙЛОВ

1 способ – с помощью текстового редактора, например Блокнот или Pascal.

2 способ – программными средствами.
Открыть файл процедурой Rewrite (f)

Процедурой REWRITE нельзя открыть запись информации в уже существующий файл. При выполнении этой процедуры старый файл с таким же именем **уничтожается** и никаких сообщений в программу не передается.

Задача № 2

Написать программу, в которой в текстовый файл записываются данные про 10 учеников: имя, вес и рост.

Перед созданием файла программа должна проверять наличие файла с таким именем на диске и спрашивать, что ей делать в случае обнаружения такого файла – прекратить работу или перезаписать файл.

```
Program Zapfile;  
Uses Crt;  
var f : text;  
    name : string [ 10 ] ;  
    ves : real;  
    ROST : real;  
    Otvet : char;  
Begin  
Clrscr;  
Assign ( f, ' deti . txt ' ) ;  
{ $ | - }  
{ проверяем, существует ли  
такой файл }  
Reset ( f ) ;  
If IOResult = 0 then  
Begin writeln ( ' файл deti. txt  
существует.  
Заменить его? ( y /n ) ' ) ;
```

```
Readln ( otvet ) ;  
if otvet = ' n ' then halt ;  
End;  
Else begin  
Close ( f ) ;  
Rewrite ( f ) ;  
For i := 1 to 10 do  
Begin  
Writeln ( ' введите имя, вес и  
рост ' );  
Readln ( name, ves, rost ) ;  
Writeln ( f, name : 10, ves : 4,  
rost : 5:2 ) ;  
End;  
End;  
Close ( f ) ;  
End.
```


Задача № 3

Написать программу, которая считывает слова из одного текстового файла и записывает их в столбик в другой текстовый файл.

Пояснение: слова разделяются символом пробел. Поэтому мы будем считывать символы из первого файла и «складывать» их в слово до тех пор, пока не встретится пробел. Потом это слово запишем во второй файл и опять начнем формировать следующее слово. И так до тех пор, пока не достигнем конца первого файла.

```
Program slovo;  
{Запись слов из файла f  
в столбик в файл h}  
uses Crt;  
var f,h:text;  
    буква:char;  
    clovo:string;  
begin clrscr;  
assign(f,'f.pas');  
assign(h,'h.pas');  
reset(f);  
rewrite(h);  
clovo:=' ';  
while not eof(f) do
```

```
begin  
    read(f, буква);  
    if буква<>' ' then  
        begin  
            clovo:=clovo+буква;  
        end  
    else  
        begin  
            writeln(h,clovo);  
            writeln(clovo);  
            clovo:=' ';  
            End ;  
        end;  
    readln;  
    Close (f);  Close (h); end.
```

Задание для самостоятельной работы:

- Написать программу, которая на диске компьютера создает файл `numbers.txt` и записывает в него 5 введенных с клавиатуры целых чисел. При помощи текстового редактора (например, БЛОКНОТА) просмотрите файл и убедитесь, что запись в файл произошла.
- Написать программу, которая дописывает в файл `numbers.txt` 5 введенных с клавиатуры целых чисел. При помощи текстового редактора (например, БЛОКНОТА) просмотрите файл и убедитесь, что запись в файл произошла.