



# Тип данных файл

1 курс



## План:

1. Понятие типа данных файл. Дескриптор файла.
2. Классификация файлов в Паскале.
3. Структура описания типа данных файл.
4. Действия с типом данных файл в Паскале.
5. Функции для удобства работы с файлами.



## Файл – это ...

Называется совокупность данных, записанных во внешней памяти под определенным именем.

Паскаль не работает напрямую с файлами, работает с дескрипторами файла

(это переменная, которая описывает конкретный файл, который Вы собираетесь обрабатывать. Дескриптор Вы должны создать сами, точно так же как Вы описываете любую другую переменную).

# Классификация файлов



# Классификация файлов по лог. структуре

- **Типизированная информация** - в файле находится (или будет находиться) информация любого рода. Но структура такой информации обязательно должна повторяться. То есть, файл должен состоять из 1 и более одинаково устроенных частей.


<имя\_переменной\_дескриптора>: **File of** <любой\_тип\_данных\_паскала>;

- **Не типизированная информация** - файл содержит любую информацию, не поддающуюся какой либо классификации. Например набор личных данных, машинный код, закодированный или сжатый блок информации.

<имя\_переменной\_дескриптора>: **File;**

- **Текстовая информация** - в файле находится (или будет находиться) текст. Под текстом я понимаю набор строк из символов. Каждая строка начинается с новой строки.

<имя\_переменной\_дескриптора>: **Text;**



# Действия над файлами:

- 1. Связать дескриптор и конкретный файл на компьютере**
- 2. Открыть файл**
- 3. Записать в файл или прочесть из него информацию**
- 4. Закрывать файл**

# 1.Связать дескриптор и конкретный файл на компьютере

Для дескрипторов любого типа связь с файлом осуществляется одной командой:

**Assign**(<имя\_переменной\_дескриптора>, <строка\_имя\_файла>);

Параметры:

<имя\_переменной\_дескриптора> - это, собственно, и есть дескриптор, который мы объявили ранее;

<строка\_имя\_файла> - это либо константа, либо переменная-строка, содержащая в себе имя файла.

*Например,*

`assign(f,'ByteData.dat');` { Теперь переменная f - это файл bytedata.dat, и состоит из байт }

`assign(tel,s+'\Phones.db');` { tel это файл с телефонами и обонентами. Находится в d:\teldata\phones.dat }

`assign(dat,filename1);` { dat - это текстовый файл myfile1.txt }

## 2.Открыть файл

**Один и тот же файл нельзя открыть одновременно двумя разными способами.**

### **Rewrite**

Эта команда создает (если файла с таким именем еще нет) и открывает файл.

### **Reset**

Эта команда открывает файл, связанный с дескриптором, и позволяет читать или писать в него информацию.

### **Append**

Формат такой же как и у двух предыдущих команд с соблюдением всех правил и указаний. Разница между ними в двух вещах: после открытия файла таким образом читать информацию из него нельзя, и второе - последующая запись в файл будет писать информацию после уже существующей в нем.

Синтаксис:

**<команда>**(<название\_дескриптора\_файла>[, <размер\_записи>]);

### **Truncate**

Синтаксис:

**Truncate**(<название\_дескриптора\_файла>);



## 3. Записать в файл или прочесть из него информацию

В паскале есть два стандартных набора команд для работы с информацией в файлах.  
Это:

1. **Write** и **Writeln**, **Read** и **Readln**
2. **BlockRead** и **BlockWrite**

Например,

```
var
  f:text;
  s1,s2,s3:string;
  v1:longint;
  c1,c2,c3:char;
begin
  assign(f,<тут имя файла>);
  reset(f);
  readln(f,s1);      {s1 <- "Это строка"}
  readln(f,v1);     {v1 <- 12321}
  readln(f,s2);     {s2 <- "Это было число"}
  readln(f,s3);     {s3 <- "А вот это будет три символа:"}
  readln(f,c1,c2,c3); {c1 <- "№", c2 <- "%", c3 <- "@"}
  ...{остальные действия}
end.
```

## 4. Закрывать файл

Файл любого типа закрывается командой **Close**, которая имеет простой синтаксис:

```
procedure Close(<дескриптор_открытого_файла>);
```

Попытка закрыть уже закрытый (либо ещё не открытый) файл не приводит ни к каким последствиям. Это бывает в некоторых случаях удобно.

## Функции для удобства работы с файлами:

1. **Seek**- *меняет в открытом файле текущее месторасположение.*
2. **FilePos**-*используется для того чтобы узнать номер текущей записи в файле.*
3. **FileSize**-*позволяет определить количество записей в файле.*
4. **Eof**-*можно узнать, достигли ли мы конца файла.*
5. **SeekEof**-*только для текстовых файлов, аналог предыдущей.*
6. **EoLn**-*временное сохранение.*
7. **SeekEoLn**-*только для текстовых файлов, аналог предыдущей.*
8. **Delete**-*удаление.*
9. **Rename**-*переименовать.*

## Домашнее задание:

Учебник «Информатика» 10 класса. А. Гремальски, §1.6-§1.8

Ответить на вопросы письменно:

1. Объясните термины файл языка Паскаль, внешний файл.
2. Для чего нужны процедуры открытия и закрытия файлов?
3. Из скольких элементов может состоять файл?
4. В каком порядке записываются и считываются элементы файла с последовательным доступом?
5. В чем разница между текстовым файлом и файлом **file of char**?
6. Записать программу для создания, считывания файла с компонентами типа запись (record).
7. Записать программу для создания, считывания и обработки текстового файла.