

ФАЙЛЫ

```
graph TD; A[ФАЙЛЫ] --> B[Текстовые]; A --> C[Типизированные]; A --> D[Нетипизированные];
```

Текстовые

- Состоят из символьных строк, символа конца строки и символа конца файла.
- Каждый элемент доступен последовательным способом, т.е. друг за другом.

Типизированные

- Элементы имеют один и тот же тип.
- Каждый элемент имеет свой номер.
- Каждый элемент доступен и последовательным и прямым способом.
- Информация хранится в машинном представлении, непригодным для просмотра в текстовом редакторе.
- Скорость ввода/вывода выше, чем в текстовый файл.
- Объём памяти меньше, чем у текстового файла.

Нетипизированные

- Элементы могут иметь любой тип.
- Каждый элемент имеет свой номер.
- Каждый элемент доступен прямым способом.
- Информация хранится в машинном представлении, непригодным для просмотра в текстовом редакторе.
- Скорость ввода/вывода выше, чем в текстовый файл.
- Объём памяти меньше, чем у текстового файла.

Каким бы не был файл последовательность обращения к нему следующая:

- Описать файловую переменную;
- Установить связь программы с файлом;
- Открыть файл для чтения или записи;
- Читать из файла или записать в файл;
- Закрыть файл.

Текстовые файлы

- Описать файловую переменную;

f:text;

- Установить связь программы с файлом;

Assign(f, 'pushkin.txt');

- Открыть файл для чтения ИЛИ записи;

ReWrite(f); // файл открыть для записи

Reset(f); // файл открыт для чтения

- Читать из файла ИЛИ записать в файл;

WriteLn(f, s); // строковая переменная s записана в файл

*ReadLn(f, s); // в строковую переменную s считана
// строка из файла*

- Закреть файл.

Close(f);

Записать в файл нескольких строк, введённых с клавиатуры.

Когда состоит только из символа '.', ввод строк заканчивается.

```
var
f:text;
s:string;
begin
assign(f, 'privet.txt');
rewrite(f);
readln(s);
while s<>'.' do
begin
writeln(f,s);
readln(s);
end;
close(f);
end.
```

Прочитать все строки из заданного файла и вывести их на экран.

```
var
f:text;
s:string;
begin
assign(f, 'privet.txt');
reset(f);
while not eof(f) do
begin
readln(f, s);
writeln(s);
end;
close(f);
end.
```

Имя файла задано в программе

- `assign(f, 'privet.txt');` // в каталог, в котором находится программа
- `assign(f, 'z:/test/lala/privet.txt');` // в заданный каталог

Имя файла вводится с клавиатуры

```
• var  
  f:text;  
  s:string;  
  name_file:string;  
begin  
  writeln('Name of file:');  
  readln(name_file);  
  assign(f, name_file);
```

Типизированные файлы

- Описать файловую переменную;

var

имя: file of базовый тип;

Пример:

var

next: file of string;

look: file of integer;

- Установить связь программы с файлом;

`Assign(f, 'pushkin.txt');`

- **Открыть файл для чтения ИЛИ записи;**

ReWrite(f); // файл открыть для записи

Reset(f); // файл открыт для чтения

- **Читать из файла ИЛИ записать в файл;**

Write(f, s); // строковая переменная s записана в файл

Read(f, s); // в строковую переменную s считана

// строка из файла

PS!!!! ~~WriteLn, readLn~~ т.к. данные записываются в одну строку

- **Закрывать файл.**

Close(f);

- Seek(имя_файловой_переменной,номер_элемента);

PS: нумерация элементов с нуля!

- FilePose(имя_файловой_переменной);// текущее положение указателя файла
- FileSize(имя_файловой_переменной);//общее количество записей

Нетипизированные файлы

- Описать файловую переменную;

var

имя: file;

Пример:

var

xt: file;

Все остальные операции обращения к файлу, такие как: установка связи между файловой переменной и файлом на диске, запись, чтение, закрытие выглядят так же как и с текстовым файлом.