

---

# Файловый тип данных Turbo Pascal

---

Операции для работы с файлами

11 класс

# Общие положения

**Множества значений или переменных с одним общим именем называются *структурированными (составными) типами.***

Основные разновидности структурированных типов:

- ***регулярный тип (массивы);***
- ***комбинированный тип (записи);***
- ***Файловый тип (файлы);***
- ***Множественный тип (множества).***

# Описание

***Файл* – это область памяти на внешнем носителе, в которой хранится некоторая информация.**

В языке Паскаль файл представляет собой последовательность элементов одного типа.

В *файлах последовательного доступа*, чтобы получить доступ к элементу, необходимо последовательно просмотреть все предыдущие.

# Объявление файловой переменной

```
Var <имя файла>: File Of <тип элементов>;
```

Например:

```
var
```

```
  f1: file of char;
```

```
  f2: file of integer;
```

```
  f3: file;
```

```
  t: text;
```

---

# Типы файлов Турбо Паскаль

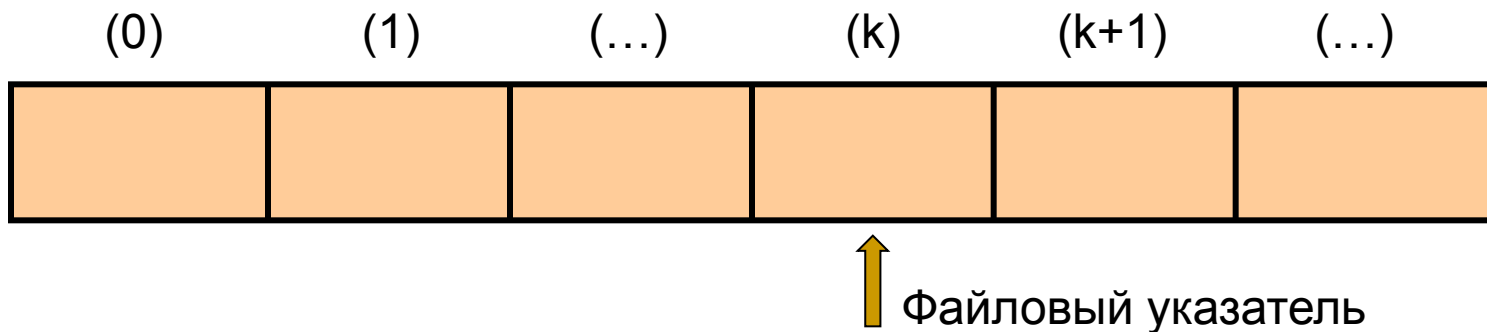
Турбо Паскаль поддерживает три файловых типа:

- *текстовые файлы;*
- *типизированные файлы;*
- *нетипизированные файлы.*

Переменную файлового типа описывают одним из трех способов:

- *file of тип* - типизированный файл (указан тип компоненты);
  - *text* - текстовый файл;
  - *file* - нетипизированный файл.
-

- Файл, в описании которого указывается тип элементов, называется *типизированным*. Все элементы файла пронумерованы начиная с нуля.
- С каждым файлом связан так называемый *файловый указатель*. Это неявно описанная переменная, которая указывает на некоторый элемент файла. Все операции производятся с элементом, на который указывает файловый указатель.



---

# Обработка файлов

- Связь переменной файлового типа с файлом на диске.
  - Чтение из файла.
  - Закрытие файла.
  - Признак конца файла.
  - Запись в файл.
  - Установка указателя.
  - Определение номера элемента.
  - Определение количества элементов в файле.
  - Удаление и переименование файлов.
-

# Связь переменной файлового типа с файлом на диске.

Любые дисковые файлы становятся доступными программе после связывания их с файловой переменной, объявленной в программе. Все операции в программе производятся только с помощью связанной с ним файловой переменной.

Для установления связи служит процедура

**Assign**(<имя файловой переменной>, '<имя файла на диске>');

Например: `Assign(F1, 'A:int.dat');`





# Чтение из файла.

- Под чтением из файла понимается пересылка данных из внешнего файла, находящегося на диске, в оперативную память.
- Для чтения из файла необходимо открыть файл для чтения с помощью процедуры **RESET**(<имя файловой переменной>);
- Чтение данных из файла выполняется процедурой **READ** (<имя файловой переменной>, '<имя переменной>');

Примечание: переменная должна иметь тот же тип, что и компоненты файла.



# Заккрытие файла.

После того как данные из файла прочитаны, его необходимо закрыть посредством процедуры **CLOSE** (<имя файловой переменной>);

# Признак конца файла.

Так как число элементов файла заранее не известно, необходимо уметь определять, что файл кончился. Для этого используется логическая функция

**EOF** (<имя файловой переменной>);

Она принимает истинное значение (True), если достигнут конец файла, и ложное (False) – в противном случае.



# Запись в файл.

Под записью в файл понимается вывод результатов программы из оперативной памяти ЭВМ в файл на диске.

Для записи в файл необходимо открыть файл для записи посредством процедуры

**REWRITE**(<имя файловой переменной>);

Запись данных в файл выполняется процедурой

**WRITE**(<имя файловой переменной>, <значение>);

После работы с файлом его необходимо закрыть.

# Пример 1

**Прочитать из файла целые числа и вывести их на экран:**

```
Assign (F1, 'A: INT.DAT' );  
{связываем файловую переменную с файлом на диске}  
Reset (F1); {открываем файл для чтения}  
While Not EOF (F1) Do  
{пока не достигнут конец файла F1}  
Begin  
    Read (F1, n); {считываем очередное число}  
    Write (n, ' '); {выводим его на экран}  
End:  
Close (F1); {закрываем файл}
```

---

# Пример 2

**Ввести с клавиатуры и записать в файл DAN1.DAT последовательность целых чисел. Признак конца ввода чисел - 0.**

```
Program Exampl_2;  
Var F: File Of Integer;  
    n:= Integer;  
Begin  
    Assign (F, 'DAN1.DAT');  
        {связываем файловую переменную с файлом на диске}  
Rewrite(F); {открываем файл для записи}  
Writeln ('Конец ввода чисел – 0');  
Repeat      {Пока не будет введен 0}  
    Writeln ('Ведите число');  
    Readln(n); {Если введено число, отличное от 0, то записываем  
        его в файл}  
If n<>0 Then Write(F, n);  
Until n=0; {Если введен 0, то выходим из цикла}  
Close (F);  
End.
```

---

# Домашнее задание

- 1) В файле DAN1.DAT (см. предыдущую задачу) записаны целые числа. Вычислить сумму элементов файла и результат вместе с исходными данными записать в файл DAN2.DAT.
  - 2) Какие из следующих операторов правильные?
    - Assign(f1, 'A:STR1.DAT') ;
    - Reset(f1,f2) ;
    - Rewrite;
    - Assign(f2,'C:\ТТ\TAB1.DAT')
    - Rewrite (f1);
-

# Текстовые файлы

- *Текстовые файлы* – это файлы, содержащие символы, разделенные на строки. Строки могут иметь различную длину, и в конце каждой строки стоит признак конца строки. Для их описания используется служебное слово **Text**:
- **Var**     A:     Text;

# Обработка текстовых файлов

Для обработки текстовых файлов используются те же процедуры и функции, что и для обработки обычных типизированных файлов.

- Для связывания файловой переменной с файлом на диске употребляется процедура **Assign**.
- Для чтения данных применяется процедура **Read**.
- Если необходимо после чтения данных перейти на следующую строку, то используется процедура **Readln**.



---

# Продолжение

- Процедура **Write** записывает данные в текущую строку.
  - Если надо записать данные и перейти к следующей строке, то можно использовать процедуру **WriteIn**, которая записывает в файл признак конца строки и устанавливает файловый указатель на начало следующей строки.
-

---

# Продолжение

- Так как в строках может быть разное количество символов, имеется логическая функция

**Eoln** (<имя файловой переменной  
текстового файла>),

которая принимает значение True, если достигнут конец строки.

---

---

# Продолжение

- Кроме перечисленных процедур и функций, к текстовым файлам применяется процедура

**Append** (<имя файловой переменной текстового файла>).

Она открывает файл для "дозаписи", помещая файловый указатель в конец файла.

---

---

## Пример 3

- Создать текстовый файл, содержащий только целые числа, в каждой строке может быть несколько чисел, которые разделяются пробелом. Вывести на экран все числа с учетом разбиения на строки и подсчитать количество элементов в каждой строке.
-

---

Пусть в файле содержится информация:

1 2 3 4 5 6 7

-1 -2 -3 -4

-1 -2

**Этот файл можно создать:**

1. Создать новый файл (меню File команда New)
2. Записать все числа в строках через пробелы.
3. Сохранить его, например, A:\UNT1.TXT

***Или написать программу для создания текстового файла.***

---

```
Program Example_3;
```

```
Var F: Text;
```

```
x, k: Integer;
```

```
Begin
```

```
Assign (F, 'UNT1.TXT');
```

```
{Связываем файловую переменную с файлом на  
диске}
```

```
Reset (F); {Открываем файл для чтения}
```

```
While Not Eof(F) Do {Пока не достигнут конец  
файла}
```

```
Begin
```

```
k:=0; {Счетчик элементов строки}
```

```
While Not Eoln(F) Do {Пока не достигнут конец  
строки}
```

```
Begin
```

```
Read(F, x); {Считываем очередное число}
```

```
Write(x, ' '); {Выводим его на экран}
```

```
Inc(k); {Увеличиваем счетчик }
```

```
End;
```

```
Writeln( ' В строке ', k, ' элементов' );
```

```
Readln(F); {Переходим к следующей строке  
файла}
```

```
End;
```

```
Close (F); {Закрываем файл}
```

```
Readln;
```

```
End.
```

# Решение