



типизированные файлы

Типизированные файлы

- это файлы, все элементы в которых одного типа

```
var f: file of integer;
```

```
    f1: file of real;
```

```
    f3: file of string[10];
```

При работе с типизированным файлом можно сначала определить тип данных, а затем определить файл:

```
type massiv=array[1..5] of integer;
```

```
var f : file of massiv;
```

В типизированном файле можно использовать любой тип кроме файлового.

- Типизированные файлы допускают как последовательный, так и прямой доступ. При использовании прямого доступа следует помнить, что элементы файла всегда нумеруются, начиная с нуля!
- Чтение из типизированного файла осуществляется **только** процедурой **Read**, а запись - **Write**
- **Read**(файловая переменная, список переменных);
- **Write**(файловая переменная, список переменных);

При считывании в каждую переменную из списка переменных указатель текущей позиции в файле перемещается на следующий элемент.



↑
указатель текущей позиции

Если указатель текущей позиции находится за последним элементом, т.е. в позиции $EOF(f)=true$, то выполнение процедуры `Read` приводит к ошибке.

При записи в файл, указатель перемещается на следующий элемент.

При достижении конца файла-файл расширяется

Процедуры и функции для работы с типизированными файлами

название	действие
FilePos функция	-определяет номер текущей позиции указателя в файле (начиная с нуля)
Seek процедура	-перемещает указатель текущей позиции в файле на элемент с заданным номером (начиная с нуля!)
FileSize функция	-определяет текущий размер файла (число элементов файла, начиная с единицы)
Truncate процедура	-усекает размер файла до текущей позиции. Все элементы, расположенные после текущей позиции, отсекаются и она становится его концом EOF(f)=true;

в режиме append.

Недостатки типизированных файлов:

- Нельзя создать или просмотреть в текстовом редакторе.
- В него нельзя дописывать данные в режиме append.

Пример.

Составить программу, которая создает типизированный файл, состоящий из случайного числа случайных целых чисел, выводит этот файл на экран компьютера. Затем сортирует элементы файла и выводит на экран уже отсортированный файл.

```

Program Filesort;
uses crt;
var f: file of integer;
    x, y : integer;
    i, j : integer;
begin
    {$I-}
    Clrscr;
    assign (f, 'sort_dat.pas');
    {$I+}    { Создаем файл случайных чисел }
    Rewrite ( f );
    Randomize;
    J :=random( 100 ); { количество элементов файла }
    For i:=1 to j do
        begin
            x:=random(100);
            write( f, x );
        End;
    close ( f );
    writeln ('исходный файл');
    reset ( f );
    for i:=1 to fileSize ( f ) do
        begin
            read ( f, x );
            write ( x:4 );
        end;
end;

```

```

writeln;
close ( f );    { сортировка }
reset ( f );
for l :=fileSize(f)-1 downto 1 do
    { всплывание очередного максимального
    элемента на 1-ую позицию}
    for j:=0 to l-1 do
        begin
            seek ( f, j );
            Read (f, x, y);
            if x>y then
                begin
                    seek(f, j);
                    write ( f, y, x );
                end;
            end;
        close ( f );
        reset ( f );
        writeln ('отсортированный файл');
        for i:=1 to fileSize ( f ) do
            begin
                read( f, x );
                write ( x:4 );
            end;
        readln;
        close ( f );
    end.

```

Задачи для самостоятельной работы:

- 1) Написать программу, которая создает типизированный файл, состоящий из случайного числа целых случайных чисел, и затем выводит содержание этого файла на экран компьютера.
- 2) Используя файл, созданный в предыдущей задаче, найти:
 - а) самое большое число в этом файле.
 - б) количество одинаковых чисел в нем.
 - в) порядковые номера чисел, кратных 3.