

Программирование на языке Паскаль

**Файлы с последовательным
доступом.**

Кулебякин В.В.

Файлы

Файл – это область на диске, имеющая имя.

Файл

ы

Текстовые

е

только текст без оформления,
не содержат управляющих
символов (с кодами < 32)

ASCII (1 байт на символ)

UNICODE (2 байта на символ)

*.txt, *.log,

*.htm, *.html

Двоичные

могут содержать любые
символы кодовой таблицы

*.doc, *.exe,

*.bmp, *.jpg,

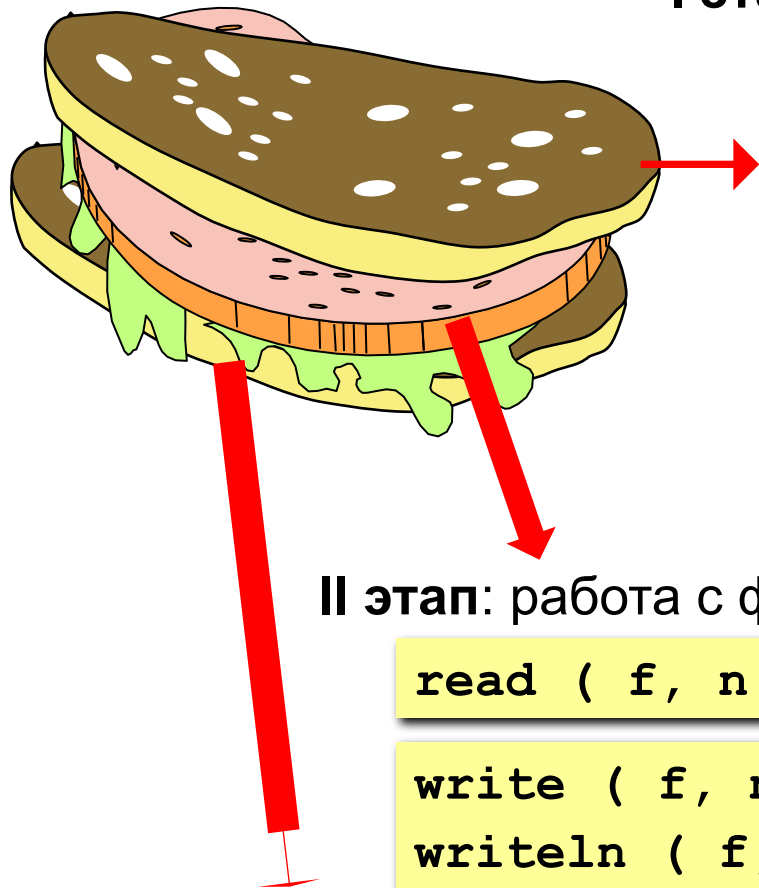
*.wav, *.mp3,

*.avi, *.mpg

Папки (каталоги)

Принцип сэндвича

3
Переменная типа
«текстовый файл»:
`var f: text;`



I этап. открыть файл :

- связать переменную `f` с файлом

```
assign(f, 'qq.txt');
```

- открыть файл (сделать его активным, приготовить к работе)

```
reset(f); {для чтения}
```

```
rewrite(f); {для записи}
```

II этап: работа с файлом

```
read ( f, n ); { ввести значение n }
```

```
write ( f, n ); { записать значение n }
```

```
writeln ( f, n ); {с переходом на нов.строку }
```

III этап: закрыть файл

```
close(f);
```

Работа с файлами

Особенности:

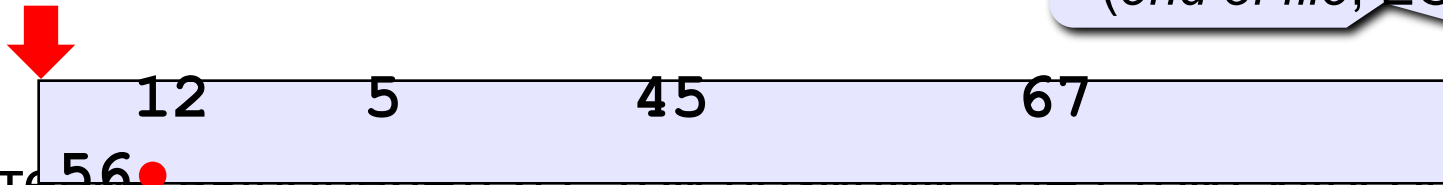
- имя файла упоминается только в команде `assign`, обращение к файлу идет через файловую переменную
- файл, который открывается на чтение, должен **существовать**
- если файл, который открывается на запись, существует, старое содержимое **уничтожается**
- данные записываются в файл в текстовом виде
- при завершении программы все файлы закрываются автоматически
- после закрытия файла переменную `f` можно использовать еще раз для работы с другим файлом

Последовательный доступ

- при открытии файла курсор устанавливается в начало

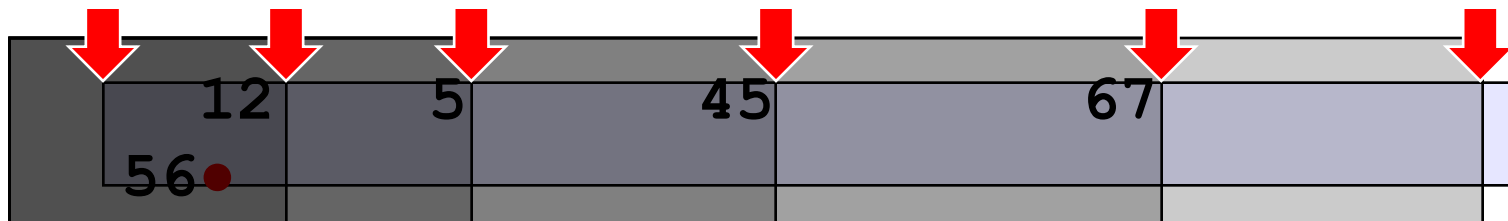
```
assign ( f, 'qq.txt' );
reset ( f );
```

конец файла
(*end of file*, EOF)



- чтение выполняется с той позиции, где стоит курсор
- после чтения курсор сдвигается на первый непрочитанный СИМВОЛ

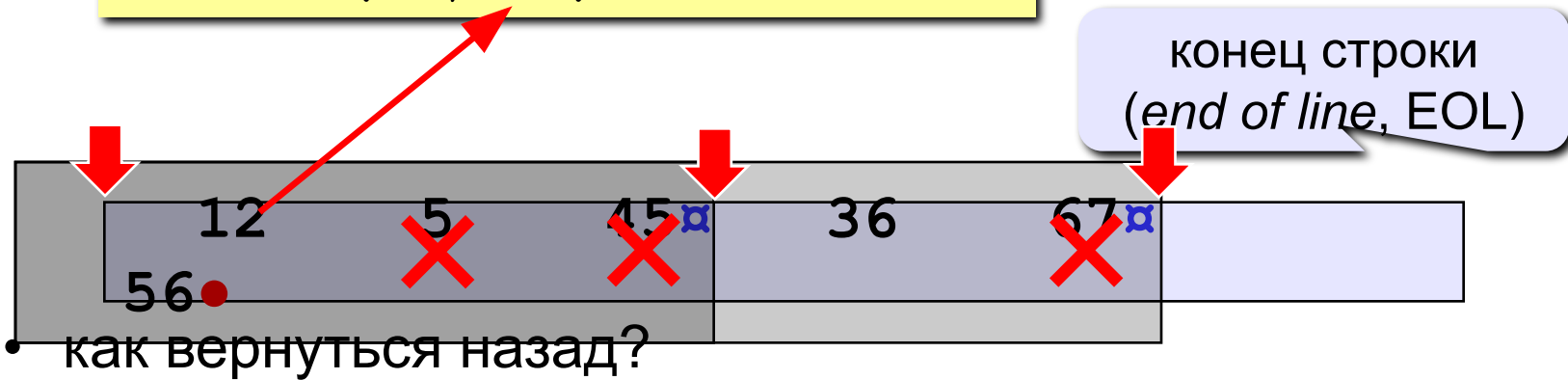
```
read ( f, x );
```



Последовательный доступ

- чтение до конца строки

```
readln ( f, x );
```



```
close ( f );
```

```
reset ( f ); { начать с начала }
```

Пример

Задача: в файле `input.txt` записаны числа (в столбик), сколько их – неизвестно. Записать в файл `output.txt` их сумму.



Можно ли обойтись без массива?

Алгоритм:

1. Открыть файл `input.txt` для чтения.
2. $S := 0$;
3. Если чисел не осталось, перейти к шагу 7.
4. Прочитать очередное число в переменную x .
5. $S := S + x$;
6. Перейти к шагу 3.
7. Закрыть файл `input.txt`.
8. Открыть файл `output.txt` для записи.
9. Записать в файл значение S .
10. Закрыть файл `output.txt`.

цикл с условием
«пока есть данные»

```
program qq;  
var s, x: integer;  
    f: text;  
begin  
    assign(f, 'input.txt');  
    reset(f);  
    s := 0;  
    while not eof(f) do begin  
        readln(f, x);  
        s := s + x;  
    end;  
    close(f);  
  
    assign(f, 'output.txt');  
    rewrite(f);  
    writeln(f, 'Сумма чисел ', s);  
    close(f);  
end.
```

логическая функция,
возвращает **True**, если
достигнут конец файла

запись результата в
файл `output.txt`

Задания

В файле `input.txt` записаны числа, сколько их – неизвестно.

- «8»: Найти среднее арифметическое всех чисел и записать его в файл `output.txt`.
- «10»: Найти минимальное и максимальное числа и записать их в файл `output.txt`.

Обработка массивов

Задача: в файле `input.txt` записаны числа (в столбик), сколько их – неизвестно, но не более 100. Переставить их в порядке возрастания и записать в файл `output.txt`.



Можно ли обойтись без массива?

Проблемы:

1. для сортировки надо удерживать в памяти все числа сразу (массив);
2. сколько чисел – неизвестно.

Решение:

3. выделяем в памяти массив из 100 элементов;
4. записываем прочитанные числа в массив и считаем их в переменной N ;
5. сортируем первые N элементов массива;
6. записываем их в файл.

Чтение данных в массив

Глобальные переменные:

```
var A: array[1..100] of integer;  
    f: text;
```

Функция: ввод массива, возвращает число элементов

```
function ReadArray: integer;  
var i: integer;  
begin  
    assign(f, 'input.txt');  
    reset(f);  
    i := 0;
```

```
    while (not eof(f)) and (i < 100) do begin  
        i := i + 1;  
        readln(f, A[i]);  
    end;  
    close(f);
```

```
    ReadArray :=  
        i;  
end;
```

цикл заканчивается, если достигнут конец файла или прочитали 100 чисел

Программа

```
program qq;  
var A: array[1..100] of integer;  
    f: text;  
    N, i: integer;  
function ReadArray: integer;  
    ...  
begin  
    N := ReadArray;  
    { сортировка первых N элементов }  
  
    assign(f, 'output.txt');  
    rewrite(f);  
    for i:=1 to N do  
        writeln(f, A[i]);  
    close(f);  
end;
```

Вывод отсортированного
массива в файл

Задания

В файле `input.txt` записаны числа (в столбик), известно, что их не более 100.

- «8»: Отсортировать массив по убыванию последней цифры и записать его в файл `output.txt`.
- «10»: Отсортировать массив по возрастанию суммы цифр и записать его в файл `output.txt`.

Обработка текстовых данных

Задача: в файле `input.txt` записаны строки, в которых есть слово-паразит «*короче*». Очистить текст от мусора и записать в файл `output.txt`.

Файл `input.txt` :

Мама, короче, мыла, короче, раму.

Декан, короче, пропил, короче, бутан.

А роза, короче, упала на лапу, короче, Азора.

Каждый, короче, охотник желает, короче, знать, где ...

Результат - файл `output.txt` :

Мама мыла раму.

Декан пропил бутан.

А роза упала на лапу Азора.

Каждый охотник желает знать, где сидит фазан.

Обработка текстовых данных

пока не кончились данные

Алгоритм:

1. Прочитать строку из файла (`readln`).
2. Удалить все сочетания "**, короче,**" (`Pos`, `Delete`).
3. Записать строку в другой файл.
4. Перейти к шагу 1.

Обработка строки `s`:

искать «, короче,»

```
repeat
  i := Pos(' , короче , ' , s);
  if i <> 0 then Delete(s, i, 9);
until i = 0;
```

удалить
9 символов

Основная идея:

надо одновременно держать открытыми два файла
(один в режиме чтения, второй – в режиме записи).

Работа с двумя файлами одновременно

```
program qq;  
var s: string;  
    i: integer;  
    fIn, fOut:  
        text;  
begin  
    assign(fIn, 'input.txt');  
    reset(fIn);  
    assign(fOut, 'output.txt');  
    rewrite(fOut);  
    { обработать файл }  
    close(fIn);  
    close(fOut);  
end.
```

файловые
переменные

открыть файл
для чтения

открыть файл
для записи

Полный цикл обработки файла

пока не достигнут конец файла

```
while not eof(fIn) do begin
```

```
  readln(fIn, s);
```

обработка строки

```
  repeat
```

```
    i := Pos(' , короче , ' , s);
```

```
    if i <> 0 then
```

```
      Delete(s, i, 9);
```

```
  until i = 0;
```

```
end;
```

запись «очищенной»
строки

Задания

В файле `input.txt` записаны строки, сколько их – неизвестно.

«4»: Заменить все слова «короче» на «в общем» и записать результат в файл `output.txt`.

«5»: Вывести в файл `output.txt` только те строки, в которых больше 5 слов (слова могут быть разделены несколькими пробелами).

