РАБОТА С ТЕКСТОВЫМИ ФАЙЛАМИ

В PASCAL

ЦЕЛЬ УРОКА: НАУЧИТЬСЯ РАБОТАТЬ С ТЕКСТОВЫМИ ФАЙЛАМИ В PASCAL

- Задачи:

1. Познакомиться с понятием «текстового файла»
2. Изучить процедуры и функции для работы с файлами
3. Применить на практике полученные знания

- Текстовые файлы предназначены для хранения текстовой информации. Именно в таких файлах хранятся, например, исходные тексты программ.

- Компоненты текстовых файлов могут иметь переменную длину, что существенно влияет на характер работы с ними.

ОПИСАНИЕ ФАЙЛОВ

- В разделе var переменные, используемые для работы с файлами (файловые переменные), описываются следующим образом:

- var f1, f2 : Text; {текстовые файлы}

- Файловая переменная не может быть задана константой.

ОПЕРИРОВАНИЕ ФАЙЛАМИ

- Процедура
- Assign(f, '<имя_файла>');
- служит для установления связи между файловой переменной f и именем того файла, за действия с которым эта переменная будет отвечать
- Строка '<имя_файла>' может содержать полный путь к файлу.
- Если путь не указан, файл считается расположенным в той же директории, что и исполняемый модуль программы.

ОПЕРИРОВАНИЕ ФАЙЛАМИ

Reset(f);	— открытие файла для считывания из него информации; если такого файла не существует, попытка открыть его вызовет ошибку и аварийную остановку работы программы. Эта же команда служит для возвращения указателя на начало файла;

ОПЕРИРОВАНИЕ ФАЙЛАМИ

Rewrite(f);	— открытие файла для записи в него информации; если такого файла не существует, он будет создан; если файл с таким именем уже есть, вся содержавшаяся в нём ранее информация исчезнет;

ОПЕРИРОВАНИЕ ФАЙЛАМИ

Append(f);	— открытие файла для записи в него информации (указатель помещается в конец этого файла). Если такого файла не существует, он будет создан; а если файл с таким именем уже есть, вся содержащаяся в нём ранее информация будет сохранена, потому что запись будет производиться в его конец.

ОПЕРИРОВАНИЕ ФАЙЛАМИ

- После того, как ваша программа закончит работу с файлом, очень желательно закрыть его:
- Close(f);
- В противном случае информация, содержащаяся в этом файле, может быть потеряна.

ОПЕРИРОВАНИЕ ФАЙЛАМИ

- Чтение данных из файла, открытого для считывания, производится с помощью команд Read() и ReadLn().

- В скобках сначала указывается имя файловой переменной, а затем — список ввода. Например:
- Read(f, a, b, c);
- ReadLn(f, a, b, c)

ОПЕРИРОВАНИЕ ФАЙЛАМИ

- Считывать из текстового файла можно только переменные простых типов: целых, вещественных, символьных, а также строковых.
- Численные переменные, считываемые из файла, должны разделяться хотя бы одним пробельным символом.

- Типы вводимых данных и типы тех переменных, куда эти данные считываются, обязаны быть совместимыми

ОПЕРИРОВАНИЕ ФАЙЛАМИ

- Считываемые переменные могут иметь различные типы.
- Например, если в файле[3] f записана строка
- 1 2.5 c
- то командой Read(f, a, b, c, c); можно прочитать одновременно значения для трёх переменных, причем все — разных типов.

ОПЕРИРОВАНИЕ ФАЙЛАМИ

- Сохранять переменные в файл, открытый для записи командами Rewrite(f) или Append(f), можно при помощи команд Write() и WriteLn().
- Так же, как в случае считывания, первой указывается файловая переменная, а за ней — список вывода:

- Write(f, a, b, c);
- WriteLn(f, a, b, c);

ОПЕРИРОВАНИЕ ФАЙЛАМИ

- К пробельным символам (присутствующим в файле, но невидимым на экране) относятся:
  - символ горизонтальной табуляции (#9);
  - символ перевода строки (#10) (смещение курсора на следующую строку, в той же позиции);
  - символ вертикальной табуляции (#11);
  - символ возврата каретки (#13) (смещение курсора на начальную позицию текущей строки; в кодировке UNIX один этот символ служит признаком конца строки);
  - символ конца файла (#26);
  - символ пробела (#32).
- **Замечание**: Пара символов #13 и #10 является признаком конца строки текстового файла (в кодировках DOS и Windows).

- Eof(f) — возвращает значение True, если уже достигнут конец файла f (указатель находится сразу за последним элементом файла), и False в противном случае;
- SeekEof(f) — возвращает значение True, если «почти» достигнут конец файла f (между указателем и концом файла нет никаких символов, кроме пробельных), и False в противном случае;
- Eoln(f) — возвращает значение True, если достигнут конец строки в файле f (указатель находится сразу за последним элементом строки), и False в противном случае;
- SeekEoln(f) — возвращает значение True, если «почти» достигнут конец строки в файле f (между указателем и концом строки нет никаких символов, кроме пробельных), и False в противном случае.

ПРАКТИЧЕСКАЯ РАБОТА

- **Задача**. В текстовом файле f.txt записаны (вперемешку) целые числа: поровну отрицательных и положительных. Используя только один вспомогательный файл, переписать в текстовый файл h.txt все эти числа так, чтобы:
  - порядок отрицательных чисел был сохранён;
  - порядок положительных чисел был сохранён;
  - любые два числа, стоящие рядом, имели разные знаки.

ПРАКТИЧЕСКАЯ РАБОТА

- **Решение**
- Если бы нам разрешили использовать два вспомогательных файла, мы бы просто переписали все положительные числа в один из них, а все отрицательные — в другой. А затем объединили бы два этих файла. В нашем же случае придётся переписать во вспомогательный файл только положительные числа. Затем при «сборке» мы будем считывать из вспомогательного файла «всё подряд», а из исходного — только отрицательные числа.

РЕАЛИЗАЦИЯ

```
program z1;
var f, g, h: Text;
  k : Integer;
begin
  Assign(f, 'f.txt');
  Assign(g, 'g.txt');
  Assign(h, 'h.txt');
 { Переписываем положительные числа в доп. файл }
 Reset(f);
 Rewrite(g);
 while not Eof(f) do
 begin
   Read(f, k);
   if k > 0 then
     Write(g, k, ' ');
 end;
```

РЕАЛИЗАЦИЯ

- { Собираем числа в новый файл h.txt }
  Reset(f); { Возвращаем указатель на начало файла f }
  Reset(g);
  Rewrite(h);
  while not Eof(g) **do**
  begin
    Read(g, k);
    Write(h, k, ' ');
    repeat
      Read(f, k)
    until k < 0;
    Write(h, k, ' ');
  end;
  Close(f);
  Close(g);
  Close(h);
  end.