

Файловые типы данных



Память человека есть лист белой бумаги: иногда напишется хорошо, а иногда дурно.

К. Прутков

Компьютерные науки

© М.Л. Цымблер

Содержание

- Классификация файлов
- Основные операции с файлами
- Примеры

Файловые типы данных

© М.Л. Цымблер

2

Классификация файлов

- *Файл* – именованная совокупность взаимосвязанных данных, хранящихся на внешних носителях. Файлы состоят из компонент, называемых *записями*.
- Виды файлов:
 - Типизированные файлы
 - Текстовые файлы
 - Нетипизированные файлы

Файловые типы данных

© М.Л. Цымблер

3

Типизированные файлы

- *Типизированный файл* представляет собой последовательность значений-записей одинакового типа во внутреннем (двоичном) представлении.
Все записи типизированного файла имеют одинаковую длину, определяемую типом значений.
- Определение переменной, представляющей типизированный файл:
<Имя переменной>: **file of** <Тип записей>
<Тип записей> – любой, кроме файлового.

Файловые типы данных

© М.Л. Цымблер

4

Пример: файл записей

type

```
Person = record
  Name: String[10];
  Gender: Char;
  Age: Byte;
end;
```

var

```
PersonFile :
  file of Person;
```

0-я запись

1-я запись

1	2	3	4	5	6	7	8	9	10	11	12	13	1	2	3	4	5	6	7	8	9	10
#6	И	в	а	н	о	в	#0	#0	#0	#0	М	#20	#8	С	и	д	о	р	о	в	а	#0

Файловые типы данных

© М.Л. Цымблер

5

Текстовые файлы

- *Текстовый файл* имеет структуру, стандартную для текстовых файлов ОС Windows.
Записи текстового файла – символьные строки переменной длины, разделенные спец. комбинацией символов, называемой «*конец строки*». Конец строки – последовательность двух управляющих символов: «**перевод каретки**» (символ **#13**) и «**перевод строки**» (символ **#10**, может отсутствовать). Файл завершается специальным кодом «*конец файла*» (символ **#26**).
- Определение переменной, представляющей текстовый файл:
<Имя переменной>: **Text**

Файловые типы данных

© М.Л. Цымблер

6

Пример: текстовый файл

```
var
  PersonFile : Text;
```

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1-я строка	И	в	а	н	о	в	#32	М	#32	2	0	#13	#10		
2-я строка	С	и	д	о	р	о	в	а	#32	Ж	#32	1	9	#13	#10
...															
n-я строка	Я	к	о	в	л	е	в	#32	М	#32	4	0	#26		

Нетипизированные файлы

- *Нетипизированный файл* представляет собой последовательность записей фиксированной длины.
В нетипизированный файл может быть записано значение любого типа, имеющее указанную длину.
- Определение переменной, представляющей типизированный файл:
<Имя переменной>: **file**

Пример: нетипизированный файл

```
const
  RecSize = 13;
```

```
var
  PersonFile : file;
```

0-я запись													1-я запись									
1	2	3	4	5	6	7	8	9	10	11	12	13	1	2	3	4	5	6	7	8	9	10
#6	И	в	а	н	о	в	М	#20	#3	Р	U	S	#8	С	и	д	о	р	о	в	а	Ж

Основные операции с файлами

- **Assign**(<Файловая переменная>, '<Имя файла>')
Назначение файловой переменной дисковый файла с указанным именем.
- **Reset**(<Файловая переменная>)
Открытие файла для чтения.
- **Rewrite**(<Файловая переменная>)
Открытие для записи с предварительной очисткой.
- **Append**(<Файловая переменная>)
Открытие для записи без предварительной очистки (только для текстовых файлов). Новые записи будут добавляться в конец файла.

Основные операции с файлами

- **Close**(<Файловая переменная>)
Закрытие файла.
- **Write**([<Файловая переменная>.] <Список вывода>)
Вывод значений из списка вывода в файл. Если файловая переменная не указана, то вывод происходит в стандартный файл вывода Output (на экран).
- **Read**([<Файловая переменная>.] <Список ввода>)
Ввод значений из файла в переменные, указанные в списке ввода. Если файловая переменная не указана, то ввод происходит из стандартного файла ввода Input (с клавиатуры).

Основные операции с файлами

- **WriteLn, ReadLn**
Определены только для текстовых файлов. Аналогичны Write и Read, но после операции ввода-вывода производят смену строки в файле.
- **Eoln**(<Файловая переменная>)
Выдает True, если достигнут конец строки (только для текстовых файлов).
- **Eof**(<Файловая переменная>)
Выдает True, если достигнут конец файла.

Пример 1: работа с текстовым файлом

<pre>{ Создание копии текстового файла с нумерованными строками. } var Inp, Out: Text; S: String; L: LongInt; const InpFName = 'input.txt'; OutFName = 'output.txt' begin Assign(Inp, InpFName); Reset(Inp);</pre>	<pre>Assign(Out, OutFName); Rewrite(Out); L := 0; while not Eof(Inp) do begin ReadLn(Inp, S); L := L + 1; WriteLn(Out, L:4, S); end; Close(Inp); Close(Out); end.</pre>
--	---

Пример 2: работа с текстовым файлом

<pre>{ Подсчет количества символов в заданном файле. } var F: Text; S, Name: String; Sum: LongInt; begin Write('Введите имя файла ==> '); ReadLn(Name);</pre>	<pre>Assign(F, Name); Reset(F); Sum := 0; while not Eof(F) do begin ReadLn(F, S); Sum := Sum + Length(S); end; Close(F); WriteLn(Name, ' содержит ', Sum, ' символов.');</pre>
--	--

Вопросы для обсуждения

- От чего зависит выбор программистом типа файла (типизированный, текстовый или нетипизированный) при обработке конкретного файла на диске: имя, расширение файла, др.?
