

Типизированные файлы

Наумов Д.А., доц. каф. КТ, ИТГД

Программирование и алгоритмические языки, 2020

Содержание лекции

1 Файлы и файловый тип данных

2 Операции с файлами

- Процедура Assign
- Процедуры Rename, Erase
- Процедуры открытия файла
- Процедура закрытия файла
- Чтение и запись в файл
- Проверка достижения конца файла
- Подпрограммы работы с типизированными файлами
- Контроль ошибок ввода-вывода

Файл

именованная сущность, для которой определены операции ввода и вывода данных.

Для работы с файлами в Pascal предусмотрены **файловые типы данных**:

- 1 типизированные: информация считывается и записывается в переменные конкретного типа (целые, вещественные, массивы и т. д.);
- 2 нетипизированные: информация считывается и записывается блоками определённого размера.
- 3 текстовые: информация обрабатывается посимвольно, но возможно чтение и запись данных в переменные строкового, целого и вещественного типа.

В данной лекции будут рассматриваться **типизированные файлы**.

Описание переменной файлового типа

```
1 type
2   TElem = real;
3   TPersonInfo = record
4     Name: string[25];
5     Age: integer;
6     Birth: record
7       Day: 1..31; Month: 1..12; Year: integer;
8     end;
9   end;
10  TValueFile = file of TElem;
11  TPersonFile = file of TPersonInfo;
12  // описание переменных файлового типа
13  var
14    F1: TValueFile; //тип элементов - TElem, т.е. real
15    F2: TPersonFile; //тип элементов - TPersonInfo
16    F3: file of integer; //тип элементов - integer
```

Assign

Процедура Assign

связывает файловую переменную с файлом.

Синтаксис: Assign(f, s)

- f - переменная файлового типа;
- s - строка, полный или частичный путь к файлу.

```
1 var
2   F1, F2: file of real;
3 begin
4   Assign(F1, 'myfile.dat');
5   Assign(F2, 'c:\ work\ myfile.dat');
```

Rename, Erase

Процедура Rename

переименовывает или перемещает не открытый файл.

Синтаксис: `Rename(f, s)`

- `f` - переменная файлового типа;
- `s` - строка, полный или частичный путь к файлу.

Процедура Erase

удаляет не открытый файл.

Синтаксис: `Erase(f)`

- `f` - переменная файлового типа.

Открытие файла

После установления связи между файловой переменной и именем файла на диске нужно открыть файл, воспользовавшись процедурами `reset`, или `rewrite`.

Процедура `Reset`

открывает файл для чтения.

Синтаксис: `Reset(f)`

- `f` - переменная файлового типа.
- файл должен существовать (иначе - ошибка ввода-вывода);
- текущей позицией для чтения становится начало файла.

```
1 var
2   F1: file of real;
3 begin
4   Assign(F1, 'myfile.dat');
5   Reset(F1);
```

Открытие файла

Процедура Rewrite

открывает файл для записи.

Синтаксис: Rewrite(f)

- f - переменная файлового типа.
- содержимое файла уничтожается;
- текущей позицией для чтения становится начало файла.

```
1 var
2   F1: file of real;
3 begin
4   Assign(F1, 'myfile.dat');
5   Rewrite(F1);
```

Указатель текущей позиции - место в файле, с которого будет осуществляться процедура чтения или записи.

Закрытие файла

Процедуры записи в файл записывают информацию в буфер. После того как буфер заполнится, вся информация из него переносится в файл. При выполнении процедуры *Close* сначала происходит запись буфера файла на диск, и только потом файл закрывается.

Процедура *Close*

закрывает файл.

Синтаксис: *Close(f)*, *f* - переменная файлового типа.

```
1 var F1: file of real;  
2 begin  
3   Assign(F1, 'myfile.dat');  
4   Rewrite(F1);  
5   Write(F1, Pi);  
6   Close(F1);  
7 end.
```

Чтение из файла

Процедура Read

выполняет чтение информации из файла.

Синтаксис: `Read(f, x1, x2, x3, ..., xn);`

- `f` - переменная файлового типа;
- `x1, x2, x3, ..., xn` - переменные.

Для типизированных файлов тип переменной должен совпадать с типом компонент файла.

```
1 var F1: file of real;
2 var v: real;
3 begin
4   Assign(F1, 'myfile.dat');
5   Reset(F1);
6   Read(F1, v); //считываем одно число
7   Write(F1, v); //выведем его на экран
8   Close(F1);
```

Запись в файл

Процедура Write

выполняет запись информации в типизированный файл.

Синтаксис: `Write(f, x1, x2, x3, ..., xn);`

- `f` - переменная файлового типа;
- `x1, x2, x3, ..., xn` - переменные.

Для типизированных файлов тип переменной должен совпадать с типом компонент файла.

```
1 var F1: file of real;  
2 var i: integer;  
3 begin  
4   Assign(F1, 'myfile.dat');  
5   Rewrite(F1);  
6   for i := 1 to 10 do  
7     Write(F1, random());  
8   Close(F1);
```

Функция Eof

возвращает True, если при чтении из файла был достигнут конец файла.

Синтаксис: Eof(f)

- f - переменная файлового типа

Пример: чтение всех данных из файла.

```
1 var F1: file of real;
2 var v: real;
3 begin
4   Assign(F1, 'myfile.dat');
5   Reset(F1);
6   while not EOF(F1) do
7     begin
8       Read(F1, v);
9       Writeln(v);
10    end;
11   Close(F1);
```

Функция Filesize(f)

возвращает для открытого типизированного файла **f** количество его компонент.

Функция Filepos(f)

возвращает текущую позицию (чтения, записи) в открытом файле **f**.

Процедура Seek(f, k)

устанавливает указатель в открытом файле, связанном с файловой переменной **f**, на компонент с номером **k** (нумерация компонентов идет от 0).

Процедура Truncate(f)

отсекает часть открытого файла, начиная с текущего компонента, и подтягивает на его место конец файла.

Контроль ошибок

При открытии файла и чтении данных осуществляет контроль ошибок ввода-вывода. Если произошла ошибка (открываемый для чтения файл не существует), то проверка может осуществляться через обращение к функции `IOResult`.

```
1 Assign(F, 'unexist.file');
2 {$I+}
3 Reset(F);
4 {$I-}
5 if IOResult = 0
6 //файл существует
7 else
8 //произошла ошибка ввода-вывода
```

Коды ошибок операций ввода-вывода

Код ошибки	Описание
2	файл не найден
3	путь не найден
4	слишком много открытых файлов
5	отказано в доступе
12	неверный режим доступа
15	неправильный номер диска
16	нельзя удалять текущую директорию
100	ошибка при чтении с диска
101	ошибка при записи на диск
102	не применена процедура AssignFile
103	файл не открыт
104	файл не открыт для ввода
105	файл не открыт для вывода
106	неверный номер
150	диск защищён от записи