

Файлы

Наумов Д.А., доц. каф. КТ, ИТГД

Алгоритмические языки и программирование, 2019

Содержание лекции

1 Файлы и файловый тип данных

2 Операции с файлами

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.

Описание переменной файлового типа

```
1 type
2   TElem = real;
//описание файловый типов данных
3   TTypedFile = file of TElem; //типизированный
4   TTextFile = text; //текстовый
5   TUntypedFile = file; //нетипизированный
//описание переменных файлового типа
6 var
7   F1: TTypeFile;
8   F2: TTextFile;
9   F3: TUntypedFile;
11  F4: file of integer;
12  F5: text;
13  F6: file;
```

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\ pascal\ 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`, `rewrite` или `append`.

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);
```


Открытие файла

Процедура Append

открывает файл для записи.

Синтаксис: Append(f)

- f - переменная текстового файла;
- файл должен существовать;
- текущей позицией для чтения становится конец файла.

```
1 var
2   F1: text;
3 begin
4   Assign(F1, 'myfile.txt');
5   Append(F1);
```

Заккрытие файла

Процедуры записи в файл записывают информацию в буфером. 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);
```

Чтение из файла

Процедура Read

выполняет чтение информации из текстового или типизированного 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.

Для текстового файла можно считывать значения символьного, целого, вещественного, строкового типов (и отрезков символьного и целого типов).

Запись в файл

Процедура Write

выполняет запись информации в текстовый или 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.

Для текстового файла можно записывать значения символьного, целого, вещественного, строкового типов (и отрезков символьного и целого типов).

Функция Eof

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

Синтаксис: Eof(f)

- f - переменная файлового типа

Процедура Readln

выполняет чтение из текстового файла до символа конца строки.

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

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

Процедура Writeln

выполняет запись в текстовый файл и добавляет символ конца строки.

Подпрограммы работы с типизированными файлами

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.

Подпрограммы работы с нетипизированными файлами

Процедуры Reset и Rewrite имеют расширенный синтаксис:

```
Reset(var f: File; BuferSize: word);  
Rewrite(var f: File; BuferSize: word) ;
```

Процедура BlockWrite

запись данных в нетипизированный файл.

Синтаксис: BlockWrite(var f: file; var X; Count: word; var WriteCount: word);

Процедура BlockRead(f)

чтение данных из нетипизированного файла.

Синтаксис: BlockRead(var f: file; var X; Count: word; var ReadCount: word);

- f - переменная файлового типа;
- x - переменная произвольного типа;

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

Варианты заданий

Задание на лабораторную работу включает в себя задание на работу с текстовыми (Часть 1) и с типизированными файлами (Часть 2).

Часть 1: доработать программу к лабораторной работе по процедурам и функциям, осуществив чтение данных из текстового файла и запись результатов в текстовый файл.

Часть 2:

- Вариант 1: определить, является ли последовательность чисел, записанных в файл, монотонной (возрастающей или убывающей).
- Вариант 2: осуществить циклический сдвиг (влево или вправо) чисел, записанных в текстовый файл, на n позиций.
- Вариант 3: определить, встречается ли в файле последовательность чисел x_1, x_2, \dots, x_n .
- Вариант 4: даны два отсортированных файла. Создать третий файл, который будет содержать отсортированные значения из первых двух файлов (выполнить сортировку слиянием).

Варианты заданий

- Вариант 5: разделить последовательность чисел, записанных в файл, на файлы, содержащие только отрицательные, только положительные и только нулевые элементы.
- Вариант 6: удалить из файла числа, выходящие за пределы трех СКО (среднеквадратичных отклонений) от среднего значения.
- Вариант 7: выполнить сортировку чисел, записанных в файл, методом выбора минимального элемента. (Ищется минимальный элемент и меняется местом с первым. В оставшейся части файла снова ищется минимальный элемент и меняется местами со вторым и т.д.)
- Вариант 8: в файле определить длину максимальной последовательности одинаковых чисел;

Варианты заданий

- Вариант 9: в файле неповторяющихся чисел определить медианное значение (такое значение, что половина чисел больше данного, а половина - меньше). Если количество чисел четное, рассчитать полусумму двух центральных значений;
- Вариант 10: определить длину максимальной последовательности чисел, которая содержится в обоих файлах.