

Продолжение темы:
**РАЗРАБОТКА И ОТЛАДКА ПРОГРАММ С ИСПОЛЬЗОВАНИЕМ
СТРУКТУРИРОВАННЫХ ТИПОВ ДАННЫХ – ФАЙЛОВ**

Типы данных, которые рассмотренные на прошлых занятиях, обладают одним существенным недостатком. Они не приспособлены для долговременного хранения информации. Можно внести в программу множество различных данных (с клавиатуры в процессе выполнения программы), однако, после закрытия программы все эти данные исчезнут. При повторном запуске программы нам по новой придётся вводить информацию о цвете автомобиля, его марке, пробеге, дате выпуска и т.д. Если необходимо ввести информацию о нескольких сотнях автомобилях? Придется каждый раз при новом запуске программы вводить всю информацию по-новому. Для долговременного хранения информации на компьютере в языке программирования Pascal используют файловый тип.

Файлом называют область во внешней памяти компьютера, которая обладает определёнными признаками:

- файл имеет имя;
- размер файла ограничен лишь ёмкостью устройства внешней памяти;
- файл содержит компоненты одного типа.

Файлы бывают 3-х видов: текстовые файлы, типизированные файлы и нетипизированные файлы.

Введение файлового типа в язык Паскаль вызвано необходимостью обеспечить возможность работы с периферийными (внешними) устройствами ЭВМ, предназначенными для ввода, вывода и хранения данных.

Файловый тип данных или файл определяет упорядоченную совокупность произвольного числа однотипных компонент.

Понятие файла достаточно широко. Это может быть обычный файл на диске, коммуникационный порт ЭВМ, устройство печати, клавиатура или другие устройства.

При работе с файлами выполняются операции ввода - вывода. Операция ввода означает перепись данных с внешнего устройства (из входного файла) в основную память ЭВМ, операция вывода - это пересылка данных из основной памяти на внешнее устройство (в выходной файл).

Файлы на внешних устройствах часто называют физическими файлами. Их имена определяются операционной системой. В программах на языке Паскаль имена файлов задаются с помощью строк. Например, имя файла на диске может иметь вид:

```
'LAB1.DAT'  
'c:\ABC150\pr.txt'  
'my_files'
```

Типы файлов Паскаль

Паскаль поддерживает три файловых типа:

- текстовые файлы;

- типизированные файлы;
- нетипизированные файлы.

Доступ к файлу в программе происходит с помощью переменных файлового типа. Переменную файлового типа описывают одним из трех способов:

- file of тип - типизированный файл (указан тип компоненты);
- text - текстовый файл;
- file - нетипизированный файл.

Примеры описания файловых переменных:

```
var  
f1: file of char;  
f2: file of integer;  
f3: file;  
t: text;
```

Стандартные процедуры и функции

Любые дисковые файлы становятся доступными программе после связывания их с файловой переменной, объявленной в программе. Все операции в программе производятся только с помощью связанной с ним файловой переменной.

Assign(f, FileName)

связывает файловую переменную f с физическим файлом, полное имя которого задано в строке FileName. Установленная связь будет действовать до конца работы программы, или до тех пор, пока не будет сделано переназначение.

Пример процедуры Assign в Паскале

```
Assign (chf, 'G:\Home\ Student\ Lang\ Pascal\ primer.dat');
```

После связи файловой переменной с дисковым именем файла в программе нужно указать направление передачи данных (открыть файл). В зависимости от этого направления говорят о чтении из файла или записи в файл.

Reset(f)

открывает для чтения файл, с которым связана файловая переменная f. После успешного выполнения процедуры Reset файл готов к чтению из него первого элемента. Процедура завершается с сообщением об ошибке, если указанный файл не найден.

Если f - типизированный файл, то процедурой reset он открывается для чтения и записи одновременно.

Rewrite(f)

открывает для записи файл, с которым связана файловая переменная f. После успешного выполнения этой процедуры файл готов к записи в него первого элемента. Если указанный файл уже существовал, то все данные из него уничтожаются.

Close(f)

закрывает открытый до этого файл с файловой переменной f. Вызов процедуры Close необходим при завершении работы с файлом. Если по какой-то причине процедура Close не будет выполнена, файл все-же будет создан на внешнем устройстве, но содержимое последнего буфера в него не будет перенесено.

EOF(f): boolean

возвращает значение TRUE, когда при чтении достигнут конец файла. Это означает, что уже прочитан последний элемент в файле или файл после открытия оказался пуст.

Rename(f, NewName)

позволяет переименовать физический файл на диске, связанный с файловой переменной f. Переименование возможно после закрытия файла.

Erase(f)

уничтожает физический файл на диске, который был связан с файловой переменной f. Файл к моменту вызова процедуры Erase должен быть закрыт.

IOResult

возвращает целое число, соответствующее коду последней ошибки ввода - вывода. При нормальном завершении операции функция вернет значение 0. Значение функции IOResult необходимо присваивать какой-либо переменной, так как при каждом вызове функция обнуляет свое значение. Функция IOResult работает только при включенном режиме проверок ошибок ввода - вывода или с ключом компиляции {\$I-}.