Практическое занятие

РАЗРАБОТКА И ОТЛАДКА ПРОГРАММ С ИСПОЛЬЗОВАНИЕМ СТРУКТУРИРОВАННЫХ ТИПОВ ДАННЫХ – ФАЙЛОВ

Цель работы: организация вычислительных процессов с использованием внешних файлов, которые в программе ассоциируются с помощью структурированных файловых или "буферных" переменных.

Общие сведения

Файл представляет собой именованный набор данных одного типа на внешнем носителе. Файл используется для размещения исходных данных, результатов вычислений, а также и текста исходной программы. Внешние носители - жёсткий диск, лазерный диск (CD). Удобство использования файла заключается в возможности сохранения информации пользователем.

Файлы объявляются либо в подразделе VAR, либо в подразделе TYPE раздела описаний Паскаль-программы, а именно:

Var имя: file of базовый тип;

или

TYPE имя типа = FILE OF базовый тип;

Var имя: имя типа;

Обмен данными между основной программой и файлом на внешнем носителе осуществляется через файловую или буферную переменную. В описании "имя" есть имя файловой переменной. В Паскале различают текстовые, типизированные и нетипизированные файлы. В данной работе речь идёт только о типизированных и текстовых файлах.

Текстовые файлы - это файлы последовательного доступа, т.е. для того, чтобы обратиться к какой-либо компоненте файла, необходимо перебрать все расположенные до неё; длина компоненты такого файла меняется. **Типизированные файлы** (описание приведено выше) - файлы прямого доступа, т.е. к любой компоненте файла можно обратиться непосредственно, поскольку длина каждой компоненты файла известна и задаётся типом буферной переменной. Таким образом, в текстовых файлах размер компоненты меняется, а для типизированных файлов он постоянен и определяется базовым типом. Описание текстового файла имеет вид:

VAR имя: TEXT;

В Паскале существуют два особых файла со стандартными именами INPUT (ввод) и OUTPUT (вывод). Оба они имеют тип TEXT. При вводе-выводе информации через стандартное устройство (экран дисплея) они формируются автоматически.

Существуют два вида взаимодействия с файлами: чтение и запись информации. Перед работой необходимо связать файловую переменную F с файлом по имени ”NAME” при помощи оператора

ASSIGN (F,’NAME’);

а после окончания работы файл закрыть

CLOSE (F);

причем имя файла может быть полным, т.е. указывается не только имя файла, но и полный путь к нему.

Чтение из файла данных:

RESET (F);

READ (F, список переменных программы для чтения, ввода);

Запись в файл данных

REWRITE (F);

WRITE (F, список переменных программы, для записи, вывода);.

Текстовые файлы открываются либо только для чтения, либо только для записи, типизированные файлы можно открывать одновременно и для записи, и для чтения. Помните, что файл, который необходимо открыть в программе, должен существовать на внешнем носителе. Кроме того, если вы пытаетесь открыть уже существующий типизированный файл с информацией оператором REWRITE (F); , то после его открытия вся информация будет стёрта.

Функции и процедуры работы с файлами

1. Типизированные файлы

Процедура ERASE(VAR F) – стирает внешний файл, ассоциированный с буферной переменной F;

процедура RENAME(VAR F, 'NEWNAME') – переименовывает внешний файл;

функция EOF(VAR F): BOOLEAN – имеет значение TRUE, если текущая позиция в файле находится за последним символом файла или если в файле нет компонент, в противном случае значение функции FALSE, т.е. функция определяет признак конца файла;

2. Текстовые файлы

Процедура APPEND(VAR F) – открывает существующий внешний текстовый файл для добавления информации, текущий указатель файла устанавливается в его конец;

функция EOLN(VAR F):BOOLEAN – определяет признак конца строки;

функция FILEPOS(VAR F):LONGINT – возвращает (определяет) текущую позицию указателя в файле;

функция FILESIZE(VAR F):LONGINT –возвращет (определяет) текущий размер файла;

процедура SEEK(VAR F, N:LONGINT) – устанавливает текущую позицию указателя файла на указанный элемент N;

процедура TRUNCATE(VAR F) – усекает размер файла до текущей позиции указателя в файле.

Примеры

1. Создайте типизированный файл для записи данных вещественного типа и считайте из уже существующего в текущем каталоге текстового файла данные целого типа.

```
PROGRAM USER11;
VAR X1, X2, X3 :REAL;
I, I1, I2, I3 :INTEGER;
XR : FILE OF REAL;
YR :TEXT;
BEGIN
    ASSIGN (XR, 'X.DAT'); {файл для записи }
    ASSIGN (YR,'Y.DAT'); { файл для чтения }
    WRITELN('ВВЕДИТЕ ТРИ ПЕРЕМЕННЫЕ');
    READ (X1, X2, X3);
    REWRITE(XR); WRITE (XR,X1,X2,X3); {запись в файл X.DAT}
    RESET (YR);
    READ(YR, I1, I2, I3); {чтение из файла Y.DAT}
    I := I1 + I2 + I3;
    WRITELN ('I1= ',I1:10,'I2 = ',I2:10,'I3 = ',I3:10, 'I=',I :10);
    CLOSE (XR);
    CLOSE (YR);
END.
```

2. Прочитайте из предварительно созданного в рабочем каталоге файла pr1.txt информацию, представленную в следующем виде:

```
мама  49
папа   50
сын 18
дочь 15
```

После чего к каждому году в программе прибавьте два года и запишите новую информацию в файл pr2.txt в следующем виде:

```
51 мама
52 папа
20 сын
17 дочь
```

```
program pr1;
var r1,r2:text;
s,s1,s2:string;
d,d1,code:integer;
begin
assign(r1,'pr1.txt'); assign(r2,'pr2.txt');
reset(r1); rewrite(r2);
while not eof(r1) do begin
while not eoln(r1) do begin
readln(r1,s);
s2:=copy(s,7,2); val(s2,d,code); writeln(s2,d);
delete(s,5,9); writeln(s); d1:=d+2;
writeln(r2,' ',d1,' ',s);
```

```
end;end;
close(r1); close(r2);
end.
```

3. Считайте из предварительно созданного в рабочем каталоге файла pr3.txt информацию, представленную в следующем виде:

```
12
14
34
56
234
-123
```

и к каждому считанному значению d прибавьте sin(pi/3).

```
program pr2;
var r1:text;
d:integer; d1:real;
begin
assign(r1,'pr3.txt'); reset(r1);
while not eof(r1) do begin
readln(r1,d); d1:=d+sin(pi/3);
writeln(' ',d,' ',d1); end; close(r1); end.
```