Файлы

- Работа с файлами на паскале выполняется с использованием файловой переменной, которая имеет файловый тип
- Файловый тип для типизированных файлов вводится следующим образом:
- Type T={любой тип, кроме файлового};
    - FT = file of T; {T-базовый тип файлового типа FT}

Процедуры для файлов

- f:FT; {f – файловая переменная}
- Assign(f,stroka); {связывание файловой переменной с конкретным файлом на внешнем устройстве, stroka – константа, выражение или переменная типа string}
- Например,
- Assign(f,’data.dat’);
- Assign(f,’c:/tmp/t1.pas’);

Процедуры для файлов

- Запись в файл
- Rewrite(f); {открытие файла на запись}
- Write(f,x); {заись значения переменной типа T в файл, связанный с файловой переменной f}
- Close(f); {освобождение файловой переменной и закрытие файла}

Процедуры для файлов

- Чтение из существующего файла
- Reset(f); {открытие файла на чтение}
- Read(f,x); {чтение значения типа T для переменной x}
- Функция eof(f) принимает два значения: истина – если прочитан «конец файла», ложь – в противном случае.

Close(f); {освобождение файловой переменной и закрытие файла}

Пример

- Написать программу чтения целых чисел с клавиатуры и запись их файл. После записи («0» - признак конца ввода) прочитать записанные в файл числа и распечатать их на экране.

Пример

```
Program example;
Type T=integer; FT=file of T;
Var f:FT; i:T;
Begin
Assign(f,'integer.num'); Rewrite(f);
Read(i);
While i<>0 do
 begin
    write(f,i); read(i);
 end;
Reset(f);
While not eof(f) do
  begin
    read(f,i); writeln(i:5);
  end;
Close(f);
End.
```

Текстовые файлы

- Это файлы, компонентами которых являются символы. Эти файлы в отличие от типизированных (двоичных) можно просматривать текстовым редактором, например Блокнотом.
- Текстовые файлы отличаются от обычных тем, что они делятся на строки и имеют тип **text**.
- Каждая строка заканчивается специальным символом – признаком конца строки, который заносится с помощью оператора writeln(f).

Текстовые файлы

- Для обнаружения признака конца строки используется функция eoln(f), принимающая значение «истина», если из файла считан символ, за которым стоит признак конца строки, и «ложь» в противном случае.
- Текстовые файлы обрабатываются с помощью тех же процедур и функций, что и типизированные файлы.

Текстовые файлы

- Дополнительно используются Readln(f,stroka); Writeln(f,stroka);
- Append(f); {открытие файла на добавление новых записей}

Нетипизированные файлы

- Нетипизированные файлы – совокупность символов или битов.
- Для описания нетипизированного файла используется зарезервированное слово **FILE**.
- Для работы с типизированными файлами кроме процедур Assign, Rewrite, Reset, Close используются BlockREad и BlockWrite.

Нетипизированные файлы

- Нетипизированные файлы применяются при разработке высокоскоростных утилит считывания, копирования и других операций, требующих высоких эксплуатационных характеристик.

Динамические структуры данных

- Статические программные объекты – переменные различных типов, которые порождаются непосредственно перед выполнением программы, существуют в течение всего времени ее выполнения и размер (в байтах) которых не меняется по ходу выполнения программы.

Динамические структуры данных

- Динамические программные объекты – объекты, которые порождаются в процессе выполнения программы, существуют в течение всего части времени ее выполнения и размер (в байтах) которых может меняться по ходу выполнения программы.
- Компоненты таких объектов на некотором уровне детализации представляют собой статические объекты, т.е. они принадлежат к одному из основных типов данных.

Ссылочная реализация

- Динамические структуры данных в современных языках программирования высокого уровня реализуются с использованием переменных ссылочного типа.
- Значением этого типа является ссылка на какой-либо программный объект, по которой осуществляется доступ к этому объекту.

Ссылочная реализация

- На машинном языке такая ссылка представляется указанием места в оперативной памяти (адресом) соответствующего объекта.
- Для описания действий над динамическим объектом сопоставляется статическая переменная ссылочного типа, с помощью которой можно создавать или уничтожать динамическую структуру данных, изменять ее размер.

Ссылочная реализация

- <задание ссылочного типа>::=^<имя типа>
- Type pointer=^integer;
- next=^char;

Описание переменных ссылочного типа:

Var a:pointer; b:next; p:^integer; q:^char;

У всех этих переменных есть общая черта – их значения указывают место в памяти соответствующего динамического объекта.

Поэтому переменные ссылочного типа часто называют указатели.

Ссылочная реализация

- Пустая ссылка nil
- p:=nil; q:=nil;
- Процедура new(<переменная ссылочного типа>) используется для порождения динамического объекта.
- Var p:^integer;
- New(p);
- Переменная с указателем p^ - динамический объект целого типа.

Ссылочная реализация

- Процедура dispose(<переменная ссылочного типа>) используется для уничтожения динамической переменной (переменной с указателем).
- Dispose(p);