

Информационные технологии

Лекция №8



Записи

Тип запись

- является структурированным типом данных
- переменные составлены из нескольких частей
- В Pascal существует возможность объединить в одну переменную данные разных типов

Пример

Пусть в переменной требуется хранить сведения о некотором человеке: ФИО, пол, адрес, телефон. Тогда для хранения этих данных будет удобен такой тип:

```
Type
  tPerson = record
    Name, Surname, SecondName: string[30];
    Male: boolean;
    Address: string[50];
    Phone: string[11];
  end;
```

Объявление переменной типа запись выполняется с помощью var.

Части записи называются полями.

Обращение к полю записи в программе производится с помощью знака '.'

Пример обращения к полям:

```
var
  emp: tPerson;
  ...
begin
  ...
  emp.Surname:='Иванов';
  emp.Name:='Иван';
  emp.SecondName:='Иванович';
  ...
end.
```

В случаях, когда приходится много раз обращаться к полям одной и той же записи, можно воспользоваться ключевым оператором `with`, который упрощает ссылку к структурированным переменным:

```
with <имя_записи> do <оператор>;
```

Пример:

```
with emp do
  begin
    Surname:=' Иванов';
    Name:='Иван';
    SecondName:='Иванович';
    ...
  end;
```

Записи можно включать в состав более сложных переменных, например массивов и других записей:

```
var  
  tStaff : array [1..30] of tPerson;
```

Работа с файлами

Определение

Файл (англ. file) — именованная область данных на носителе информации

В паскале работа с файлами осуществляется через специальные файловые типы
Файловые типы определяют тип файла, то есть указывают его содержимое.

С помощью этой переменной, которой присвоен необходимый тип, и осуществляется вся работа с файлами - открытие, запись, чтение, закрытие и т.п.

Порядок действий

1. Создание (описание) файловой переменной;
2. Связывание этой переменной с конкретным файлом на диске или с устройством ввода-вывода (экран, клавиатура, принтер и т.п.);
3. Открытие файла для записи либо чтения;
4. Действия с файлом: чтение либо запись;
5. Закрытие файла.

В Pascal имеется три типа файловых переменных:

1. `text` - текстовый файл
2. `file of _любой_тип_` - "типизированные" файлы
3. `file` - нетипизированный файл.

text - текстовый файл

Из переменной такого типа мы сможем читать строки и символы

```
Var
  txt: text;
  s: string;
Begin
  Assign(txt, 'war_and_peace.txt');
  Reset(txt);
  While not Eof(txt) do
    Begin
      readln(txt, s);
      writeln(s);
    End;
  Close(txt);
End.
```

Фактически, текст - это файл, состоящий из строк (разной длины)

Типизированные файлы

file of `_любой_тип_` - так называемые "типизированные" файлы, то есть файлы, имеющие тип. Этот тип определяет, какого рода информация содержится в файле и задается в параметре `_любой_тип_`.

Например:

```
F: file of integer;
```

Файл F содержит числа типа `integer`; Соответственно, читать из такого файла можно только переменные типа `integer`, ровно как и писать.

Пример

```
type
  A = record
    I,J: Integer;
    S: String[20];
  end;
var
  F: File of A;
```

Нетипизированные файлы

Когда мы указываем в качестве типа файла просто File, то есть без типа:

```
F: File;
```

То получаем "нетипизированный" файл, чтение и запись в который отличается от работы с файлами других типов.

Эти действия производятся путем указания количества байт, которые нужно прочитать, а также указанием области памяти, в которую нужно прочитать эти данные.

Связывание переменной с файлом

Выполняется одинаково для всех типов файлов:

```
assign(<переменная_файлового_типа>, '<путь к файлу>');
```

В качестве параметров задаются переменная любого файлового типа и строка - путь к файлу:

```
var
  T: text;
  F1: file of integer;
  F2: file;
begin
  Assign(T, 'text.txt');
  Assign(F1, 'int.txt');
  Assign(F2, 'file1.dat');

end.
```

Открытие файла

Режим использования: запись или чтение

```
Reset (<любая_файловая_переменная>);
```

Открывает файл на чтение. В качестве параметра - файловая переменная любого из перечисленных выше типов. Это может быть текстовый, типизированный либо не типизированный файл. В случае с текстовым файлом, он открывается только на чтение. В случае с типизированным и нетипизированным файлом - он открывается на чтение и запись.

```
Append(T: Text);
```

Эта процедура открывает текстовый файл (только текстовый!) на запись. Запись производится в конец файла (добавляются строки)

```
ReWrite(F);
```

создает новый файл либо перезаписывает существующий.

Необходимо быть осторожным при использовании этой процедуры, т.к. файл, открытый таким образом будет полностью перезаписан!

Заккрытие файла

Заккрытие файла производится с помощью процедуры `Close(F)`, где `F` - это переменная файлового типа. Эта процедура одна для всех типов файлов.

```
Var
  f: file;
Begin
  Assign(f, 'file.dat');
  reset(f);
  close(f);
end.
```

Чтение файлов

Чтение файлов производится с помощью процедур `read` и `readln`. Они используются также, как и при чтении информации с клавиатуры.

```
read(F, C);
```

где `F` - дескриптор файла, `C` - переменная (`char`, `string` - для текстовых, любого типа - для типизированных файлов).

```
var
  F: file of Integer;
  I: Integer;
Begin
  Assign(F, 'int.dat');
  Reset(F);
  read(F, I);
  Close(F);
end.
```

Запись в файлы

Запись в файлы производится с помощью процедур `write` и `writeln`.

```
write(F, S);
```

где `F` - дескриптор, `S` - переменная.

Переменная должна соответствовать типу файла.

```
program cat;
```

```
var
```

```
  f: text;
```

```
  c: char;
```

```
begin
```

```
  assign(f, 'prog.pas');
```

```
  rewrite(f);
```

```
  writeln(f, 'program prog;');
```

```
  close(f);
```

```
end.
```

Важные функции

В программе необходимо определить, дошёл ли указатель файла до конца строки или до конца файла. В этом случае полезно использовать такие функции:

```
eoln(f: text): boolean;  
eof(f: text): boolean;
```

Первая принимает значение true (истина), если указатель стоит на конце строки, вторая - то же самое для конца файла.

```
If eof(f) then  
  writeln('конец файла');
```

Чтение из файлов без типа

связывания файловой переменной с внешним файлом и его открытие ничем чем отличаются от обычного порядка действий.

```
var
F: File;
begin
Assign(F, '1.dat');
Reset(F);
end.
```

Чтение производится с помощью процедуры blockread:

```
blockread(F: file, Buf: var, size: word, result: word)
```

F: file; - переменная типа file; Именно из этой переменной и происходит чтение данных.

Buf: var; - переменная любого типа. В эту переменную помещаются прочитанные данные.

Size: word; - количество считываемых байт.

Result: word; - в эту переменную помещается реальное количество байт, которые были прочитаны.

Запись в файлы без типа

```
BlockWrite(F: File, Buf: Var, Size: Word, Result: Word)
```

```
F: File; //переменная типа File;
```

```
Buf: Var; // переменная любого типа. Начиная с этой переменной, данные будут записываться в файл.
```

```
Size: Word; // количество записываемого блока данных в байтах.
```

```
Result: Word; // в эту переменную помещается реальное количество байт, которые были записаны.
```

```
var
  Fin, fout : file;
  NumRead, NumWritten : word;
  Buf : Array[1..2048] of byte;
  Total : longint;
begin
  Assign (Fin, Paramstr(1));
  Assign (Fout, Paramstr(2));
  Reset (Fin, 1);
  Rewrite (Fout, 1);
  Total:=0;
  Repeat
    BlockRead (Fin, buf, Sizeof(buf), NumRead);
    BlockWrite (Fout, Buf, NumRead, NumWritten);
    inc(Total, NumWritten);
  Until (NumRead=0) or (NumWritten<>NumRead);
  Write ('Copied ', Total, ' bytes from file ', paramstr(1));
  Writeln (' to file ', paramstr(2));
  close(fin);
  close(fout);
end.
```