Структура программы

Общие сведения

В общем случае программа имеет вид:

{ описательная часть }

begin

{ исполнительная часть }

end

Заголовок

В начале программы может находиться заголовок, состоящий из зарезервированного слова program, имени программы и параметров, с помощью которых программа взаимодействует со своим внешним окружением.

```
program ИмяПрограммы;
```

Разделы программы

{ заголовок }

{ описательная часть }

- раздел подключаемых библиотечных модулей;
- раздел объявления меток;
- раздел объявления констант;
- раздел объявления типов;
- раздел объявления переменных;
- раздел объявления процедур и функций

{ исполнительная часть}

раздел инструкций (операторов) программы, заключаемый в слова begin и end.

Описательная часть предназначена для объявления всех встречающихся в программе данных и их характеристик (имена данных, их тип, возможные значения и др.).

В ***исполнительной части*** (разделе операторов) записывается последовательность исполняемых операторов. Каждый оператор выражает действие, которое необходимо выполнить. Исполняемые операторы, как мы уже упоминали, отделяются друг от друга символом ; (точка с запятой

Структура программы

```
program ИмяПрограммы;
uses
      ИмяМодуля1, . .;
label
      ИмяМетки1, .. ;
const
      ИмяКонстанты = ЗначениеКонстанты;
type
      ИмяТипа = ЗначенияТипа;
var
      ИмяПеременноЙ : Тип;
    ( объявления процедур и функций программиста )
begin
( инструкции основной программы }
end.
```

Раздел USES

Раздел uses позволяет подключать стандартные и пользовательские библиотечные модули. Он начинается с зарезервированного слова uses и имеет следующий вид:

```
uses
    ИмяМодуля1, ИмяМодуля2,.;
```

Например:

```
uses crt;
```

Раздел описания меток

Перед любым оператором Turbo Pascal можно поставить метку, что позволяет выполнить прямой переход на этот оператор с помощью оператора goto из любого места программы.

label ИмяМетки1, ИмяМетки2, ...;

Раздел описания констант

Хранение констант не требует памяти, компилятор помещает их значения прямо в текст исполняемой программы. Каждая константа принадлежит к определенному типу данных, однако при определении константы его обычно не указывают.

```
const ИмяКонстанты = ЗначениеКонстанты;
```

Например:

```
const g=9.8; { вещественная константа}
nmax=100; {целая константа }
nmin=-nmax;
s='абвгд'; { строковая константа }
kod=$123; {шестнадцатиричная константа }
```

Раздел описания типов данных

Раздел описания типов данных — это раздел описания типов, определяемых пользователем, поэтому в простых программах он часто отсутствует.

Раздел начинается с зарезервированного слова type и имеет вид:

type

ИмяТипа1 = ОписаниеТипа1;

ИмяТипа2 = ОписаниеТипа2;

Например:

type matr = array [1..maxrow, 1..maxcol] of real; {задан тип matr — таблица с maxrow строк и maxcol столбцов }

Раздел описания переменных

Все переменные, используемые в программе, должны быть перечислены в разделе описания переменных. Описание должно предшествовать использованию переменной. После того как переменная описана, она может быть опознана компьютером, а в тексте программы к ней можно обратиться по имени. Однако содержимое переменной пока еще не определено, поэтому переменные часто инициализируют, присваивая им начальное значение.

```
var
ИмяПеременной1,, ИмяПеременнойN: ТипПеременной;
```

Например:

```
var xl,x2: integer;
yl: longint; { целочисленные переменные }
sum: real; root: double; { вещественные переменные }
znak: char; { символьная переменная }
flag: boolean; { логическая переменная }
```

Раздел операторов

Раздел операторов является основным, т. к. именно в нем с предварительно описанными константами, переменными, значениями функций выполняются действия, позволяющие получить результат, ради которого и создавалась программа.

```
begin
Оператор1;
Оператор2; . . .
ОператорN;
end.
```