

Структура программ на Паскале

В Паскале существует очень важное правило – любое имя (кроме ключевых слов), используемое в программе, обязательно должно быть предварительно описано. Именно такой подход обусловил наличие в программе двух частей – сначала идет описательная, или декларативная часть, а затем – исполняемая часть.

Описать имя – это означает, в первую очередь, присвоить ему соответствующий тип. Последнее еще раз свидетельствует о том, как важно при программировании на Паскале учитывать природу операндов, описываемую каждым типом.

Программа всегда начинается с заголовка:

Program <имя программы>;

Имя программы выбирается произвольно, но оно не должно совпадать с каким-либо ключевым словом и должно быть уникальным.

Далее следует описательная часть, которая может включать в себя несколько разделов.

Раздел описания констант начинается с ключевого слова `const`, за которым следует сами описания. (Напомним, что константы не меняют своих значений в процессе работы программы.)

Например:

<code>const</code>	{заголовок раздела описания констант}
<code>f=15;</code>	{целочисленная константа}
<code>a="&;</code>	{символьная константа}
<code>lg="Computer";</code>	{строковая константа}
<code>s12_a=3.8521254</code>	{вещественная константа}
<code>tt=TRUE;</code>	{булевская (логическая) константа}

Комментарии, в отличие от Бейсика, заключаются в фигурные скобки. Их можно размещать в программе везде, где есть пробелы. Это часто оказывает полезным для документирования программы, да и просто в качестве памятки. На работу программы комментарии никак не влияют.

НАЧАЛО ПРОГРАММЫ	
<code>program</code> <имя программы> [параметры]	
ОПИСАТЕЛЬНАЯ ЧАСТЬ	РАЗДЕЛ ОПИСАНИЯ ТИПОВ type
	РАЗДЕЛ ОПИСАНИЯ ПЕРЕМЕННЫХ var
	РАЗДЕЛ ОПИСАНИЯ КОНСТАНТ const
	РАЗДЕЛ ОПИСАНИЯ МЕТОК label
	ОПИСАНИЕ ПОДПРОГРАММ (ПРОЦЕДУР И ФУНКЦИЙ ПОЛЬЗОВАТЕЛЯ)
ИСПОЛНЯЕМАЯ ЧАСТЬ	НАЧАЛО ИСПОЛНИТЕЛЬНОЙ ЧАСТИ begin
	Операторы исполняемой части
	ЗАВЕРШЕНИЕ ПРОГРАММЫ end

Раздел описания переменных начинается с ключевого слова `var`, за которым следует сами описания. (Напомним, что переменные – это поименованные данные, которые в процессе работы программы могут принимать различные значения.) Например:

<code>var</code>	{заголовок раздела описания переменных}
<code>i:integer;</code>	{целая переменная типа Integer}
<code>yt:byte;</code>	{целая переменная типа Byte}
<code>s:real;</code>	{вещественная переменная типа Real}
<code>b67:char;</code>	{символьная переменная}
<code>str:string;</code>	{строковая переменная}
<code>R_T:boolean;</code>	{булевская (логическая) переменная}

Обратите внимание: после имени переменной всегда ставится двоеточие, а любая команда в языке Паскаль заканчивается точкой с запятой.

Раздел описания типов начинается с ключевого слова `type` и включает в себя описания нестандартных (пользовательских) типов данных. Например, пусть нам для создания картотеки нужно описать такой тип данных, который включал бы поля `avt` (автор), `nazv` (название), и `god` (год издания). Такого типа среди стандартных явно нет, поэтому опишем свой. (Такие структуры называются записями – `record`.)

```
type                {заголовок раздела описания типов}
bibl=record         {называем новый тип «bibl» и указываем, что это запись}
avt:string;        {после записи, хранящее фамилию автора – строковый тип}
nazv:string;       {поле записи, хранящее название}
god:word;          {поле записи, хранящее год издания}
end;               {любое описание типа заканчивается ключевым словом «end»}
```

Когда нужный тип описан, мы можем использовать его наряду со стандартными. Например:

```
var
sw:integer;
poi:char;
z:bibl;
a123:longint;
```

Это означает, что переменная `z` будет иметь тип `bibl`, т.е. состоять из полей: `z.avt`, `z.nazv`, и `z.god`.

Раздел описания меток начинается с ключевого слова `label`, за которым перечисляются все имеющие метки. Они ставятся в программе перед теми операторами, на которые осуществляется переход. Например, если оператор: `Z:=X+Y`; и на него предполагается переход, то его нужно предварить меткой – любым именем, заканчивающимся двоеточием, например `M11`:. Тогда оператор будет иметь вид:

```
M11:Z:=X+Y;
```

Эта метка должна быть описана в разделе описания меток:

Label

```
M11
```

Если в программе имеется несколько меток, то они могут быть описаны в одной строке:

Label

```
M11, F15, D, G123, INV
```

Если программа использует подпрограммы (процедуры и функции), созданные пользователем, то они тоже должны быть описаны до начала исполняемой части (которая, как для процедур и функций, так и для основной программы, начинается с оператора `begin`).

Для процедуры:

```
procedure <имя процедуры> [параметры]
{описательная часть процедуры}
begin
{исполняемая часть процедуры}
end;
```

Для функций:

```
function <имя функции> [параметры]
{описательная часть функции}
begin
{исполняемая часть функции}
end;
```

Итак, в описательной части, помимо описания подпрограмм, могут иметься четыре раздела: `const`, `var`, `type` и `label`. Нужно помнить, что точка с запятой после названия раздела не ставится. В описательной части может быть несколько разделов с одним и тем же заголовком. В этом случае все они рассматриваются как один соответствующий раздел.

Главное, чтобы все описания, в том числе описания подпрограмм (процедур и функций), предшествовали началу исполняемой части программы. Однако совершенно не обязательно стремиться в самом начале работы, до написания исполнительской части, описать все это. Лучше уже в процессе написания программы по мере создания новых констант, переменных, типов, меток и подпрограмм возвращаться к описательной части программы и дописывать все необходимое.