

Паскаль. Линейные программы

Язык программирования Паскаль

Структура программы

Язык программирования Паскаль был разработан в конце 1960-х годов швейцарским ученым **Никлаусом Виртом** и назван в честь французского философа и математика XVII века Блеза Паскаля. Одной из целей создания языка Паскаль Никлаус Вирт считал обучение студентов структурному программированию.

В настоящее время используются в основном системы программирования **Free Pascal** и **Pascal ABC**, а также последняя версия - **PascalABC.NET**, сочетающая простоту языка Паскаль и огромные возможности платформы .NET.

Структура программы:

```
program <имя программы>;  
  label <описание меток>;  
  const < описание констант>;  
  type < описание типов>;  
  var <описание переменных>;  
  <описание процедур и функций>  
begin  
  <раздел операторов>  
end.
```

Замечания:

- ✓Любой раздел, кроме раздела операторов, может отсутствовать;
- ✓Раздел операторов также называется **телом программы**;
- ✓Порядок следования разделов описаний может быть различным, но для сохранения стиля желательно придерживаться приведенного выше порядка.

Имя программы, а также все имена констант, типов и переменных - это просто их обозначения в программе, по которым их можно отождествить – **идентифицировать**. Поэтому в программировании вместо термина «имя» обычно используется термин **«идентификатор»**.

Идентификаторы (имена) могут состоять из **латинских букв, цифр и знаков подчеркивания**, но **первой обязательно должна быть буква (имя не может начинаться с цифры!)**. При этом заглавные и строчные буквы не различаются.

Таким образом, имена могут включать

- латинские буквы (A-Z)

заглавные и строчные буквы не различаются

- цифры

имя не может начинаться с цифры

- знак подчеркивания _

Желательно вкладывать в имя смысл, и знак подчеркивания обычно используется вместо пробела в том случае, если имя состоит из нескольких слов.



**Величины и выражения.
Оператор присваивания**

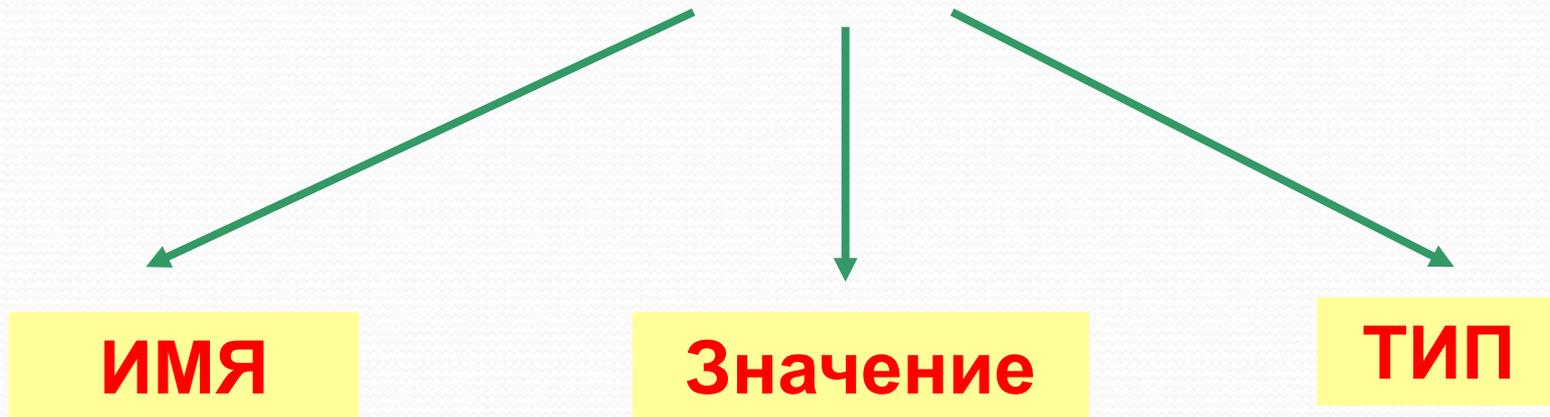
Понятие величины

Любая программа предназначена для обработки некоторых данных, которые хранятся в памяти компьютера в виде величин.

То есть, под величинами в языках программирования обычно понимаются именованные области памяти для хранения данных, с которыми работает программа

- **Величина имеет три основных характеристики:**

ВЕЛИЧИНА



Имя величины – это обозначение («метка», «бирка») той области памяти, которая отведена под данную величину.

Значение величины – это содержимое соответствующей величине области памяти.

Тип величины – это характеристика, определяющая множество допустимых значений величины и множество допустимых операций над ней, а также форму внутреннего представления величины в памяти компьютера.

Величины делятся на:

КОНСТАНТЫ и ПЕРЕМЕННЫЕ

Константа – постоянная величина, то есть ее значение не изменяется в процессе работы программы.

Переменная – это величина, значение которой можно изменять во время работы программы.

Примеры описания констант:

`const`

`i2 = 45; { целое число }`

`pi = 3.14; { вещественное число }`

целая и дробная часть отделяются точкой

`s = '***снег*'; { строка символов }`

`L = True; { логическая величина }`

Примеры описания переменных:

Типы величин:

- integer { целая }
- real { вещественная }
- char { один символ }
- boolean { логическая }

Объявление переменных (выделение памяти):

```
var a, b: integer;  
    Q: real;  
    s1, s2: char;
```

Оператор присваивания

Для присваивания величине какого-либо значения или для изменения существующего значения величины определена команда (оператор) **присваивания**, которая имеет следующий **формат**:



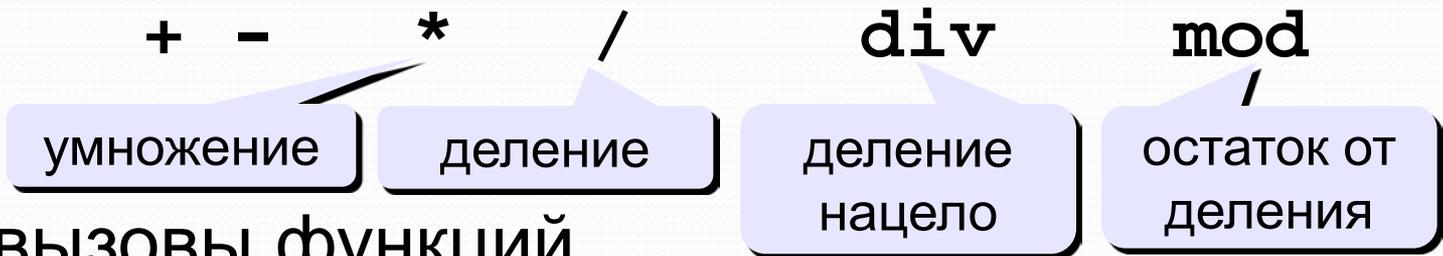
```
<имя переменной> := <выражение>;
```

При выполнении команды присваивания вычисляется значение выражения, стоящего справа от знака присваивания ($:=$), и полученное значение заносится в область памяти с соответствующим именем величины (стоящим слева от знака присваивания).

Выражение – это конструкция языка, определяющая порядок вычисления некоторого значения. Различают **арифметические и логические выражения**.

Арифметическое выражение может включать

- КОНСТАНТЫ
- имена переменных
- знаки арифметических операций:



- ВЫЗОВЫ функций
- круглые скобки ()

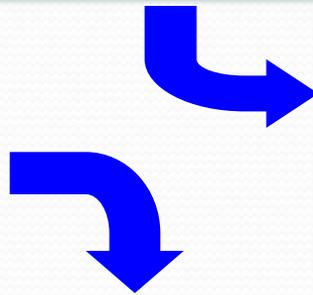
Порядок выполнения операций

- вычисление выражений в скобках
- умножение, деление, **div**, **mod** слева направо
- сложение и вычитание слева направо

2 3 5 4 1 7 8 6 9

```
z := (5*a*c+3*(c-d))/a*(b-c)/ b;
```

$$x = \frac{a^2 + 5c^2 - d(a+b)}{(c+d)(d-2a)}$$



$$z = \frac{5ac + 3(c-d)}{ab} (b-c)$$

2 6 3 4 7 5 1 12 8 11 10 9

```
x := (a*a+5*c*c-d*(a+b))/((c+d)*(d-2*a));
```



**Организация ввода и вывода в
программах на языке Паскаль.
Линейные программы.**

Линейной называется программа, последовательность записи команд в которой совпадает с последовательностью их выполнения.

Простейшие линейные программы обычно сводятся к тому, что:

- пользователь с клавиатуры вводит значения некоторых величин (числа или символы),
- эти значения обрабатываются, обычно с помощью команды присваивания,
- полученный результат выводится на экран компьютера.

Оператор вывода

Формат оператора:

Write[ln] (<список вывода>);

Замечание: Здесь и далее в квадратных скобках будем записывать необязательную часть каких-либо конструкций языка (она может быть, а может и отсутствовать).

В данном случае сочетание **ln** означает перевод курсора на следующую строку экрана по окончании вывода всех элементов списка. При её отсутствии по окончании вывода курсор остается в текущей позиции строки вывода.

Список вывода может содержать три вида элементов:

- 1) Имена переменных
- 2) Константы (числовые и строковые)
- 3) Выражения

Разделителем в списке служит запятая.

При выполнении оператора вывода на экран соответственно выводятся:

- 1) Значение переменной
- 2) Сама константа
- 3) Значение выражения

Примеры

`writeln (a);` { вывод значения
переменной `a` и переход
на новую строку }

`writeln ('Привет!');` { вывод текста-
строковой константы }

`writeln ('Ответ: ', c);` { вывод
текста и значения переменной `c` }

`writeln (a, '+', b, '=', c);`

Форматы вывода

К именам переменных в списке вывода могут добавляться через знак двоеточие **числа, обозначающие количество позиций на экране, отводимых под соответствующие значения переменных.**

Для переменных **целого и символьного** типа указывается **одно число**, а для переменных **вещественного** типа – **два числа**: первое определяет общее количество позиций, отводимых под число на экране, а второе – количество позиций, отводимых под дробную часть числа.

Пример:

```
program primer_format_vivoda;  
var i: integer;  
    x: real;  
begin  
    i := 15;  
writeln ( '>', i:5, '<' );  
    x := 12.345678;  
writeln ( '>', x:7:2, '<' );  
end.
```

ВСЕГО
СИМВОЛОВ

ВСЕГО
СИМВОЛОВ

В дробной
части

ABC Pascal ABC

Файл Правка Вид Программа Сервис Помощь



•Program1.pas

```
program primer_vivod;  
    var a,b: integer; c:real; d:char;  
begin  
    a:=3; b:=5; c:=37.6; d:='@';  
    writeln(a,b);  
    writeln(a:8,b:10);  
    writeln('a',d:6,b:6);;  
    writeln(c);  
    writeln(c:10:4);  
    writeln('a+b=',a+b);  
    writeln('a=',a,'    b=',b,d,d,d,2*a-b)  
end.
```

35

```
      3      5  
a      @      5  
37.6  
      37.6000  
a+b=8  
a=3    b=5@@@1
```

Оператор ввода

Формат оператора:

Read[ln] (<список имен переменных>);

- Разделителем в списке служит запятая.
- Сочетание *ln* обеспечивает переход курсора к новой строке по окончании ввода.

Например:

```
readln( a, b ); { ввод значений  
                  переменных a и b }
```

При выполнении оператора ввода программа приостанавливает свою работу, и ждет ввода соответствующих (указанных в списке переменных) значений. **Числовые значения набираются через пробел, а символьные – без пробела. По окончании нажимается клавиша ввода: “ ↵ ” - Enter.**

Число имен переменных в списке и количество набираемых на клавиатуре значений должны быть равны, их типы и порядок следования также должны быть согласованы.

Для того, чтобы пользователю было понятно, что следует вводить (набирать на клавиатуре), перед оператором ввода обычно помещают оператор вывода с наводящим сообщением – подсказкой о смысле вводимых данных, например:

```
Write('введите три числа: '); {подсказка}  
Readln(a,b,c); {ввод}
```

Пример 1. Составить программу, вычисляющую площадь круга и длину окружности заданного радиуса (радиус окружности вводится с клавиатуры).

Текст программы:

```
Program krug;  
  Const pi=3.1415;  
  Var   R, S, L: real;  
Begin  
  Write ('введите радиус ');  
  Readln (R);  
  S:= pi*R* R;  
  L:=2*pi*R;  
  Writeln ('S=', S:5:2, ' L=', L:5:2);  
End.
```