
Программирование на языке Паскаль

Тема 1. Введение

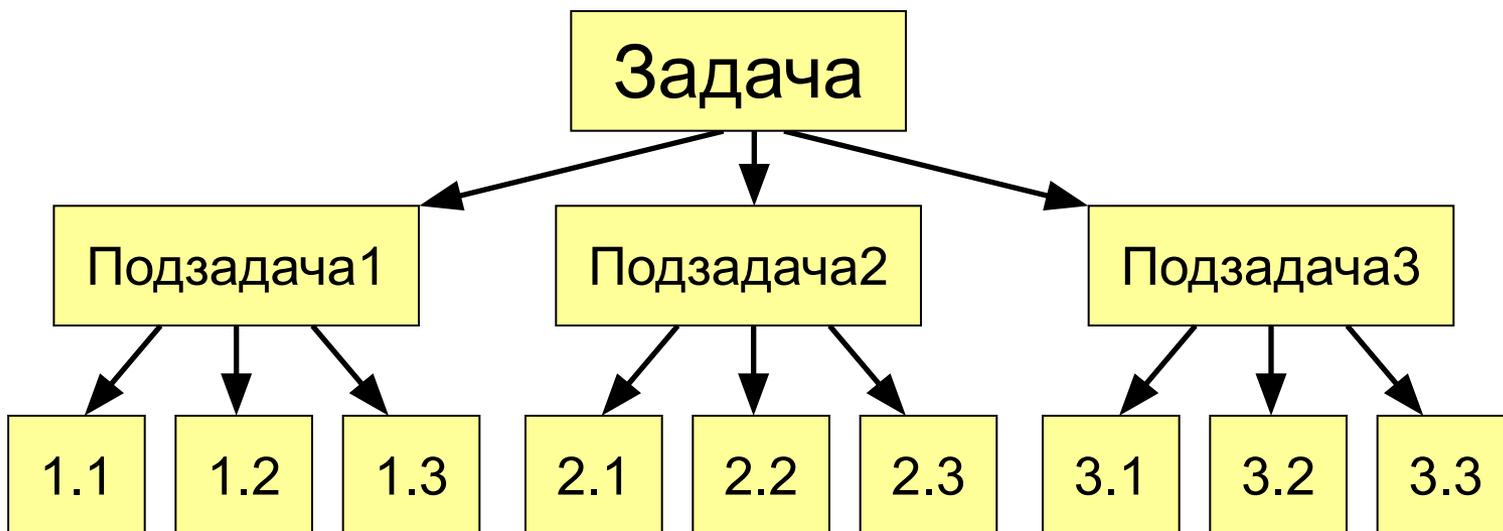
-
- **Алгоритм** - это последовательность действий, которые необходимо выполнить, чтобы решить поставленную задачу.
 - **Программа** же представляет собой набор команд на языке, понятном исполнителю, реализующий некоторый *алгоритм*. В нашем случае исполнителем является компьютер, а языком программирования будет **ЯЗЫК ВЫСОКОГО УРОВНЯ Pascal**. К сожалению, любой язык высокого уровня удобен только человеку, пишущему или отлаживающему *программу*, но совершенно непонятен компьютеру. *Программа* на таком языке называется исходным текстом и хранится во внешнем файле с расширением `.pas`.
-

-
- Для перевода программы на *язык низкого уровня*, понятный исполнителю-компьютеру, существуют специальные программы-переводчики - **компиляторы**. Результатом работы *компилятора* (иными словами, результатом процесса компиляции) является **исполняемый код**, который записывается в файл с расширением `.exe`.
-

Язык Паскаль

1970 – Никлаус Вирт (Швейцария)

- язык для обучения студентов
- разработка программ «сверху вниз»

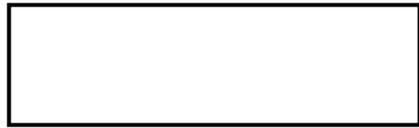


- разнообразные структуры данных (массивы, структуры, множества)

Синтаксическая диаграмма -

является направленным графом, при прохождении которого **автоматически** строится синтаксически правильная программа.

В диаграммах используется два типа блоков:



1. каждое понятие, заключенное в прямоугольник, требует в свою очередь некоторого определения;



2. содержит понятие, не требующее дополнительного определения.

Программа:



Идентификаторы

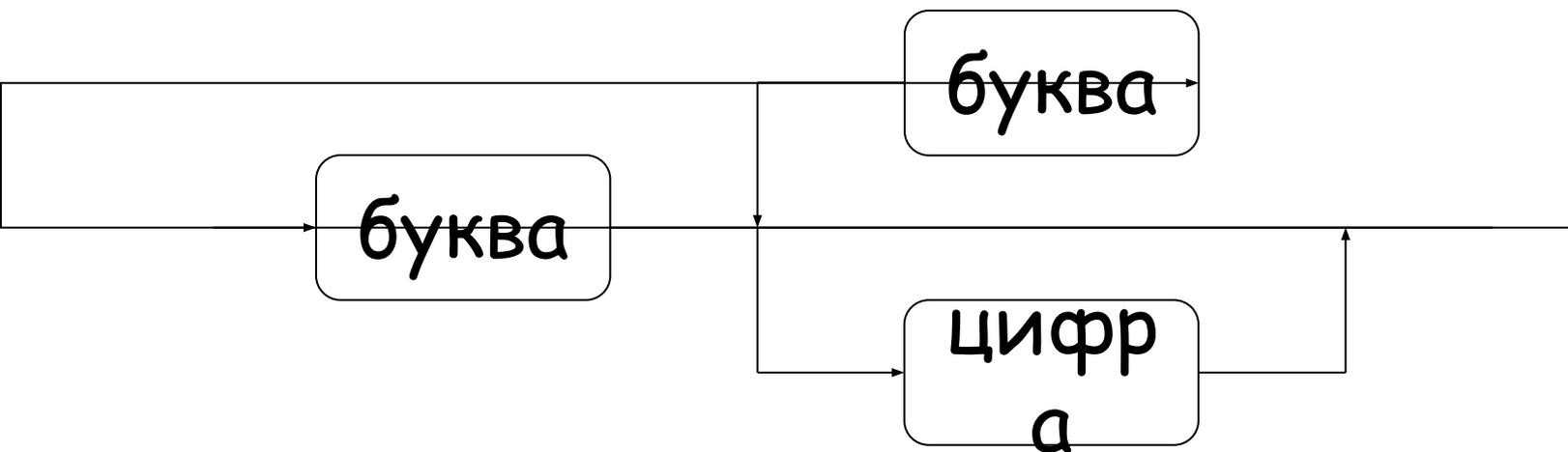
Имена, даваемые программным объектам (*константам*, типам, *переменным*, функциям и процедурам, да и всей *программе* целиком) называются **идентификаторами**. Они могут состоять только из цифр, латинских букв и знака "_" (подчеркивание). Однако цифра не может начинать имя. Идентификаторы могут иметь любую длину, но если у двух имен первые 63 символа совпадают, то такие имена считаются идентичными.

Идентификаторы

Вы можете давать программным объектам любые имена, но необходимо, чтобы они отличались от *зарезервированных слов*, используемых языком Pascal, потому что *компилятор* все равно не примет *переменные* с зарезервированными именами.

Примеры.

Идентификатор:



Константы

Константа - это объект, значение которого известно еще до начала работы программы.

В языке Pascal существует три вида констант:

- *неименованные константы* (цифры и числа, символы и строки, множества);
 - *именованные нетипизированные константы* ;
 - *именованные типизированные константы*.
-

Неименованные константы

Неименованные константы не имеют имен, и потому их не нужно описывать.

- любая последовательность цифр (возможно, предваряемая знаком "-" или "+" или разбиваемая одной точкой) воспринимается *компилятором* как *неименованная константа* - число (целое или вещественное);
- любая последовательность символов, заключенная в апострофы, воспринимается как *неименованная константа* - строка;
- любая последовательность целых чисел либо символов через запятую, обрамленная квадратными скобками, воспринимается как *неименованная константа* - множество.

Примеры использования *неименованных* *констант*:

int1 := -10; real2 := 12.075 + x;

char3 := 'z'; string4 := 'abc' + string44;

set5 := [1,3,5] * set55; boolean6 := true;

Именованные нетипизированные константы

Именованные константы, как следует из их названия, должны иметь имя. Стало быть, эти имена необходимо сообщить *компилятору*, то есть описать в специальном разделе ***const***.

Если не указывать тип константы, то по ее внешнему виду *компилятор* сам определит, к какому (базовому) типу ее отнести. Любую уже описанную константу можно использовать при объявлении других констант, переменных и типов данных.

Именованные нетипизированные константы

Примеры описания *нетипизированных именованных констант*:

```
const n = -10;  
      m = 1000000000;  
      mmm = n*100;  
      x = 2.5;  
      c = 'z';  
      s = 'string';  
      b = true;
```

Именованные типизированные константы

Типизированные именованные константы представляют собой *переменные (!)* с начальным значением, которое к моменту старта *программы* уже известно. Следовательно, во-первых, *типизированные константы* нельзя использовать для определения других констант, *типов данных и переменных*, а во-вторых, их значения можно изменять в процессе работы *программы*.

Именованные типизированные константы

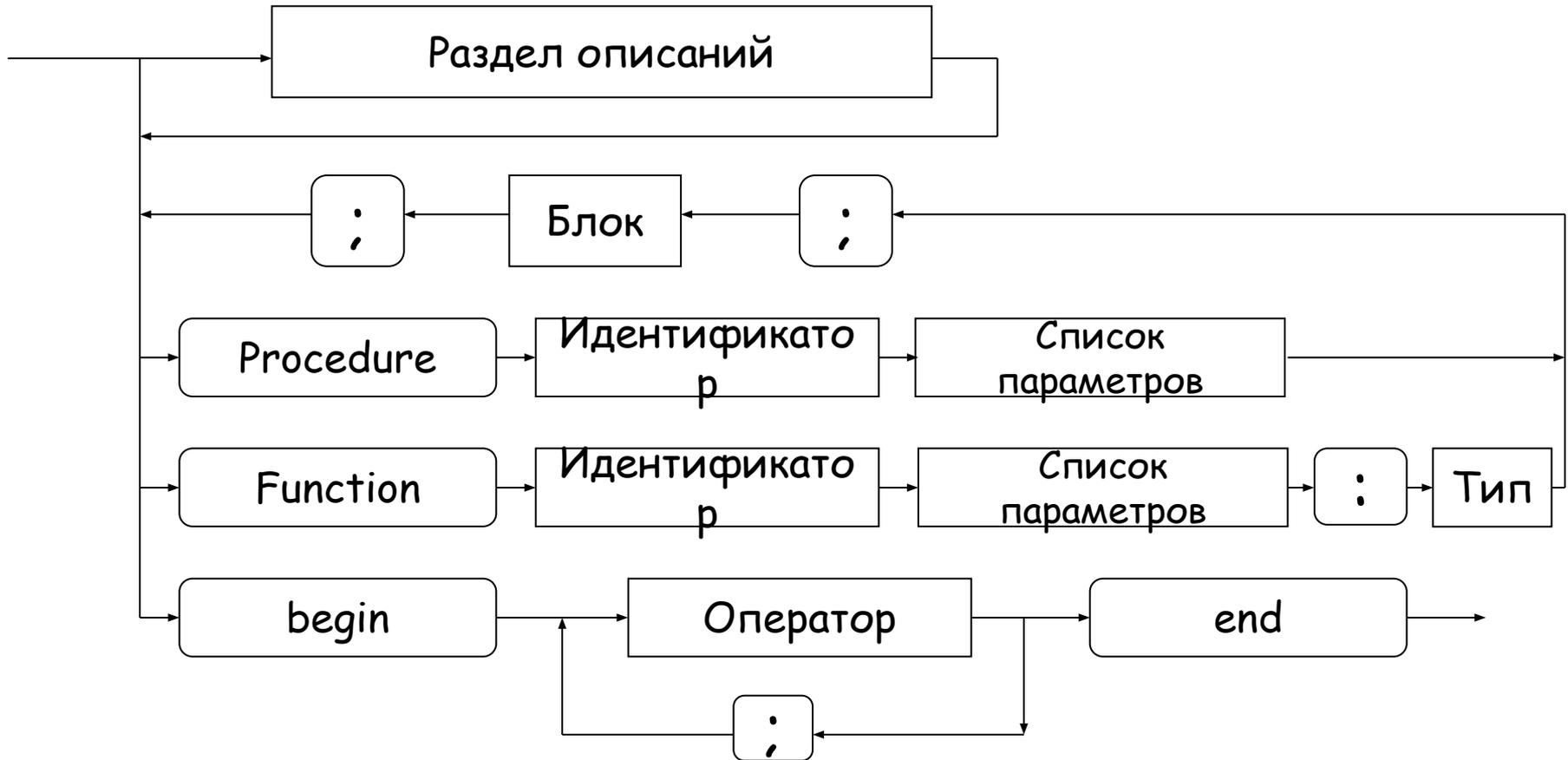
Описание *типизированных констант* производится по следующему шаблону:

```
const <имя_константы> : <тип_константы>  
= <начальное_значение>;
```

Именованные типизированные константы

```
const n: integer = -10;  
      x: real = 2.5;  
      c: char = 'z';  
      b: boolean = true;
```

Блок:



Из чего состоит программа?

program <имя программы>;

const ...; { константы }

var ...; { переменные }

{ процедуры и функции }

begin

... { основная программа }

end.

комментарии в фигурных скобках
не обрабатываются

Имена программы, констант, переменных

Имена могут включать

- латинские буквы (A-Z)

заглавные и строчные буквы не различаются

- цифры

имя не может начинаться с цифры

- знак подчеркивания _

Имена **НЕ** могут включать

- русские буквы
- пробелы
- скобки, знаки +, =, !, ? и др.

Какие имена правильные??

AXby R&B 4Wheel Вася "PesBarbos"

~~TU154 [QuQu] _ABBA A+B~~

Константы

`const`

`i2 = 45; { целое число }`

`pi = 3.14; { вещественное число }`

целая и дробная часть отделяются точкой

`qq = 'Вася'; { строка символов }`

можно использовать русские буквы!

`L = True; { логическая величина }`

может принимать два значения:

- True (истина, "да")
- False (ложь, "нет")

Переменные

Переменная – это величина, имеющая имя, тип и значение. Значение переменной можно изменять во время работы программы.

Типы переменных:

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

Объявление переменных (выделение памяти):

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

Как изменить значение переменной?

Оператор – это команда языка программирования высокого уровня.

Оператор присваивания служит для изменения значения переменной.

Пример:

```
program qq;
```

```
var a, b: integer;
```

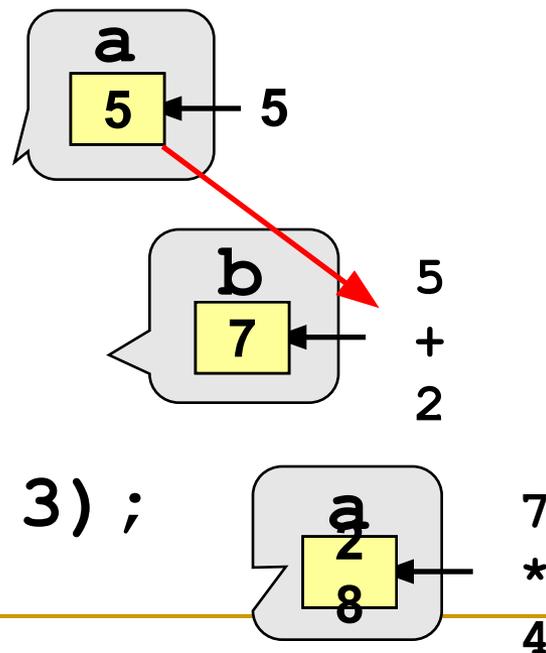
```
begin
```

```
  a := 5;
```

```
  b := a + 2;
```

```
  a := (a + 2) * (b - 3);
```

```
end.
```



Оператор присваивания

Общая структура:

<имя переменной> := <выражение>;

Арифметическое выражение может включать

- КОНСТАНТЫ
- имена переменных
- знаки арифметических операций:

+ - * / div mod

умножение

деление

деление
нацело

остаток от
деления

- ВЫЗОВЫ функций
- круглые скобки ()

Какие операторы неправильные?

```
program qq;  
var a, b: integer;  
    x, y: real;  
begin  
    a := 5;  
    10 := x;  
    y := 7,8;  
    b := 2.5;  
    x := 2*(a + y);  
    a := b + x;  
end.
```

имя переменной должно
быть слева от знака :=

целая и дробная часть
отделяются **точкой**

нельзя записывать
вещественное значение в
целую переменную

Ручная прокрутка программы

```
program qq;  
var a, b: integer;  
begin  
  a := 5;  
  b := a + 2;  
  a := (a + 2) * (b - 3);  
  b := a div 5;  
  a := a mod b;  
  a := a + 1;  
  b := (a + 14) mod 7;  
end.
```

a	b
?	?
5	
	7
28	
	5
3	
4	
	4

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;

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

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

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));

Сложение двух чисел

Задача. Ввести два целых числа и вывести на экран их сумму.

Простейшее решение:

```
program qq;  
var a, b, c: integer;  
begin  
    read ( a, b );  
    c := a + b;  
    writeln ( c );  
end.
```

Оператор вывода

`write (a);` { вывод значения
переменной `a` }

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

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

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

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

Форматы вывода

```
program qq;  
var i: integer;  
    x: real;  
begin  
    i := 15;  
    writeln ( '>', i, '  
    writeln ( '>', i:5, '<' );  
    x := 12.345678;  
    writeln ( '>', x, '<' );  
    writeln ( '>', x:10, '<' );  
    writeln ( '>', x:7:2, '<' );  
end.
```

ВСЕГО
СИМВОЛОВ

```
>15<  
> 15<  
  
>1.234568E+001<  
> 1.23E+001<  
> 12.35<
```

ВСЕГО
СИМВОЛОВ

В дробной
части

Полное решение

```
program qq;  
var a, b, c: integer;  
begin  
    writeln('Введите два целых числа');  
    read ( a, b );  
    c := a + b;  
    writeln ( a, '+', b, '=', c );  
end.
```

Протокол:

это выводит компьютер

Введите два целых числа

25 30

это вводит пользователь

25+30=55