

Соловьев С.Ю.  
soloviev@glossary.ru

**А**лгоритмы и  
**А**лгоритмические языки

[www.park.glossary.ru/pascal/](http://www.park.glossary.ru/pascal/)

*Лекция No.5*

2022

# Напоминание



# Вещественный тип

(A) Переменные вещественного типа

```
var R : real;  
var X, Y : real;
```

(B) <вещественное> – синтаксис вещ.: **1.5**, **-3.14**, **1E-20**.

Область значений определяется реализацией.

(B.1) Операции над вещественными: **+** | **-** | **\*** | **/**

(B.2) Восемь стандартных функций: **REAL** → **REAL**

1. <b>abs (x)</b>	x	5. <b>exp (x)</b>	$e^x$
2. <b>sqr (x)</b>	$x^2$	6. <b>ln (x)</b>	$\ln(x)$
3. <b>sin (x)</b>	$\sin x$	7. <b>sqrt (x)</b>	$\sqrt{x}$
4. <b>cos (x)</b>	$\cos x$	8. <b>arctan (x)</b>	$\text{arctg}(x)$

# Преобразования REAL ↔ INTEGER

Пусть `var X : real;`  
`N : integer;`

**REAL → INTEGER** – функции `trunc` и `round`

`N := trunc(x)`  
отбрасывание  
дробной части

`N := round(x)`  
ближайшее целое

x	trunc(x)	round(x)
5.2	5	5
5.8	5	6
-5.2	-5	-5
-5.8	-5	-6

$$\text{round}(x) \stackrel{\text{def}}{=} \begin{cases} \text{trunc}(x+0.5), & \text{если } 0 \leq x \\ \text{trunc}(x-0.5), & \text{если } 0 > x \end{cases}$$

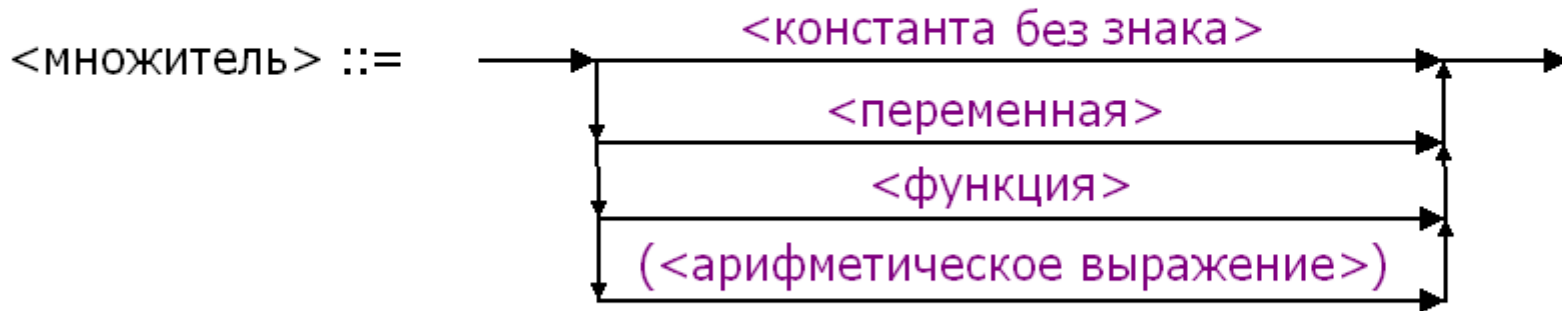
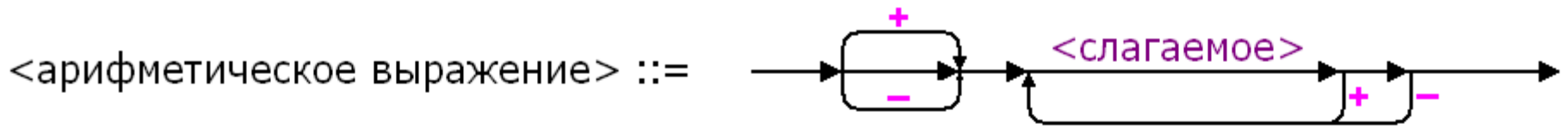
**INTEGER → REAL** – соглашения

- Целые на местах вещественных: `sin(N)`
- Правила определения типов операций: `X := N`

<переменная> := <выражение>

## Арифметические выражения

Тип выражения – тип результата. Арифм. вып. – integer или real.



константы без знака      функция.

+3 \* (maxInt / 7 + 6.7) - abs (N+K)

(арифм. вып.)      переменные

- Старшинство операций
- Правила определения типа вып.

# Булевский тип (введение)

“Саше 5 лет”, “ $x > 5$ ”, “Сократ – человек”

Предложение **истинное** или **ложное** в конкретных обстоятельствах.

**true**

**false**

Конструирование таких предложений

“простые”

$<$

$<$

$\leq$

$\leq$

$>$

$>$

$\geq$

$\geq$

$=$

$=$

$\neq$

$\neq$

odd

“сложносочиненные”

$\&$

and

$\vee$

or

$\neg$

not

$a < x < b \Rightarrow (a < x) \text{ and } (x < b)$

$\langle \text{арифм.выр} \rangle \leq \langle \text{арифм.выр} \rangle$

$\text{odd}(\langle \text{арифм.выр} \rangle)$

# Булевский тип

(А) Переменные булевского типа

```
var X : boolean;  
var A, B : boolean;
```

(Б) Значения: **true**, **false**.

(В) Операции над булевскими: **not** | **and** | **or** | **=** | **<>**

A	not A
false	true
true	false

A	B	A and B
false	false	false
false	true	false
true	false	false
true	true	true

A	B	A or B
false	false	false
false	true	true
true	false	true
true	true	true

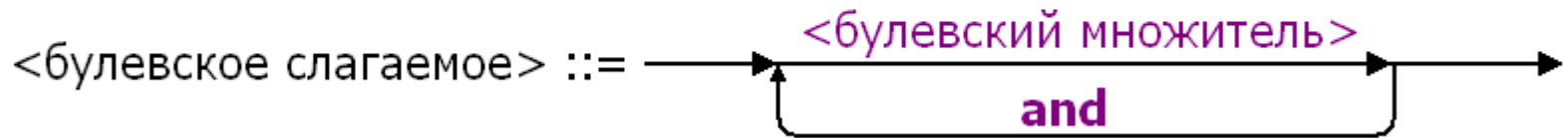
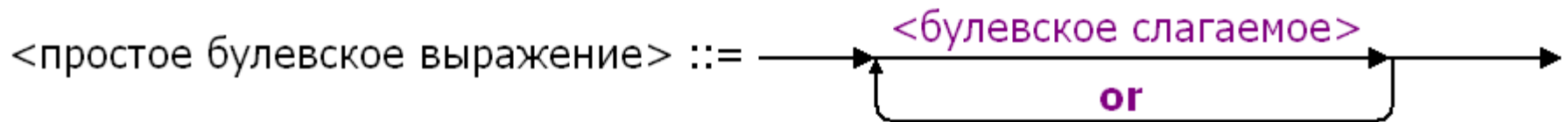
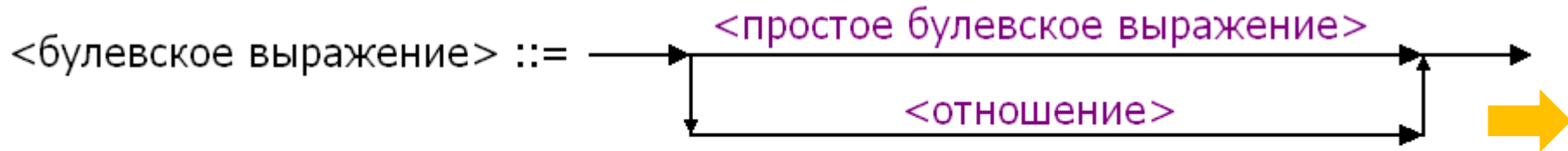
A	B	A = B
false	false	true
false	true	false
true	false	false
true	true	true

A	B	A <> B
false	false	false
false	true	true
true	false	true
true	true	false

# Булевские выражения

Пусть

```
var X, Y : boolean;  
    A, B, C : integer;
```

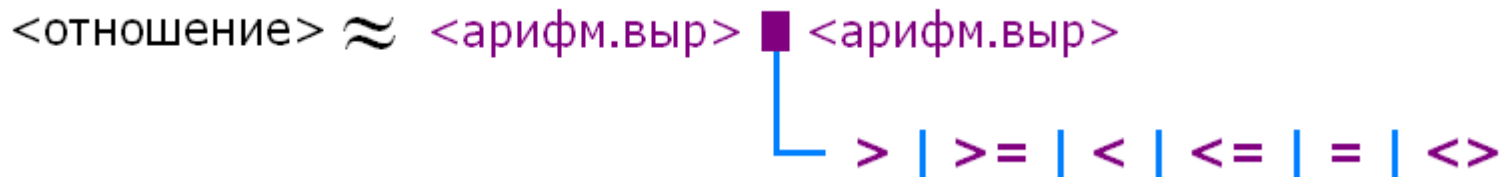
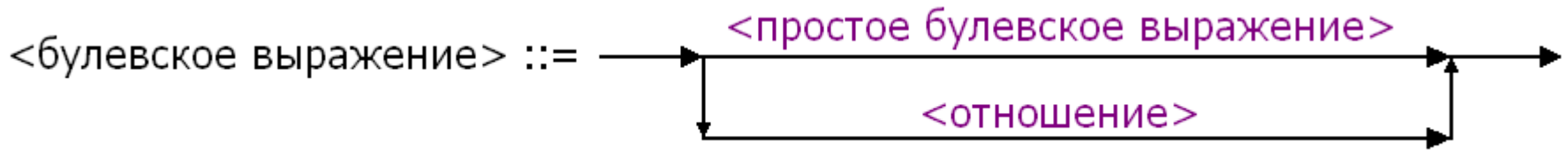


<булевский множитель> есть

- либо переменная булевского типа: **X, Y**
- либо константа булевского типа: **true, false**
- либо функция булевского типа: **odd(A)**
- либо **not** <булевский множитель>: **not X**
- либо (<булевское выражение>): **(X and not Y)**



# Булевские выражения – 2

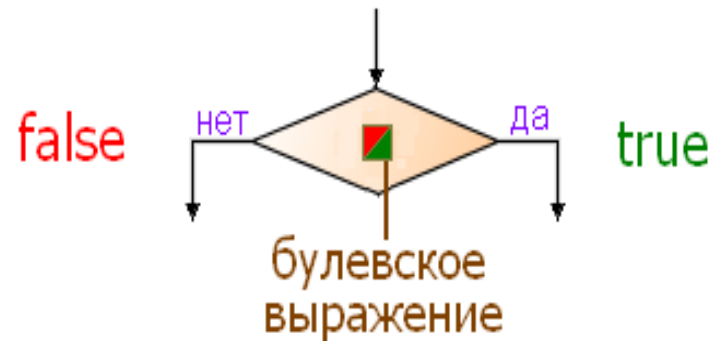


Примеры:  $A+5 < 2*B$ ,  $X \text{ or } (A+5 < 2*B)$ ,  
 $A >= 3$ ,  $(A >= 3) \text{ and } Y \text{ or } (B <= 4)$

Приоритеты операций

- 1- not
- 2- \* / div mod and
- 3- + - or
- 4- > >= < <= = <>

Прагматика:



# Символьный тип – 1

(A) Переменные символьного типа

```
var C : char;  
var D, X : char;
```

(Б.1) Значения – доступные литеры: 'A', 'a', '#', . . .  
особый случай: ''

(B.1) Операции над символьными: `ord: char → integer`

(B.2) Отношения: `< | <= | = | <> | > | >=`

`C < D ⇔ ord(C) < ord(D)`

`C <= D ⇔ ord(C) <= ord(D) . . .`

(Б.2) Обязательные свойства множества значений:

`'0' < '1' < '2' < '3' < '4' < '5' < '6' < '7' < '8' < '9' и ord('9') - ord('0') = 9 ;`

`'a' < 'b' < . . . < 'y' < 'z' ;`

`'A' < 'B' < . . . < 'Y' < 'Z'`

# Символьный тип – 2

(B.3) Операции над булевскими: **chr: integer → char**

СВОЙСТВО:  $\text{chr}(\text{ord}(C)) = C$

(B.4) Функции:

**pred: char → char**  $\Leftrightarrow \text{pred}(C) = \text{chr}(\text{ord}(C) - 1)$

**succ: char → char**  $\Leftrightarrow \text{succ}(C) = \text{chr}(\text{ord}(C) + 1)$



**ord: char → integer ?**



КОДИРОВКИ

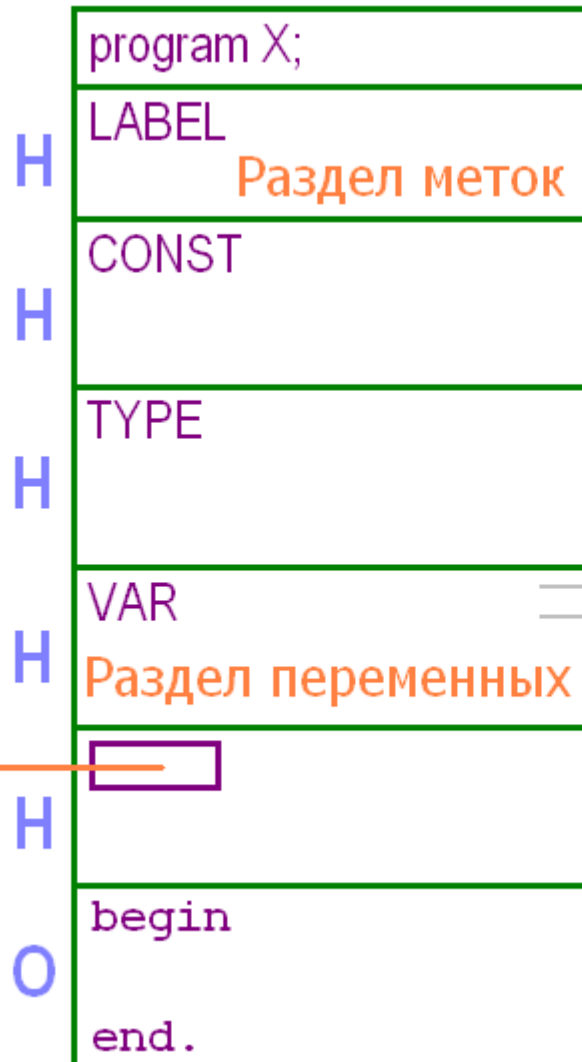
ASCII

CP nnn

KOI8 xx

UTF-8

# Программа :: Уточнение No. 1



<раздел меток> ::=  
 <пусто> | label <метка> {, <метка> };

<пусто> ::=

var I : integer;  
 var N, M : integer;

var R : real;  
 var X, Y : real;

procedure  
 или  
 function

Программа как строка

1. Конец строки-в-файле
2. Расстановка пробелов
3. Комментарии: (\* \*) | { }

с оговорками

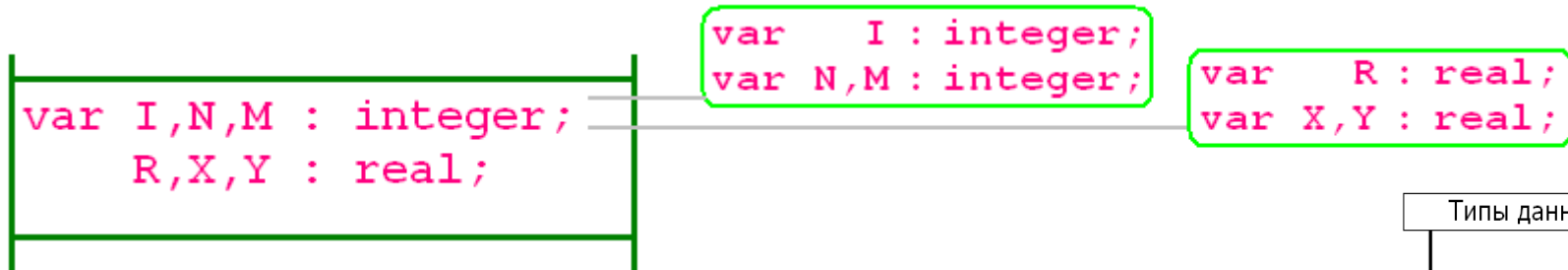
# Программа :: Раздел переменных

<раздел переменных> ::= <пусто> |

**var** <описание переменных>; { <описание переменных>; }

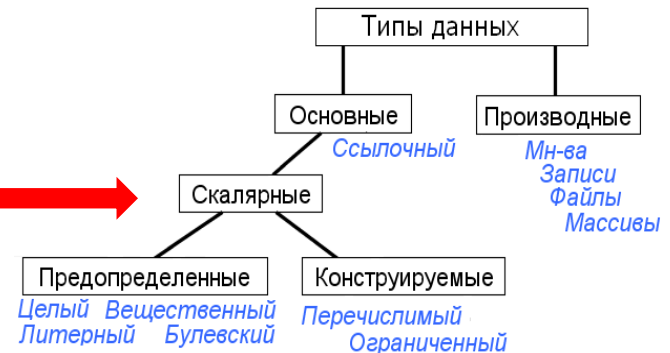
<описание переменных> ::=

<имя переменной> {, <имя переменной> } : <тип>



*Требования к программе*

1. Переменные скалярного типа должны быть описаны.
2. Разные переменные – разные имена.
3. Описание должно предшествовать использованию.
4. Первоначально у переменных нет значений.



Операторы: присваивания, ввода, вывода, пустой,  
составной, условный, перехода, цикла, ...

## Оператор ввода (с клавиатуры)

Read / Readln

`Read(X1, X2, ..., XN);`

Элемент списка ввода: `char | integer | real`

Пусть: `var X1 : integer; X2 : char; X3 : real;`

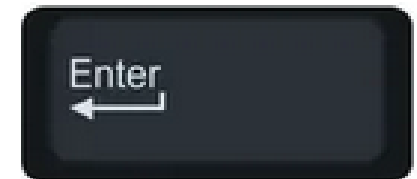
`Read (X1 , X2 , X3) ;`

Входной поток: `100_A_0.77_`

Результат: `X1 = 100; X2 = 'A'; X3 = 0.77`

`Readln (X1 , X2 , X3) ;`

Входной поток: `100_A_0.77<Enter>`



# Оператор вывода (на экран)

write / writeln

$\text{write}(X_1, X_2, \dots, X_N); \equiv \text{write}(X_1); \text{write}(X_2); \dots; \text{write}(X_N);$

└─ real | integer | boolean | char + строки

Вместо  $\text{write}(X)$  можно использовать  $\text{write}(X:\langle\text{целое}\rangle_1)$  и

(если  $X$  – вещественное)

$\text{write}(X:\langle\text{целое}\rangle_1:\langle\text{целое}\rangle_2)$

```
R:=0.00025;  
writeln(R);  
writeln(R:10:5);  
writeln(R:10:4);  
writeln(R:10:2);
```

```
└2.5000000000E-04  
└└└0.00025  
└└└└0.0003  
└└└└└0.00
```

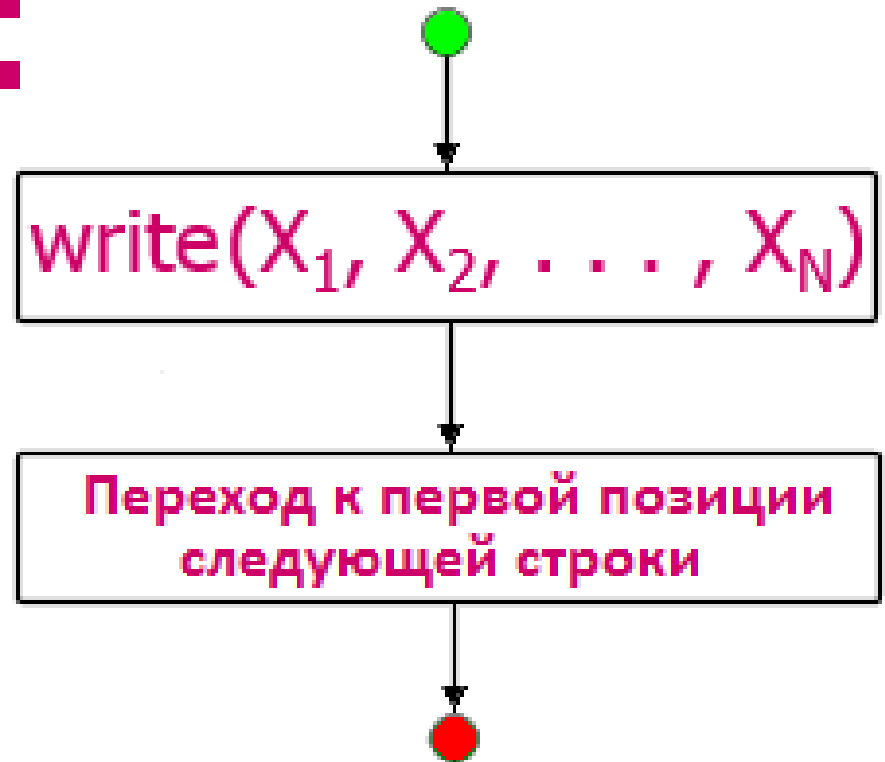
┆  
ширина  
поля

┆  
к-во знаков  
в дробной

Соглашение:  $\text{write}('И', 'в', 'а', 'н') = \text{write}('Иван')$

# Оператор вывода (на экран) - 2

`writeln(X1, X2, . . . , XN)` ■



Допускается оператор `writeln` без аргументов



## Обзор лекции No.5

Преобразования `real` → `integer` и `integer` → `real`

Арифметические выражения

Булевский тип данных

Булевские выражения

Операции над булевыми выражениями

Приоритеты операций в булевских выражениях

Символьный тип данных

Свойства множества значений символьного типа

Преобразования `char` → `integer` и `integer` → `char`

ПАСКАЛЬ-программа как строка

Комментарии в ПАСКАЛЬ-программе

Раздел меток ПАСКАЛЬ-программы

Раздел переменных ПАСКАЛЬ-программы

Оператор ввода

Оператор вывода

**--- Конец лекции No. 5 ---**