

# Язык программирование Pascal ABC

**Цель:**

- Познакомить с понятием язык программирования Pascal и основными конструкциями языка.
- Сформировать навыки использования разных типов данных при решении задач.
- Показать комплексное использование компьютера (источник информации, исполнитель алгоритмов )

*Учитель информатики  
гимназии №12 г. Тюмени*

*Бугаева Елена  
Викторовна ©*

# Язык программирования –

это совокупность набора символов системы (алфавит), правил образования (синтаксис) и истолкования конструкций из символов (семантика) для задания алгоритмов с использованием символов естественного языка.

*Pascal* - процедурный язык, имеющий блочную структуру. Набор операторов языка отражает принципы структурного программирования.

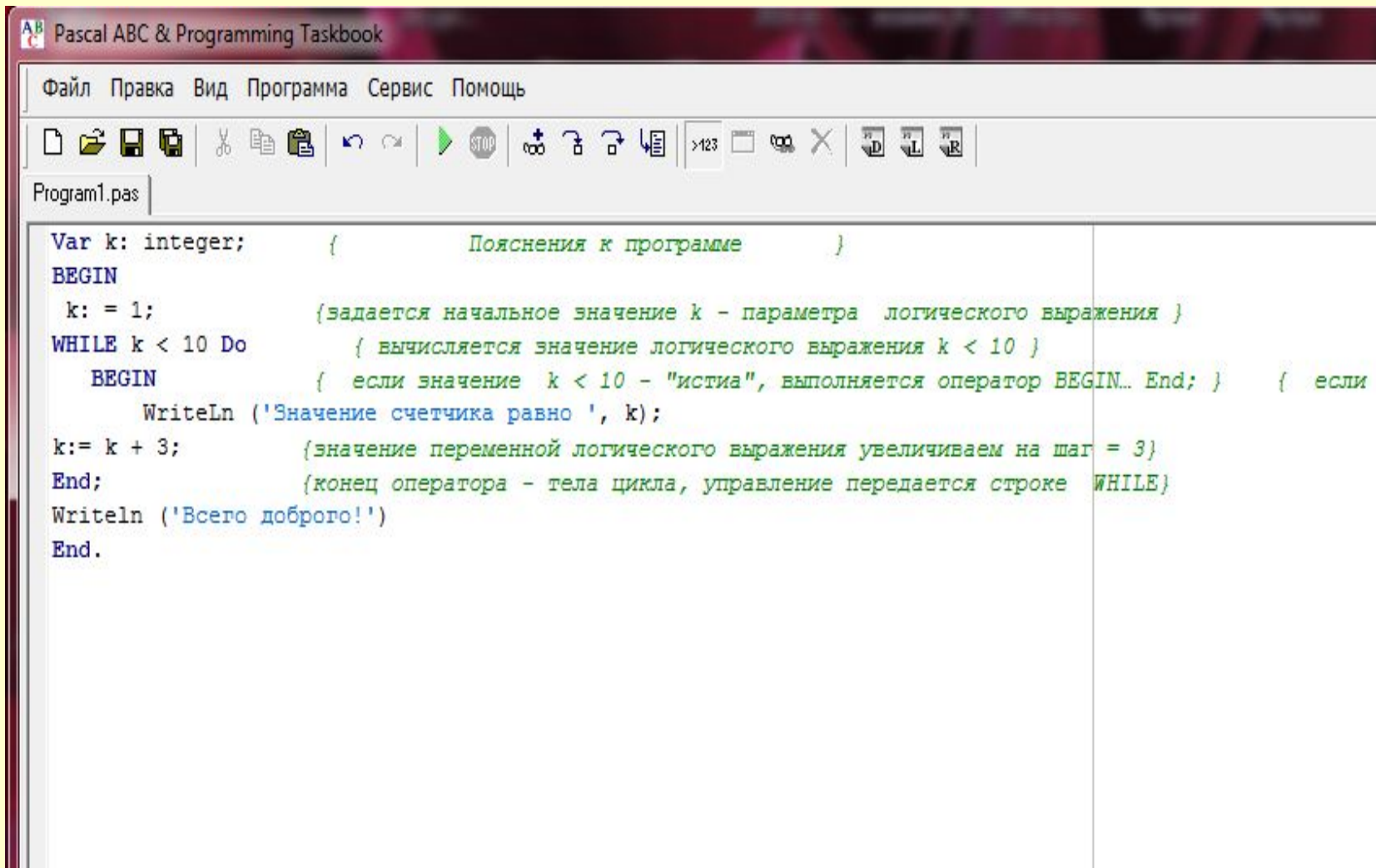
Существует два "вида" переводчиков языков программирования, основная задача которых, перевести программу с языка программирования в машинные коды понятные компьютеру:

- **Компилятор** переводит всю программу сразу и только после этого, если в ней нет ошибок, запускает её на выполнение.

- **Интерпретатор** переводит текст программы построчно и сразу построчно её выполняет.

# Запуск Pascal ABC

C:\Program Files\РАВС\PascalABC.exe

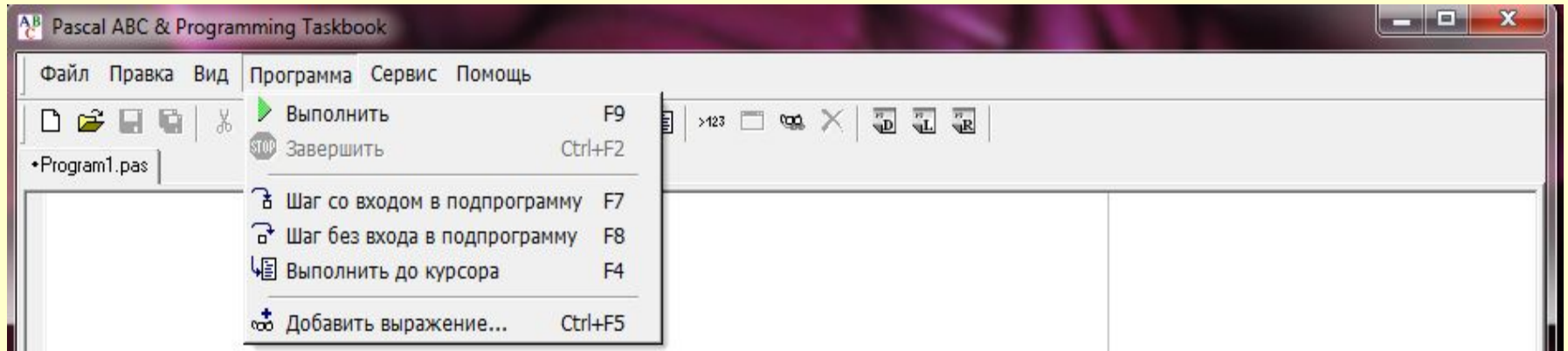


The screenshot shows a window titled "Pascal ABC & Programming Taskbook". The menu bar includes "Файл", "Правка", "Вид", "Программа", "Сервис", and "Помощь". The toolbar contains various icons for file operations and execution. The main text area displays the following Pascal code with Russian comments:

```
Var k: integer;      { Пояснения к программе }
BEGIN
  k: = 1;            { задается начальное значение k - параметра логического выражения }
  WHILE k < 10 Do    { вычисляется значение логического выражения k < 10 }
    BEGIN           { если значение k < 10 - "истина", выполняется оператор BEGIN.. End; } { если
      WriteLn ('Значение счетчика равно ', k);
    k:= k + 3;       { значение переменной логического выражения увеличиваем на шаг = 3}
  End;              { конец оператора - тела цикла, управление передается строке WHILE}
  WriteLn ('Всего доброго!')
End.
```

# Управление средой

Управление средой осуществляется с помощью меню, которое аналогично меню текстового редактора



# Алфавит языка

- 1) 26 латинских букв
- 2) Цифры
- 3) Арифметические операции:  $+$ ,  $-$ ,  $/$ ,  $*$ ,  
MOD, DIV
- 4) Знаки отношений:  $<$ ,  $>$ ,  $=<$ ,  $>=$ ,  $<>$ ,  
OR, AND, NOT
- 5) Специальные символы:

.	отделяет целую часть от дробной
, ;	- отделяют данные в программе
'	(апостроф) используют для комментариев в программе
{ }	используют для комментариев в программе которые не влияют на результат выполнения программы
..	Разделитель в обозначении диапазона
:=	знак присваивания
()	в арифметических выражениях

# Структура программы

Программа на языке Pascal состоит из двух основных частей:



**описания всех**

**данных,**

с которыми производятся  
действия

**описания**

**самих**

**действий**



## В разделе описания могут быть следующие объекты:

<code>Program</code>	имя программы ; {заголовков программы}
<code>Label</code>	{раздел описания меток}
<code>Const</code>	{раздел описания констант}
<code>Type</code>	{раздел описания типов}
<code>Var</code>	{раздел описания переменных}

## В разделе действий программы

<code>Begin</code>	{тело программы, представляет собой последовательность операторов, разделенных символом «;»}
<code>End.</code>	



Program1.pas

```
Var k: integer;  
BEGIN  
  k := 1;  
  WHILE k < 10 Do  
    BEGIN  
      WriteLn ('Значение k = ' + IntToStr(k));  
      k := k + 3;  
    End;  
  WriteLn ('Всего доброго!');  
End.
```

Описание данных

Описание действий

# Конструкции языка

Из символов алфавита конструируются объекты называемые данными.

- *Данные* – величины, которые обрабатываются командами и операторами языка.

## Данные Pascal :

- Модули;
- Константы;
- Переменные;
- Функции;
- Выражения;
- Массивы;
- Метки.

# Типы Данных

В любой задаче используются, обрабатываются какие-либо данные числа, целые или вещественные, массивы, символы, слова и т.д. Все они характеризуются своим типом

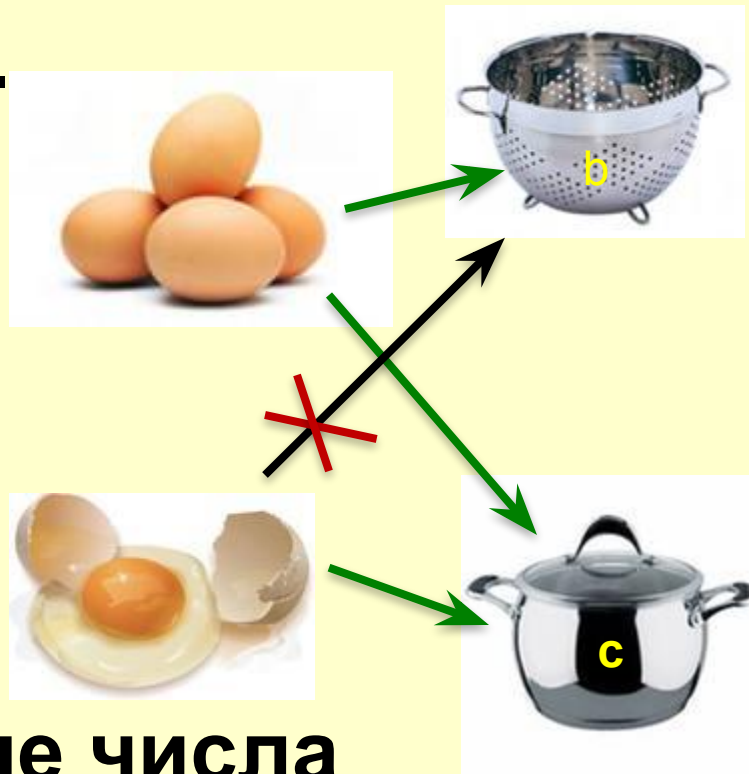
Первичными в иерархии типов являются стандартные скалярные, представляющие традиционные типы в языках программирования:

# Стандартные скалярные типы данных

- целые типы (**Integer**, Диапазон допустимых значений -32768.. 32767, размер памяти 2 байта );
- вещественные типы (**Real**, диапазон допустимых значений  $2.9 \text{ E } -39.. 1.7 \text{ E } 38$ , размер памяти 6 байт)
- символьный тип (**Char**, Значениями символьного типа являются символы из множества ASCII, размер памяти, занимаемый символом, - 1 байт.
- строковый тип (**String**, символы из множества ASCII, занимаемый несколько символом.
- логический тип (**Boolean**, основу математической логики составляют две константы: True (Истина, 1) и False (Ложь, 0). Логические данные широко используются при сравнении величин.

# Типы данных

**integer** – целые числа  
1, 100, - 1000 и т.д.



**real** – действительные числа  
1.5, 104, - 1000.5674 и т.д.

Д.З:

- Выучить лекцию

# **Описание данных в программе**



**Переменные** - величины значение которых меняется в ходе выполнения действий над ними

Пример: (переменной F присвоено значение 56): **F := 56**

Имя переменной  
(идентификатор)

Значение  
переменной

Знак  
присваивания

Имя переменной состоит из одного или нескольких символов латинского алфавита.

Недопустимые имена переменных:

21A	Начинается не с буквы
A C	Содержит пробелы
АНЯ	Содержит русские буквы
A&B	Содержит специальные символы
Char	Содержит команду

Раздел описания переменных начинается со служебного слова **var**, само описание переменной содержит два элемента:

- имя переменной (идентификатор) ;
- ее тип

**Var** имя переменной : тип переменной;

Эти элементы разделяются двоеточием, после указания типа ставится разделитель - СИМВОЛ «;».

# Например:

Описание переменных

```
Var al: integer;  
    b,c: real;  
    x: char;  
    y: string;  
    a,b: boolean;
```

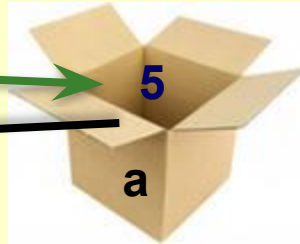
Действие над  
переменными в  
программе

```
al:=4;  
b:=3.5; c:=0.0034;  
x:='A'; y:='алф' ;  
a:=True;  
b:=False;
```

# Переменная в программе

•Program1.pas

```
var a: integer;  
begin  
a:=5;  
write(a);  
end.
```



CRT - программа завершена

5\_

**Константы** - постоянные величины изменение которых в программе не предусмотрено.

*Константы* по внешнему виду похожи на переменные, однако в отличие от них не могут изменить своего значения.

Описание констант начинается со служебного слова **const**

В языке имеются стандартные константы, которые можно использовать без предварительного объявления:

**Пример:**

**const**

```
max = 1000;
```

```
Pi = 3.1415;
```

```
C = Pi*2;
```

```
min = - max;
```

pi	3.141592
True	«ИСТИНА»
false	«ЛОЖЬ»

# Выражения

Выражения используются для вычисления новых значений.

*Операнды* - это некоторые «элементарные» значения: переменные, константы, вызовы функций.

***Примеры выражений:***

$$a+b-x;$$

$$y -(2-y+c)*y;$$

$$\sin(x) -2 + 56;$$



При составлении выражений необходимо знать следующие правила:

- всё выражение должно быть записано в строку. Например:

$$\frac{ax - a^2}{x_1 - x_2} \Rightarrow ((a * x) - a^2) / (x_1 - x_2);$$

- в выражении в качестве разделителей можно использовать только круглые скобки;
- нельзя записывать подряд два знака.  
Например, форма записи  $a+b/-c$  неправильна, правильной является форма  $a + b /(-c)$ .

# Стандартные функции языка

Pascal	Математическое определение и результат работы
<b>SQR(X)</b>	функция возведения в квадрат, где $x$ - переменная целого или вещественного типа.
<b>SQRT(X)</b>	$\sqrt{X}$ - функция извлечения квадратного корня, где $x$ - переменная целого или вещественного типа.
<b>ABS(X)</b>	$ X $ -функция определения модуля числа $x$ , где $x$ переменная целого или вещественного типа
<b>sin(x), cos(x), arctan(x)</b>	тригонометрические функции, аргументы и значения которых всегда вещественного типа
<b>round (x)</b>	операция округления вещественного числа $x$ до ближайшего целого (round (5 .67) =6; round (5 . 47) =5).
<b>exp (x)</b>	- <i>экспонента в степени x</i> , аргумент и значение всегда вещественного типа.
<b>trunc (x)</b>	- отбрасывание вещественной части $x$ , $x$ - вещественное значение (trunc(5.67)=5).
<b>In (x)</b>	- натуральный логарифм $x$ , аргумент и значение всегда вещественного типа.

# Арифметические операции

**+, -, \*, /, div, mod**

Операцию « / » целых чисел использовать нельзя, так как результат при этом будет выходить за рамки целого типа. Вместо деления к целочисленным данным применяют две операции:

операция «деление нацело» с отбрасыванием дробной части, обозначается служебным словом **div**

$(14 \text{ div } 3 = 4, \quad 5 \text{ div } 8 = 0)$ ;

операция «взятие остатка от целочисленного деления», обозначается служебным словом **mod**

$(14 \text{ mod } 3 = 2, \quad 5 \text{ mod } 8 = 5)$ .

11 div 5

$$\begin{array}{r} 11 \overline{) 5} \\ -10 \phantom{2} \\ \hline 1 \phantom{2} \end{array}$$

11 mod 5

Пример с отрицательными числами

$$-7 \text{ div } 2 = -3$$

$$-7 \text{ mod } 2 = -1$$

-3  
-1

$$-7 = (-3) * 2 + (-1)$$

Функции  $\ln(x)$  и  $\exp(x)$  могут быть использованы для возведения в степень по правилу:

$$x^n = \exp ( n \ln(x) )$$

Например, значение  $x^9$  вычисляется по формуле  $\exp (9 \ln (x))$ .

**Д/з: ВІУЧИТЬ ЛЕКЦІЮ**

# Повторение

1. Какие типы определены в языке Pascal для работы с числами?

(Integer ,real)

2. Вычислите значения выражений:

$$20 \operatorname{div} 6 = \quad ;$$

3

$$2 \operatorname{mod} 5 = \quad ;$$

2

$$80 \operatorname{mod} 6 = \quad ;$$

2

$$500 \operatorname{div} 10 =$$

50

3. Найдите значение выражения

$$A \bmod (B \operatorname{div} C + 5) - 10:$$

$$\text{при } A = 35, B = 6, C = 4.$$

Ответ -5

$$\text{при } A = 100, B = 20, C = 3.$$

Ответ -9

Отметьте выражения, которые относятся к  
целому типу:

$$1 + 2.0;$$

$$\operatorname{trunc}(3.1415);$$

$$20/5;$$

$$\operatorname{sqrt}(4);$$

$$\operatorname{sqr}(5.0);$$

$$5 + \operatorname{round}(12.7).$$



Запишите выражения на языке Pascal.

$$y = 1 + x + \frac{x^2}{2}$$

$$y = 1 + |x| + |1 + x|$$

$$y = \sqrt{1 + \sqrt{|x|}}$$

$$y = \frac{a + b}{c + d} - 2,5$$

Д/з: записать на языке Паскаль  
выражения.

1)  
у= 
$$\left( \frac{87}{54 \div 3} + 4,89 \right) \div \frac{7,86 - \sqrt{8} + 14^2}{-34 \cdot 5,9}$$

2)

В= 
$$\left( \frac{8,044}{1 + 3,04^2} + 0,82 \right) \div \frac{\sqrt{4}}{25 + 4}$$

Выполните задание на  
компьютере:  
«Выражение в паскаль»

**Тема: Операторы языка.  
Обработка числовых и  
СИМВОЛЬНЫХ ДАННЫХ**

# Задание

1) Определите какие типы данных представлены в примере. Где допущены ошибки:

204.57

2A=15

46=:X1

D:='нота'

204,57

C4:=5

SЯ=4/\$

+89/- 28

75=A

**2) Найдите ошибки в записи переменных :**

QЯN

2A

A2!b

CLS

Л4

C\$K

В программировании для решения какой-либо задачи составляют программу, которая состоит из команд и операторов языка.

*Программа- представляет собой последовательность операторов записанных друг за другом с новой строки.*

*Оператор*- это конструкция (предложение) для описания величин, оформленное в виде обозначения какого-то действия, для которого требуется выполнить целый ряд машинных операций.

# Типы операторов

Простые

Составные

Пример:  
присваивание  
 $y := \text{Sin}(\text{Pi}/3)$

Обычно состоят из  
открывающего оператора,  
тела, составного оператора, и  
закрывающегося оператора

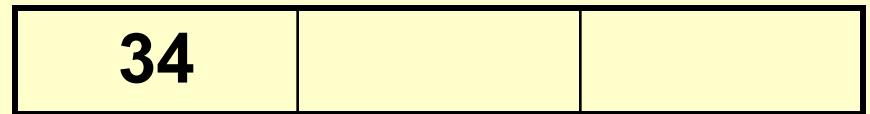
Пример:  
IF  $X > Y$  THEN  $Z = X - Y$ ;



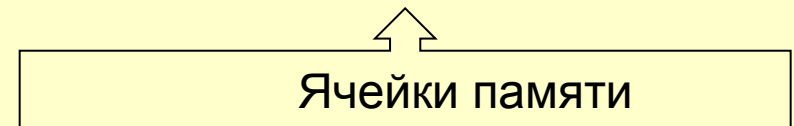
# Оператор присваивания

Для изменения значения переменной используют оператор присваивания. Он не имеет ключевого слова и определяется знаком **:=**

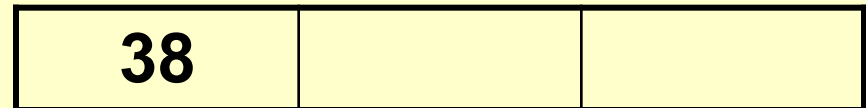
Min:=34



Min:= Min+4



Min:=> 38



# Операторы присваивания

Оператор присваивания предназначен для присваивания переменной значения выражения. Оператор обозначается символом «:=». Оператор считается синтаксически правильным, если тип выражения в правой части совпадает с типом переменной.

# Примеры использования оператора присваивания:

с переменными целого типа -  $x := 200$ ,

вещественного -  $y := \text{Sin}(\text{Pi}/3)$ ,

символьного -  $c := '6'$ ;

булевского -  $\text{test} := \text{false}$

# Изменение значений переменной

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

```
...
```

```
a := 5;
```

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

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

```
b := b + 1;
```

5

7

2  
8

b

8

5  
+  
2

7  
\*  
4

# Процедуры ввода и вывода

Для того чтобы программа могла работать в диалоговом режиме (позволяла бы вводить данные и видеть результаты работы на экране), существуют процедуры **ввода и вывода**.

Процедура вивода

Существует две формы процедуры  
вывода:

1) `write (a, b ) ;`

2) `writeln (x, z) ;`

В результате выполнения данной  
процедуры  
значения переменных будут выведены  
на экран монитора.

# Примеры вывода данных

```
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 );
```



## Запись на языке Паскаль

## результат работы программы

```
X1:=4; X2:=8; X3:=x1+x2;
```

```
write (x1, x2 , x3);
```

4 8 12

```
x:=5;
```

```
write('Значение x=', x);
```

Значение x=5

```
x:=4; y:=3;
```

```
Writeln ('Знач. выражения =',  
x+sqr(y));
```

«Значение выражения =  
13»

```
Writeln ('Hello, World!');  
write ( 'a=', a, 'b=', b, 'c=', c);
```

Hello, World! a=17 b=34 c=80

В процедурах вывода `write` и `writeln` имеется возможность установить ширину поля вывода

`write(y :i: j)`

печатает значения `y` с отступом в позиции `i` и количество знаков после запятой- `j`.

Пример:

`y := 6.205853678`

`Write (y :4: 2) ⇒`

6.20

Ширина поля

Количество знаков после запятой

Процедура ввода

Существует две формы  
процедуры ввода:

`read` и `readln`:

**Примеры**

`read(a);`

`read(a, b, c);`

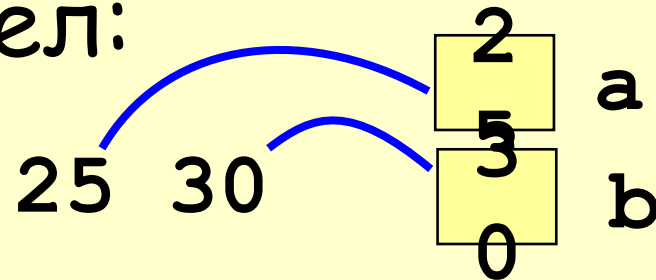
`Readln (x, y);`

Процедура **read** обеспечивает ввод данных разных типов с клавиатуры. В скобках указываются имена переменных, которым присвоятся введенные значения. При вводе нескольких значений с помощью одной процедуры, они разделяются символом «пробел». Для окончания ввода (завершения работы одной процедуры `read`) нажимается клавиша <Enter>.

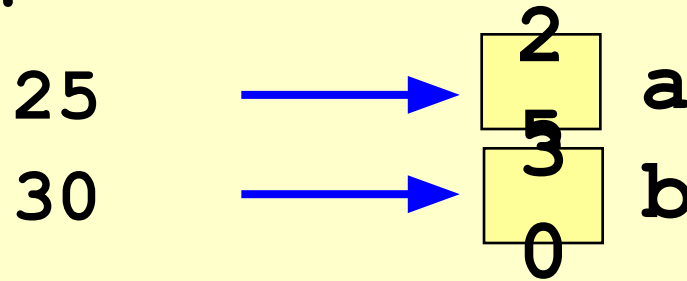
```
read ( a, b );
```

Ввод значений  
двух переменных  
(через пробел или  
*Enter*).

через пробел:



через *Enter* :



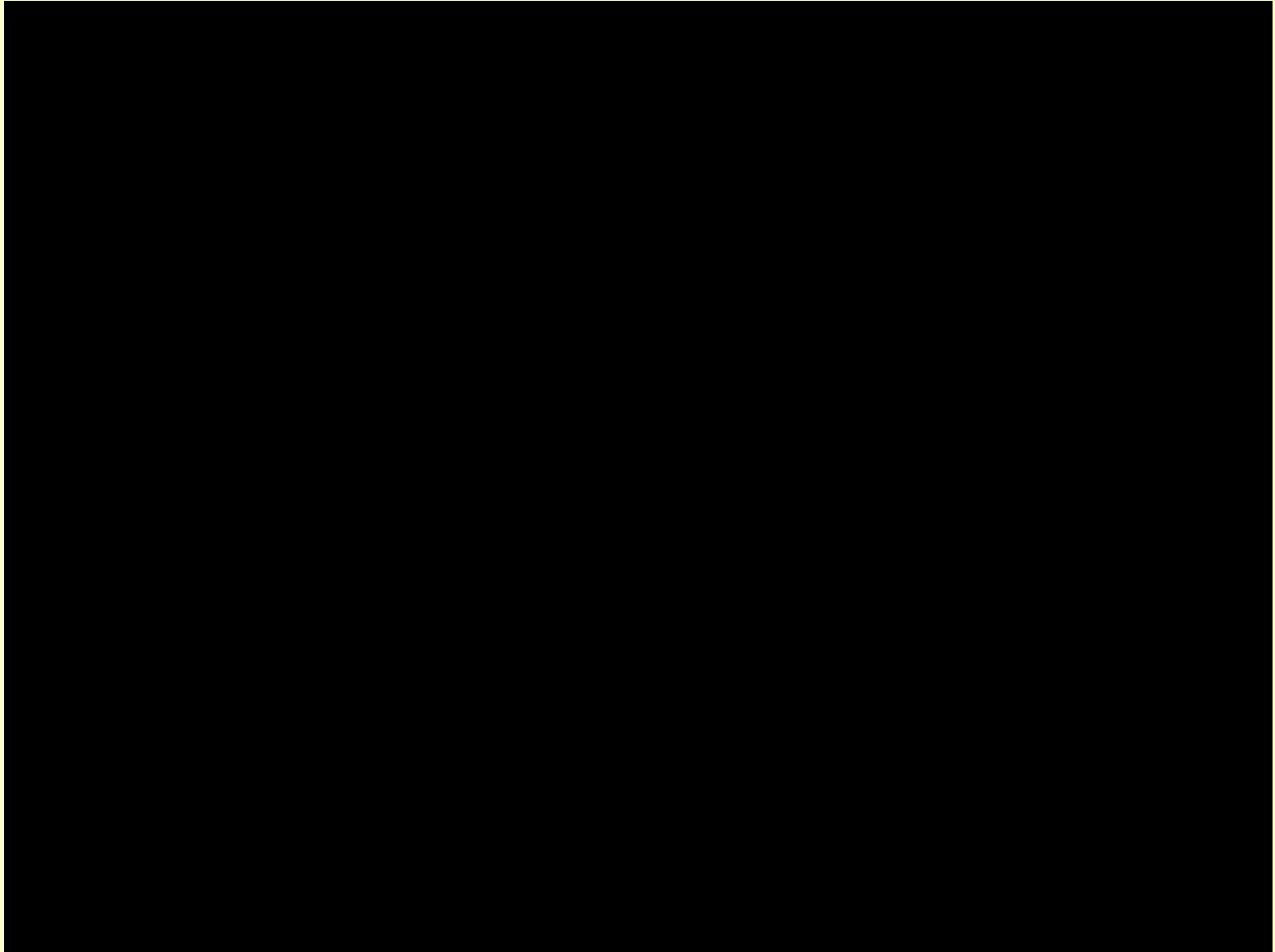
**Создание программ**

Пример: программа нахождения суммы 2-х  
чисел

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



Как работает программа



# Сложение чисел: простое

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



Что плохо?

# Сложение чисел: полное решение

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

Протокол:

КОМПЬЮТЕР

Введите два целых числа

25 30

ПОЛЬЗОВАТЕЛЬ

25+30=55

Задание к уроку:

Составьте программу  
вычисляющую значение  
выражения, запишите ее в  
соответствии с правилами  
Pascal.

**a, b, c, f- целые числа**

$$x = \left( \frac{a}{b + 3,04^2} + fc \right) \div \frac{\sqrt{1024}}{25 + b^2}$$

Примера должен  
быть записан по  
всем правилам  
записи выражений  
на языке  
программирования

Г ВЫПОЛНЕНИЯ ПРОГРАММЫ:

Решаем пример:

$$x = \left( \frac{a}{b + 3,04^2} + fc \right) \div \frac{\sqrt{1024}}{25 + b^2}$$

Введите значения переменных **a**, **b**, **c**, **f**:

4 5 2 9

Ответ: 34,230

```
Program Prim_1;
Var a, b, c, f: Integer ;
    x: real;
Begin
Writeln ( 'Решаем пример ' );
Writeln ( ' ( ( a / b... ) ) ' );
Writeln ( 'Введите переменные a, b, f, c
' );
readln (a, b, f, c);
X := ( a / ( b... ) );
Writeln ( 'Ответ: ' , x :3 :3 );
End.
```

Выражение  
запишите  
самостоятельно, по  
всем правилам  
записи выражений  
на языке  
программирования

Вычислите выражения

1 вариант)

$$c = \left( \frac{a}{54 \times b} + 4,89 \right) \div \frac{-7,86 - \sqrt{1024} + a^2}{b \cdot 3,9}$$

2 вариант)

$$c = \left( \frac{b^2}{1,44 + a^2} + 0,82 \right) \div \frac{a\sqrt{256}}{ba}$$



# **Дополнительные возможности в паскаль**

# Модули

Модули предназначены для разбиения текста программы на несколько файлов. В модулях описываются переменные, константы, типы, классы, процедуры и функции.

Для того чтобы эти объекты можно было использовать в вызывающем модуле, следует указать имя файла модуля в разделе **uses** вызывающего модуля.

Например: **uses** GraphABC;

Модуль **CRT** предназначен для работы с текстовым окном.

Модуль описывается до раздела описания переменных.

**uses crt;**

### **Константы модуля CRT**

**Clrscr;** –очистка фона экрана;

**TextColor (n);** - цвет букв; ( $0 < n < 15$ )

**TextBackGround ( n);** –цвет фона.

# Константы цвета

Название константы (цвет)	Значение	Название константы (цвет)	Значение
Black (черный)	0	DarkGray (темно-серый)	8
Blue (синий)	1	LightBlue (ярко-синий)	9
Green (зеленый)	2	LightGreen (ярко-зеленый)	10
Cyan (голубой)	3	LightCyan (ярко-голубой)	11
Red (красный)	4	LightRed (розовый)	12
Magenta (фиолетовый)	5	LightMagenta (малиновый)	13
Brown (коричневый)	6	Yellow (желтый)	14
LightGray (светло-серый)	7	White (белый)	15

# Очистка экрана, цвет текста и фона

```
program p1;  
  uses crt;  
  var c,a,b:integer;  
      r:real;  
begin  
  clrscr;  
  TextColor (1);  
  textbackground(7);
```

# Программа «Диалог с компьютером»

# Результат программы «Диалог»

1) Привет, меня зовут КОМПЬЮТЕР.

Как тебя зовут?

Ответ: **Катя**

2) Очень приятно, **Катя!** Сколько тебе лет?

Ответ:

3) Чудесный **15** возраст! А у тебя есть хобби!? ...

# Программа «Диалог с компьютером»

```
Program Dialog;  
uses crt;  
    var b1,b2,b3... : string;  
    {переменные строкового типа}  
Begin  
    clrscr;  
    Writeln( ' Привет, как тебя зовут? ' );  
    Write( 'Твой ответ:' );  
    Readln (b1);
```

Количество  
вопросов =  
количество  
переменных

Первый  
вопрос и  
ответ



```
WriteLn('Очень приятно!', b1, 'Сколько тебе  
лет?');
```

```
Write('ТВОЙ ОТВЕТ:');
```

```
ReadLn(b2);
```

```
...
```

```
End.
```