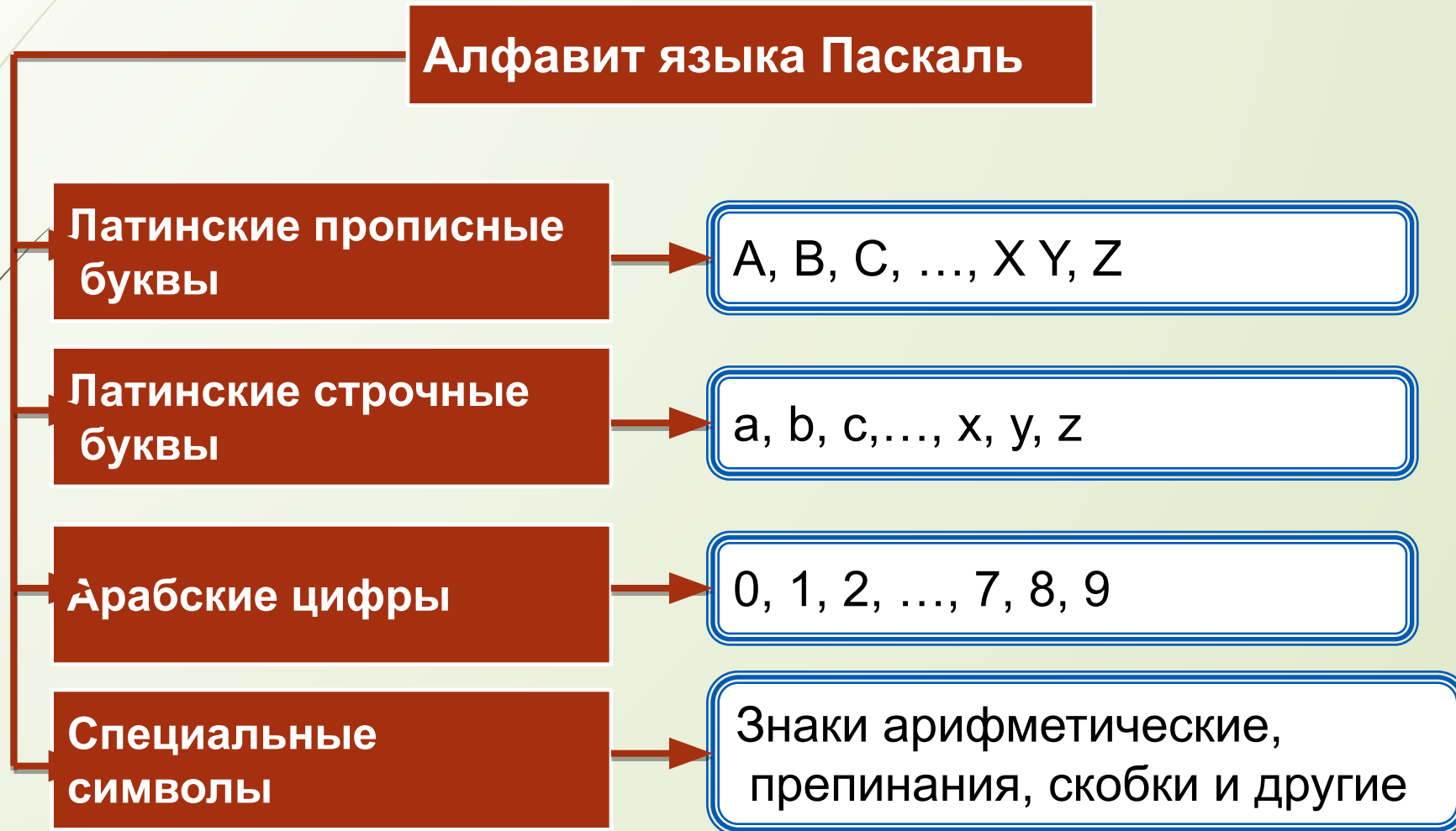




# Язык программирования Паскаль. 1 урок

# Алфавит языка

Алфавит языка программирования Паскаль - набор допустимых символов, которые можно использовать для записи программы.



# Алфавит языка

В алфавит языка Паскаль включены неделимые элементы (составные символы).



# Словарь языка

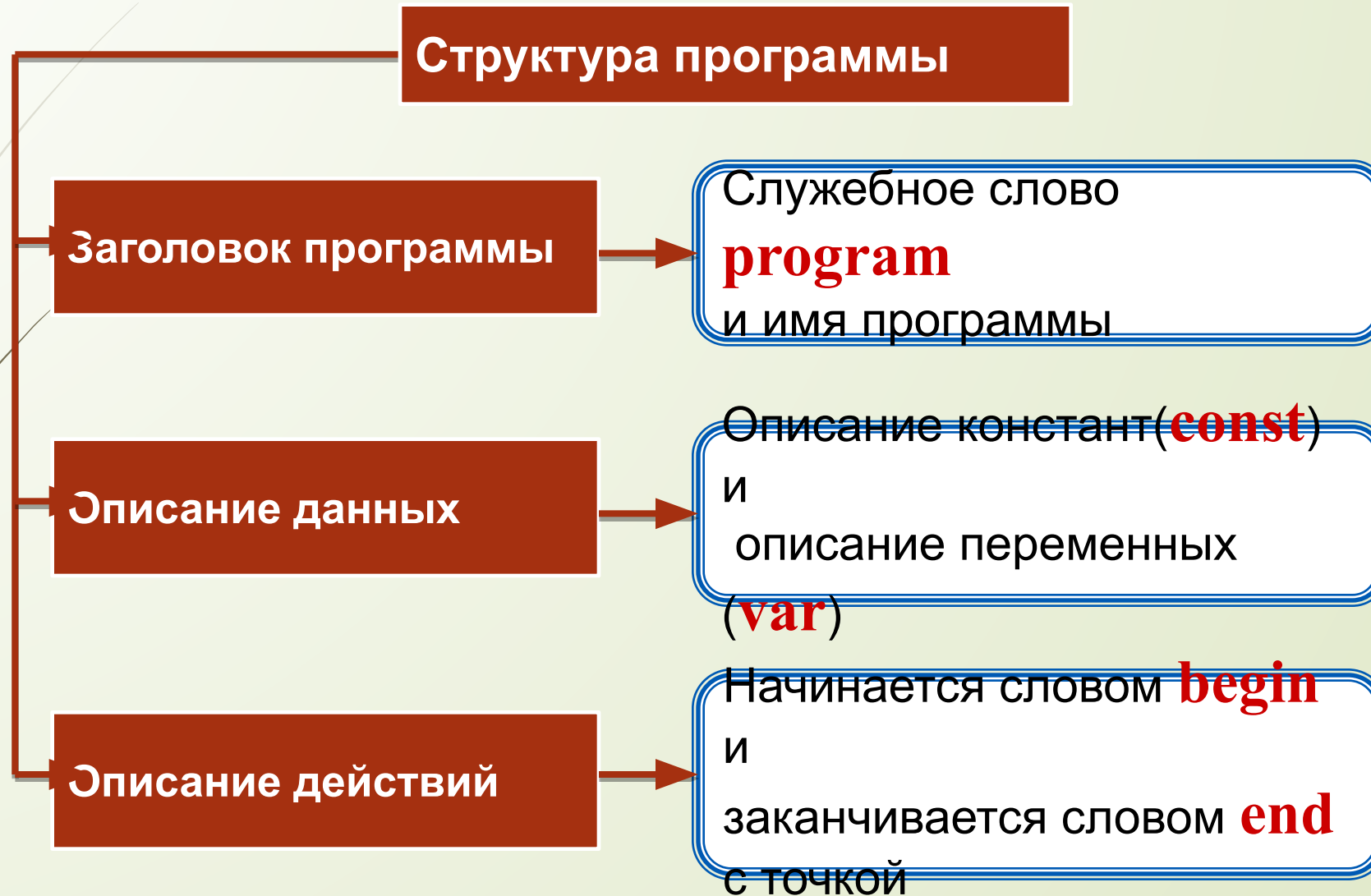
Служебное слово языка Паскаль	Значение служебного слова
<b>and</b>	и
<b>array</b>	массив
<b>begin</b>	начало
<b>do</b>	выполнить
<b>else</b>	иначе
<b>for</b>	для
<b>if</b>	если
<b>of</b>	из
<b>or</b>	или
<b>procedure</b>	процедура
<b>program</b>	программа
<b>repeat</b>	повторять
<b>then</b>	то
<b>to</b>	до (увеличивая до)
<b>until</b>	до (до тех пор, пока)
<b>var</b>	переменная
<b>while</b>	пока

# Простые типы данных

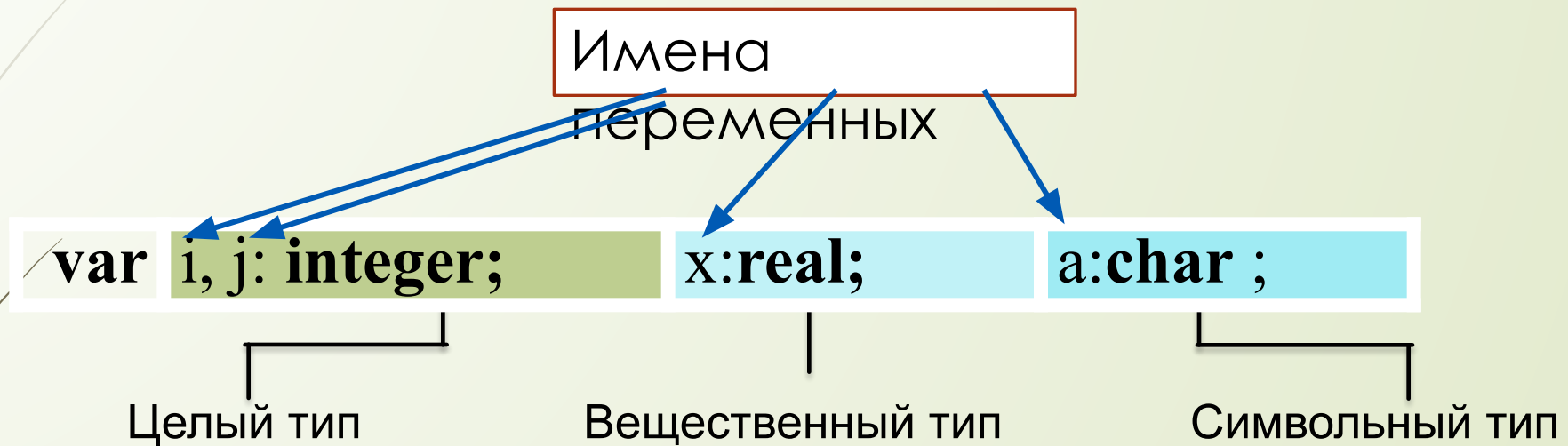
Название	Обозначение	Допустимые значения	Область памяти
Целочисленный	integer	- 32 768 ... 32 767	2 байта со знаком
Вещественный	real	$\pm(2.9 * 10^{-39} \dots 1.7 * 10^{+38})$	6 байтов
Символьный	char	Произвольный символ алфавита	1 байт
Строковый	string	Последовательность символов длиной меньше 255	1 байт на символ
Логический	boolean	True и False	1 байт



# Структура программы на языке Паскаль



# Раздел описания переменных



# Общий вид программы

```
program <имя программы>;  
  const <список постоянных значений>;  
  var <описание используемых переменных>;  
  begin <начало программного блока>  
    <оператор 1>;  
    <оператор 2>;  
    ...  
    <оператор n>  
  end.
```

**Операторы** - языковые конструкции для записи действия, выполняемого над данными в процессе решения задачи.



# Оператор присваивания

Основное преобразование данных, выполняемое компьютером, - присваивание переменной нового значения, что означает изменение содержимого области памяти.

Общий вид оператора:

**<имя переменной>:=<выражение>**

**Команда присваивания**



Файл "SWF"



# *Процедуры вывода*

## *Write и WriteLine*

(переводится – «пиши» и «пиши строку»)

С помощью данных операторов изображают на экране ту или иную информацию, состоящую из символов.

Выводить на экран можно не только числа, но и результаты вычисления арифметических выражений, а также тексты, которые, в отличие от чисел и выражений, нужно брать в одинарные кавычки.

# 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 );
```



## Примеры:

<i>Как пишем</i>	<i>Что видим</i>
Write(-500)	-500
Write(2*2-1)	3
Write('Хорошо!')	Хорошо!

## Примеры:

<i>Как пишем</i>	<i>Что видим</i>
<code>Write('Это',4+4,'Кошек')</code>	Это8Кошек
<code>Write('Это ',4+4,' кошек')</code>	Это 8 кошек
<code>Write('16+17=',16+17)</code>	16+17=33
<code>Write(3+2,' ',4)</code>	5 4
<code>Write(3+2,4)</code>	54
<code>Write('125+1',5+1,'=',120+21)</code>	125+16=141



## Пример:

Программа на Паскале. Результат выполнения

```
BEGIN
```

```
Write('Начали!');
```

```
Write(8+1);
```

```
Write(5);
```

```
END.
```

Начали!95

## *Задача 1*

Определить, что напечатает программа:

```
Begin
```

```
Write(1992);
```

```
WriteLn(' Мы начинаем!');
```

```
WriteLn(6*8);
```

```
WriteLn;
```

```
WriteLn('Шестью шесть ',6*6,'.Арифметика:',(6+4)*3);
```

```
End.
```

---

---

---

---

# Операторы ввода

- Ввод исходных данных с клавиатуры происходит по оператору `read` (читать) или `readln` (читать строку).  
Пример: **`read (список переменных)`**.





# Сложение двух чисел

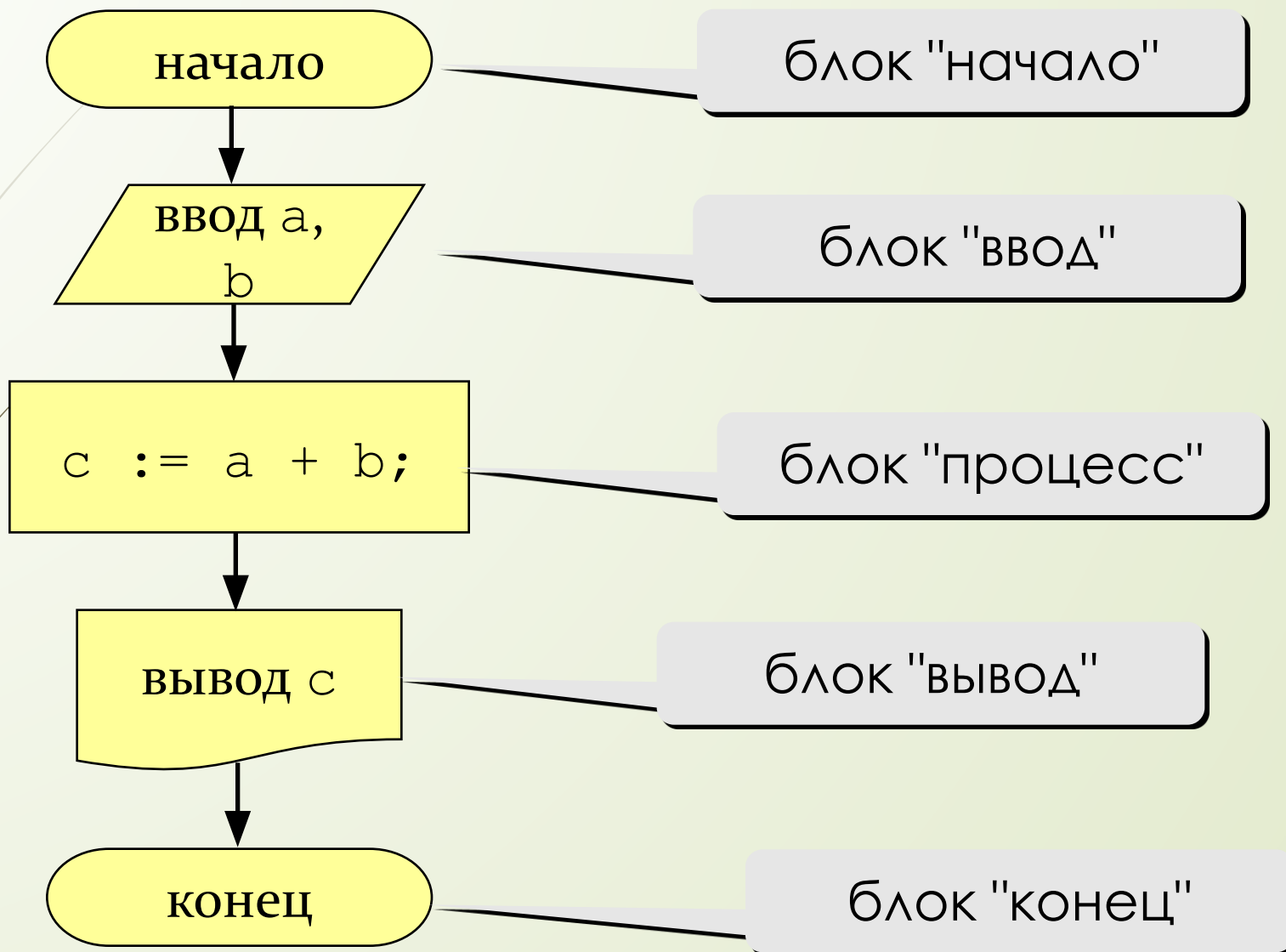
---

**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.
```

# Блок-схема линейного алгоритма





## Пример:

Begin

a:=2\*3+4;

b:=a;

y:=a+b+1;

Write('y=',y)

End.

a:=10;

b:=10;

y:=10+10+1;

y=21

Замечание. Если переменная принимает новое значение, то старое значение автоматически стирается

## 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






# ЗАДАЧА 1.

- ВЫВЕСТИ НА ЭКРАН ТРИ ВВЕДЕННЫХ С КЛАВИАТУРЫ ЧИСЛА В ПОРЯДКЕ, ОБРАТНОМ ИХ ВВОДУ.




```
□ program WriteThree;  
□ var a, b, c: integer;  
□ begin  
□   readln(a, b, c);  
□   writeln(c, ' ', b, ' ', a)  
□ end.
```



**Задача 2.** Дано натуральное число меньше 256.  
Сформировать число, представляющее собой его квадрат.

```
□ program SqrOfNum;  
□ var a: word;  
□ begin  
□   readln(a);  
□   a := a * a;  
□   writeln(a)  
□ end.
```





▣ **Задача 3** Даны 3 целых числа  $A$ ,  $B$ ,  $C$ .  
Напишите одну программу вычисления их суммы и произведения. Программа должна попросить вас ввести числа  $A$ ,  $B$ ,  $C$ . Помните, числа вводить надо через пробел и нажать после ввода последнего числа Enter

▣ **Задача 4** Даны 2 целых числа  $A$ ,  $B$ .  
Напишите программу, которая поменяет местами числа в этих переменных местах. Программа должна попросить вас ввести числа  $A$ ,  $B$ .

▣ **Задача 5** Напишите программу, которая вычисляет y по формуле:

$$\frac{5 \cos x, 3}{\ln}$$



**Задача 6.** Введите свой возраст, рост и вес.

Напишите программу, которая выводит на экран сообщение:

*Для ... - летнего возраста рост ...  
сантиметров и вес ... кг очень подходят!*

**Задача 7.** Напишите программу, которая вычисляет среднее арифметическое и среднее геометрическое 2-х чисел.