



ОБЩИЕ СВЕДЕНИЯ О ЯЗЫКЕ ПРОГРАММИРОВАНИЯ ПАСКАЛЬ

НАЧАЛА ПРОГРАММИРОВАНИЯ

9 класс

Языки программирования - это формальные языки, предназначенные для записи алгоритмов, исполнителем которых будет компьютер.

Записи алгоритмов на языках программирования называются **программами**.

Язык Паскаль – универсальный язык программирования.



Никлаус Вирт (род. в 1934 г.) - швейцарский учёный, специалист в области информатики, один из известнейших теоретиков в области разработки языков программирования, профессор информатики (компьютерных наук). Разработчик языка Паскаль и ряда других языков программирования.

Алфавит языка

В Паскале используются

Алфавит языка Паскаль

Латинские прописные
буквы

A, B, C, ..., X Y, Z

Латинские строчные
буквы

a, b, c, ..., x, y, z

Арабские цифры

0, 1, 2, ..., 7, 8, 9

Специальные
символы

Знаки арифметические,
препинания, скобки и другие

Десятичные числа записываются с точкой
2.5; 3,6

Операции над числами в языке Паскаль:

Операция	Обозначение
Сложение	+
Вычитание	-
Умножение	*
Деление	/

Десятичные числа записываются с точкой
2.5; 3,6

1. Имя переменной (записать в тетрадь)

Имена (констант, переменных, программ и других объектов) - любые отличные от служебных слов последовательности букв, цифр и символа подчеркивания, начинающиеся с буквы или символа подчеркивания.

- Именем переменной может быть:
 - Английская буква (латинская буква)
 - Сочетание английских (латинских) букв
 - Сочетание букв и цифр, начинающиеся с
- Примеры:
 - A, b, c, D, x, y
 - Summa, raznoct, aa, adc
 - A1, aa12, aa4, adc1, a2b4

Алфавит и словарь языка

Имена (констант, переменных, программ и других объектов) - любые отличные от служебных слов последовательности букв, цифр и символа подчеркивания, начинающиеся с буквы или символа подчеркивания.

Правильные имена

x
velichina
zzz
polnaja_summa
tri_plus_dva
s25
_k1
a1b88qq
oshibka



Неправильные имена

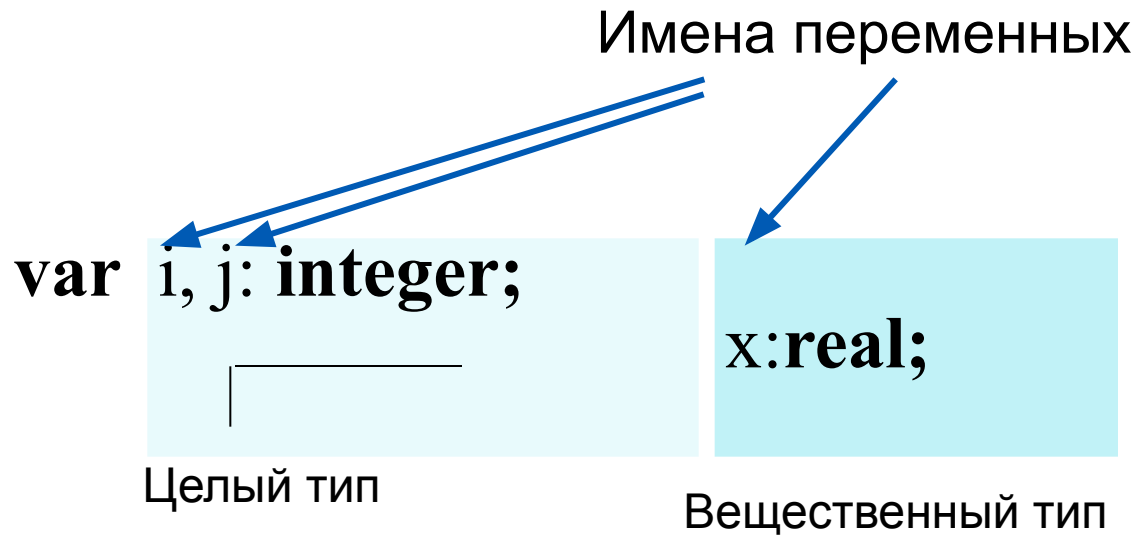
Ж - буква не латинского алфавита
polnaja summa - содержится символ (пробел), не являющийся буквой, цифрой или знаком подчеркивания.
2as - начинается с цифры
Domby&Son - содержится символ &, не являющийся буквой, цифрой или знаком подчеркивания

Длина имени может быть любой.

Простые типы данных (записать в тетрадь 2 типа целочисленный и вещественный)

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

Раздел описания переменных (записать в тетрадь)



Общий вид программы (записать в тетрадь)

```
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 (записать в

тетрадь общий вид оператора и примеры)

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

Общий вид оператора:

<имя переменной>:=<выражение>

ПРИМЕРЫ

X:=5;

X:=5*10;

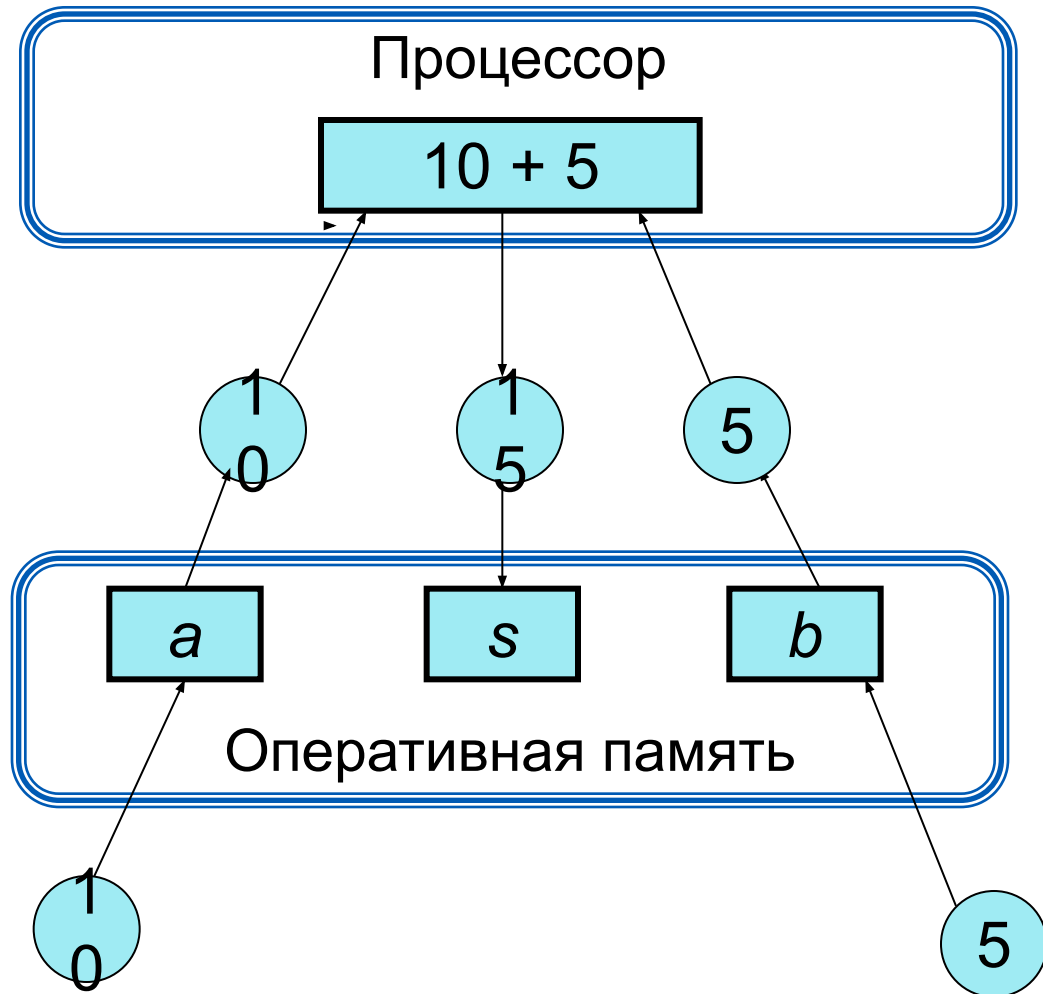
X:=Y;



Объект "SWF"

Выполнение оператора присваивания

a:=10;
b:=5;
s:=a+b



Вывод данных (записать в тетрадь)

Вывод данных из оперативной памяти на экран монитора выполняют операторы вывода ***write*** и ***writeln***.

write <выражение 1> , < выражение 2> , ..., < выражение N>

СПИСОК ВЫВОДА

Примеры	На экране
Write(5);	5
Write('текст');	текст
Write(x);	Значение x, например 7
Write(x,y);	Значение x, y, например 78
Write('x=',x);	X=7
Write(5*10);	50
Writeln(x);	7

Варианты организации вывода

Вариант организации вывода	Оператор вывода	Результат
Без разделителей	<code>write (1, 20, 300).</code>	120300
Добавить разделители – запятые	<code>write (1, ',', 20, ', ', 300)</code>	1, 20, 300
Добавить разделители – пробелы	<code>write (1, ' ', 2, ' ', 3)</code>	1 20 300

Ввод данных с клавиатуры

(записать в тетрадь)

Ввод в оперативную память значений переменных :

read (<имя переменной1>, ..., <имя переменной N>)



список ввода

Примеры

```
read(x);
```

```
read(x,y);
```

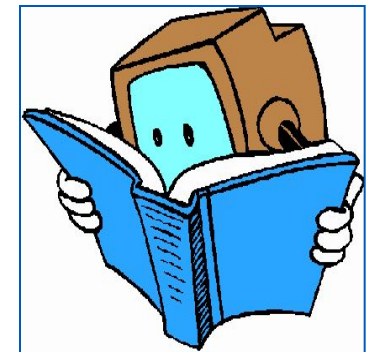
```
readln(x);
```

Ввод данных с клавиатуры

read (<имя переменной1>, ..., <имя переменной N>)

Выполнение оператора **read**:

- 1) компьютер переходит в режим ожидания данных:
- 2) пользователь вводит данные с клавиатуры:
 - несколько значений переменных числовых типов могут вводиться через пробел или через запятую;
 - при вводе символьных переменных пробел и запятую ставить их нельзя;
- 3) пользователь нажимает клавишу **Enter**.



Ввод данных с клавиатуры

! *Типы вводимых значений* должны *соответствовать типам переменных*, указанных в разделе описания переменных.

```
var i, j: integer; x: real;  
read (i, j, x);
```

варианты организации входного потока:

```
1 0 2.5 <Enter>
```

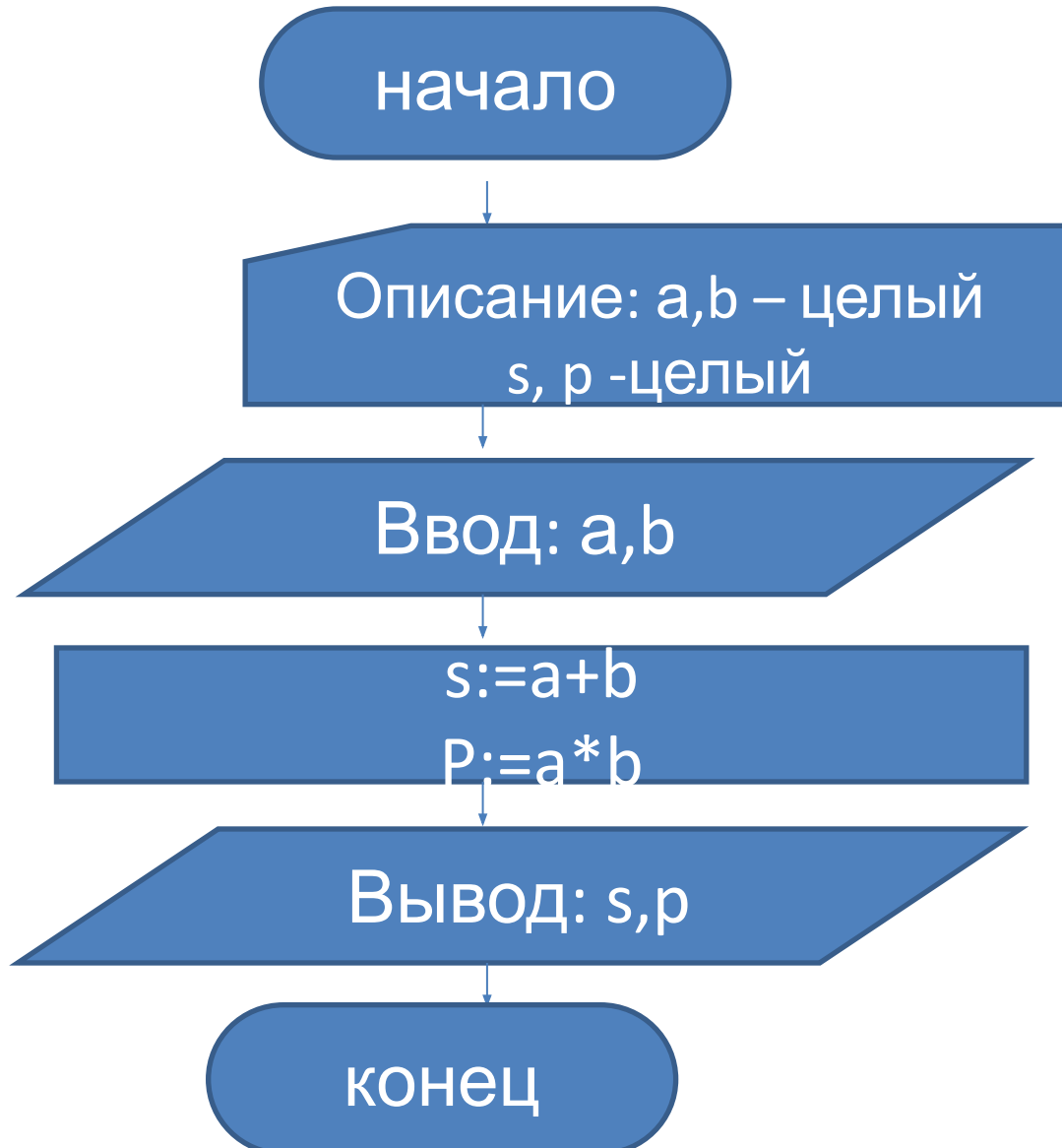
После выполнения оператора **readln** курсор переходит на новую строку.



Задача

Заданы 2 числа a, b . Найти сумму этих чисел и произведение.

- Заданы 2 числа a, b . Найти сумму этих чисел и произведение – **блок-схема**



программа

- Program n1;
- Var a,b,s,p: integer;
- Begin
- Readln(a,b);
- s:=a+b;
- p:=a*d;
- Writeln('s=',s,'p=',p)
- End.
- Название программы
- Описание переменных
- Программная скобка (начало)
- Ввод переменных
- Оператор присваивания
- Оператор присваивания
- Вывод переменных
- Конец программы