

**ОБЩИЕ СВЕДЕНИЯ О ЯЗЫКЕ**

**ПРОГРАММИРОВАНИЯ**

**ПАСКАЛЬ**

**Языки программирования –**

**это формальные языки,**

**предназначенные для записи**

**алгоритмов, исполнителем**

**которых будет компьютер.**

**Записи алгоритмов**

**на языках**

**программирования**

**называются программами.**

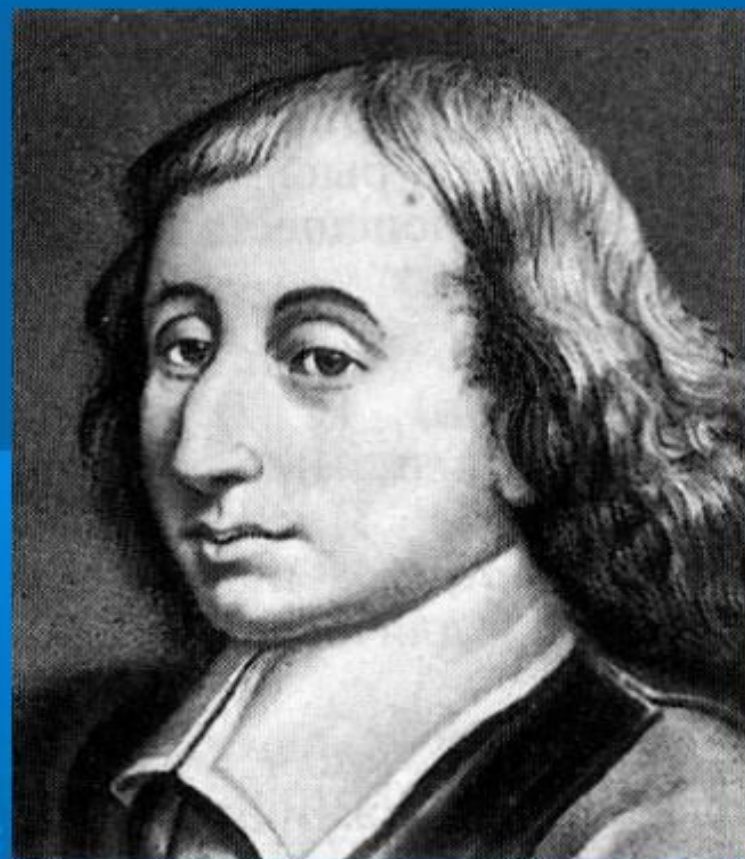


**Никлаус Вирт** (нем. *Niklaus Wirth*, род. 15 февраля, род. 15 февраля 1934 года, род. 15 февраля 1934 года) — швейцарский, род. 15 февраля 1934 года) — швейцарский учёный, специалист в области информатики, род. 15 февраля 1934 года) — швейцарский учёный, специалист в области информатики, один из известнейших теоретиков в области разработки языков, род. 15 февраля 1934 года) — швейцарский учёный, специалист в области информатики, один из известнейших теоретиков в области разработки



языко род. 15 февраля швейцарский учёный, специ ки, один из извест асти разработки профессор, специалист в известнейших теорет и Швейцарской ориха (ETHZ), Лауре род. 15 февраля 1934 года) — швейцарский учёный, специалист в области информатики, один из

Французский учёный  
**Блез Паскаль,**  
изобретатель счетной машины.



# Алфавит и словарь 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, ...8, 9)

□ Специальные символы:

знак подчёркивания; знаки препинания;  
круглые, квадратные и фигурные скобки;  
знаки арифметических операций и др.

**Существуют неделимые**

**последовательности символов:**

**:= (присваивание)**

**>= и <=**

**\* и \* (начало и конец комментария)**

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



Для обозначения констант,  
переменных, программ и др.  
объектов используются **имена** –  
любые отличные от служебных слов  
последовательности букв, цифр и  
символа подчеркивания, но  
начинаются не с цифр.

# Типы данных в языке Паскаль

*var <идентификатор>: <тип>;*

Тип	Обозначение	Допустимые значения	Область памяти	Пример записи
Целочисленный	<i>integer</i>	-32 768...32767	2 байта со знаком	<i>A, D, M: integer;</i>
Вещественный	<i>real</i>	$\pm(2,9 \cdot 10^{-39} \dots 1,7 \cdot 10^{+38})$	6 байтов с фиксированной точкой с плавающей точкой <i>мантисса E порядок</i>	<i>X1, X2: real;</i>  0.13, 4.671 2E3, 5.17E-7
Символьный	<i>char</i>	Произвольный символ алфавита	1 байт значения закljučаются в апострофы	<i>Wx, Ux: char;</i>  <i>X='B'</i>
Строковый	<i>string</i>	Последовательность символов длиной меньше 255	1 байт на символ	<i>Text: string;</i>
Логический	<i>boolean</i>	<i>true</i> и <i>false</i>	1 байт	<i>L: boolean</i>

# Структура программы на языке Паскаль

▣ Заголовок программы: *program* <заголовок>

▣ Блок описания используемых данных:

*uses* <описание внешних модулей>

*label* <описание меток>

*const* <описание констант>

*type* <описание типов переменных>

*var* <описание переменных>

*procedure* <описание процедур>

*function* <описание функций>

**Пример 1.** *var i, j: integer; x: real; a: char*

▣ Блок описания действий по преобразованию данных  
(программный блок)

*begin* <раздел операторов>

*end.* (с точкой)

## Общий вид программы

```
program    <имя программы>;  
    const    <список постоянных значений>;  
    var <описание используемых переменных>;  
begin <начало программного блока>;  
        <оператор 1>;  
        <оператор 2>;  
        <оператор 3>  
end.
```

Точка с запятой – разделитель между операторами.

Перед *end.* точку с запятой не ставят.

**Комментарии:** { } или (\* \*)

**Пример:**

1. {Вычисление переменной X}
2. (\*Не найден ни один элемент массива. Начало нового поиска\*)

## Описание меток.

Перед любым оператором можно поставить метку. Метка может состоять из букв и цифр длиной до 127 символов.

*label*

*M1, Tok, llab, 123;*

В разделе операторов после идентификатора метки ставится двоеточие

*M1: <оператор>*

*Tok:<оператор>*

## Описание констант:

*const*

*Inf=1024;* (Целая константа)

*Comunik= «Связь установлена»;* (Строковая  
константа)

*Ref=13.45;* (Вещественная константа)



## Домашнее задание

**№1. Запишите раздел описания переменных, необходимых для вычисления:**

- 1) Значения функции  $y=3x^2+5$
- 2) Площади круга
- 3) Суммы нескольких одинаковых монет

**№2. Запишите оператор для:**

- 1) Значения функции  $y=3x^2+5$
- 2) Площади круга
- 3) Суммы нескольких одинаковых монет

**Организация  
ВВОДА И ВЫВОДА  
ДАННЫХ**

**Пример 1.** *write (U, V, ..Z);*

где *U, V, ..Z* – это выражения типа *integer, real* и т.д.

**Пример 2.** *write ('s=',s);*

если *s* целочисленное, = 15, то на экране появится *s=15*

если *s* – вещественное, то *s=1.5E+01*

**Пример 3.** *write (1, 20, 300);* на экране появится  
120300

Сделать ответ более читаемым:

*write (1, ',', 20, ',', 300);* - добавили запятые

*write (1, ' ', 20, ' ', 300);* – добавили пробелы

*write (1:3, 20:4, 300:5);* - указали формат вывода

**Для вывода вещественного числа в списке вывода для каждого выражения указываются два параметра:**

- Общее количество позиций, отводимых под число**
- Количество позиций в дробной части числа**

## Пример 4.

*write ('s=', s:2:0); s=15*

*write ('s=', s:3:1); s=15.0*

*write ('s=', s:5:1); s= 15.0*

При выполнении нового оператора **write** вывод продолжается в той же строке.

Чтобы осуществить переход к новой строке, используется оператор **writeln.**

**Пример 5.** Вывести на экран «Привет, друзья!»

```
program Hello;
```

```
begin
```

```
Writeln ('Привет, друзья!'); Writeln ('Как дела?');
```

```
end.
```



**Пример 6.** Вычислить длину окружности и площадь круга, где  $r=5,4$  см,  $c=2\pi r$ ,  $s=\pi r^2$

*program n\_6;*

*const pi=3.14;*

*var r, c, s: real;*

*begin*

*r:=5.4;*

*c:=2\*pi\*r;*

*s:=pi\*r\*r;*

*writeln ('c=', c:6:4);    writeln ('s=', s:6:4)*

*end.*

*Ответ (c=33.9120, s=91.5624)*

*Для выполнения программы  
запустить компиляцию **Alt+F9**  
или меню **Compile-Compile**  
(исправление ошибок),  
сохранить программу,  
запустить на выполнение  
**Ctrl+F9** или меню **Run-Run***

## Пример 7.

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

**организуем входной поток тремя способами:**

**1 0 2,5 a <Enter>**

**1,0,2.5<Enter>**

**1<Enter>**

**a <Enter>**

**0 <Enter>**

**2.5<Enter>**

**a <Enter>**

**Усовершенствуем пример 6. Введем данные с помощью оператора *read*, выведем текстовые сообщения *writeln***

```
program n_8;
```

```
  const pi=3.14;
```

```
  var r, c, s: real;
```

```
begin
```

```
  writeln ('вычисление длины окружности и площади круга');
```

```
  writeln ('введите r>>');
```

```
  readln (r);
```

```
  c:=2*pi*r;
```

```
  s:=pi*r*r;
```

```
  writeln ('c=', c:6:4);   writeln ('s=', c:6:4)
```

```
end.
```

```
r=8.5
```

```
ОТВЕТ (c=53.3800, s=226.8650)
```

## **Домашнее задание**

**Написать программу вычисления площади и периметра прямоугольника .**