

---

# **Элементы языка Паскаль. Стандартные математические функции**

# История Паскаля

Язык программирования Паскаль был создан швейцарским профессором Никлаусом Виртом в 1969 году как язык обучения студентов структурной методике программирования. Язык получил свое название в честь Блеза Паскаля – изобретателя первого вычислительного механического устройства. Позднее фирма Borland International, Inc (США) разработала систему программирования Турбо Паскаль для ПК, которая вышла за рамки учебного применения и стала использоваться для научных и производственных целей.



[Никлаус Вирт](#)



[Блез Паскаль](#)

# Язык Паскаль

## Причины популярности языка Паскаль

Прост для обучения.

Отражает фундаментальные идеи алгоритмов в легко воспринимаемой форме, что предоставляет программисту средства, помогающие проектировать программы.

Позволяет четко реализовать идеи структурного программирования и структурной организации данных.

Использование простых и гибких структур управления: ветвлений, циклов.

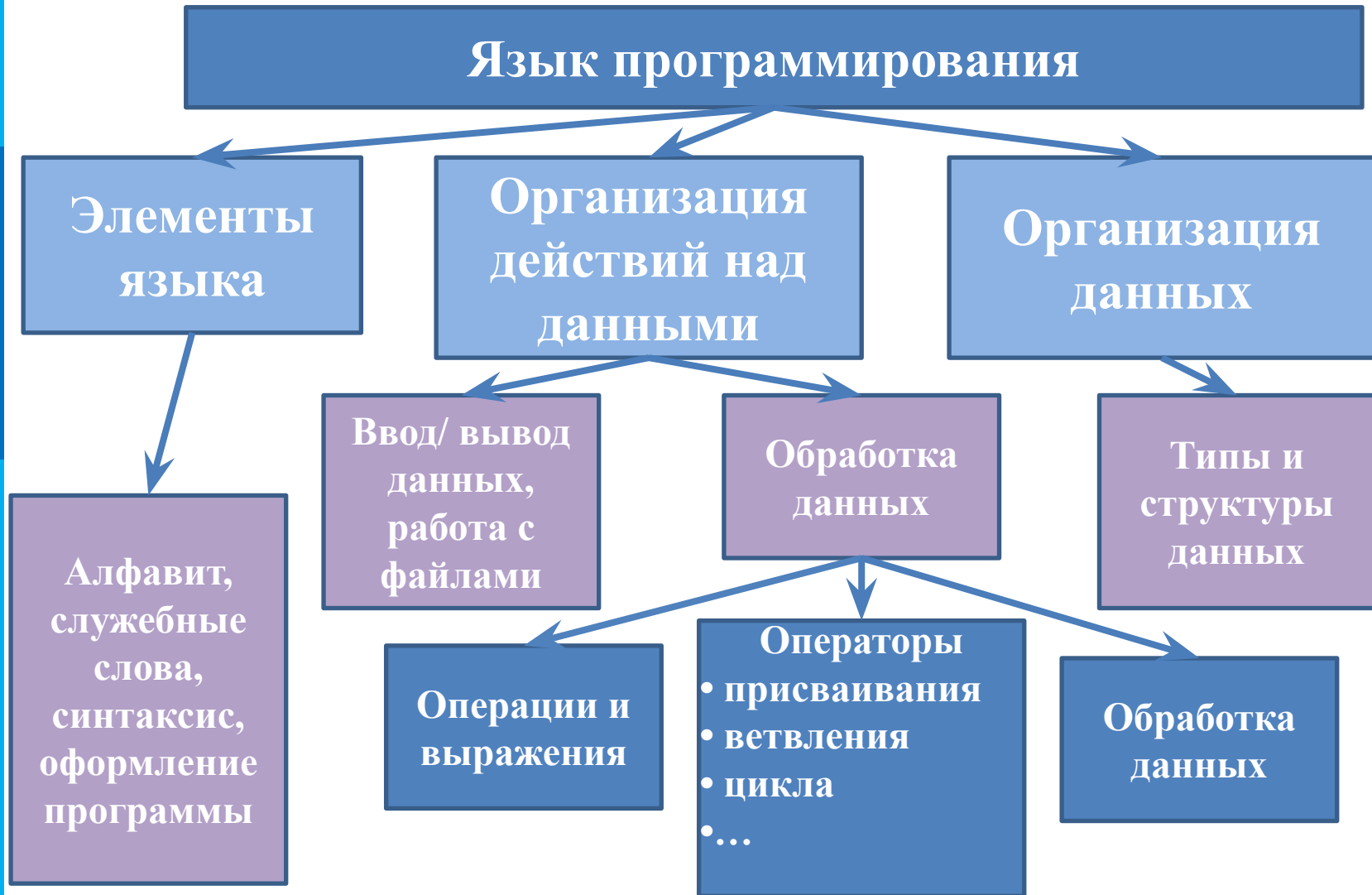
Надежность разрабатываемых программ.

# Языки программирования

---

- **Машинно-ориентированные (низкого уровня)** - каждая команда соответствует одной команде процессора (ассемблер)
- **Языки высокого уровня** – приближены к естественному (английскому) языку, легче воспринимаются человеком, не зависят от конкретного компьютера
  - *для обучения*: Бейсик, ЛОГО, Паскаль
  - *профессиональные*: Си, Фортран, Паскаль
  - *для задач искусственного интеллекта*: Пролог, ЛИСП
  - *для Интернета*: JavaScript, Java, Perl, PHP, ASP

# Структура процедурных языков программирования

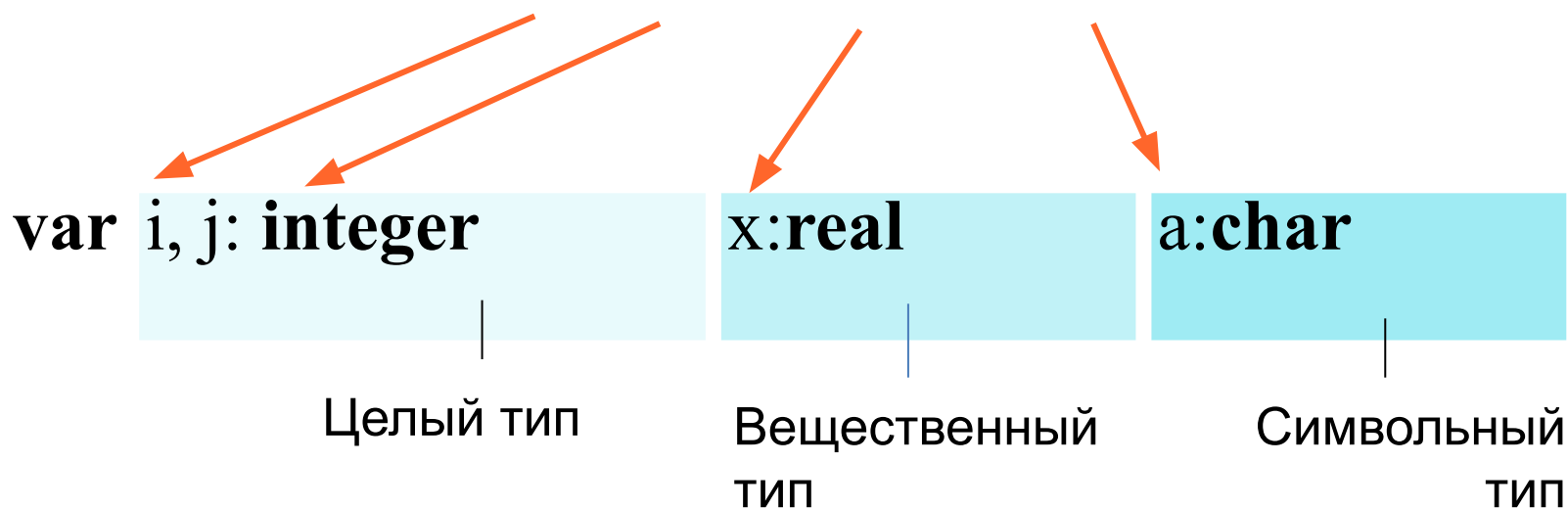


# Структура программы на языке Паскаль



# Раздел описания переменных

## Имена 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.



# Пример программы на языке Паскаль

```
Program z1;
```

```
{ Программа для вычисления площади прямоугольника по  
заданным сторонам }
```

```
Var
```

```
    a,b,s : integer;
```

```
Begin
```

```
    writeln ('Введите стороны A и B');
```

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

```
    s:=a*b;
```

```
    write ('S=',s,' кв.см.')
```

```
End.
```

# Элементы языка Pascal



## Алфавит языка —

набор допустимых символов, которые можно использовать для записи программы.

## Неделимые элементы

составные символы, которые нельзя

разрывать (разделять пробелом):

`:=` знак присваивания;

`>=` больше или равно;

`<=` меньше или равно;

`(* *)` ограничители комментариев (наряду с `{ }`);

`<>` не равно;

`(..)` эквивалент `[ ]`.

## Алфавит языка Pascal:

- латинские прописные буквы (A, B, ... Z);
- латинские строчные буквы (a, b, ... z);
- арабские цифры (0, 1, 2, 3, 4, 5, 6, 7, 8, 9);

## Специальные символы:

- знаки препинания: «,», «.», «:» и т. д.;
- знак подчёркивания «\_»;
- круглые скобки «( )»;
- квадратные скобки «[ ]»;
- фигурные скобки «{ }»;
- знаки арифметических



# Служебные слова

**Служебные слова** — цепочки символов, которые рассматриваются как единые смысловые элементы с фиксированным значением.

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

# Идентификаторы

Имена, даваемые программным объектам (константам, типам данных, переменным, функциям и процедурам, да и всей программе целиком) называются **идентификаторами**.

**Должно удовлетворять следующим требованиям:**

длина имени не должна превышать 63 символа,

первым символом не может быть цифра,

переменная не может содержать пробел;

имя не должно совпадать с зарезервированным (служебным) словом;

прописные и строчные буквы воспринимаются одинаково.

# Идентификаторы

## Правильные имена

x

velichina

zzz

polnaja\_summ

a

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



# Комментарии

**Комментарий** - это строка (или несколько строк) из произвольных символов, заключенная в фигурные скобки:

*{ комментарий }*

Другой вариант оформления комментария:

*(\* комментарий \*)*

Внутри самого комментария символы } или \*) встречаться не должны.

Буквы русского алфавита употребляются только в комментариях, символьных и текстовых константах.

# Из чего состоит программа?

---

**Константа** – постоянная величина, имеющая ИМЯ.

**Переменная** – изменяющаяся величина, имеющая имя (ячейка памяти).

**Процедура** – вспомогательный алгоритм, описывающий некоторые действия (рисование окружности).

**Функция** – вспомогательный алгоритм для выполнения вычислений (вычисление квадратного корня,  $\sin$ ).

# Константы

```
const
```

```
    i2 = 45; { целое число }
```

```
    pi = 3.14; { вещественное число }
```

целая и дробная часть отделяются точкой

```
    qq = 'Вася'; { строка символов }
```

можно использовать русские буквы!

```
    L = True; { логическая величина }
```

может принимать два значения:

- True (истина, «да»)
- False (ложь, «нет»)



# Переменные

---

**Переменная** – это величина, имеющая имя, тип и значение. Значение переменной можно изменять во время работы программы.

**Типы переменных:**

- integer                    { целая }
- real                        { вещественная }
- char                        { один символ }
- string                      { символьная строка }
- boolean                    { логическая }

**Объявление переменных (выделение памяти):**

```
var a, b: integer;  
    Q: real;  
    s1, s2: string;
```

# Как изменить значение переменной?

---

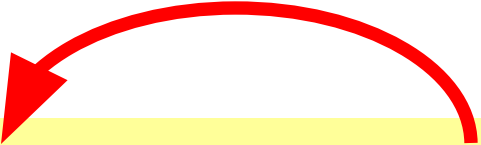
**Оператор** – это команда языка программирования высокого уровня.

**Оператор присваивания** служит для изменения значения переменной.

**Пример:**

```
program qq;  
var a, b: integer;  
begin  
    a := 5;  
    b := a + 2;  
    a := (a + 2) * (b - 3);  
end.
```

# Оператор присваивания



```
<имя переменной> := <выражение>;
```

Арифметическое выражение может включать

- КОНСТАНТЫ
- имена переменных
- знаки арифметических операций:

+ - \* /

умножение

деление

div

деление  
нацело

mod

остаток от  
деления

- ВЫЗОВЫ функций
- круглые скобки ( )

# 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.
```

# Бинарные арифметические операции



Знак	Выражение	Типы операндов	Тип результата	Операция
+	$A+B$	R, R I, I I, R; R, I	R I R	Сложение
-	$A-B$	R, R I, I I, R; R, I	R I R	Вычитание
*	$A*B$	R, R I, I I, R; R, I	R I R	Умножение
/	$A/B$	R, R I, I I, R; R, I	R R R	Вещественное деление
div	$A \text{ div } B$	I, I	I	Целочисленное деление

# Бинарные арифметические операции



A:=5.

B:=2.

C:=A div B.

A:=7.

B:=2.

C:=A div B.

# Бинарные арифметические операции



Знак	Выражение	Типы операндов	Тип результата	Операция
+	$A+B$	R, R I, I I, R; R, I	R I R	Сложение
-	$A-B$	R, R I, I I, R; R, I	R I R	Вычитание
*	$A*B$	R, R I, I I, R; R, I	R I R	Умножение
/	$A/B$	R, R I, I I, R; R, I	R R R	Вещественное деление
div	$A \text{ div } B$	I, I	I	Целочисленное деление
mod	$A \text{ mod } B$	I, I	I	Остаток от целочисленного деления

# Бинарные арифметические операции



A:=5.

B:=2.

C:=A mod B.

A:=8.

B:=3.

C:=A mod B.



# Стандартные математические функции



Обращение	Тип аргумента	Тип результата	Функция	Обращение	Тип аргумента	Тип результата	Функция
$\pi$	–	R	Возвращает число $\pi$ с 19 знаками после запятой	<b>random</b>	–	R	Псевдослучайное число в интервале [0, 1)
<b>abs (a)</b>	I, R	I, R	Модуль аргумента	<b>random (a)</b>	I	I	Псевдослучайное число в интервале [0, a)
<b>arctan (a)</b>	I, R	R	Арктангенс (в радианах)	<b>round (a)</b>	R	I	Округление до ближайшего целого числа
<b>cos (a)</b>	I, R	R	Косинус (в радианах)	<b>sin (a)</b>	I, R	R	Синус (в радианах)
<b>exp (a)</b>	I, R	R	$e^x$ – экспонента	<b>sqr (a)</b>	I, R	I, R	Квадрат a
<b>frac (a)</b>	I, R	R	Дробная часть a	<b>sqrt (a)</b>	I, R	R	Корень квадратный
<b>int (a)</b>	I, R	R	Целая часть a	<b>trunc (a)</b>	R	I	Ближайшее целое, не превышающее a по модулю
<b>ln (a)</b>	I, R	R	Натуральный				

# Пример



Заполнить таблицу, вписав в столбцы значения выражений.

Выражение	Значение выражения	Выражение	Значение выражения
abs (5.7)		round (6.3)	
abs (-3.14)		round (4.8)	
frac (4.5)		sqr (4)	
int (6.98)		sqrt (9)	
int (6.1)		trunc (8.5)	

# Пример



Заполнить таблицу, вписав в столбцы значения выражений.

Выражение	Значение выражения	Выражение	Значение выражения
<b>abs (5.7)</b>	5.7	<b>round (6.3)</b>	6
<b>abs (-3.14)</b>	3.14	<b>round (4.8)</b>	5
<b>frac (4.5)</b>	0.5	<b>sqr (4)</b>	16
<b>int (6.98)</b>	6	<b>sqrt (9)</b>	3
<b>int (6.1)</b>	6	<b>trunc (8.5)</b>	8

# Арифметические выражения



Арифметическое выражение –  
последовательность числовых  
констант,  
переменных, математических  
функций,

соединённых знаками

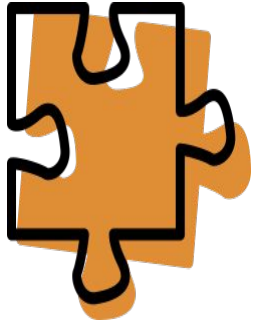
арифметических (переменная)

операций и формул арифметического  
выражения.





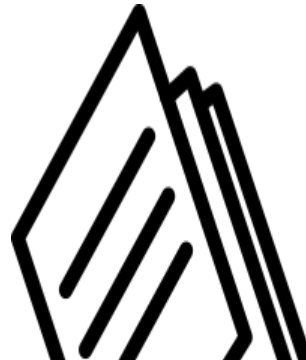
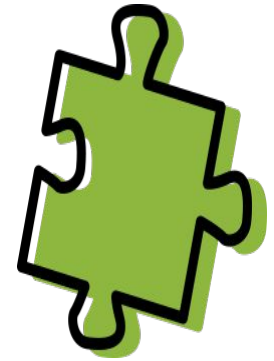
## Пример



$$\frac{\ln(y) + \sin x}{2y + \sqrt{y} + x}$$

На языке Pascal:

$(\ln(y) + \sin(x)) / (2 * y + \text{sqrt}(y) + x)$



# Правила записи арифметических выражений



- I. Все символы нужно писать в строчку на одном уровне.  
Все знаки операций должны проставляться.
- II. При записи выражений нельзя допускать два идущих друг за другом арифметических знака.
- III. Необходимо соблюдать порядок выполнения действий:
  - 1) функции;
  - 2) унарная операция изменения знака (–) и возведение в степень (^);
  - 3) \*, /, div, mod;
  - 4) +, –.
- IV. Если несколько операций одинакового приоритета записаны подряд, то нужно выполнять их слева направо.
- V. Выражение в скобках вычисляется в первую очередь.

# Правила записи арифметических выражений



$$(\ln(y)+\sin(x))/(2*y+\text{sqrt}(y)+x)$$

# Порядок выполнения операций

---

```
z := (5*a*c+3*(c-d)) / (a*b);
```



# Порядок выполнения операций

---

$$x = \frac{a^2 + 5c^2 - d(a + b)}{(c + d)(d - 2a)}$$

**Подготовиться  
к тесту**