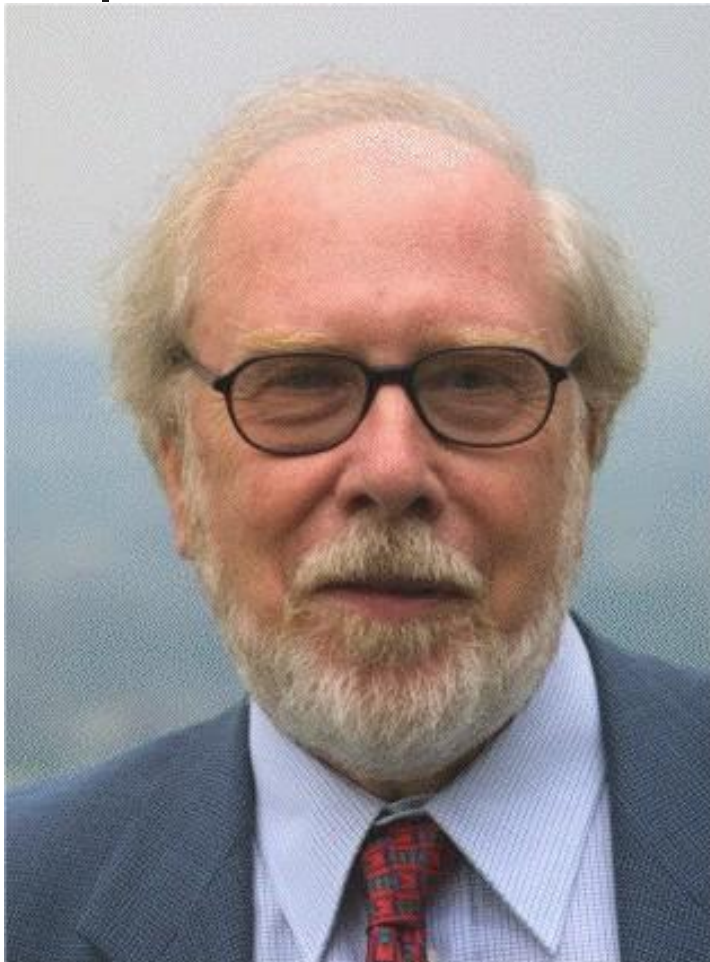




Язык Паскаль

Основы программирования

Немного истории...



Язык Паскаль был создан Никлаусом Виртом в 1968–1969 годах. Он был опубликован в 1970 году как небольшой и эффективный язык. Язык был назван в честь французского ученого Блеза Паскаля.



Алфавит языка

1. Прописные и строчные буквы латинского алфавита от **A до z**, а также символ подчеркивания (**_**), который тоже считается буквой.

2. цифры **0 1 2 3 4 5 6 7 8 9**;

3. Специальные одиночные знаки: **+ - * / = < > . , : ; ^ \$ # @**;

4. Специальные парные знаки: **[] () { } ; < = > = < >**.

Структура языка Паскаль:



```
program <имя программы>;
```

```
  var...; {описание переменных}
```

```
  Begin {начало}
```

```
    ... {основная программа}
```

```
  end. {конец}
```

Имена программы, переменных

Имена могут включать

1. латинские буквы (A-Z)
2. Цифры от 0 до 9

ИМЯ НЕ МОЖЕТ НАЧИНАТЬСЯ С ЦИФРЫ

3. знак подчеркивания _

Какие имена правильные??

AXby R&B 4Wheel Вася "PesBarbos" TU154
[QuQu] _ABBA A+B

Переменные

Переменная – это величина, имеющая имя,

Типы переменных:

тип и значение..

- `integer` { целая }
- `real` { вещественная }
- И. д.

Объявление переменных (выделение памяти):

`variable` –
переменная

тип – целые

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

список имен переменных

Оператор присваивания

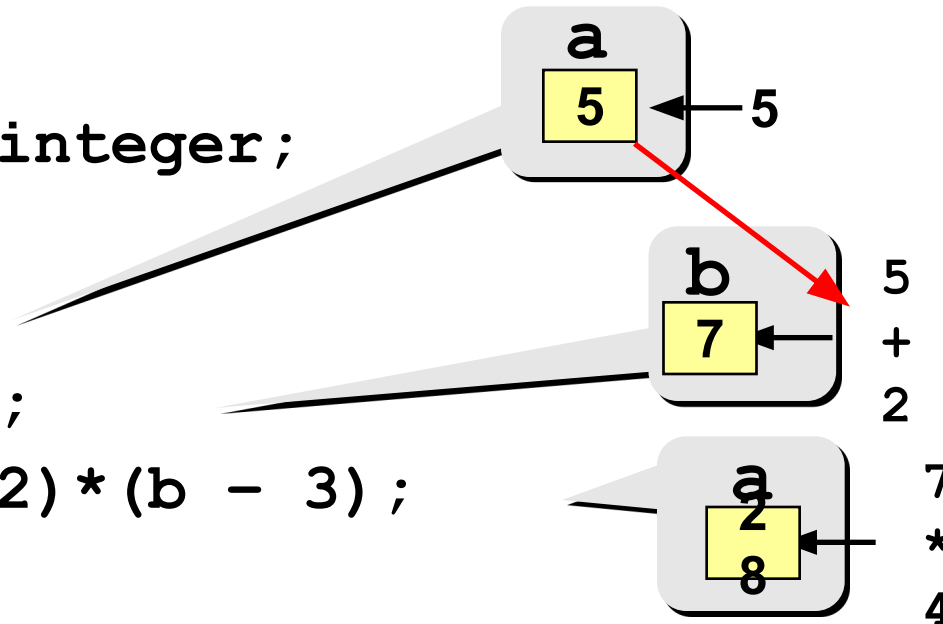
Оператор присваивания служит для изменения значения переменной.

Структура:

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

Пример:

```
program PR1;  
  var a, b: integer;  
begin  
  a := 5;  
  b := a + 2;  
  a := (a + 2) * (b - 3);  
end.
```



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

Вычисление математических выражений

Арифметическое выражение может включать

- константы
- имена переменных
- знаки арифметических операций:

+ - * / div mod

- вызов: умножение
- круглые скобки ()

деление

деление
нацело

остаток от
деления

Порядок выполнения операций

- вычисление выражений в скобках
- умножение, деление, div, mod слева направо
- сложение и вычитание слева направо

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

2 6 3 4 7 5 1 12 8 11 10

9

```
x:=(a*a+5*c*c-d*(a+b))/((c+d)*(d-2*a));
```

Примеры

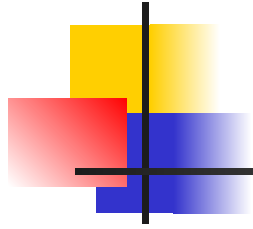
Математическая запись	Запись на языке Pascal
$37(25+87,5)-17(4,6+1,9)$	$37*(25+87.5)-17*(4.6+1.9)$
$\frac{ab}{c} + d^4$	$a*b/c+d*d*d*d$
$\frac{a+2b-3c}{5a+4}$	$(a+2*b-3*c)/(5*a+4)$
$\frac{a+b}{a-b} + \frac{ab}{3,14}$	$(a+b)/(a-b)+a*b/3.14$

Десятичная запятая в Pascal обозначается точкой

Стандартные функции

Математическая запись	Функция Паскаля	Название
$ X $	Abs(x)	Модуль числа X
X^2	Sqr(x)	Возведение числа в квадрат
e^x	Exp(x)	Экспонента
	Sqrt (x)	Вычисление квадратного корня
$\sin x$	Sin(x)	Вычисление синуса
$\cos x$	Cos(x)	Вычисление косинуса
$\ln x$	Ln(x)	Вычисление логарифма

Вывод данных



Вывод данных из оперативной памяти на экран монитора :

Write (WriteIn) ('список вывода') ;

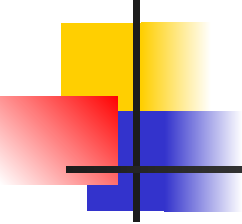
Разница между операторами лишь в том, что **writeln** перемещает курсор на начало новой строки, а **write** нет.

Пример :

```
Write ( 'введите текст' ) ;
```

```
Writeln ( 's=' , s ) ;
```

Вывод данных



```
Program pr1;  
BEGIN  
  Write('Начали!');  
  Write(8+1);  
  Write(5);  
END.
```

Результат выполнения

Начали!95

```
Program pr1;  
BEGIN  
  Writeln('Начали!');  
  Writeln(8+1);  
  Writeln(5);  
END.
```

Результат выполнения

Начали!

9

5



Оператор вывода

```
Writeln('Мне16 лет');
```

На экране появится: Мне 16 лет

```
      a:=11;
```

```
Writeln('Я учусь в ', a, ' классе');
```

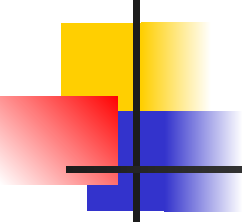
На экране появится: Я учусь в 11 классе

```
      x:=3; y:=6;
```

```
Writeln(x+y, ' ', 'рублей');
```

На экране появится: 9 рублей

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



Read (Readln) (имя переменной) ;

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

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

Задача. Вычислить сумму двух чисел

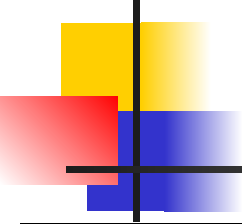
```
program pr_1;  
Uses Crt;  
var a, b, c: integer;  
begin  
    Clrscr;  
    writeln('Введите два целых числа');  
    read ( a, b );  
    c := a + b;  
    writeln ( a, '+', b, '=', c );  
    Введите два целых числа  
end.
```

Протокол:

25 30

25+30=55

Избежание ошибок



F9 – Компиляция +
проверка
Ctrl+F9 – Запуск



Задания

Ввести три числа, найти их сумму и произведение.

Пример:

Введите три числа:

4 5 7

$$4+5+7=16$$

$$4*5*7=140$$