

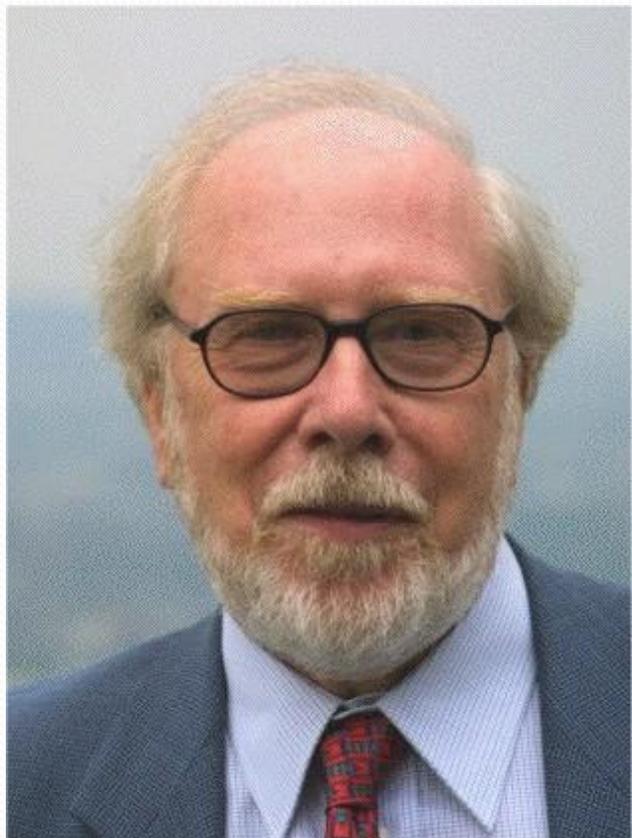
# ОСНОВЫ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ PASCAL

# Немного истории...



Язык назван в честь выдающегося французского математика, физика, литератора и философа Блеза Паскаля, который создал первую в мире механическую машину, складывающую два числа.

# Немного истории...



Язык Pascal был создан Никлаусом Виртом в 1968—1969 годах. Он был опубликован в 1970 году как небольшой и эффективный язык, чтобы способствовать хорошему стилю программирования, использовать структурное программирование и структурированные данные.

# Пунктуация Pascal

Необходимо строгое соблюдение правописания (синтаксиса) программы. В частности, в Паскале однозначно определено назначение знаков пунктуации.

**Точка с запятой (;)** ставится в конце заголовка программы, в конце раздела описания переменных, является разделителем операторов.

**Запятая (,)** является разделителем элементов во всевозможных списках: списке переменных в разделе описания, списке вводимых и выводимых величин.

Строгий синтаксис в языке программирования необходим потому, что компьютер является формальным исполнителем программы. Если, допустим, разделителем в списке переменных должна быть запятая, то любой другой знак будет восприниматься как ошибка.

# Структура программы

**Program** Имя программы;  
**Const** Введение констант;  
**Var** Описание переменных;  
**Begin**  
    Тело основной программы;  
**End.**

# Константы

*Константы* – это данные, значения которых в процессе выполнения программы не могут изменяться.

Константы вводятся в блоке `const`:

```
const c=5; b=1E-3/a;
```

В общем виде:

**ИМЯ КОНСТАНТЫ = выражение;**

# Переменные

*Переменные* – это данные, которые могут изменяться в процессе выполнения программы.

Переменные имеют имя, тип и значение.

Описание переменных происходит в блоке `var`:

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

В общем виде:

**имя переменной: тип переменной;**

# Типы переменных

Некоторые простые типы:

- 1. Целые типы** (ShortInt, Integer, LongInt, Byte, Word).
- 2. Вещественные типы** (Real, Single, Double, Extended, Comp).
- 3. Логический** (Boolean).
- 4. Символьный** (Char).
- 5. Строковые типы** (String, String [n]).

# Целочисленные типы:

| Тип      | Диапазон                                  | Формат      | Размер в байтах |
|----------|---|-------------|-----------------|
| Byte     | 0..255                                    | Беззнаковый | 1               |
| Word     | 0..65535                                  | Беззнаковый | 2               |
| Integer  | -32768..32767                             | Знаковый    | 2               |
| Cardinal | =LongWord                                 | Беззнаковый | 4               |
| LongWord | 0..4294967295                             | Беззнаковый | 4               |
| LongInt  | -2147483648..2147483647                   | Знаковый    | 4               |
| Int64    | -9223372036854775808..9223372036854775807 | Знаковый    | 8               |
| QWord    | 0..18446744073709551615                   | Беззнаковый | 8               |

# Вещественные типы:

| Тип      | Диапазон   | Количество значащих цифр | Размер в байтах |
|----------|--|--------------------------|-----------------|
| Double   | $5.0 \times 10^{-324} \dots 1.7 \times 10^{308}$   | 15-16                    | 8               |
| Real     | $2.9 \times 10^{-39} \dots 1.7 \times 10^{38}$     | 11-12                    | 6               |
| Single   | $1.5 \times 10^{-45} \dots 3.4 \times 10^{38}$     | 7-8                      | 4               |
| Extended | $1.9 \times 10^{-4932} \dots 1.1 \times 10^{4932}$ | 19-20                    | 10              |
| Comp     | $-2 \times 10^{64} + 1 \dots 2 \times 10^{63} - 1$ | 19-20                    | 8               |
| Currency | $-922337203685477.5808 \dots 922337203685477.5807$ | 19-20                    | 8               |

# Оператор присваивания:

Общий вид:

**переменная := выражение;**

Может быть: переменной,  
элементом массива и др.

Не может быть:  
константа, выражение.

Может быть: константой,  
переменной, элементом  
массива, арифметическим  
или логическим  
выражением.

Работа оператора: если справа стоит выражение, то сначала вычисляется его значение, а затем это значение пересылается в переменную стоящую слева.

**Арифметические выражения** – это конструкции, содержащие данные, знаки математических операций, математические функции.

## Арифметические операции

+

сложение

-

вычитание

\*

умножение

/

деление

div

деление  
нацело

mo  
d

остаток от  
деления

## Функции

ABS (x)

|x|

SQR (x)

$x^2$

SQRT (x)

$\sqrt{x}$

SIN (x)

*Sin x*

COS (x)

*Cos x*

# Операторы ввода и вывода

Оператор ввода:

```
read (список переменных);  
readln (список переменных);
```

Оператор вывода:

```
write ('сообщение', список переменных);  
writeln ('сообщение', список переменных);
```

Отличие операторов **read** и **write** от операторов **readln** и **writeln** состоит в том, что после выполнения операторов **readln** и **writeln** курсор переводится на новую строку.

# Домашнее задание

§ 35 (№9 в тетради)