



Basic

Урок 1

# Языки программирования

Под *языком программирования* мы будем понимать совокупность средств и правил представления алгоритма в виде, приемлемом для компьютера.

Существует разделение всех языков программирования на две большие группы — языки *высокого* и *низкого* уровня.

*Уровень языка программирования* определяется степенью его удаленности от формального языка процессора и естественного языка человека. Чем выше уровень, тем дальше от компьютера и ближе к человеку.

# Уровни языков программирования

Низкого уровня

Высокого уровня

**Машинные языки**  
Команды процессора

**Машинно-ориентированные языки**  
Ассемблер, Автокол

**Машинно-независимые языки**

**Процедурные языки**  
Бейсик, Паскаль

**Декларативные языки**

**Объектно-ориентированные языки**  
Джава, Делфи

**Логические языки**  
Пролог

**Функциональные языки**  
Лисп

# Немного истории

B - *Beginner's*  
В - *Вводная*

A - *Assembly*  
А - *Ассемблея*

S - *Symbolic*  
S - *Символических*

I - *Instruction*  
I - *команд*

C - *Code*  
C - *для начинающих*

# Немного истории

В середине 60-х годов (*1 мая 1964 г.*) два профессора Дартмутского Колледжа (США), *Дж. Кемени* и *Т.Куртс*, создали язык программирования для людей, которые не являются профессиональными программистами, но имеют дело с компьютерами.

Ныне Бейсик вышел за рамки языка для начинающих, и его могучий потомок — **Visual Basic** позволяет творить на компьютере чудеса.

# Основные части программы

1. **Входной блок.** Программа должна получить данные от пользователя.
2. **Блок данных.** Процедура ввода в этом блоке выполняется один раз, при наборе программы, а данные можно использовать многократно. Эта первые два блока программы можно менять между собой местами.
3. **Вычислительный блок.** В нем происходит обработка данных (вычисления).
4. **Выходной блок** - вывод результата на экран.

# Алфавит языка

- все прописные (A-Z) и строчные (a-z) буквы латинского алфавита;
- все прописные (А-Я) и строчные (а-я) буквы русского алфавита;
- цифры от 0 до 9;
- знаки арифметических операций: + - \* / ^ \ (деление нацело);
- знаки операций отношения: = < >;
- знаки препинания и разделители: . , : ; ' (апостроф) () [] \_
- специальные знаки: % ! # \$ и т.д.

# Имена переменных

При написании программы вам потребуется *переменные*, т.е области оперативной памяти, которые имеют имя, данное вами, значения которых может меняться.

Имя *переменной* в ходе выполнения программы постоянно, а значение может меняться многократно.



# Правила для имен переменных

- Имя переменной должно состоять не более чем из сорока символов;
- В качестве символов можно использовать только латинские буквы, цифры
- Любое имя начинается с латинской буквы.
- С помощью одной буквы можно образовать большое количество разных имён: A1; A2; A3; ...A56; ...
- В имени можно использовать несколько букв, например: Name; FOOL12 и т.д.
- В качестве имени нельзя использовать название команд, операторов и функций.

# Арифметика в Бейсике

- ❑ В отличии от арифметики, выражение должно быть записано в одну строку безо всяких числителей и знаменателей;
- ❑ Для записи арифметических действий допустимо использовать только перечисленные ниже знаки:

- +**    *Сложение*
- *Вычитание*
- \***    *Умножение*
- /**    *Деление*
- ^**    *Возведение в степень*

# Арифметика в Бейсике

- ❑ Недопустим пропуск знака умножения между коэффициентом и переменной как это возможно в алгебре. Например, нельзя писать **2х**, а надо **2\*х**.
- ❑ Дробная часть отделяется от целой точкой, а не запятой. Например нельзя писать **3,14**, а надо **3.14**.
- ❑ Допустимо опускать в записи в десятичной дроби ноль, стоящий перед точкой. Например **0.123** можно **.123**

# Приоритет действий:

- 1) сначала выполняются действия ***в скобках*** (в Бейсике скобки используются только круглые, в сложных выражениях они могут быть и двойные, и тройные, и т.д.);
- 2) далее вычисляются ***функции***, если они есть;
- 3) затем выполняется ***возведение в степень***;
- 4) потом ***умножение*** и ***деление***;
- 5) в последнюю очередь — ***сложение*** и ***вычитание***.

Действия одинаковой очередности выполняются ***слева направо***.

# Арифметика в Бейсике

Примеры:

- ❑ В арифметике  $\frac{1}{x}$  Бейсике `1/x`
- ❑ В арифметике  $2x^2$  Бейсике `2*x^2`
- ❑ В арифметике  $\frac{3+4}{8-6}$  Бейсике `(3+4)/(8-6)`

# Арифметика в Бейсике

**Деление нацело.** Для этого действия используется знак \ (так называемый обратный слэш). Компьютер в этом случае делит числа как обычно, но в качестве результата деления представляет только целую часть, отбрасывая дробную. Например:

$$\begin{aligned}41 \backslash 4 &= 10 \\ 25 \backslash 9 &= 2\end{aligned}$$

**Нахождение целого остатка от деления.** Для этого действия нет специального знака и оно выполняется при помощи оператора MOD. Оператор MOD действует как и предыдущий, но в качестве результата представляет целочисленный остаток от деления. Например:

$$\begin{aligned}41 \text{ MOD } 4 &= 1 \\ 25 \text{ MOD } 9 &= 7\end{aligned}$$

Действия деления нацело и нахождения целочисленного остатка выполняются до обычных умножения и деления.