

## 2. Имена, величины и выражения

---

### 2.1. Имена

#### 2.1.1. Общие сведения

#### 2.1.2. Слова

#### 2.1.3. Ключевые слова

#### 2.1.4. Многословные не-имена

### 2.2. Типы величин

### 2.3. Константы

#### 2.3.1. Виды констант

#### 2.3.2. Целые константы

#### 2.3.3. Вещественные константы

#### 2.3.4. Логические константы

#### 2.3.5. Символьные и литеральные константы

### 2.4. Величины

#### 2.4.1. Общие сведения

#### 2.4.2. Простые величины и таблицы. Описания величин

#### 2.4.3. Описания величин

#### 2.4.4. Область действия описаний

### 2.5. Выражения

#### 2.5.1. Общие сведения

#### 2.5.2. Операции в языке Кумир

#### 2.5.3. Базовые операции

#### 2.5.4. Тип выражения. Согласованность типов

#### 2.5.5. Вырезка из строки

#### 2.5.6. Функции

#### 2.5.7. Примеры записи выражений

---

## 2.1. Имена

### 2.1.1. Общие сведения

Имя бывает у величин, таблиц, алгоритмов и исполнителей. *Имя* – это последовательность слов, разделенных пробелами. Первое слово имени не должно начинаться с цифры. Ни одно из слов не должно быть ключевым словом.

Примеры имен: т, погода на завтра, Ноябрь 7, Седьмое ноября, дом\_576.

Примеры неправильных имен:

- 7е ноября (первое слово начинается с цифры);
- альфа-бета ([символ «минус»]) – недопустимый символ);
- альфа или омега (или – ключевое слово).

Примечание. Ключевое слово **не** можно вставлять внутрь многословных логических имен.

### 2.1.2. Слова

*Слово* – это последовательность разрешенных (словарных) символов. *Словарными* символами являются:

- буквы (кириллические и латинские, прописные и строчные);
- цифры;
- два специальных знака: @ \_.

Примеры слов: бета123, 3кг, мама, Linux, КоСтЯ, kumir@infomir\_ru.

Примеры не слов: альфа-123, ма%ма, С++.

### 2.1.3. Ключевые слова

*Ключевые слова* языка Кумир – это: алг, нач, кон, исп, кон\_исп, дано, надо, арг, рез, аргрез, знач, цел, вещ, лог, сим, лит, таб, целтаб, вещтаб, логтаб., литтаб, и, или, не, да, нет, утв, выход, ввод, вывод, нс, если, то, иначе, все, выбор, при, нц, кц, кц\_при, раз, пока, для, от, до, шаг.

### 2.1.4. Многословные не-имена

В отрицаниях логических величин, таблиц и алгоритмов функций ключевое слово **не** можно вставлять между словами многословного имени.

[Скопировать пример](#)

```
лог л, завтра будет четверг
л := не завтра будет четверг | Правильно
л := завтра не будет четверг | Правильно
л := завтра будет не четверг | Правильно
л := завтра будет четверг не | Неправильно
л := не завтра не будет четверг | Неправильно
```

#### Пример 2.1. Использование многословных не-имен

Первые три присваивания присваивают логической величине л значение, противоположное значению логической величины завтра будет четверг. Четвертая строка синтаксически

неверна – **не** нельзя ставить после имени. Последняя строка также неверна: нельзя использовать более одного.

## 2.2. Типы величин

Величины, с которыми работает Кумир-программа, подразделяются на несколько типов. Величина каждого из типов может принимать свой набор значений. В языке КуМир предусмотрены следующие типы величин:

- **цел** – принимает целые значения от  $-\text{МАКСЦЕЛ}-1$  до  $\text{МАКСЦЕЛ}$ , где  $\text{МАКСЦЕЛ} = 2147483647$ ;
- **вещ** – принимает вещественные значения от  $-\text{МАКСВЕЩ}$  до  $\text{МАКСВЕЩ}$ , где  $\text{МАКСВЕЩ}$  – это число, немного меньшее, чем  $1.797693 \times 10^{308}$ ;
- **лог** – принимает значения **да** или **нет**;
- **сим** – значением может быть любой литеральный символ ;
- **лит** – значением может быть строка литеральных символов.

Типы **цел** и **вещ** называются числовыми; типы **сим** и **лит** – текстовыми.

Значения величин **МАКСЦЕЛ** и **МАКСВЕЩ** определяются способом представления чисел в современных компьютерах и примерно одинаковые в большинстве современных языков программирования.

Язык Кумир содержит встроенные функции преобразования числовых типов в текстовые и наоборот . При необходимости значения целого типа автоматически переводятся в вещественные, а символьные – в строковые. Для преобразования вещественных значений в целые используется встроенная функция `int`.

## 2.3. Константы

### 2.3.1. Виды констант

Константы бывают целые, вещественные, логические, символьные и литеральные.

### 2.3.2. Целые константы

Целые константы бывают положительные и отрицательные. Целая константа должна быть в диапазоне от  $-\text{МАКСЦЕЛ}$  до  $\text{МАКСЦЕЛ}$ , где  $\text{МАКСЦЕЛ} = 2147483647$ . Целые константы можно записывать в десятичной и 16-ричной форме. Шестнадцатеричные константы начинаются с символа `$`.

Примеры: 123, -100000, \$100.

### 2.3.3. Вещественные константы

Вещественные константы бывают положительные и отрицательные. Вещественная константа по абсолютной величине должна быть меньше  $2^{1023}$ . Вещественные константы можно записывать в десятичной и экспоненциальной форме. В качестве разделителя в экспоненциальной записи можно использовать любой вариант буквы `e`: строчный или прописной, латинский или кириллический.

Ограничения для вещественных констант определяются стандартом IEEE 754-2008.

Примеры: 123, -100000, \$100.

### 2.3.4. Логические константы

Логическая константа — это одно из ключевых слов **да**, **нет**.

### 2.3.5. Символьные и литеральные константы

В символьной константе допустим любой символ, который можно набрать на стандартной клавиатуре. Такие символы называются *допустимыми*.

Символьная константа имеет вид 'с' или "с" (здесь с – допустимый символ).

Примеры: 'а', "%", '""', '"', 'Это я', "It's me".

Литеральная (текстовая) константа имеет вид 'Т' или "Т". Здесь Т – строка, состоящая из допустимых символов. При этом, если константа Т ограничена простыми кавычками, то Т не содержит простую кавычку, а если Т ограничена двойными кавычками, то она не содержит двойную кавычку.

## 2.4. Величины

### 2.4.1. Общие сведения

Каждая величина имеет *имя*, *тип*, *вид* и *значение*.

Имя величины служит для обозначения величины в алгоритме .

Тип величины показывает, какие значения может принимать величина, и какие операции можно с ней выполнять .

Вид величины показывает ее информационную роль в алгоритме. Например, аргументы содержат исходную информацию, необходимую для работы алгоритма, а промежуточные величины предназначены для хранения текущей информации, которую обрабатывает алгоритм.

Во время выполнения алгоритма в каждый конкретный момент величина имеет какое-то *значение* либо *не определена*.

Имя, тип и вид величины можно однозначно определить по тексту алгоритма. Это *статические* характеристики величины.

Значение определяется только во время выполнения. Это *динамическая* характеристика.

### 2.4.2. Простые величины и таблицы. Описания величин

В языке Кумир используются *простые* и табличные величины – *таблицы*.

Для таблиц определена *размерность* (бывают таблицы размерностей 1, 2 и 3). Для каждого измерения определены *границы* изменения *индекса* таблицы по этому измерению – два целых числа.

### 2.4.3. Описания величин

Каждая величина должна иметь описание. Это может быть сделано:

- с помощью оператора описания;
- при задании формальных параметров алгоритма .

В описании задаются перечисленные выше статические характеристики переменной.

Кроме того, в алгоритмах-функциях используется простая переменная знач, ее тип определяется типом функции . Явного описания переменная знач не имеет. Ее область действия – тело соответствующего алгоритма-функции.

Команда описания простой величины состоит из ключевого слова нужного типа (цел, вещ, сим, лит, лог), за которым следует список имен величин.

[Скопировать пример](#)

```
цел j, k, n
вещ длина, ширина
лит мой текст
```

#### Пример 2.2. Пример объявления простых величин

Для описания таблиц после описания типа нужно указать ключевое слово таб (слитно или раздельно с ключевым словом типа). Размерность таблицы и границы изменения индексов указываются после имени каждой величины.

[Скопировать пример](#)

```
цел таб k[-5:5]
вещтаб tab[1:4, 1:12]
```

#### Пример 2.3. Пример объявления табличных величин

Здесь k – линейная таблица, состоящая из 11 элементов целого типа. Индексы элементов принимают значения от -5 до 5. Таблица tab – прямоугольная. В ней 48 элементов – 4 строки и 12 столбцов.

#### 2.4.4. Область действия описаний

В зависимости от способа описания и места описания в программе, где описана величина, определена ее *область действия* описания – та часть текста программы, где допустимо использование этой величины.

Если величина описана во вступлении к программе, ее можно использовать в любом алгоритме **этой** программы.

Если величина описана в заголовке алгоритма, то ее можно использовать в теле этого алгоритма, а также в заголовке – после этого описания.

Если переменная описана в теле алгоритма, то ее можно использовать только в теле этого алгоритма **после** места описания.

```
алг
нач
  п := 1 | Так нельзя!
  цел п
  п := 1 | Так можно
кон
```

### Пример 2.4. Объявление величин в алгоритме

## 2.5. Выражения

### 2.5.1. Общие сведения

Выражение в языке КуМир описывает новое значение, полученное из уже известных значений с помощью предусмотренных в языке КуМир *операций*.

Примеры:

- $(a+b)*(a-b)$ ;
- **да или нет**;
- $(\sin(\text{альфа}))^{**2}+(\cos(\text{альфа}))^{**2}$ .

В Кумир-программе выражения могут появляться в:

- правой части оператора присваивания;
- в индексе таблицы;
- в аргументе (типа **арг**) вызова функции;
- в качестве подвыражения другого выражения;
- в команде **ВЫВОД**.

### 2.5.2. Операции в языке Кумир

Операции в языке Кумир – это:

- базовые операции (арифметические, логические, текстовые);
- вырезка из строки;
- операции, задаваемые алгоритмами-функциями.

Для каждой операции известны:

- количество значений-аргументов;
- их типы;
- тип результата.

### 2.5.3. Базовые операции

В зависимости от типов аргументов и результата, базовые операции делятся на следующие классы:

- арифметические операции (аргументы и результат – числового типа);
- сравнение арифметическое (аргументы – числового типа, результат – **ЛОГ**);
- сравнение текстовое (аргументы – текстового типа, результат – **ЛОГ**);
- логические операции (аргументы – **ЛОГ**, результат – **ЛОГ**);
- текстовые операции (аргументы и результат – текстового типа).

Каждой базовой операции соответствует свой символ. В некоторых случаях приходится применять составной символ, состоящий из двух обычных символов:

- **\*\*** – возведение в степень;
- **<=** – меньше или равно;
- **>=** – больше или равно;
- **<>** – не равно.

### 2.5.4. Тип выражения. Согласованность типов

Типом выражения называется тип результата операции, которая выполняется последней при вычислении этого выражения.

Типы всех подвыражений должны быть согласованы с типами аргументов выполняемых операций.

Рассмотрим выражение  $\text{гамма}(x) - \text{дельта}(2y+1, z)$ , где **гамма** и **дельта** – описанные в программе алгоритмы-функции.

Это должны быть функции числового типа. Если обе они имеют тип **цел**, то и все выражение имеет тип **цел**. В противном случае выражение имеет тип **вещ**.

#### Пример 2.5. Тип выражения

### 2.5.5. Вырезка из строки

Операция вырезки из строки имеет 3 аргумента: **лит** строка, **цел** старт, **цел** финиш и результат: **лит** вырезка. В отличие от базовых операций, аргументы вырезки из строки имеют разные типы. Поэтому способ записи вырезки из строки отличается от способа, принятого для базовых операций.

[Скопировать пример](#)

**лит** строка, вырезка

```
строка = "строка"  
вырезка := строка[3:5]  
УТВ вырезка = "рок"
```

### Пример 2.6. Вырезка из строки

#### 2.5.6. Функции

В выражениях языка Кумир можно использовать:

- встроенные алгоритмы-функции, например:  $\sin(x)$ , `длин("ХВОСТ")`;
- алгоритмы-функции встроенных исполнителей, например: `температура`;
- алгоритмы-функции программы пользователя (в том числе – алгоритмы-функции исполнителей пользователя).

У каждой функции есть имя, для нее фиксировано количество параметров, параметры перенумерованы. Для каждого параметра функции и ее результата фиксированы их типы; тип результата называется типом функции.

Вызов функции с именем `имя_функции` и аргументами, заданными выражениями  $X_1, \dots, X_n$  записывается так: `имя_функции( $X_1, \dots, X_n$ )`.

#### 2.5.7. Примеры записи выражений

Ниже приведены различные способы записи выражений.

Таблица 2.1. Запись различных математических выражений

Математическая формула	Выражение на языке Кумир
	$-1/x^{**2}$
	$a/(b*c)$
	$a/b*c$ или $(a/b)*c$
	$2^{**}(2^{**}n)$
	$(-b+\text{sqrt}(b^{**2}-4*a*c))/(2*a)$
	$\text{sqrt}(p*(p-a)*(p-b)*(p-c))$
	$(a+b+c)/2$



$$\sqrt{a^2+b^2-2ab\cos(\gamma)}$$

$$(a+d+b*c)/(b*d)$$

$$\sin(\alpha)*\cos(\beta)+\cos(\alpha)*\sin(\beta)$$