

### 3.1. Присваивание

#### 3.2. Контроль выполнения

##### 3.2.1. Команды с условиями

##### 3.2.2. Команды-действия

#### 3.3. Ввод и вывод

##### 3.3.1. Вывод

##### 3.3.2. Ввод

#### 3.4. Вызов алгоритма

#### 3.5. Выход

---

### 3.1. Присваивание

Команда присваивания предназначена для изменения значения простых переменных и элементов таблиц и имеет общий вид **ВЕЛИЧИНА := ВЫРАЖЕНИЕ**, где

- **ВЕЛИЧИНА** – это имя простой величины или описание элемента таблицы;
- **ВЫРАЖЕНИЕ** – это выражение, составленное из величин, констант, вызовов алгоритмов-функций и знаков операций.

Тип выражения должен быть согласован с типом величины.

[Скопировать пример](#)

```
n := 0
m := n
m := m+1
m := 2*длин(t)+div(n,2)
c := (x+y)/2
площадь:=a*b*sin(C)/2
d:=b**2-4*a*c
x[1]:=(-b+sqrt(d))/(2*a)
a[i]:=2*a[i-2]+a[i-1]
b[i,j]:=-b[j,i]
```

#### Пример 3.1. Присваивания

Все переменные должны быть описаны, а их типы – согласованы с типами операций и их аргументов.

### 3.2. Контроль выполнения

Команды контроля выполнения в языке КуМир делятся на две категории:

1. команды контроля выполнения с условиями,

- команды управления выполнением (или команды-действия).

### 3.2.1. Команды с условиями

В языке Кумир существует три команды контроля выполнения с условиями: **утв**, **дано**, **надо**.

Формат вызова:

- утв** ЛОГ ВЫРАЖЕНИЕ
- дано** ЛОГ ВЫРАЖЕНИЕ
- надо** ЛОГ ВЫРАЖЕНИЕ

Все три команды выполняются так.

- Проверяется условие ЛОГ ВЫРАЖЕНИЕ
- Если условие не соблюдается (равно **нет**), то Кумир прекращает выполнение алгоритма и сообщает, что возник отказ
- Если же условие соблюдается, то выполнение алгоритма нормально продолжается так, как если бы команды контроля не было вовсе.

Команда **дано** проверяет условие в начале выполнения алгоритма, команда **надо** – в конце выполнения алгоритма, а командой **утв** можно проверить условие в процессе выполнения алгоритма.

[Скопировать пример](#)

```
алг абс (рез вещь x)
дано x<=0
надо x>=0
нач
  x := -x
кон
```

Пример 3.2. Использование «дано» и «надо»

[Скопировать пример](#)

```
алг вещь кв (вещ x)
нач
  вещь k
  k := x*x
  утв k>=0
  знач := k
кон
```

Пример 3.3. Использование «утв»

### 3.2.2. Команды-действия

В КуМире существуют две команды управления выполнением программы **пауза** и **стоп**:

- Команда **пауза** приостанавливает выполнение программы в той строке, в которой она была вызвана. Действие этой команды аналогично действию точки остановки.
- Команда **стоп** принудительно завершает работу программы. Действие этой команды аналогично действию пункта **Остановить выполнение** меню **Выполнение**.

### 3.3. Ввод и вывод

#### 3.3.1. Вывод

Формат вызова:

**вывод** выражение-1, ... , выражение-N

или

**вывод** величина-файлового-типа, выражение-1, ... , выражение-N

Каждое выражение может быть либо арифметическим, логическим или текстовым выражением, либо командой перехода на новую строку (ключевое слово **НС**). Значения выражений выводятся последовательно в строку области ввода-вывода без разделителей.

Если первой величиной после команды **ВЫВОД** является величина типа **файл**, то вместо вывода на экран, значения выражений будут записаны в соответствующий файловый объект.

Ключевое слово **НС** в качестве одного из выражений команды **ВЫВОД** является признаком перехода на новую строку.

[Скопировать пример](#)

```

алг
нач
  цел а
  а := 3
  вещ б
  б := 1.3
  нц 5 раз
    вывод а, " ", б, "Привет!", нс
  кц
кон

```

Результат вывода на экран:

[Скопировать пример](#)

```

3 1.3Привет!
3 1.3Привет!
3 1.3Привет!
3 1.3Привет!
3 1.3Привет!

```

**Пример 3.4.** Вывод на экран

[Скопировать пример](#)

```
использовать Файлы
алг
нач
  файл ф
  ф := открыть на запись("test.txt")
  цел а
  а := 3
  вещ б
  б := 1.3
  нц 5 раз
    вывод ф, а, " ", б, "Привет!", нс
  кц
  закрыть(ф)
кон
```

Содержимое файла «test.txt» после выполнения программы:

[Скопировать пример](#)

```
3 1.3Привет!
3 1.3Привет!
3 1.3Привет!
3 1.3Привет!
3 1.3Привет!
```

### Пример 3.5. Вывод в файл

#### 3.3.2. Ввод

Формат вызова:

**ввод** выражение-1, ... , выражение-N

или

**ввод** величина-файлового-типа, выражение-1, ... , выражение-N

При выполнении команды **ввод**, если первым аргументом не является величина типа **файл**, то система Кумир делает активным окно ввода-вывода и ждет, пока пользователь введет соответствующие значения. По окончании введенные значения присваиваются указанным величинам. В качестве имени величины можно указать имя простой величины или имя элемента таблицы с указанием значений индексов. Признаком конца ввода служит нажатие на клавишу Enter. При вводе нескольких значений, они отделяются друг от друга пробелом.

Если первым аргументом является величина типа **файл**, то производится чтение данных из текстового файла.

Использование **нс** в качестве одного из выражений команды **ввод** изменяет разделитель вводимых значение с пробела на символ новой строки (Enter). Например, если **нс** в списке параметров команды **ввод** находится между значениями **а** и **б**, то после ввода **а** нужно нажать Enter чтобы перейти ко вводу **б**. Значение, введенное после **а** и разделённое с **а** пробелом потеряется.

[Скопировать пример](#)

```
алг
  #использование нс в качестве
  #парамтера команды ввод.
нач
  вещь а, б
  ввод а, нс, б
  вывод а, нс, б
кон
```

**Пример 3.6.** Использование «нс» в качестве параметра команды «ввод»

Если на вход команды ввод мы подадим три числа, написанные так:

```
121 125
2
```

На выходе программы получим:

```
121
2
```

Итак, число 125, введённое через пробел, потерялось.

### 3.4. Вызов алгоритма

Вызов алгоритма-процедуры является отдельной командой алгоритмического языка и имеет вид:

имя алгоритма-процедуры

или

имя алгоритма-процедуры(список параметров вызова)

[Скопировать пример](#)

```
алг
нач
  подпрограмма
кон

алг подпрограмма
нач
  вывод "Мы в подпрограмме", нс
кон
```

**Пример 3.7.** Вызов подпрограммы без параметров

[Скопировать пример](#)

```
алг
нач
  сумма(2.4, 7.6)
кон
алг сумма(вещ а, вещь б)
нач
  вывод "Сумма = ", а+б, нс
кон
```

**Пример 3.8.** Вызов подпрограммы с параметрами

### 3.5. Выход

Команда **ВЫХОД** используется для выхода из цикла или для окончания работы текущего алгоритма. Если команда **ВЫХОД** выполняется внутри цикла, то выполнение продолжается с первой команды после тела этого цикла. Если команда **ВЫХОД** используется во вложенных циклах, то завершается самый внутренний цикл. Если команда **ВЫХОД** выполняется вне циклов, то она приводит к завершению выполнения текущего алгоритма.

[Скопировать пример](#)

```
алг
нач
  нц
    нц
      вывод "-2-", нс
      выход
    кц
      вывод "-1-", нс
      выход
  кц
    вывод "-0-", нс
    выход
    вывод "-F-", нс
кон
```

При выполнении этой программы будет выведено на экран:

[Скопировать пример](#)

```
-2-
-1-
-0-
-F-
```

**Пример 3.9.** Использование команды «выход»