

4.1. Команды ветвления

4.1.1. если-то-иначе-все

4.1.2. выбор-при-иначе-все

4.2. Команды цикла

4.2.1. Цикл "для"

4.2.2. Цикл "пока"

4.2.3. Цикл "до тех пор"

4.2.4. Цикл "N раз"

4.2.5. Цикл "нц-кц"

4.2.6. Принудительное завершение цикла

4.1. Команды ветвления

4.1.1. если-то-иначе-все

Общий вид команды:

```
если условие
  то серия1
  иначе серия2
все
```

серия2 вместе со служебным словом **иначе** может отсутствовать. В этом случае команда имеет вид:

```
если условие
  то серия1
все
```

При выполнении команды **если** Кумир сначала проверяет условие, записанное между **если** и **то**. При соблюдении этого условия выполняется **серия1**, в противном случае – **серия2** (если она есть), после чего Кумир переходит к выполнению команд, записанных после слова **все**.

Если условие не соблюдается, а **серия2** вместе с **иначе** отсутствует, то Кумир сразу переходит к выполнению команд, записанных после слова **все**.

[Скопировать пример](#)

```
если a<b
  то b:=b-a; p:=p+q
  иначе a:=a-b; q:=q+p
все
```

Пример 4.1. Конструкция «если-то-иначе-все»

```
если x>m
  то
    m:=x
    n:=n+1
все
```

Пример 4.2. Конструкция «если-то-все»

4.1.2. выбор-при-иначе-все

Общий вид команды:

```
выбор
  при условие 1 : серия 1
  при условие 2 : серия 2
  ...
  при условие n : серия n
  иначе серия n+1
все
```

Ключевое слово **иначе** вместе с соответствующей серией команд может отсутствовать:

```
выбор
  при условие_1 : серия_1
  при условие_2 : серия_2
  ...
  при условие_n : серия_n
все
```

Кумир сначала проверяет **условие 1**. Если оно соблюдается, то Кумир выполняет команды из **серии 1**, после чего переходит к выполнению команд, записанных после слова **все**. В противном случае Кумир делает то же самое с **условием 2** и командами из **серии 2** и т. д.

Команды, записанные после слова **иначе**, выполняются в том случае, когда не соблюдено ни одно из условий.

В команде **выбор** всегда выполняется не более одной серии команд, даже если несколько условий окажутся истинными. Выполнение команды **выбор** заканчивается после того, как найдено первое (по порядку следования) условие со значением **да** (и выполнена соответствующая серия команд).

```
выбор
  при a>1: i:=i+1
  при a<0: j:=j-1
  иначе t:=i; i:=j; j:=t
все
```

Пример 4.3. Конструкция «выбор-при-иначе-все»

```
выбор
  при a[i]>1000 : b[i]:=3; c[i]:=3.141
  при a[i]>100 :
    b[i]:=2; c[i]:=3.14
  при a[i]>10 :
    b[i]:=1
    c[i]:=3.14
все
```

В данном примере при $a[i]=1812$ будут выполнены присваивания: $b[i]:=3$ и $c[i]:=3.141$.

Пример 4.4. Конструкция «выбор-при-все»

4.2. Команды цикла

4.2.1. Цикл «для»

Общий вид цикла *для*:

```
нц для i от i1 до i2
  тело_цикла
кц
```

Здесь i – величина типа **цел** (она называется *параметром цикла*), а $i1$ и $i2$ – целые выражения, т. е. выражения типа **цел**. При выполнении цикла **для** тело цикла выполняется последовательно для $i = i1, i = i1 + 1, \dots, i = i2$. Если $i1 = i2$, то тело цикла выполнится один раз для $i = i1$. Если же $i1 > i2$, то тело цикла не выполнится ни разу.

Общий вид цикла *для* с шагом:

```
нц для i от i1 до i2 шаг d
  тело_цикла
кц
```

Если шаг $i3$ (который также должен быть целым выражением) является положительным значением, то тело цикла будет выполняться последовательно для $i = i1, i = i1 + d, i = i1 + 2d, \dots$ до тех пор, пока значение i удовлетворяет условию $i \leq i2$.

```
нц для j от 1 до длин(t)
  t1[j]:=t[длин(t)+1-j]
кц
```

Пример 4.5. Цикл «для» без указания шага

```
нц для i от 1 до 100 шаг 2
  a[i+1]:=a[i]
кц
```

Пример 4.6. Цикл «для» с положительным шагом

[Скопировать пример](#)

```
нц для i от 100 до 1 шаг -2
  a[i]:=a[i-1]
кц
```

Пример 4.7. Цикл «для» с отрицательным шагом

4.2.2. Цикл «пока»

Общий вид цикла *пока*:

```
нц пока условие
  тело_цикла
кц
```

При выполнении цикла **пока** Кумир циклически повторяет следующие действия:

- Проверяет записанное после служебного слова **пока** условие.
- Если условие не соблюдается, то выполнение цикла завершается и Кумир начинает выполнять команды, записанные после **кц**.

Если же условие соблюдается, то Кумир выполняет тело цикла, снова проверяет условие и т. д.

[Скопировать пример](#)

```
нц пока a<10
  a := a + 1
кц
```

Пример 4.8. Цикл «пока»

4.2.3. Цикл «до тех пор»

Общий вид цикла *до тех пор*:

```
нц
· тело_цикла
кц при условие
```

При выполнении цикла **до тех пор** Кумир циклически повторяет следующие действия:

- Выполняет тело цикла.

- Проверяет записанное после служебного слова **пока** условие.
- Если условие не соблюдается, то выполнение цикла завершается и Кумир начинает выполнять команды, записанные после **кц**.

Если же условие соблюдается, то Кумир выполняет тело цикла, снова проверяет условие и т. д.

[Скопировать пример](#)

```
нц
  x := 2*x
кц_при x>100
```

Пример 4.9. Цикл «до тех пор»

Условие окончания цикла может быть добавлено в любую команду повторения, например, в цикл *N раз*.

[Скопировать пример](#)

```
нц 5 раз
  ввод x, y, z
  вывод нс, "Координаты:", x, y, z
кц при x+y+z>100
```

Пример 4.10. Цикл с условием «при»

В старых версиях системы Кумир использовалась запись **кц_при**. В новой версии Кумир записи **кц_при** (с символом подчеркивания) и **кц при** (с пробелом) эквивалентны.

4.2.4. Цикл «N раз»

Общий вид цикла *N раз*:

```
нц N раз
  тело_цикла
кц
```

Здесь *N* – целое выражение, задающее число повторений. При выполнении алгоритма последовательность команд циклически повторяется указанное число раз.

[Скопировать пример](#)

```
нц 4 раз
  ввод x, y, z
  вывод нс, "Координаты:", x, y, z
кц
```

Пример 4.11. Цикл «N раз»

4.2.5. Цикл «нц-кц»

Общий вид цикла:

```
нц
  тело_цикла
кц
```

[Скопировать пример](#)

```
нц
  а := а + 1
  если а>100 то выход все
кц
```

Пример 4.12. Цикл «нц-кц»

Кумир не проверяет, встречается ли в теле цикла команда **выход**. Если такой команды нет, то цикл *нц-кц* будет выполняться до бесконечности.

4.2.6. Принудительное завершение цикла

В теле любого из циклов может быть использована команда **выход**(см. [Выход](#)). При выполнении этой команды, содержащий ее цикл будет завершен.