

## Цикл N раз

Лучшее качества компьютеров проявляются не тогда, когда они рассчитывают значения сложных выражений, а когда многократно, с незначительными изменениями, повторяют сравнительно простые операции. Даже очень простые расчеты могут поставить человека в тупик, если их надо повторить тысячи раз, а повторять операции миллионы раз человек совершенно не способен.

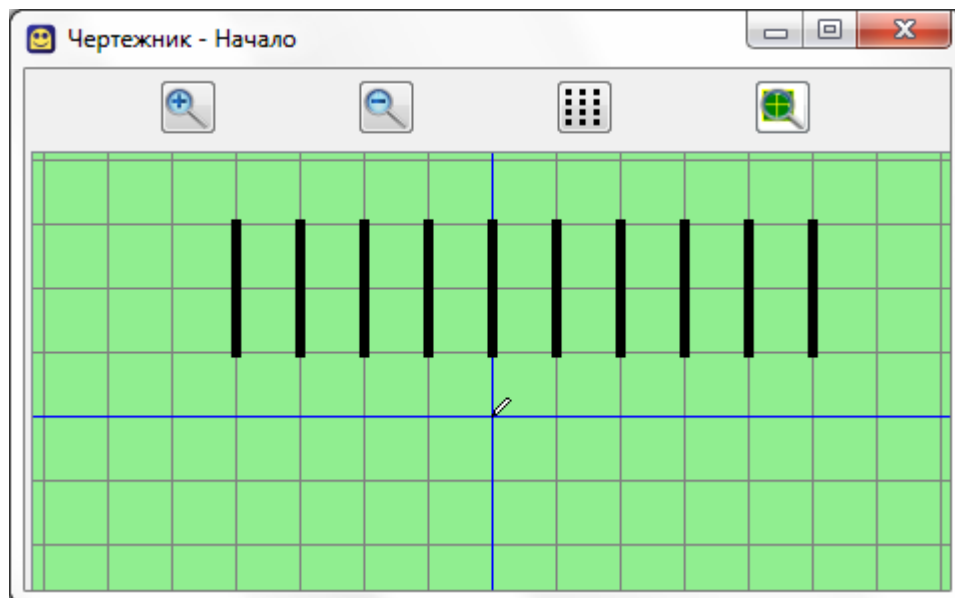
С необходимостью повторяющихся вычислений программисты сталкиваются постоянно. Например, если надо подсчитать, сколько раз буква "о" встречается в тексте необходимо перебрать все буквы. При всей простоте этой программы исполнить ее человеку очень трудно, а для компьютера это задача на несколько секунд.

*Описание действий, которые могут повторяться указанное число раз или пока не выполнено заданное условие, называется **циклом**.*

При составлении алгоритмов довольно часто встречаются случаи, когда некоторую последовательность команд нужно выполнить несколько раз подряд. Для упрощения записи алгоритма в таких случаях можно использовать специальную конструкцию **повторения N раз**.

Общий вид цикла:

```
нц <количество раз> раз  
    <тело цикла (последовательность команд) >  
кц
```



**алг** черточки2

нач

• **C**

. опустить перо

. сместиться на

. поднять перо

. Сместиться на

. опустить перо

. Сместиться на

. поднять перо

. СМЕСТИТЬСЯ НА

. опустить перо

. сместиться на

. поднять перо

. СМЕСТИТЬСЯ НА

. опустить перо

. Сместиться на

. поднять перо

. сместиться на

. опустить перо

. **сместиться на**  
\_\_\_\_\_

. поднять перо

• сместиться на  
систему бор

опустить перо  
сместить ст. на вектор (0, 2)

• СМЕСТИТЬСЯ НА  
ПОВЫШЬ ПОРО

сместить ся на вектор  $(1 \ 2)$

спустить пар

сместиться на вектор  $(0, -2)$

- поднимать перо

• сместиться на вектор  $(1, 2)$

```

. опустить перо
. сместиться на вектор(0,-2)
. поднять перо
. сместиться на вектор(1,2)
. опустить перо
. сместиться на вектор(0,-2)
. поднять перо
. сместиться на вектор(1,2)
. опустить перо
. сместиться на вектор(0,-2)
. поднять перо
. сместиться на вектор(1,2)
. поднять перо
. сместиться в точку(0,0)
кон

```

Это алгоритм может быть записан так:

**использовать Чертежник**

**алг** черточки<sup>1</sup>

**нач**

```

. сместиться в точку(-4,3)
. нц 10 раз
. . опустить перо
. . сместиться на вектор(0,-2)
. . поднять перо
. . сместиться на вектор(1,2)
. кц
. поднять перо
. сместиться в точку(0,0)

```

**кон**

Этот цикл выполнится 10 раз, т. е. команды между «нц» и «кц» повторятся 10 раз. Перечень повторяющихся действий называют *телом цикла*. Однократное выполнение тела цикла называется *итерацией*.

Разница заметна на глаз. Вместо того чтобы 10 раз подряд писать

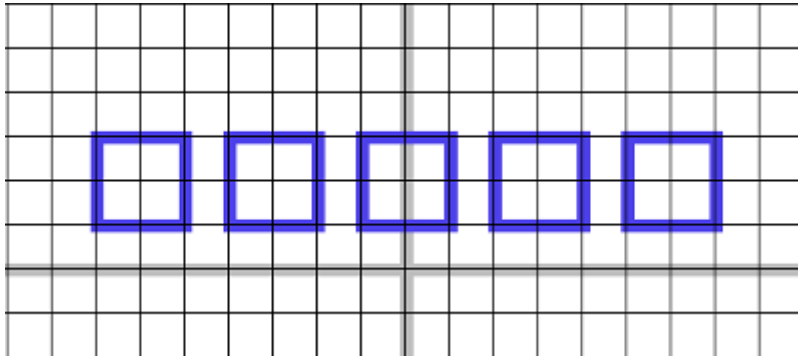
```

. опустить перо
. сместиться на вектор(0,-2)
. поднять перо
. сместиться на вектор(1,2)

```

можно записать, что этот блок команд надо повторить 10 раз. Это не только короче, но еще и понятнее.

Рассмотрим еще один пример. Вспомним рисование квадратов. Рисование квадрата можно написать как тело цикла, а можно вынести в отдельный алгоритм и в цикле его вызывать. После рисования каждого квадрата, в этом примере, нужно сместиться на вектор(3,0), чтобы следующий квадрат был правее этого. Объясните, почему команда сместиться в точку в этом цикле не даст нужного результата.



Таким образом, программу можно записать так:

**использовать Чертежник**

**алг** квадраты

**нач**

. сместиться в точку  $(-7, 1)$

. **нц** 5 **раз**

. . квадрат

. . сместиться на вектор  $(3, 0)$

. **кц**

. сместиться в точку  $(0, 0)$

**кон**

**алг** квадрат

**нач**

. опустить перо

. сместиться на вектор  $(0, 2)$

. сместиться на вектор  $(2, 0)$

. сместиться на вектор  $(0, -2)$

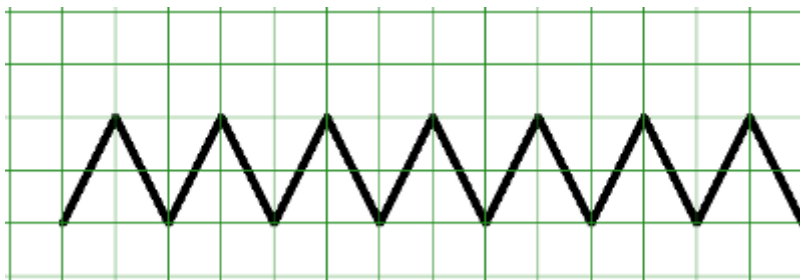
. сместиться на вектор  $(-2, 0)$

. поднять перо

**кон**

**Задание 1.** Измените предыдущий алгоритм, чтобы рисовалось не 5, а 8 квадратов.

**Задание 2.** Используя цикл, нарисуйте рисунок:



**Задание 3.** Используя вспомогательный алгоритм рисования доски, нарисуйте забор. Используйте цикл **N раз**.

