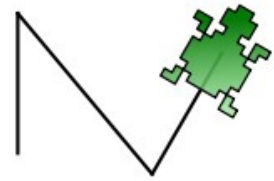


KTURTLE. СПРАВКА ПО КОМАНДАМ

СРЕДА ПРОГРАММИРОВАНИЯ

KTurtle — среда программирования, в которой применяется «черепашня графика» и мы управляем виртуальной черепашкой с помощью программы. **KTurtle** является аналогом языка **Logo**. Отличия только в синтаксисе. Все основные команды управления черепашкой и принцип рисования точно такой же: черепашка перемещается по холсту и оставляет следы в виде линий.



КОМАНДЫ УПРАВЛЕНИЯ ЧЕРЕПАШКОЙ

Каждую команду рекомендуется писать в отдельной строке. Отступы перед командами и пустые строки ни на что не влияют. Регистр букв важен.

Если команда требует **аргумент**, то он пишется после команды через пробел. В качестве аргумента подставляется **число**. Число может быть любым. Число может быть отрицательным. Число может быть дробным (разделитель в десятичной дроби — точка).

Команда	Что делает
вперёд 50	Черепашка перемещается вперёд на расстояние 50 и оставляет след (линию). Число обозначает условные единицы расстояния.
назад 50	Черепашка перемещается назад на расстояние 50 и оставляет след (линию). Число обозначает условные единицы расстояния.
направо 50	Черепашка поворачивается направо на 50 градусов, при этом она остаётся на месте.
налево 50	Черепашка поворачивается налево на 50 градусов, при этом она остаётся на месте.
направление 50	Черепашка поворачивается так, чтобы смотреть в направлении 50. То же самое, что азимут: вверх — север — азимут 0, вправо — восток — азимут 90, и тд.
перо_подними	Черепашка поднимает перо и при последующих перемещениях НЕ будет оставлять след. Не требует аргументов.
перо_опусти	Черепашка опускает перо и при последующих перемещениях будет оставлять след. Не требует аргументов.
толщина_пера 50	Задать толщину линии. Толщина задаётся для того, что будет нарисовано после этой команды. Команда не влияет на уже нарисованные линии.
повтори 50 { «команды» }	Цикл. Черепашка повторит «команды» 50 раз. В фигурных скобках может быть любое количество команд. Вложенные циклы допустимы.
сброс	Холст очищается, черепашка сбрасывает все настройки пера, возвращается в центр холста, поворачивается так, чтобы смотреть вверх (на север).

С команды **сброс** должна начинаться любая программа. Это гарантирует, что черепашка начнёт выполнять программу с «чистого листа», из начального состояния.

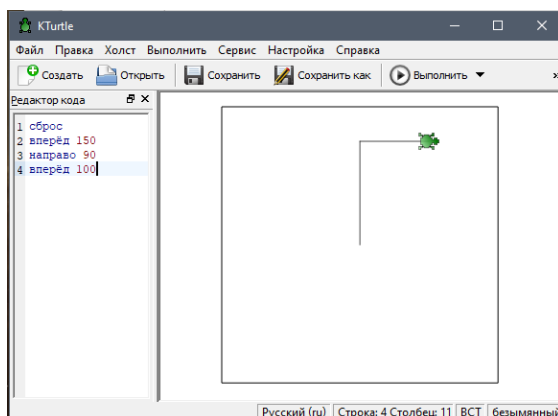
Например, типичная программа для черепашки:

Пример программы

```
сброс  
вперёд 150  
направо 90  
вперёд 100
```

Черепашка

- 1) займёт исходную позицию
- 2) пройдёт вперёд на 150
- 3) повернётся направо на прямой угол (90 градусов)
- 4) пройдёт вперёд на 100.



МАТЕМАТИКА

Вместо любого числа может быть записано **математическое выражение** с математическими операциями (+, -, *, /) и скобками. Например, команда

Пример подстановки математических операций

```
вперёд 3*(20+50/5)
```

заставит черепашку пройти вперёд на 90. Черепашка сначала вычислит математическое выражение, а потом выполнит команду вперёд с получившимся числом.

КОММЕНТАРИЙ

В черепашке есть однострочный комментарий. Он создаётся с помощью символа #. Всё, что написано после решётки в одной строке, будет считаться комментарием.

Пример комментариев

```
# часть 1  
вперёд 3*(20+50/5) #вычисление длины первой линии
```

ПЕРЕМЕННЫЕ

Черепашка умеет работать с переменными.

■ Создание переменной и запись в неё

Имена переменных в черепашке должны начинаться с символа `$`. После него должен быть хотя бы один символ — имя переменной.

Запись в переменную

```
$t = 123 #в переменную $t запишется число 123
$вася777 = 5 #в переменную $вася777 запишется число 5
$dлина = 7*9+5 #в переменную $dлина запишется число 68
```

В имени можно использовать английские или русские буквы, цифры и нижнее подчёркивание.

■ Подстановка переменной

Переменную можно подставить в любую команду туда, где записывается число. Вместо переменной подставится то число, которое записано в переменную.

Подстановка переменной

```
вперёд $t
направо $вася777
назад $dлина
```

В данном случае черепашка пройдёт вперёд на 123, повернёт направо на 5 градусов и пройдёт назад на 68.

■ Изменение переменной

Число в переменной можно изменять. В том числе выполнять инкремент.

Изменение переменной

```
$t = 12 #в переменную $t запишется число 12
$t = 7 #в переменную $t запишется число 7, а число 12 пропадёт
$t = $t + 1 #в переменную $t запишется 8
$t = $t + 1 #в переменную $t запишется 9
$t = $t * 2 #в переменную $t запишется 18
```