

Программа Kturtle. Язык TurtleScript, перемещение Черепашки

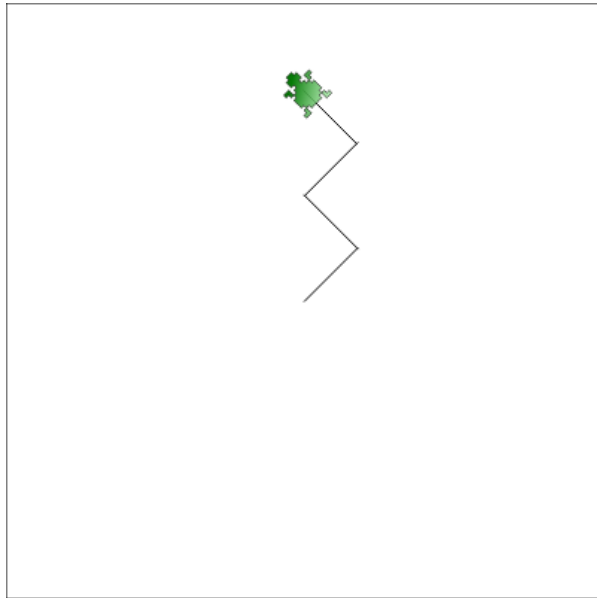
KTurtle обладает замечательными особенностями, которые помогут начать программировать легко и непринуждённо. В **KTurtle** используется язык программирования **TurtleScript**, используя который можно сообщить, что необходимо сделать программе.

Команды управления и перемещения Черепашки:

центр
нпр (направление) X
иди X, Y
иди_гор X
иди_верт Y
вп (вперёд) X
нд (назад) X
пр (направо) X
лв (налево) X
пж (покажи)
сч (спрячь)

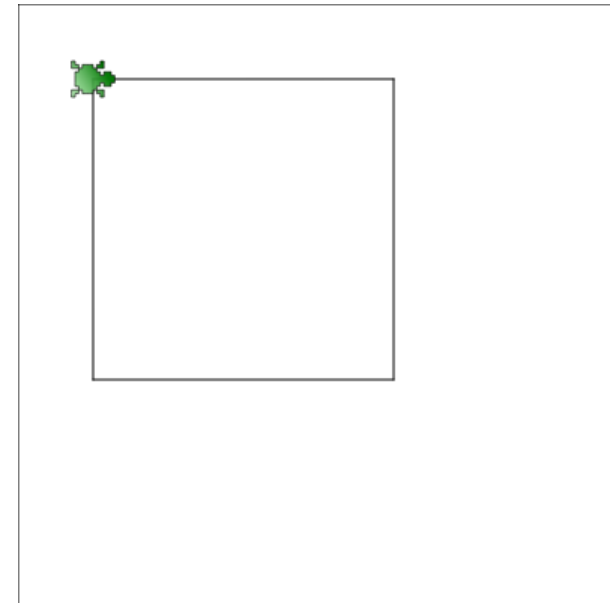
Задания 1-6:

1. Создайте программу, которая рисует нижеприведенный зигзаг:



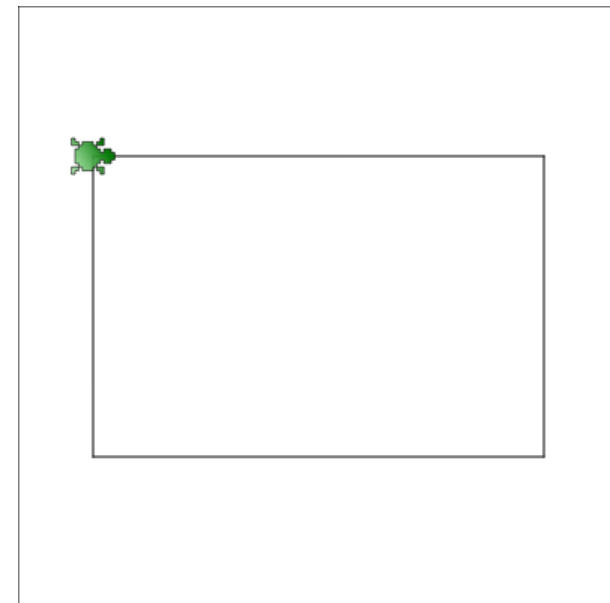
[решение тут](#)

2. Создайте программу, которая рисует нижеприведенный квадрат:



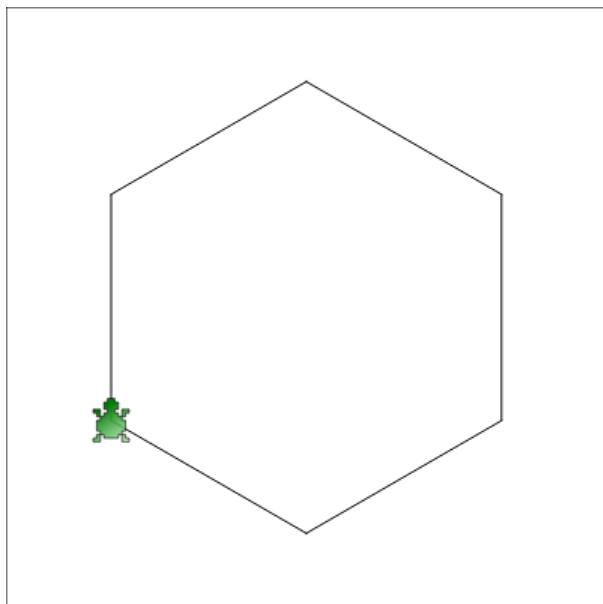
[решение тут](#)

3. Создайте программу, которая рисует нижеприведенный прямоугольник:



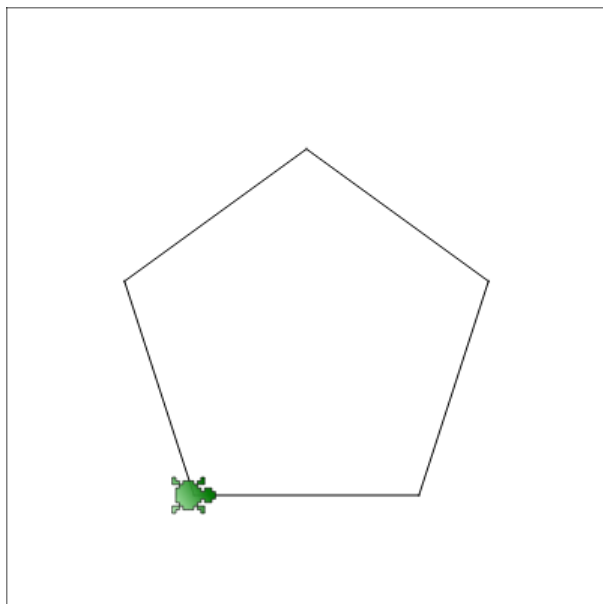
[решение тут](#)

4. Создайте программу, которая рисует нижеприведенный шестиугольник:



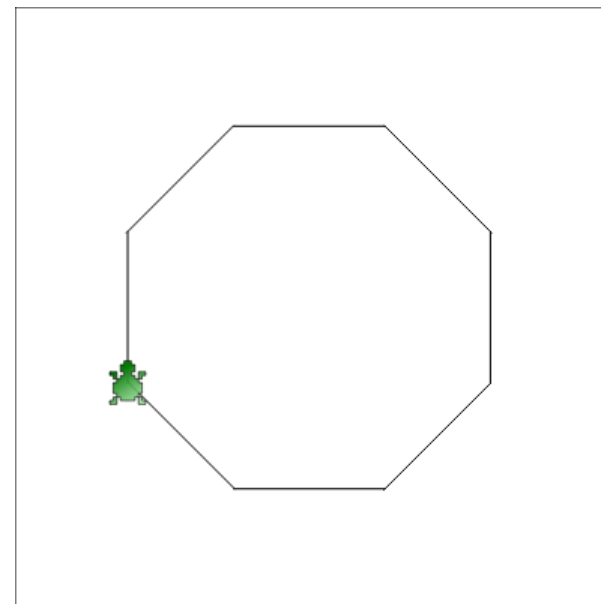
[решение тут](#)

5. Создайте программу, которая рисует нижеприведенный пятиугольник:



[решение тут](#)

6. Создайте программу, которая рисует нижеприведенный восьмиугольник:



[решение тут](#)

Холст, перо, цвета в KТurtle

Холст – это пространство Черепашки. Здесь она рисует в соответствии с командами, которые получает. Существует несколько команд для работы с холстом:

- рх** (размер_холста) X, Y
- цх** (цвет_холста) R, G, B
- очс** (очисти)
- сброс**

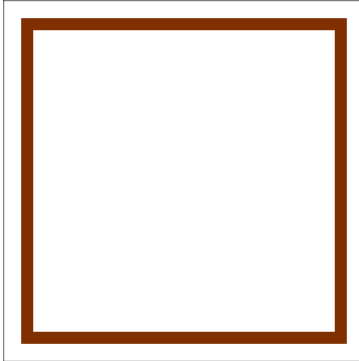
У Черепашки есть **перо**, которым она рисует линию во время перемещения. Есть несколько команд для управления пером:

- пп** (перо_подними)
- по** (перо_опусти)
- тп** (толщина_пера) X
- цп** (цвет_пера) R, G, B

Входным параметром **цвета** (**цх** или **цп**) является **RGB** комбинация - сочетания **красного, зеленого и синего** составляющих.

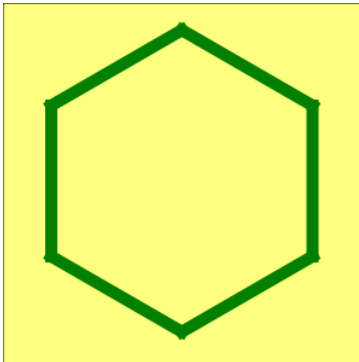
Задания 7-10:

7. Создайте программу, которая рисует нижеприведенный квадрат:



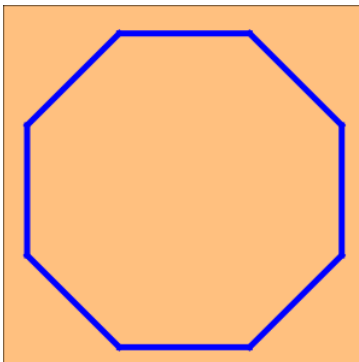
[решение тут](#)

8. Создайте программу, которая рисует нижеприведенный шестиугольник:



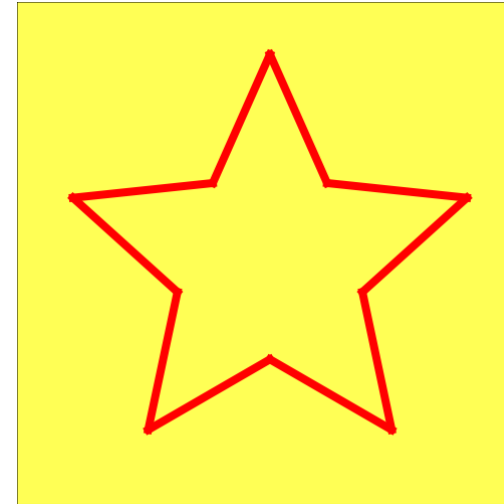
[решение тут](#)

9. Создайте программу, которая рисует нижеприведенный восьмиугольник:



[решение тут](#)

10. Создайте программу, которая рисует пятиконечную звезду:



[решение тут](#)

Цикл «повтори»

инструкция **повтори X { ... }** – служит для повторения команд в скобках **X** раз.

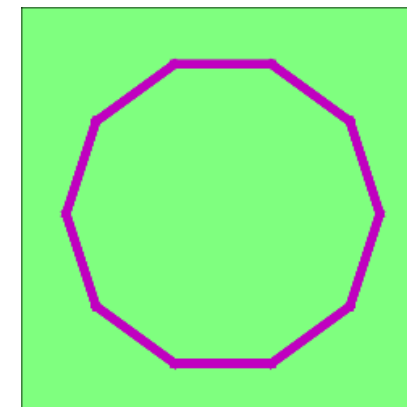
Команды: **жди X** – указывает Черепашке **подождать X секунд**;

прекрати – немедленно **завершает** выполнение текущего цикла;

закончить – **заканчивает** выполнение программы.

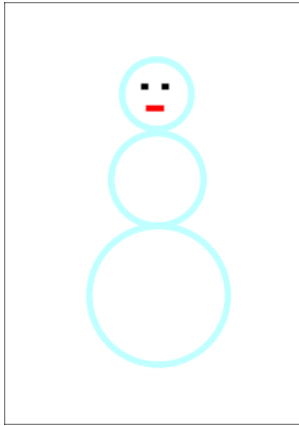
Задания 11-14:

11. Создайте программу, которая рисует нижеприведенный десятиугольник:



[решение тут](#)

12. Создайте программу, которая рисует снеговика:



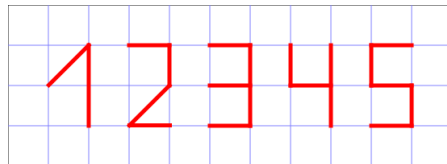
[решение тут](#)

13. Создайте программу, которая рисует радугу:



[решение тут](#)

14. *Создайте программу, которая на клетчатом холсте рисует цифры по нижеприведенному образцу:



[решение тут](#)

Печать на холсте Вычисления в KТurtle, переменные

Для указания Черепашке **написать** что-либо на холсте используется команда **напиши, размер_шрифта** – устанавливает **размер шрифта**, используемый для печати.

Для вычислений используются **математические операторы**: сложение (+), вычитание (-), умножение (*), деление (/) и возведение в степень (^). В вычислениях могут использоваться скобки. Выражение в скобках будет вычислено первым.

При вычислениях могут использоваться **переменные**. **Переменные** – это слова, начинающиеся с символа «\$», которым можно присвоить значения используя символ «=». Для использования переменной, её надо предварительно назначить, например: $\$D = 0$.

В вычислениях можно использовать дополнительные **математические команды** KТurtle:

округли (X) – округляет число **X** до ближайшего целого;

случайное (слч) X, Y – выдает число между **X** и **Y**;

остаток X, Y – выдает **остаток** после деления первого числа на второе;

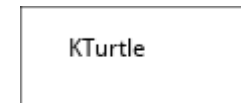
sqrt X – используется для нахождения **квадратного корня** от числа **X**;

pi – выдает константу **Пи**.

Есть и тригонометрические функции: **sin X, cos X, tg X, arcsin X, arccos X, arctg X**.

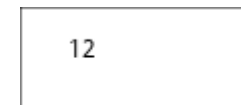
Задания 15-24:

15. Создайте программу, которая выводит на холст нижеприведенный текст:



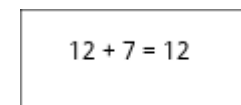
[решение тут](#)

16. Создайте программу, которая вычисляет и выводит на холст сумму чисел 5 и 7:



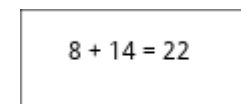
[решение тут](#)

17. Создайте программу, которая вычисляет и выводит на холст запись суммирования чисел 5 и 7:



[решение тут](#)

18. Создайте программу, которая вычисляет и выводит на холст запись суммирования двух случайных целых чисел от 2 до 20:



[решение тут](#)

19. Создайте программу, которая выводит на холст значение числа Пи:

3.14159

[решение тут](#)

20. Создайте программу, которая выводит на холст 2 целых случайных числа - от 10 до 20 и от 2 до 5, а также остаток от деления первого числа на второе в виде:

Делимое - 10
Делитель - 4
Остаток - 2

[решение тут](#)

21. Создайте программу, которая вычисляет длину гипотенузы прямоугольного треугольника с катетами 24 и 10 см и выводит на холст значения катетов и гипотенузы:

Катет 1 - 24 см
Катет 2 - 10 см
Гипотенуза - 26 см

[решение тут](#)

22. *Создайте программу, которая определяет и выводит на холст синус, косинус и тангенс угла 25 градусов:

Синус угла 25 градусов - 0.422618
Косинус угла 25 градусов - 0.906308
Тангенс угла 25 градусов - 0.466308

[решение тут](#)

23. *Создайте программу, которая определяет и выводит на холст величину угла, тангенс которого 1.2:

Тангенс угла 50 градусов ~ 1.2

[решение тут](#)

24. *Создайте программу, которая определяет и выводит на холст значения следующих углов - синус первого угла 0.6, косинус второго угла 0.4:

Синус угла 37 градусов ~ 0.6
Косинус угла 66 градусов ~ 0.4

[решение тут](#)

Условное выполнение

При записи команды:

если условие { ... } иначе { ... }

инструкции, расположенные между первыми скобками, будут выполнены только, если **условие** будет **истинным**, если наоборот – будут выполнены инструкции, находящиеся между вторыми скобками.

Задания 25-28:

25. Создайте программу, которая сравнивает число x с числом 4 и выводит результат сравнения:

6 больше 4!

[решение тут](#)

26. Создайте программу, которая выводит, кратно ли число x на 3:

11 не кратно 3!

[решение тут](#)

27. Создайте программу, которая определяет, оканчивается ли число x на 0 или 5:

число 15 оканчивается на 5.

[решение тут](#)

28. Создайте программу, которая сравнивает число x с числом y и выводит результат сравнения:

x и y равны.

[решение тут](#)

Считающие циклы Для и Пока

Цикл «для» – это цикл «со счётчиком», то есть он сохраняет **счётчик** для вас. Для записи:

для А = X до Y { ... }

при первом проходе **число А** принимает значение **X**. С каждым проходом **значение А увеличивается на 1** до того момента, **пока не достигнет числа Y**.

Пример записи:

```
для $x = 1 до 10 {
  напиши $x * 7
  вперёд 15
}
```

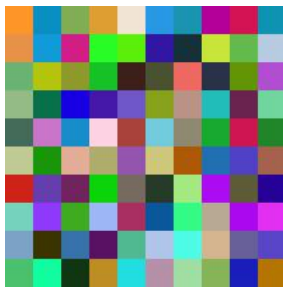
Цикл «пока» будет повторять команды, расположенные между скобками, до тех пор, пока логическое выражение не выдаст значение «ложь».

Пример записи:

```
$x = 1
пока $x < 5 {
  вперёд 10
  жди 1
  $x = $x + 1
}
```

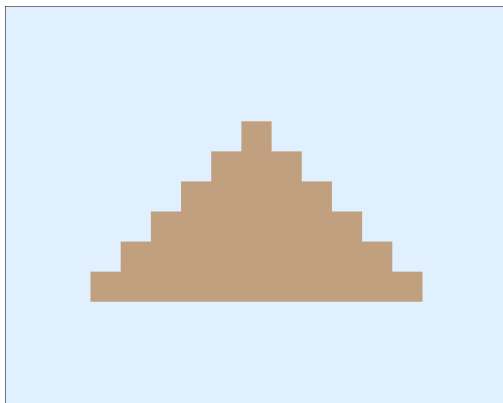
Задания 29-37:

29. Создайте программу, которая рисует мозаику:



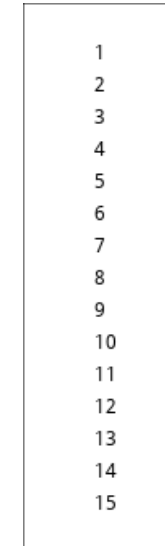
[решение тут](#)

30. Создайте программу, которая рисует нижеприведенную пирамиду:



[решение тут](#)

31. Создайте программу, которая выводит столбиком в порядке возрастания натуральные числа, сумма которых не меньше 120:



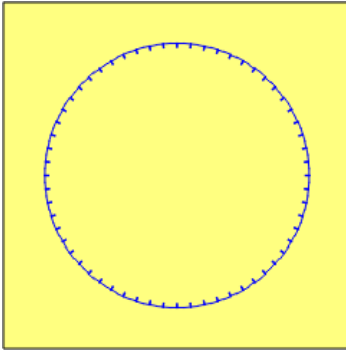
[решение тут](#)

32. *Создайте программу, которая выводит на холст таблицу умножения от 2 до 9:

Таблица умножения			
2 x 2 = 4	3 x 2 = 6	4 x 2 = 8	5 x 2 = 10
2 x 3 = 6	3 x 3 = 9	4 x 3 = 12	5 x 3 = 15
2 x 4 = 8	3 x 4 = 12	4 x 4 = 16	5 x 4 = 20
2 x 5 = 10	3 x 5 = 15	4 x 5 = 20	5 x 5 = 25
2 x 6 = 12	3 x 6 = 18	4 x 6 = 24	5 x 6 = 30
2 x 7 = 14	3 x 7 = 21	4 x 7 = 28	5 x 7 = 35
2 x 8 = 16	3 x 8 = 24	4 x 8 = 32	5 x 8 = 40
2 x 9 = 18	3 x 9 = 27	4 x 9 = 36	5 x 9 = 45
6 x 2 = 12	7 x 2 = 14	8 x 2 = 16	9 x 2 = 18
6 x 3 = 18	7 x 3 = 21	8 x 3 = 24	9 x 3 = 27
6 x 4 = 24	7 x 4 = 28	8 x 4 = 32	9 x 4 = 36
6 x 5 = 30	7 x 5 = 35	8 x 5 = 40	9 x 5 = 45
6 x 6 = 36	7 x 6 = 42	8 x 6 = 48	9 x 6 = 54
6 x 7 = 42	7 x 7 = 49	8 x 7 = 56	9 x 7 = 63
6 x 8 = 48	7 x 8 = 56	8 x 8 = 64	9 x 8 = 72
6 x 9 = 54	7 x 9 = 63	8 x 9 = 72	9 x 9 = 81

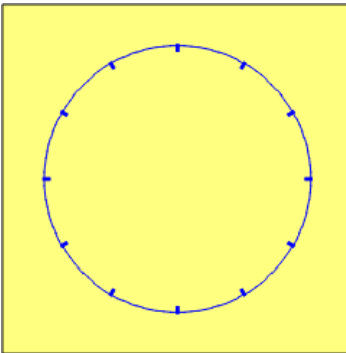
[решение тут](#)

33. Создайте программу, которая рисует заготовку циферблата часов с засечками минут:



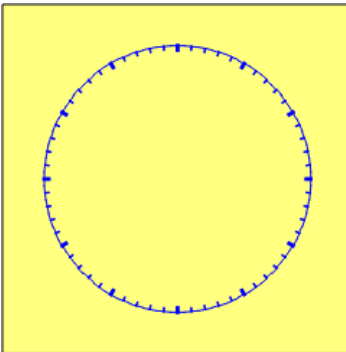
[решение тут](#)

34. *Создайте программу, которая рисует заготовку циферблата часов с засечками часов:



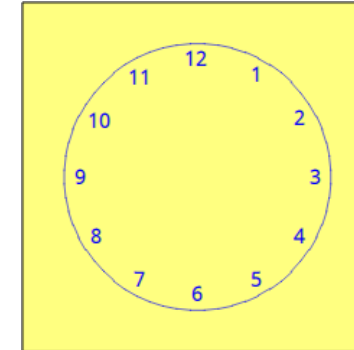
[решение тут](#)

35. *Создайте программу, которая рисует заготовку циферблата часов с засечками часов и минут:



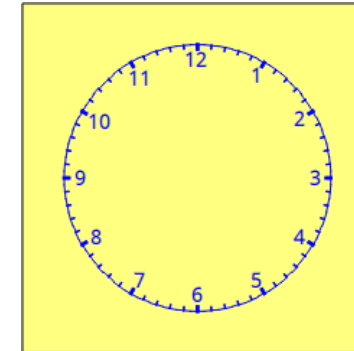
[решение тут](#)

36. Создайте программу, которая рисует заготовку циферблата часов с указанием цифр часов:



[решение тут](#)

37. *Создайте программу, которая рисует циферблат часов с засечками часов и минут, с указанием цифр часов:



[решение тут](#)

Создание собственных команд

Команда «**выучи**» предназначена для создания ваших **собственных команд**. Создаваемые вами команды могут принимать **входные параметры** и выполнять различные действия. Пример записи:

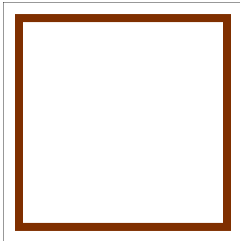
```

выучи круг $X {
    повтори 36 {
        вперёд $X
        налево 10
    }
}
иди 200, 200
круг 20
иди 300, 300
круг 40
    
```

В результате выполнения программы будут нарисованы два круга.

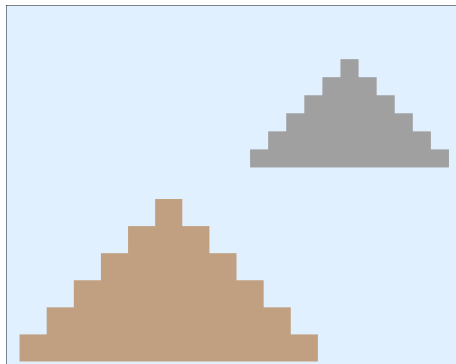
Задания 38-41:

38. Создайте программу, которая, используя Вашу собственную команду, нарисует нижеприведенный квадрат:



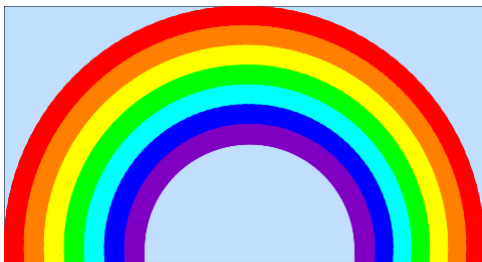
[решение тут](#)

39. Создайте программу, которая, используя Вашу собственную команду, нарисует нижеприведенные пирамиды:



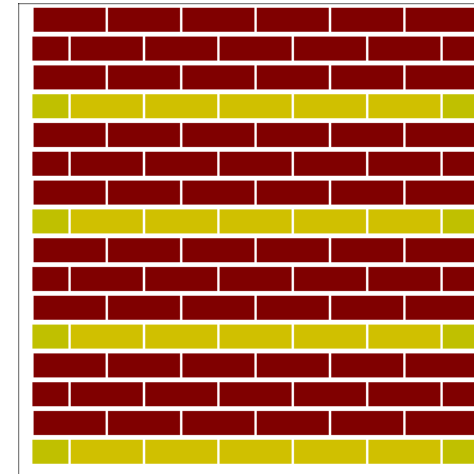
[решение тут](#)

40. Создайте программу, которая, используя Вашу собственную команду, нарисует нижеприведенную радугу:



[решение тут](#)

41. *Создайте программу, которая, используя Вашу собственную команду, нарисует нижеприведенную стену:



[решение тут](#)

Диалоги в KТurtle

Диалог – маленькое **окно**, которое **обеспечивает обратную связь** или **запрашивает ввести** что-либо. В KТurtle есть две команды для диалога:

сообщение X

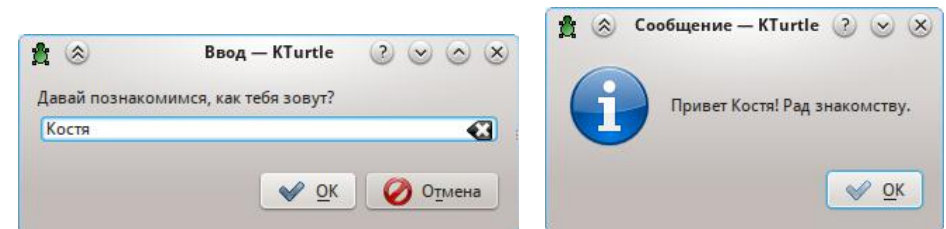
– **сообщение** будет показана в появившемся **окне**, и

спроси X

– показывает в окне диалога **строку** сообщения (**вопроса**), и строку из **поля ввода** принимает на вход.

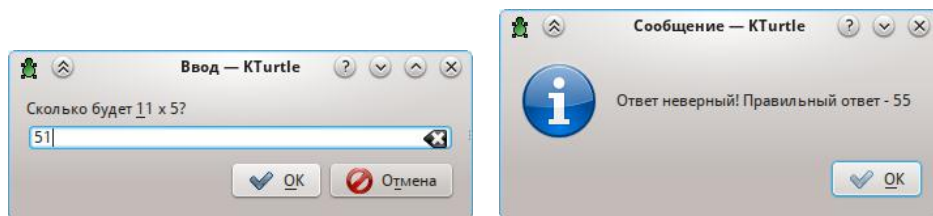
Задания 42-45:

42. Создайте программу, которая через диалоговые окна запрашивает имя и выводит сообщение:



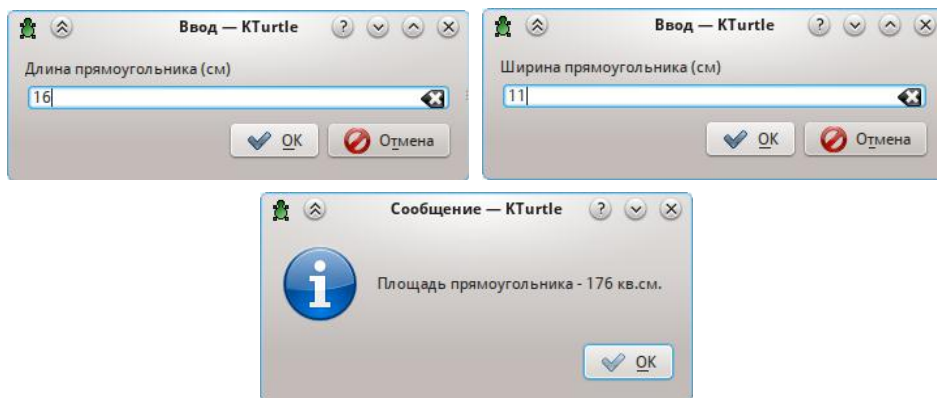
[решение тут](#)

43. Создайте программу, использующую диалоговые окна, которая проверяет математические знания на умножение чисел (в качестве множителей должны быть случайные числа от 2 до 15):



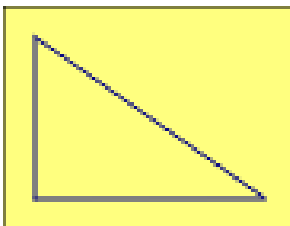
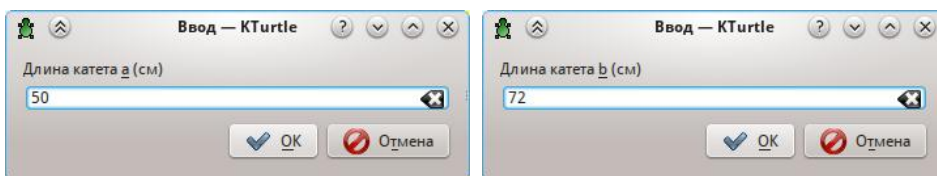
[решение тут](#)

44. Создайте программу, которая через диалоговые окна запрашивает стороны прямоугольника и выводит его площадь:



[решение тут](#)

45. *Создайте программу, которая через диалоговые окна запрашивает длины катетов прямоугольного треугольника и рисует его на холсте:



[решение тут](#)

Литература:

1. Cies Breijs. Руководство пользователя KТurtle, 2003-2007. Перевод на русский: В.Давыдов.