

acm.mipt.ru

олимпиады по программированию на Физтехе

Раздел «Язык Си» . Pyth1Func :

- [Функции \(на примере черепахи\)](#)
 - [Функции без параметров](#)
 - [Задача 1 - крест](#)
 - [Задача 2 - снежинка](#)
 - [Функция с одним параметром](#)
 - [Задача 3 - квадраты разных размеров](#)
 - [Функция с несколькими параметрами](#)
 - [Вызов функции из функции](#)
 - [Вызов математических функций sqrt, sin, cos](#)
 - [Задача 4 - солнце](#)
 - [Задача 5.1. Луч снежинки](#)
 - [Задача 5.2 - снежинка](#)
 - [Задачи на функции](#)
 - [Очень простые задачи - прямоугольник](#)
 - [Задача 1.](#)
 - [Задача 2.](#)
 - [Задача 4.](#)
 - [КП1. Прямоугольник в \(0,0\) левым верхним углом](#)
 - [КП2. Прямоугольник с центром в \(0,0\)](#)
 - [KR1. Повернуть прямоугольник в \(0, 0\)](#)
 - [KR2. Повернуть прямоугольник вокруг центра](#)
 - [Что дальше?](#)

Поиск

Поиск

Раздел «Язык Си»

[Главная](#)
[Зачем учить C?](#)
[Определения](#)

Инструменты:

[Поиск](#)
[Изменения](#)
[Index](#)
[Статистика](#)

Разделы

[Информация](#)
[Алгоритмы](#)
[Язык Си](#)
[Язык Ruby](#)
[Язык](#)
[Ассемблера](#)
[Ei Judge](#)
[Парадигмы](#)
[Образование](#)
[Сети](#)
[Objective C](#)

Logon>>

Автор задач раздела: Овсянникова Т.В.

Функции (на примере черепахи)

В задачах этого раздела циклы использовать ЗАПРЕЩЕНО

Функция – это именованный кусок кода.

Для чего:

- использовать этот код много раз;
- разбить задачу на подзадачи.

Функции без параметров

Можно научить черепаху новой команде.

Для этого надо придумать имя новой функции и записать в нее известные функции.

```
import turtle

t = turtle.Turtle()    # сделали черепаху, назвали черепаху t
t.shape("turtle")      # как черепаха выглядит
t.width(5)

# это новая функция
# Она называется sqfill (от square - "квадрат" и fill - "закрасить", filled square)
# def пишется с начала строки без пробелов и <Tab>
# рисовать квадрат будет черепаха t,
# поэтому пишем функцию там, где уже знаем о черепахе (после того, как ее сделали)
def sqfill():
    t.pencolor("red")
    t.fillcolor("yellow")

    t.begin_fill()      # начинаем рисовать кривую, внутри которой будем красить
    t.fd(100)
    t.lt(90)
    t.fd(100)
```

```

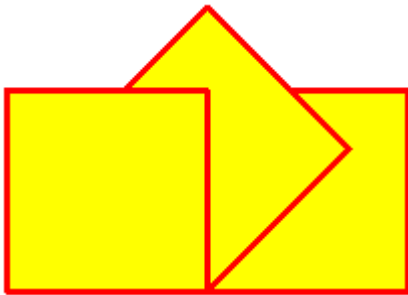
    t.lt(90)
    t.fd(100)
    t.lt(90)
    t.fd(100)
    t.lt(90)
    t.end_fill()      # закончили рисовать кривую, закрасили внутри нее
# Здесь новая команда sqfill закончилась.
# Закончили ставить <Tab>

# Тут место для выполнения старых и новых команд.
# выполняем команды. Пишем БЕЗ пробелов и <Tab>

sqfill()              # ВЫЗВАЛИ (call) функцию sqfill - исполнили все ее команды
t.lt(45)
sqfill()              # ВЫЗВАЛИ (call) функцию sqfill - исполнили все ее команды
t.lt(45)
sqfill()              # ВЫЗВАЛИ (call) функцию sqfill - исполнили все ее команды
t.lt(45)

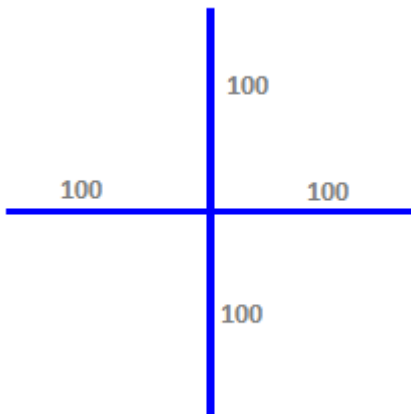
t.ht()    # hideturtle
turtle.done()

```



Задача 1 – крест

Написать функцию **cross()**, которая рисует крест со стороной 100

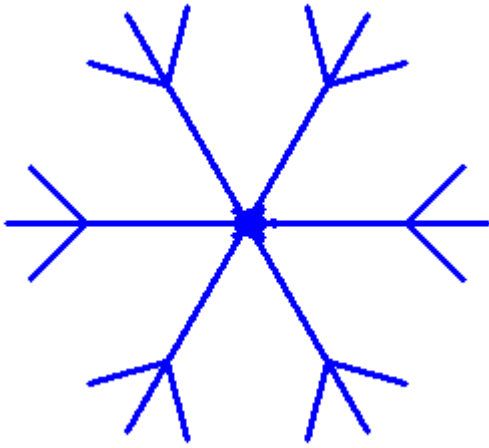


Задача 2 – снежинка

Написать функцию **snowline()** (snow – снег), которая рисует 1 часть снежинки.



Потом нарисовать такую снежинку.



Функция с одним параметром

Надо нарисовать 3 квадрата со сторонами 200, 150 и 100. Надо написать 3 функции? Нет! Можно создать одну функцию и передавать в нее аргументом размер стороны квадрата.

Напишем функцию `sq(size)`, которая рисует квадрат со стороной `size` линией красным цветом.

```
import turtle

t = turtle.Turtle()    # сделали черепаху, назвали черепаху t
t.shape("turtle")      # как черепаха выглядит
t.width(5)

# это новая функция
# Она называется sq, параметр назовем size
# def пишется с начала строки без пробелов и <Tab>
def sq(size):
    t.fd(size)
    t.lt(90)
    t.fd(size)
    t.lt(90)
    t.fd(size)
    t.lt(90)
    t.fd(size)
    t.lt(90)

# Здесь новая команда sq закончилась.
# Закончили ставить <Tab>

# Тут место для выполнения старых и новых команд.
# выполняем команды. Пишем БЕЗ пробелов и <Tab>

sq(200)                # ВЫЗВАЛИ (call) функцию sq, size=200 - исполнили все ее команды
sq(150)                # ВЫЗВАЛИ (call) функцию sq, size=150 - исполнили все ее команды
sq(100)                # ВЫЗВАЛИ (call) функцию sq, size=100 - исполнили все ее команды

t.ht()                # hideturtle
turtle.done()
```

Задача 3 – квадраты разных размеров

Введите размеры 2 квадратов с клавиатуры. Нарисуйте эти квадраты.

Функция с несколькими параметрами

Много параметров пишутся через запятую (,)

Каждый параметр имеет свое имя.

В параметры можно передавать числа.

Напишем функцию **rect(size, col)**, которая рисует прямоугольник. У него линии всегда красные и одна сторона в 3 раза больше другой.

В функцию передадим цвет внутри прямоугольника col и размер большой стороны size .

Размер маленькой стороны посчитаем size/3 .

С параметрами можно делать разные математические действия: +, -, * (умножить), / (разделить), // (взять целую часть от деления)

```
import turtle
from time import sleep

# Это прямоугольник (rectangle).
# Одна сторона у него в 3 раза больше другой.
# Линии красные.
# 2 параметра: первый col (цвет внутри) , второй size (размер)
def rect (size, col):
    # "red" - цвет линии, col - цвет внутри
    t.pencolor("red")
    t.fillcolor(col)

    t.begin_fill()
    t.forward(size)
    t.left(90)
    t.forward(size/3)
    t.left(90)
    t.forward(size)
    t.left(90)
    t.forward(size/3)
    t.left(90)
    t.end_fill()

t = turtle.Turtle()
t.shape("turtle")
t.width(3)
t.speed(0)

rect(300, "yellow")
t.left(90)
rect(120, "violet")

t.hideturtle()
turtle.done()
```



Вызов функции из функции

Новые функции можно писать в других новых функциях.

Нарисуем два квадрата одинакового размера

```
import turtle          # познакомили программу с пакетом turtle (черепаха)

t = turtle.Turtle()    # сделали черепаху, назвали черепаху t
t.shape("turtle")      # как черепаха выглядит
t.width(3)

# это новая функция
# Она называется sq (квадрат)
```

```

# def пишется с начала строки без пробелов и <Tab>
# рисовать лист будет черепаха t,
# поэтому пишем функцию там, где уже знаем о черепахе (после того, как ее сделали)
def sq():
    t.forward(100)
    t.left(90)
    t.forward(100)
    t.left(90)
    t.forward(100)
    t.left(90)
    t.forward(100)
# Здесь новая команда sq закончилась.

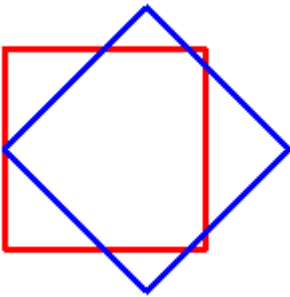
# Началась другая новая функция sq2 (два квадрата)
def sq2():
    t.color("red")
    sq()          # это наша функция sq
    t.back(50)
    t.left(45)
    t.color("blue")
    sq()          # это наша функция sq
# Здесь новая команда sq2 закончилась.

# Тут место для выполнения старых и новых команд.
# выполняем команды. Пишем БЕЗ пробелов и <Tab>

sq2()

turtle.done()

```



Вызов математических функций sqrt, sin, cos

Для того, чтобы посчитать квадратный корень, нужно вызвать функцию **sqrt** библиотеки **math**

Нарисуем квадрат со стороной size и вписанный в него ромб.

```

import turtle
from math import sqrt, sin, cos # из библиотеки math взяли функции sqrt, sin и cos

t = turtle.Turtle() # сделали черепаху, назвали черепаху t
t.shape("turtle")   # как черепаха выглядит
t.width(3)

# началась функция sq
def sq(size):
    t.fd(size)
    t.lt(90)
    t.fd(size)
    t.lt(90)
    t.fd(size)
    t.lt(90)
    t.fd(size)
    t.lt(90)
# закончилась функция sq

# функция sq2(size, col1, col2) рисует два квадрата, один вписан в другой,

```

```
# size - размер стороны большого квадрата
# col1 - цвет большого квадрата
# col2 - цвет вписанного квадрата
def sq2(size, col1, col2):
    t.color(col2)
    sq(size)

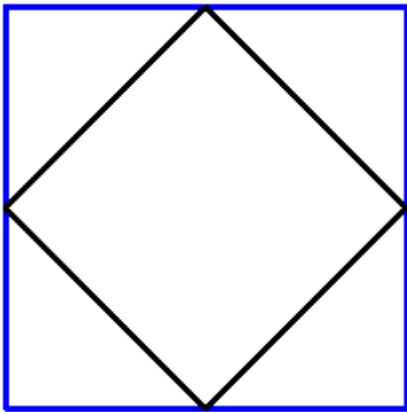
    t.fd(size/2)
    t.lt(45)

    # размер нового квадрата - это размер старого квадрата,
    # деленный на корень из 2.
    t.color(col2)
    sq(size / sqrt(2))

# закончилась функция sq2

# пишем код дальше
sq(200, "blue", "black")

turtle.done()
```



Задача 4 – солнце

Нарисовать солнце.

Написать функцию `sun1(r, size, angle, col)`

`r` – радиус круга солнца

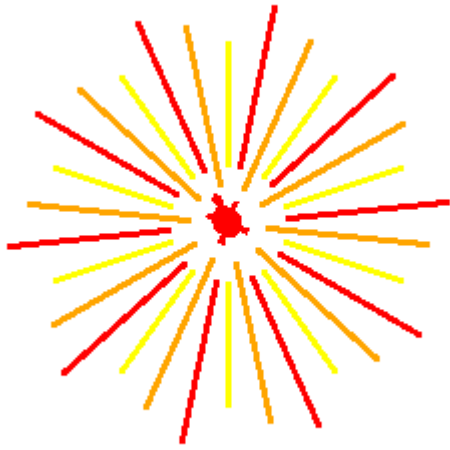
`size` – длина луча

`angle` – на сколько повернуть первый луч

`col` – цвет луча.

Написать вторую функцию `sun(r, size)` которая использует `sun1(r, size, angle, col)`

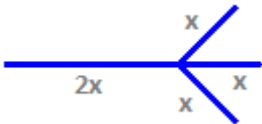
Получить картинку:



Задача 5.1. Луч снежинки

У луча снежинки ветки в 3 раза меньше, чем длина луча.

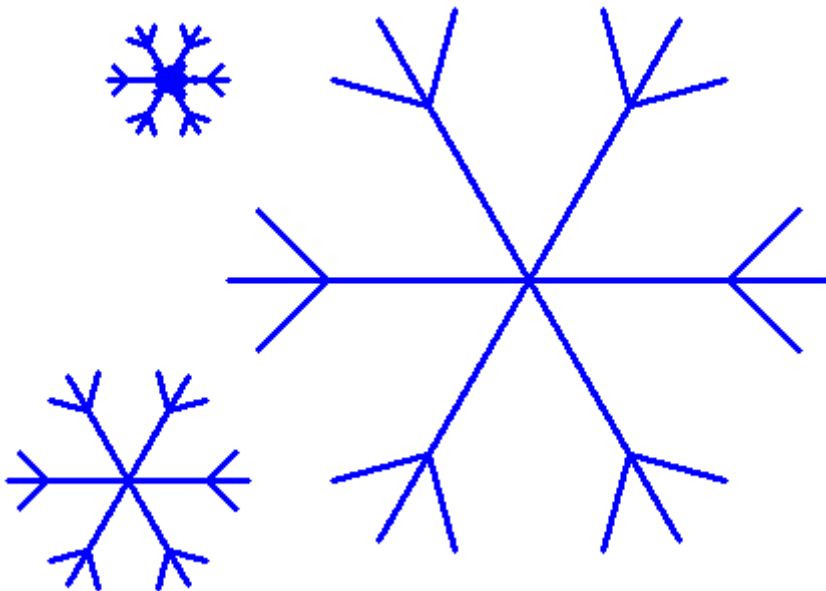
Написать функцию рисования 1 луча с параметром длина ветки. `snowline(x)`



Задача 5.2 – снежинка

Написать функцию рисования снежинки с параметром размер снежинки. `snowflake(size)`

Нарисовать несколько разных снежинок.



Задачи на функции

В этом разделе подумайте, какие функции надо создать и как их вызвать.

Рекомендуем прочитать сразу все задачи раздела, а потом начинать решать первую задачу.

Очень простые задачи – прямоугольник

Задача 1.

Даны длины сторон прямоугольника a и b . Нарисовать прямоугольник с этими сторонами (из центра координат).

Задача 2.

Даны длины сторон прямоугольника a и b .

- Нарисовать прямоугольник с этими сторонами.
- Передвинуть черепаху в центр этого прямоугольника.
- Поставить туда точку другим цветом

Задача 4.

Даны координаты точки x и y . Даны длины сторон прямоугольника a и b .

- Нарисовать точку с координатами x и y . Это центр прямоугольника.
- Нарисовать прямоугольник с центром в данной точке.

КП1. Прямоугольник в $(0,0)$ левым верхним углом

Даны x , y , w , h .

- Нарисовать прямоугольник с левой верхней точкой (x, y) , шириной w и высотой h .
- Поставить точку в его центр.
- Передвинуть прямоугольник (move) так, чтобы его левая верхняя точка была в точке $(0, 0)$. Нарисовать этот прямоугольник другим цветом.

КП2. Прямоугольник с центром в $(0,0)$

Даны x , y , w , h .

- Нарисовать прямоугольник с левой верхней точкой (x, y) , шириной w и высотой h .
- Поставить точку в его центр.
- Передвинуть прямоугольник (move) так, чтобы его центр был в точке $(0, 0)$. Нарисовать этот прямоугольник другим цветом.

КР1. Повернуть прямоугольник в $(0, 0)$

Даны w , h .

- Нарисовать прямоугольник с центром в точке $(0, 0)$, шириной w и высотой h .
- Повернуть его на 90 градусов вокруг $(0,0)$.
- Нарисовать новый прямоугольник другим цветом.

КР2. Повернуть прямоугольник вокруг центра

Даны x , y , w , h .

- Нарисовать прямоугольник с левой верхней точкой (x, y) , шириной w и высотой h .
- Поставить точку в его центр.
- Повернуть его на 90 градусов вокруг центра.
- Нарисовать новый прямоугольник другим цветом.

Что дальше?

Дальше можно прочитать про [рекурсивный вызов функций](#) и сделать задачи этого раздела.

-- TatyanaDerbysheva - 17 Sep 2018

(с) Материалы раздела "Язык Си" публикуются под лицензией [GNU Free Documentation License](#).