Лабораторная работа 4 Функции

1. Даны четыре действительных числа: x1, y1, x2, y2. Напишите функцию distance(x1, y1, x2, y2), вычисляющая расстояние между точкой (x1,y1) и (x2,y2). Считайте четыре действительных числа и выведите результат работы этой функции.

**Sample Input 1:**

0

0

1

1

**Sample Output 1:**

1.4142135623730951

**Sample Input 2:**

0

0

1

0

**Sample Output 2:**

1.0

**Sample Input 3:**

3

-2

-1

7

**Sample Output 3:**

9.848857801796104

1. Напишите функцию `miles\_to\_km`, которая будет переводить мили в километры. В одной миле примерно 1.609 километров.
2. Напишите функцию mean, которая будет считать среднее арифметическое переданных в нее аргументов.

mean(1,2,2,4) == 2.25

mean(1) == 1.0

mean(1, 10, 10) == 7.0

1. Напишите функцию concat, которая будет принимать произвольное число строк и объединять их в одну.

concat("a") == "a"

concat("a", "b") == "ab"

concat("a", "asdf") == "aasdf"

1. Напишите функцию call, которая будет принимать произвольную функцию, аргументы для неё и делать вызов переданной функции.

def add(a, b):

return a + b

call(add, 2, 3) == 5

call(add, a = 4, b = 1) == 5

call(max, [3,2,4,7]) == 7

1. Написать модуль, в котором содержатся все эти функции, а также отдельные скрипты проверки для каждой функции

Объектно-ориентированное программирование

1. Реализуйте класс Point (точка). У этого класса должны быть конструктор, принимающий два числа x и y, координаты точки на плоскости; атрибуты x и y через которые можно будет получить координаты точки; метод dist, который принимает еще один объект класса Point и находит эвклидово расстояние между двумя точками.

Примеры вызова:

p1 = Point(1.5, 1)

p2 = Point(1.5, 2)

p1.dist(p2) == 1

1. Напишите класс Circle, представляющий круг. У этого класса должны быть: конструктор принимающий объект класса Point, точку центра круга, и число, радиус круга; атрибуты center и radius, возвращающие центр и радиус круга соответственно; метод square, который возвращает площадь круга.

circ = Circle(Point(0, 3), 4)

circ.radius == 4

circ.center.x == 0

Circle(Point(0, 0), 0).square() == 0

1. Доработайте класс Circle. Добавьте ему метод do\_intersect, который будет принимать другой объект класса Circle и возвращать True или False в зависимости от того, пересекаются круги или нет.

Весь остальной код класса вы можете скопировать из решения предыдущей задачи.

1. Добавьте возможность сравнивать объекты класса Circle. Больше будет тот круг, у которого площадь больше.

sm = Circle(Point(12, 12), 425)

bg = Circle(Point(9, 1), 10293.4)

sm < bg

sm == sm

Когда вы их напишите, вы сможете не только сравнивать круги, но и сортировать списки таких объектов:

mid = Circle(Point(0,0), 500.4)

circ\_l = [bg, sm, mid]

circ\_l.sort()

circ\_l[0] is sm and circ\_l[1] is mid and circ\_l[2] is bg

1. Все животные имеют между собой много общего. Они умеют издавать звуки и перемещатся. А еще у них есть имена, которые им дают при рождении. Это общее можно описать следующим классом:

class Animal:

def \_\_init\_\_(self, name):

self.name = name

def make\_sound(self):

print(self.name + " says " + self.sound)

def move(self):

raise NotImplementedError()

Создайте классы Dog и Snake, унаследовав их от Animal.  
Объект класса Dog должен издавать звук woof. Ходят собаки пешком. Поэтому при вызове метода move должен выводится текст "<Name> walks", где <Name> – это имя животного.  
Объект класса Snake должен издавать звук hsss. Змеи ползают. Поэтому при вызове метода move должен выводится текст "<Name> crawles", где <Name> – это имя животного.  
  
Примеры проверки работы написанных классов

>>> snake = Snake('John')

>>> snake.make\_sound()

John says hss

>>> snake.move()

John crawles

>>>

>>> dog = Dog('Mike')

>>> dog.make\_sound()

Mike says woof

>>> dog.move()

Mike walks

>>>