Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования

«Курский государственный университет»

Кафедра КТиИО

Направление подготовки **Информатика и вычислительная техника**

Направленность(профиль) **Прикладной искусственный интеллект**

Факультет **Физики, математики, информатики**

Дисциплина **Программирование**

Форма обучения **очная**

**ОТЧЕТ**

по лабораторной работе №4

«Декораторы функций.»

Выполнил: студент 1 курса Алиев А.М.о.

Проверил: к.п.н., доцент Костенко И.Е.

Курск – 2024

## Содержание

[Содержание 2](#_Toc162805349)

[**Лабораторная работа №4** 3](#_Toc162805350)

[Задание №1 3](#_Toc162805351)

[Задание №2 4](#_Toc162805352)

[Задание №3 6](#_Toc162805353)

# **Лабораторная работа №****4**

**ДЕКОРАТОРЫ ФУНКЦИЙ**

**РЕШЕНИЕ ЗАДАЧ**

## Задание №1

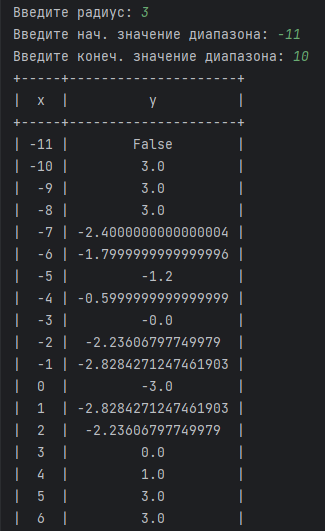
**Условие**

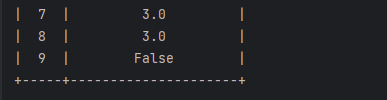
Сделать декоратор, который проверяет значение аргумента и либо не запускает функцию, либо запускает ее с каким-то стандартным значением в том случае, если аргумент не входит в область определения.

**Код решения**

from prettytable import PrettyTable  
  
  
def decorate(func):  
 def check(x):  
 if x < -10 or x > 8:  
 return x, False  
 else:  
 return func(x)  
 return check  
  
  
@decorate  
def function(x):  
 if x <= -8:  
 y = r  
 elif x > -8 and x < -r:  
 y = ((r / 5) \* x + (r \*\* 2)/ 5)  
 elif x < r and x >= -r:  
 y = -(r \*\* 2 - x \*\* 2) \*\* 0.5  
 elif x >= r and x < 5:  
 y = x - r  
 elif x >= 5 and x <= 8:  
 y = r  
 return (x, y)  
  
  
r = float(input("Введите радиус: "))  
start, stop = int(input("Введите нач. значение диапазона: ")), int(input("Введите конеч. значение диапазона: "))  
mytable = PrettyTable()  
mytable.field\_names = ['x', 'y']  
for i in range(start, stop):  
 x, y = function(i)  
 mytable.add\_row([x, y])  
print(mytable)

**Тестирование**





## Задание №2

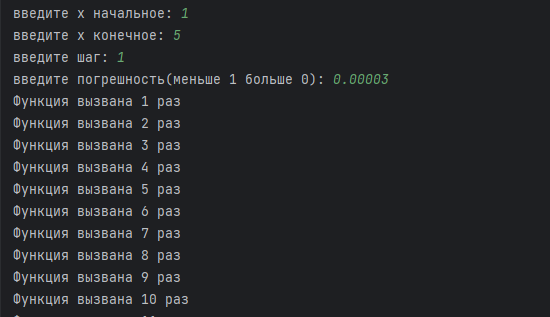
**Условие**

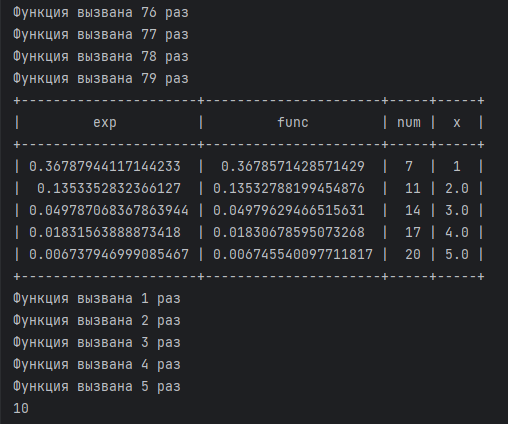
Изменить функцию вычисления разложения, сделать ее рекурсивной. Делаем декоратор, который выводит журнал запуска функции.

**Код решения**

from math import exp, factorial  
from prettytable import PrettyTable  
  
  
def count(func):  
 c = 0  
 def wrapper(\*args, \*\*kwargs):  
 nonlocal c  
 c += 1  
 print(f"Функция вызвана {c} раз")  
 return func(\*args, \*\*kwargs)  
 return wrapper  
  
  
@count  
def f(x, eps, n = 0, s = 0):  
 sl = (((-1) \*\* n) \* (x \*\* n)) / factorial(n)  
 if abs(sl) > eps:  
 return f(x, eps, n + 1, s + sl)  
 else:  
 return (s, n - 1)  
  
  
  
@count  
def fi(x):  
 x -= 1  
 if x > 10:  
 return fi(x)  
 elif x < 10:  
 x += 3  
 return fi(x)  
 else:  
 print(x)  
  
  
x0 = int(input("введите х начальное: "))  
x1 = int(input("введите х конечное: "))  
dx = float(input("введите шаг: "))  
eps = float(input("введите погрешность(меньше 1 больше 0): "))  
mytable = PrettyTable()  
mytable.field\_names = ['exp', 'func', 'num', 'x']  
while x0 <= x1:  
 s, n = f(x0, eps)  
 mytable.add\_row([exp(-x0), s, n, x0])  
 x0 += dx  
print(mytable)  
fi(6)

**Тестирование**





## 

## Задание №3

**Условие**

Для проверки координат требуется пароль, в случае если пароль правильный проверка запускается и пароль больше не требуется, иначе проверка не запустится и будет заново просить пароль

**Код решения**

def password(func):  
 flag = 0  
 def check(x):  
 nonlocal flag  
 if flag == 0:  
 if input("Введите пароль: ") != "admin":  
 print("Неправильный пароль!")  
 return check(x)  
 else:  
 flag = 1  
 return func(x)  
 else:  
 return func(x)  
 return check  
  
@password  
def test(x, y, r):  
 if x < 0 and y > 0:  
 if (x \*\* 2 + y \*\* 2 < r \*\* 2):  
 return True  
 # print("точка попадает в заштрихованную область!")  
 else:  
 return False  
 elif x > 0 and y < 0:  
 if (y > (-2 \* x) and (y > (2 \* x - 8))):  
 return True  
 # print("точка попадает в заштрихованную область!")  
 else:  
 return False  
 # print("Не попадает в закрашенную область!")  
 else:  
 return False  
 # print("Не попадает в закрашенную область!")  
  
  
n = int(input("Введите кол-во координат: "))  
r = abs(float(input("Введите радиус: ")))  
for \_ in range(n):  
 x = float(input("Введите координату x: "))  
 y = float(input("Введите координату y: "))  
 if test(x, y, r) == 1:  
 print("точка попадает в заштрихованную область!")  
 else:  
 print("Не попадает в закрашенную область!")

**Тестирование**

