МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ОРЛОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИМЕНИ И.С. ТУРГЕНЕВА»

Кафедра программной инженерии

**ОТЧЕТ**

по лабораторной работе № 2

на тему: «Функциональное программирование»

по дисциплине: «Программирование на языке Python»

Вариант № 18

Выполнил: Шорин В.Д. Шифр: 171406

Институт приборостроения, автоматизации и информационных технологий

Направление: 09.03.04 «Программная инженерия»

Группа: 71-ПГ

Проверили: Захарова О.В., Раков В.И.

Отметка о зачете:

Дата: «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2019 г.

Орел, 2019 г.

**Задание:**

1. Написать программу, вычисляющую выражение (лабораторная работа № 1, вариант по списку студентов, задание 1). Значения всех переменных задавать с клавиатуры. При задании неверных данных выдать сообщение об ошибке. Использовать модуль math. При разработке программы не использовать управляющие конструкции. Использовать функции первого класса и высшего порядка.

2. Написать программу, вычисляющую площадь фигур. Функционал программы разработать в соответствии с первой лабораторной работой, вариантом по списку студентов, заданием 3 (например для первого варианта: вычисление площади прямоугольника («R»); вычисление площади прямоугольного треугольника («T»); вычисление площади многоугольника («M»), выход из программы («E»); в случае ввода неверных данных выдать сообщение об ошибке).

Входные данные задать в виде одного списка. Программа должна обеспечить возможность вычислять площадь фигур до тех пор, пока в списке есть входные данные. При разработке программы не использовать управляющие конструкции. Использовать функции первого класса и высшего порядка.

Пример списка с входными данными:

L = [ ['R', 'r', 'M', 'T', 'T', 'E'], [1, 2, 3, 4, 5, 6], <входные данные > ]

**Код:**

**«main.py»**

import task1

import task2

while True:

print("Choose task:")

print("1 - 1")

print("2 - 2")

print("0 - exit")

task\_number = input("Your choose: ")

print()

if task\_number == '1':

task1.TaskOneMain()

elif task\_number == '2':

task2.TaskTwoMain()

elif task\_number == '0':

break

else:

print(task\_number + "\_Undefined command")

**«task1.py»**

import math

def TaskOneMain():

while True:

while True:

a = input("a = ")

try:

a = float(a)

except ValueError:

print("Its not float")

continue

break

while True:

b = input("b = ")

try:

b = float(b)

except ValueError:

print("Its not float")

continue

break

while True:

c = input("c = ")

try:

c = float(c)

except ValueError:

print("Its not float")

continue

break

while True:

n = input("n = ")

try:

n = float(n)

except ValueError:

print("Its not float")

continue

break

while True:

x = input("x = ")

try:

x = float(x)

except ValueError:

print("Its not float")

continue

break

break

calculate = lambda a, b, c, n, x: ((5 \* math.pow(a, n \* x))

/ math.fabs(b + c)) \

- math.sqrt(math.fabs(math.sin(math.pow(x, n \* x))))

print("Result: " + str(calculate(a, b, c, n, x)))

input("Press any key...")

**«task2.py»**

import math

import re

def TaskTwoMain():

while True:

print("Ellipse - e (enter two values for ellipse)")

print("Triangle - t (enter two values for triangle)")

print("Rectangle - r ( enter two values for rectangle)")

print("Quit - q")

commands = list(filter(

lambda command: command.lower() in ['e', 't', 'r', 'q'],

[i for i in input().split()]

))

#

#

print("Enter a, b:")

values = list(filter(

lambda value: re.match("[0-9.0-9]", value),

[j for j in input().split()]

))

results = list\

(

map

(

lambda command:

(

command.lower() == 'e' and

(

float(values[0]) \* float(values[1]) \* math.pi

)

or command.lower() == 't' and

(

1/2 \* float(values[0]) \* float(values[1])

)

or command.lower() == 'r' and (float(values[0]) \* float(values[1]))

or command.lower() == 'q' and "exit"

),

commands

)

)

list(map

(

lambda command, result: print(f'{command}: {result}'),

commands,

results

)

)

list(map(

lambda result:

(

result == "exit" and (exit(0))

),

results

))