

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО 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»

Выполнил: Евдокимов Н.А.

Институт приборостроения, автоматизации и информационных технологий

Направление: 09.03.04 «Программная инженерия»

Группа: 71-ПГ

Проверила: Захарова О.В.

Отметка о зачете:

Дата: « \_\_\_\_ » \_\_\_\_\_ 2019 г.

Орёл, 2019

## Задание

### Лабораторная работа № 2 Функциональное программирование

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 («Т»); вычисление площади многоугольника («М»); выход из программы («Е»); 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

1.

```
import math as m
```

```
func = lambda x, y, n: 3 * m.pow(m.cos(x - m.pi / 6), n) / (1 / 2 + m.sin(y))
```

```
x = None
```

```
while x is None:
```

```
    try:
```

```
        x = float(input('Введите x: '))
```

```
    except ValueError:
```

```
        x = None
```

```
y = None
```

```
while y is None:
```

```
    try:
```

```
        y = float(input('Введите y: '))
```

```
    except ValueError:
```

```
        y = None
```

```
n = None
```

```
while n is None:
```

```
    try:
```

```
        n = float(input('Введите n: '))
```

```
    except ValueError:
```

```
        n = None
```

```
print('Результат вычислений:', func(x, y, n))
```

2.

```
import textwrap as tw
```

```
import math as m
```

```

def print_menu():
    menu = """
        R) Площадь прямоугольника
        T) Площадь прямоугольного треугольника
        M) Площадь правильного многоугольника
        E) Выйти
    """
    print(tw.dedent(menu))

def rectangle_area():
    a, b = None, None

    while a is None or b is None:
        try:
            user_input = input('Введите две стороны через пробел: ')
            splitted_input = user_input.lstrip().rstrip().split(' ')
            a, b = splitted_input[0], splitted_input[1]

            a = float(a)
            b = float(b)
        except ValueError:
            a = None
            b = None

    area = lambda a, b: a * b
    print(area(a, b))

def right_triangle_area():
    a, b = None, None

    while a is None or b is None:
        try:
            user_input = input('Введите две стороны через пробел: ')
            splitted_input = user_input.lstrip().rstrip().split(' ')
            a, b = splitted_input[0], splitted_input[1]

            a = float(a)
            b = float(b)
        except ValueError:
            a = None
            b = None

    area = lambda a, b: a * b / 2
    print(area(a, b))

def right_polygon_area():
    n, a = None, None

    while n is None or a is None:
        try:
            user_input = input('Введите количество сторон и их длину через пробел: ')
            splitted_input = user_input.lstrip().rstrip().split(' ')
            n, a = splitted_input[0], splitted_input[1]

            n = float(n)
            a = float(a)
        except ValueError:
            n = None

```

```
a = None
```

```
area = lambda n, a: n / 4 * a**2 * (1 / m.tan(m.pi / n))  
print(area(n, a))
```

```
def namestr(obj, namespace):  
    return [name for name in namespace if namespace[name] is obj]
```

```
def run():  
    should_run = True  
    rect = lambda a, b: a * b  
    right_tr = lambda a, b: a * b / 2  
    right_pol = lambda n, a: n / 4 * a**2 * (1 / m.tan(m.pi / n))  
  
    list = [[rect, rect, right_tr, rect, right_pol, rect], [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]]  
  
    for func in list[0]:  
        result = func(list[1][0], list[1][1])  
        print('Result of {}() with args [{}, {}] is {}'.  
              .format(namestr(func, locals())[0], list[1][0], list[1][1], result))
```

```
if __name__ == '__main__':  
    run()
```