

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

**ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И
ПРОГРАММИРОВАНИЯ**

ЛАБОРАТОРНАЯ РАБОТА №4

Выполнил:

Попович Виталий Сергеевич
студент 1 курса группы М3100

Проверил:

Андреев Николай Владимирович

Цель тестирования

Функциональное тестирование

Получение верно рабочей версии программы

1.Проверка верности формул из кода

2.Проверка верности работы написанного кода(синтаксиса)

Тестируемый продукт

Тестируем 4 модуля: circle.py, rectangle.py, square.py, triangle.py

В каждом модуле есть по 2 функции, которые мы и будем тестировать:

area() – вычисление площади

perimeter() – вычисление периметра

Область тестирования

1)Circle.py

def area(r) – принимает радиус круга, возвращает площадь круга

def perimeter(r) – принимает радиус круга, возвращает периметр круга

Circle.py

```
import math

def area(r):
    return math.pi * r * r

def perimeter(r):
    return 2 * math.pi * r
```

2)Rectangle.py

def area(a, b) – принимает длину сторон прямоугольника, возвращает площадь прямоугольника

def perimeter(a, b) – принимает длину сторон прямоугольника, возвращает периметр прямо

Rectangle.py

```
def area(a, b):  
    return a * b  
  
1 usage  
def perimeter(a, b):  
    return (a + b) * 2
```

3)Square.py

def area(a) – принимает длину стороны квадрата, возвращает площадь квадрата

def perimeter(a) – принимает длину стороны квадрата, возвращает периметр квадрата

Square.py

```
1  
2 def area(a):  
3     return a * a  
4  
5  
6 def perimeter(a):  
7     return 4 * a  
8
```

4)Triangle.py

def area(a, h) – принимает длину стороны и высоты треугольника, возвращает площадь треугольника

def perimeter(a, b, c) – принимает длины сторон треугольника, возвращает периметр треугольника

Triangle.py

```
1 def area(a, h):  
2     return a * h / 2  
3  
4 def perimeter(a, b, c):  
5     return a + b + c
```

Стратегия тестирования :

На основе входных данных и формул вычисления площадей и периметров рассчитаем ожидаемый результат программы. Запустим программу и сравним ответы. В случае несовпадения сообщим об этом.

Критерий приемки

Ответы вычисленные программой должны совпадать с ответами заранее вычисленными вручную

Ожидаемые результаты

Каждая функция из каждого модуля принимает неотрицательные данные(длины сторон/высот) фигур в типе данных float или integer
Возвращают также в типе данных float(с точностью до 5 знаков) или integer

Тестирование

Circle.py

Код тестов

```
2 usages
def area(r):
    return math.pi * r * r

2 usages
def perimeter(r):
    return 2 * math.pi * r

class CircleTestcase(unittest.TestCase):
    def test_zero_mul(self):
        res = area(0)
        self.assertEqual(res, 0)
        res = perimeter(0)
        self.assertEqual(res, 0)

    def test_calculation_area(self):
        res = round(area(7), 5)
        self.assertEqual(res, 153.93804)

    def test_calculation_perimeter(self):
        res = round(perimeter(4), 5)
        self.assertEqual(res, 25.13274)
```

Результат тестирования

```
(venv) PS C:\Vitaly\projectLabPython> python.exe -m unittest circle.py
...
-----
Ran 3 tests in 0.000s

OK
(venv) PS C:\Vitaly\projectLabPython> 
```

Square.py

Код тестов

```
import unittest
2 usages
def area(a):
    return a * a

2 usages
def perimeter(a):
    return 4 * a

class SquareTestcase(unittest.TestCase):
    def test_zero_mul(self):
        res = area(0)
        self.assertEqual(res, 0)
        res = perimeter(0)
        self.assertEqual(res, 0)

    def test_calculation_area(self):
        res = area(4)
        self.assertEqual(res, 16)

    def test_calculation_perimeter(self):
        res = perimeter(4)
        self.assertEqual(res, 16)
```

Результат тестирования

```
(venv) PS C:\Vitaly\projectLabPython> python.exe -m unittest square.py
...
-----
Ran 3 tests in 0.000s

OK
(venv) PS C:\Vitaly\projectLabPython>
```

Triangle.py

Код тестов

```
1 import unittest
2
3 2 usages
4 def area(a, h):
5     return a * h / 2
6
7 2 usages
8 def perimeter(a, b, c):
9     return a + b + c
10
11 class TriangleTestcase(unittest.TestCase):
12     def test_zero_mul(self):
13         res = area(0, 5)
14         self.assertEqual(res, 0)
15         res = perimeter(0, 0, 0)
16         self.assertEqual(res, 0)
17
18     def test_calculation_area(self):
19         res = area(8, 6)
20         self.assertEqual(res, 24)
21
22     def test_calculation_perimeter(self):
23         res = perimeter(5, 6, 8)
24         self.assertEqual(res, 19)
```

Результат тестирования

```
(venv) PS C:\Vitaly\projectLabPython> python.exe -m unittest triangle.py
...
-----
Ran 3 tests in 0.000s

OK
(venv) PS C:\Vitaly\projectLabPython> 
```

Rectangle.py

Код тестов

```
import unittest

2 usages
def area(a, b):
    return a * b

2 usages
def perimeter(a, b):
    return (a + b) * 2

▶ class RectangleTestcase(unittest.TestCase):
▶     def test_zero_mul(self):
        res = area(0, 5)
        self.assertEqual(res, 0)
        res = perimeter(0, 0)
        self.assertEqual(res, 0)

▶     def test_calculation_area(self):
        res = area(5, 6)
        self.assertEqual(res, 30)

▶     def test_calculation_perimeter(self):
        res = perimeter(5, 6)
        self.assertEqual(res, 22)
```

Результат тестирования

```
(venv) PS C:\Vitaly\projectLabPython> python.exe -m unittest rectangle.py
...
-----
Ran 3 tests in 0.000s

OK
(venv) PS C:\Vitaly\projectLabPython>
```

Исходники кода можно найти тут: [ссылка на репозиторий Git Hub](#)
Commit с изменениями лабораторной 4: [ссылка на КОММИТ](#)