

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

Санкт-Петербургский национальный исследовательский университет ИТМО

Мегафакультет трансляционных информационных технологий

Факультет информационных технологий и программирования

Лабораторная Работа №4

По дисциплине «Инструментальные средства разработки ПО»

Выполнил студент группы №М3102

Ерофеев Иван Константинович

Проверил

Кириллюк Денис Алексеевич



УНИВЕРСИТЕТ ИТМО

Санкт-Петербург

2024

1. Вот мои функции для вычисления характеристик произвольного прямоугольника, заданного двумя сторонами.

```
def area(a, b):  
    return a * b  
  
def perimeter(a, b):  
    return 2 * (a + b)  
  
def diagonal(a, b):  
    return math.sqrt(a**2 + b**2)  
  
def inradius(a, b):  
    return (a * b) / (a + b) if a + b != 0 else 0  
  
def outradius(a, b):  
    return math.sqrt(a**2 + b**2) / 2
```

2. Моей задачей было написать тесты, позволяющие оценить правильность выполнения данных функций. Для этого я использовал библиотеку unittest которая позволяет написать стресс тесты для программы
3. Функции, которые тестируются:

area(a, b) — вычисляет площадь прямоугольника.

perimeter(a, b) — вычисляет периметр прямоугольника.

diagonal(a, b) — вычисляет длину диагонали прямоугольника.

inradius(a, b) — вычисляет радиус вписанной окружности.

outradius(a, b) — вычисляет радиус описанной окружности

4. Тестируются в том числе пограничные и некорректные значения, например когда сторона отрицательная или равна нулю. Также тестируются частные случаи, например когда прямоугольник является квадратом
5. Написал тесты для проверки случая, когда одна сторона равна нулю

```
def test_one_side_is_zero(self):  
    self.assertEqual(area(10, 0), 0)  
    self.assertEqual(area(0, 0), 0)  
    self.assertEqual(area(-1, 0), 0)  
    self.assertEqual(area(-1.5, 0), 0)  
    self.assertEqual(area(1000000000039, 0), 0)
```

6. Написал тесты для проверки площади, когда прямоугольник является квадратом

```
def test_square_area(self):
    self.assertEqual(area(1, 1), 1);
    self.assertEqual(area(2, 2), 4);
    self.assertEqual(area(10, 10), 100);
    self.assertEqual(area(11, 11), 121);
    self.assertEqual(area(100000000039, 100000000039), 100000000039*100000000039)
```

7. Написал тесты для проверки площади в общем виде

```
# tests for rectangle area
def test_rectangle_area(self):
    self.assertEqual(area(3, 4), 12)
    self.assertEqual(area(5, 10), 50)
    self.assertEqual(area(1.5, 2.5), 3.75)
```

8. Написал тесты для нахождения диагонали

```
def test_diagonal(self):
    self.assertAlmostEqual(diagonal(3, 4), 5.0)
    self.assertAlmostEqual(diagonal(5, 12), 13.0)
    self.assertAlmostEqual(diagonal(1.5, 2.5), math.sqrt(1.5**2 + 2.5**2))
```

9. Написал тесты для нахождения вписанной окружности

```
def test_inradius(self):
    self.assertAlmostEqual(inradius(3, 4), 1.2)
    self.assertAlmostEqual(inradius(5, 12), 3.75)
    self.assertAlmostEqual(inradius(1.5, 2.5), 0.9375)
```

10. Написал тесты для нахождения описанной окружности

```
def test_outradius(self):
    self.assertAlmostEqual(outradius(3, 4), 2.5)
    self.assertAlmostEqual(outradius(5, 12), 6.5)
    self.assertAlmostEqual(outradius(1.5, 2.5).math.sqrt(1.5**2 + 2.5**2) / 2)
```

11. Результат работы программы

```
PS C:\Users\User\Desktop\ITMO\ISRPO\Lab_№4> python -m unittest tests.py
.....
-----
Ran 7 tests in 0.001s

OK
PS C:\Users\User\Desktop\ITMO\ISRPO\Lab_№4>
```

12. Данные тесты позволят проверять любые функции вычисляющие параметры произвольно заданного прямоугольника. Они не только корректно работают и сообщают об ошибках, но и сравнивают значения с эталонным учитывая то, что числа в памяти не могут храниться с бесконечной точностью

13. Для того чтобы протестировать значения необходимо написать `python -m unittest <name>.py` И тогда все тесты будут автоматически проведены.

14. Продемонстрирую работу тестов когда функция работает с ошибкой

```
-----
AIL: test_inradius (tests.RectangleTestCase.test_inradius)
-----
Traceback (most recent call last):
  File "C:\Users\User\Desktop\ITMO\ISRPO\Lab №4\tests.py", line 35, in test_inradius
    self.assertEqual(inradius(3, 4), 3.428571, places=6)
    ~~~~~^~~~~~
AssertionError: 3.7142857142857144 != 3.428571 within 6 places (0.2857147142857146 difference)
-----
- 7 tests in 0.001s
```

Видно, что даже небольшие расхождения с ответом не будут пропущены тестирующей программой. Кроме того все ошибки будут подсвечены для того, чтобы пользователь мог сразу понять в каком месте программы он имеет расхождения