

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 (Тесты)

По дисциплине «Инструментальные средства разработки ПО»

Выполнил студент группы №М3102

Знак Владислав Вячеславович

Проверил

Кирилюк Денис Алексеевич



УНИВЕРСИТЕТ ИТМО

Санкт-Петербург

2024

1. Задача: Определение основных целей тестирования и задач, которые необходимо достичь в процессе тестирования

В качестве цели тестирования выбрана отладка функций area и perimeter из модуля geometric_lib

Задачи тестирования: Написание unit-тестов на языке python и проведение unit-тестирования исходного проекта

2. Задача: Обзор функциональности, особенностей и требований к продукту, которые должны быть протестированы

Исходный функционал взят из модуля

https://github.com/Znakous/geometric_lib/tree/new_features_465976 . Функционал исходного модуля описан в README.md

Описание функций	
<code>area()</code>	<code>Perimeter()</code>
Используется для нахождения площади фигур	Используется для нахождения периметра фигур

3. Задача: Определение конкретных функций, модулей или компонентов продукта, которые будут исследованы в тестирование:

В качестве компонентов тестирования выбраны функции area и perimeter

4. Задача: Описание общего подхода к тестированию:

Использован модуль unittest языка python, в частности элементы assertEquals и assertNotEqual для сверки ожидаемых ответов функций с фактическими

```
# tests for rectangle.py
class TestRectangle (unittest.TestCase):
    def test_area_zero(self):
        res = rectangle.area(10, 0)
        self.assertEqual(res, 0)
    def test_area_one(self):
        res = rectangle.area(1, 1)
        self.assertEqual(res, 1)
    def test_area_sum(self):
        a = rectangle.area(10, 15)
        b = rectangle.area(20, 25)
        c = rectangle.area(10+20, 15)
        d = rectangle.area(20, 25-15)
        self.assertEqual(a+b, c+d)
```

Пример исходного кода тестов

5. Задача: Определение условий и критериев, которые должны быть выполнены для успешного завершения тестирования и приемки продукта:

Тестирование считается успешным при успешном завершении 100% тестов (32/32)

6. *Задача: Указание ожидаемых результатов тестирования:*

```
C:\Users\znaks\Desktop\ISRPO\Lab4_ISRPO\geometric_lib>python.exe -m unittest tests.py
.....
-----
Ran 32 tests in 0.001s

OK
```

Все тесты прошли без ошибок, значит тестирование было успешным

Использованные тест-кейсы

Тесты для rectangle.py

Test	Input	Output	Result
test_area_zero	area(10, 0)	0	Passed
test_area_one	area(1, 1)	1	Passed
test_area_sum	area(10, 15)+area(20, 25), area(30, 15)+area(20, 10)	650, 650 (equal)	Passed
test_area_not_eq	area(10, 90), area(10, 9)	900, 90 (not equal)	Passed
test_perimeter_zero	perimeter(0, 0)	0	Passed
test_perimeter_one	perimeter(1, 1)	4	Passed
test_perimeter_sum	perimeter(30, 70), perimeter(10, 30) + perimeter(20, 40)	400, 400 (equal)	Passed
test_perimeter_not_eq	perimeter(10, 20), perimeter(1, 20)	200, 20 (not equal)	Passed

Тесты для square.py

Test	Input	Output	Result
test_area_zero	area(0)	0	Passed
test_area_one	area(1)	1	Passed
test_area_sum	area(30)+area(40), area(50)	2500, 2500 (equal)	Passed

test_area_not_eq	area(9), area(10)	81, 100 (not equal)	Passed
test_perimeter_zero	perimeter(0)	0	Passed
test_perimeter_one	perimeter(1)	4	Passed
test_perimeter_sum	perimeter(30), perimeter(20) + perimeter(10)	120, 120 (equal)	Passed
test_perimeter_not_eq	perimeter(10), perimeter(20)	40, 80 (not equal)	Passed

Тесты для circle.py

Test	Input	Output	Result
test_area_zero	area(0)	0	Passed
test_area_one	area(1)	Pi	Passed
test_area_sum	area(3)+area(4), area(5)	25Pi, 25Pi (equal)	Passed
test_area_not_eq	area(9), area(10)	81Pi, 100Pi (not equal)	Passed
test_perimeter_zero	perimeter(0)	0	Passed
test_perimeter_one	perimeter(1)	2Pi	Passed
test_perimeter_sum	perimeter(30), perimeter(20) + perimeter(10)	60Pi, 60Pi (equal)	Passed
test_perimeter_not_eq	perimeter(10), perimeter(20)	20Pi, 40 (not equal)	Passed

Тесты для triangle.py

Test	Input	Output	Result
test_area_zero	area(10, 0)	0	Passed
test_area_one	area(1, 2)	1	Passed
test_area_sum	area(10, 15)+area(20, 25), area(30, 15)+area(20, 10)	325, 325 (equal)	Passed

test_area_not_eq	area(10, 90), area(10, 9)	450, 45 (not equal)	Passed
test_perimeter_zero	perimeter(0, 0, 0)	0	Passed
test_perimeter_one	perimeter(1, 1, 1)	3	Passed
test_perimeter_sum	perimeter(10, 20, 30), perimeter(3, 7, 16) + perimeter(7, 13, 14)	60, 60 (equal)	Passed
test_perimeter_not_eq	perimeter(10, 20, 5), perimeter(1, 20, 5)	35, 26 (not equal)	Passed

Исходные коды тестов и всего проекта размещены на github

<https://github.com/Znakous/code-styler-for-google-rules>