**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 №M3102

Знак Владислав Вячеславович

Проверил

Кирилюк Денис Алексеевич



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

1. ***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

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описаниеИзображение выглядит как текст, Шрифт, снимок экрана, Графика

Автоматически созданное описание

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

В качестве компонентов тестирования выбраны функции area и perimeter

1. ***Задача: Описание общего подхода к тестированию:***

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

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описаниеПример исходного кода тестов

1. ***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)

1. ***Задача: Указание ожидаемых результатов тестирования:***

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

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>