

**Министр 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

Написание Unit тестов на Python

Выполнила студентка группы № М3119

Великсар Александр Денисович

Подпись:



Проверил:

Повышев Владислав Вячеславович

Санкт-Петербург
2024

План тестирования для функции perimeter

1. Цели и задачи тестирования

Цели:

- Обеспечить корректное функционирование функции perimeter.
- Проверить правильность вычисления периметра для различных значений аргументов.

Задачи:

- Написать и выполнить тесты для функции perimeter.
- Выявить и устранить возможные ошибки в реализации функции.

2. Описание тестируемого продукта

Продукт:

- Функция perimeter(a, b), которая вычисляет периметр прямоугольника по заданным сторонам a и b.

Функциональность:

- Принимает два аргумента: длину и ширину прямоугольника.
- Возвращает периметр прямоугольника, вычисляемый как $(a + b) * 2$.

3. Область тестирования

Функции и компоненты:

- Функция perimeter(a, b).
- Классы тестов RectangleTestCase.

4. Стратегия тестирования

Подходы и методы:

- **Функциональное тестирование:** Проверка корректности вычислений функции perimeter.
- **Методы тестирования:** Юнит-тестирование с использованием фреймворка unittest.

Типы тестирования:

- Тестирование положительных сценариев.
- Тестирование отрицательных сценариев (например, проверка поведения при некорректных значениях).

5. Критерии приемки

Условия успешного завершения тестирования:

- Все тесты проходят без ошибок.
- Функция perimeter корректно обрабатывает различные комбинации аргументов.

6. Ожидаемые результаты

Результаты тестирования:

- Генерация отчетов о прохождении/непрохождении тестов.
- Статус тестирования каждой функции.
- Метрики качества, такие как процент прохождения тестов.

Пример отчета о дефектах:

- При неуспешном прохождении тестов должны быть зафиксированы все выявленные дефекты, включая описание ошибки, шаги для воспроизведения, и ожидаемый результат.

Код тестов по которому был написан данный отчет:

```
test.py > RectangleTestCase > test_square_plus
You, 5 дней назад | 1 author (You)
1 import unittest
2
3 def perimeter(a, b):
4     return (a + b) * 2
5
You, 5 дней назад | 1 author (You)
6 class RectangleTestCase(unittest.TestCase):
7     def test_rectangle_plus(self):
8         res = perimeter(10, 20)
9         self.assertEqual(res, 60)
10
11     def test_square_plus(self):
12         res = perimeter(10, 10)
13         self.assertEqual(res, 40)
```

Ссылка на репозиторий:

https://github.com/AlexVeliksar/isrpo_lab_4-5.git