**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 №M3102

Ерофеев Иван Константинович

Проверил

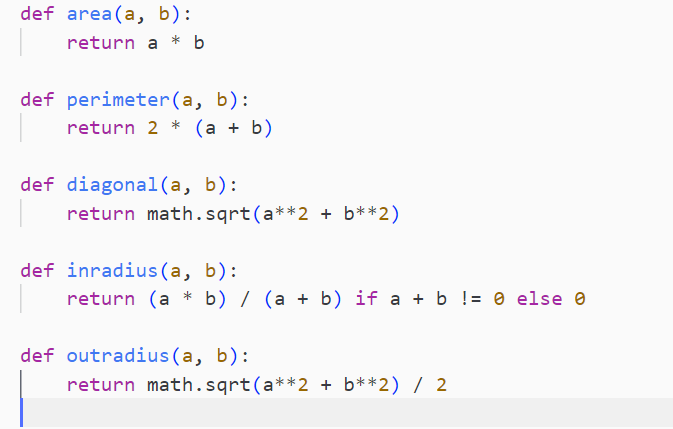
Кирилюк Денис Алексеевич



Санкт-Петербург

2024

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



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

**area(a, b)** — вычисляет площадь прямоугольника.

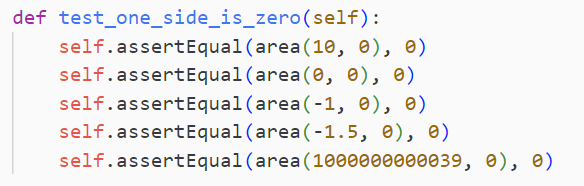
**perimeter(a, b)** — вычисляет периметр прямоугольника.

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

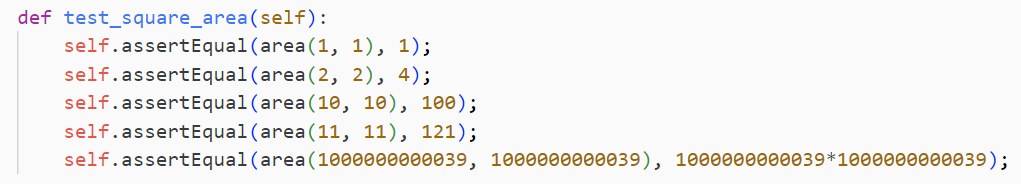
**inradius(a, b)** — вычисляет радиус вписанной окружности.

**outradius(a, b)** — вычисляет радиус описанной окружности

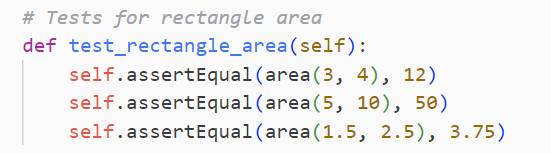
1. Тестируются в том числе пограничные и некорректные значения, например когда сторона отрицательная или равна нулю. Также тестируются частные случаи, например когда прямоугольник является квадратом
2. Написал тесты для проверки случая, когда одна сторона равна нулю



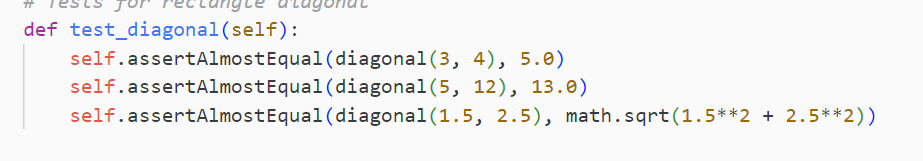
1. Написал тесты для проверки площади, когда прямоугольник является квадратом



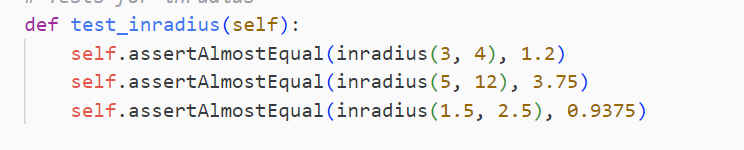
1. Написал тесты для проверки площади в общем виде



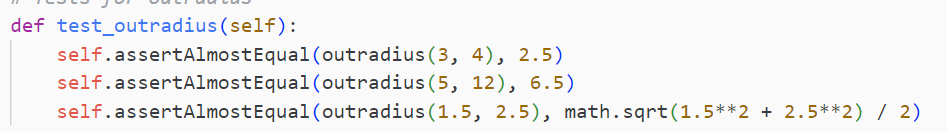
1. Написал тесты для нахождения диагонали



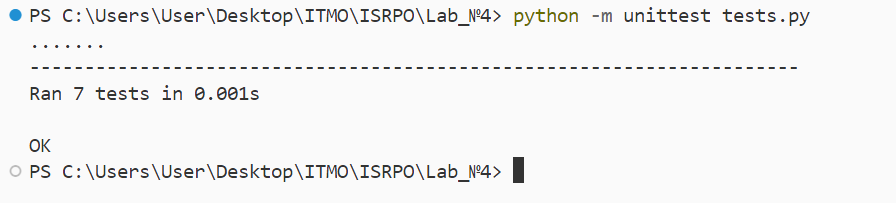
1. Написал тесты для нахождения вписанной окружности

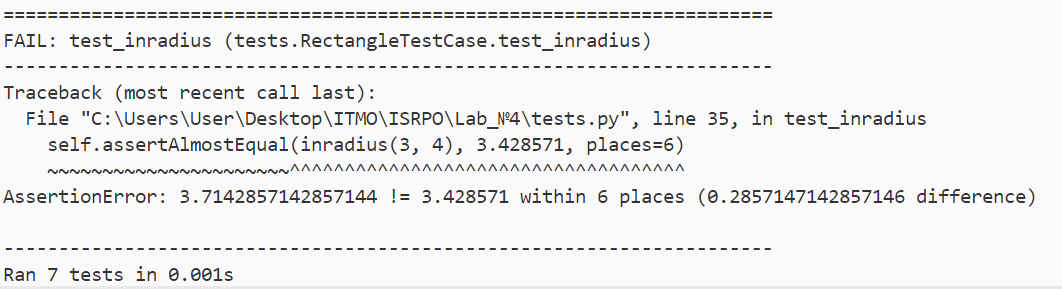


1. Написал тесты для нахождения описанной окружности



1. Результат работы программы



1. 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 хранится с бесконечной точностью
2. Для того чтобы протестировать значения необходимо написать python -m unittest <name>.py И тогда все тесты будут автоматически проведены.
3. Продемонстрирую работу тестов когда функция работает с ошибкой

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