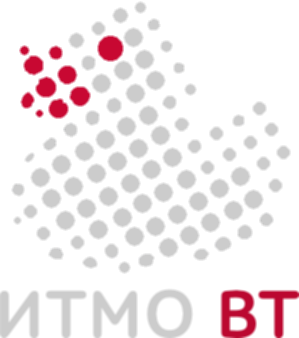
Санкт-Петербургский Национальный Исследовательский Университет Информационных Технологий, Механики и Оптики

ПИиКТ

Лабораторная работа №2

по дисциплине

«Тестирование программного обеспечения»



Выполнил: Студент группы P33112

Почикалин Владислав

Преподаватель: Яркеев Александр Сергеевич

Санкт-Петербург

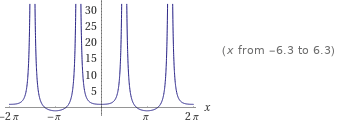
2021 г.

**Вариант 33166:**

**Изображение выглядит как текст

Автоматически созданное описание**

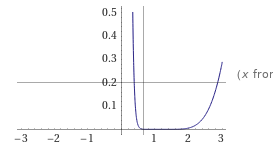
**Функция №1:**



Данная функция является периодичной с периодом 2PI.

Для анализа эквивалентности были выбраны 2 промежутка, где функция ведет себе одинаково – от -PI до -PI/2 – функция возрастает, от -PI/2 до 0 – функция убывает, причем в точке -PI/2 – функция не существует (точка разрыва). В точках -PI,0 – функция принимает однозначные значения (-1, 1 соответственно). Модульные тесты были разработаны с учетом этих особенностей.

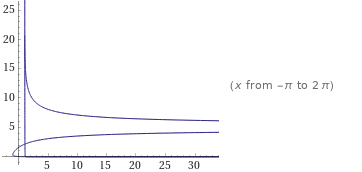
**Функция №2:**

****

Данная функция имеет 2 промежутка эквивалентности –

от 0 до 1, где значение функции убывает, от 1 до +infinity. Точка 1 является экстремумом функции (точка минимума). В ней функция достигает значения 0. Модульные тесты были написаны с учетом этой особенности.

**Система уравнений:**

****

Данная функция имеет решения, однако при входящим условии x>0 и x<=0 данная система никогда не будет иметь решения. Таким образом, в модульных тестах необходимо доказать, что решений нет.

**Интеграционное тестирование**

В качестве стратегии интеграции была выбрана нисходящий метод интеграций. С учетом этого, отпала надобность в драйверах, однако необходимость создания заглушек осталось. Заглушки были сделаны с помощью Mockito. Во всех тестах я придерживался такого алгоритма :

1. Сначала делаем заглушку на весь метод подсчета функции
2. Потом углубляемся внутрь метода, и заминаем все его составные модули заглушками.
3. Постепенно убираем заглушки, заменяя их реальными классами
4. В итоге получаем тест, в котором все заглушки отсутствуют.

**UML**

**Изображение выглядит как текст

Автоматически созданное описание**

**Вывод**

В ходе выполнения работы был опробован способ нисходящего интеграционного тестирования, а также библиотека, ускоряющая разработку заглушек, Mockito.