

Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное
учреждение высшего образования

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Факультет программной инженерии и компьютерной техники

Дисциплина: Тестирование программного обеспечения

Отчёт по лабораторной работе №2

Вариант: **9082**

Выполнил: Нгуен Нгок Дык

Студент группы: Р33302

Преподаватель: Харитонов А.Е

Санкт-Петербург

2023 г.

1. Описание

Вариант **9082**

Провести интеграционное тестирование программы, осуществляющей вычисление системы функций (в соответствии с вариантом).

$$x \leq 0 : (((((\cos(x) * \sec(x)) + \sin(x)) * \cos(x)) / \tan(x)) ^ 3)$$

$$x > 0 : (((((\ln(x) + \log_2(x)) ^ 3) ^ 2) / (((\log_3(x) ^ 2) / (\log_{10}(x) + \log_3(x))) ^ 3)) + \log_2(x))$$

1.1 Правила выполнения работы:

1. Все составляющие систему функции (как тригонометрические, так и логарифмические) должны быть выражены через базовые (тригонометрическая зависит от варианта; логарифмическая - натуральный логарифм).
2. Структура приложения, тестируемого в рамках лабораторной работы, должна выглядеть следующим образом (пример приведён для базовой тригонометрической функции $\sin(x)$):
3. Обе "базовые" функции (в примере выше - $\sin(x)$ и $\ln(x)$) должны быть реализованы при помощи разложения в ряд с задаваемой погрешностью. Использовать тригонометрические / логарифмические преобразования для упрощения функций ЗАПРЕЩЕНО.
4. Для КАЖДОГО модуля должны быть реализованы табличные заглушки. При этом, необходимо найти область допустимых значений функций, и, при необходимости, определить взаимозависимые точки в модулях.
5. Разработанное приложение должно позволять выводить значения, выдаваемое любым модулем системы, в csv файл вида «X, Результаты модуля (X)», позволяющее произвольно менять шаг наращивания X. Разделитель в файле csv можно использовать произвольный.

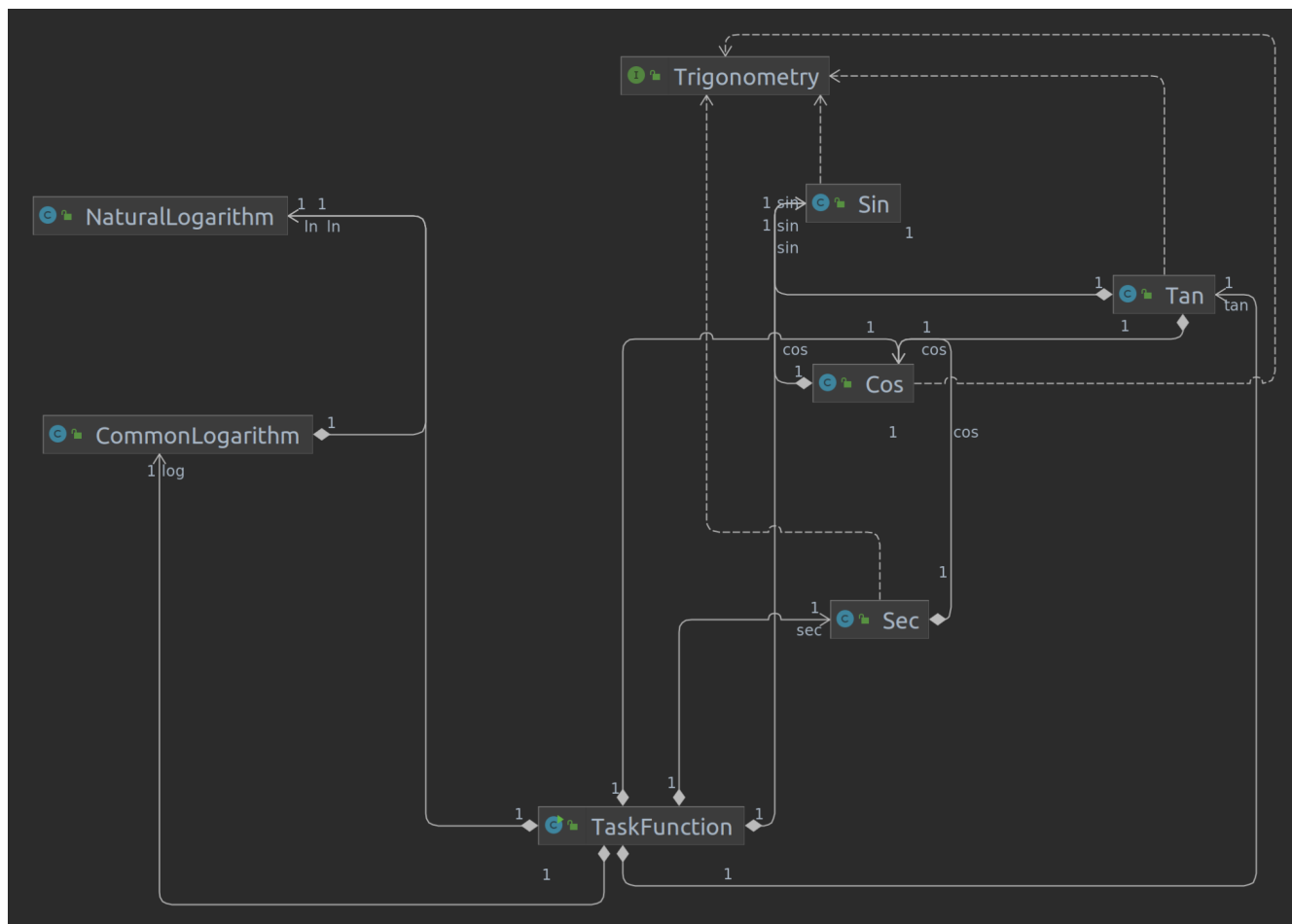
1.2 Порядок выполнения работы:

1. Разработать приложение, руководствуясь приведёнными выше правилами.
2. С помощью JUNIT4 разработать тестовое покрытие системы функций, проведя анализ эквивалентности и учитывая особенности системы функций. Для анализа особенностей системы функций и составляющих ее частей можно использовать сайт <https://www.wolframalpha.com/>.
3. Собрать приложение, состоящее из заглушек. Провести интеграцию приложения по 1 модулю, с обоснованием стратегии интеграции, проведением интеграционных тестов и контролем тестового покрытия системы функций.

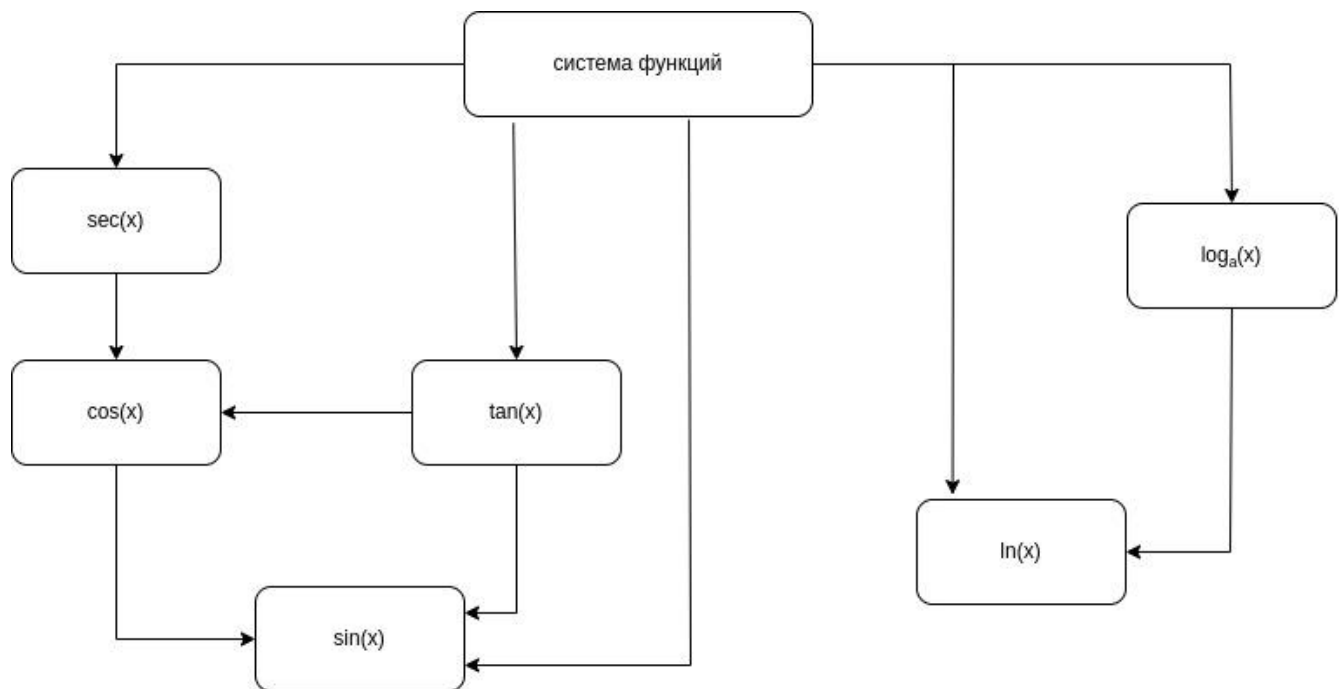
2. Выполнение

Исходный код: https://github.com/ndwannafly/Integration_Testing

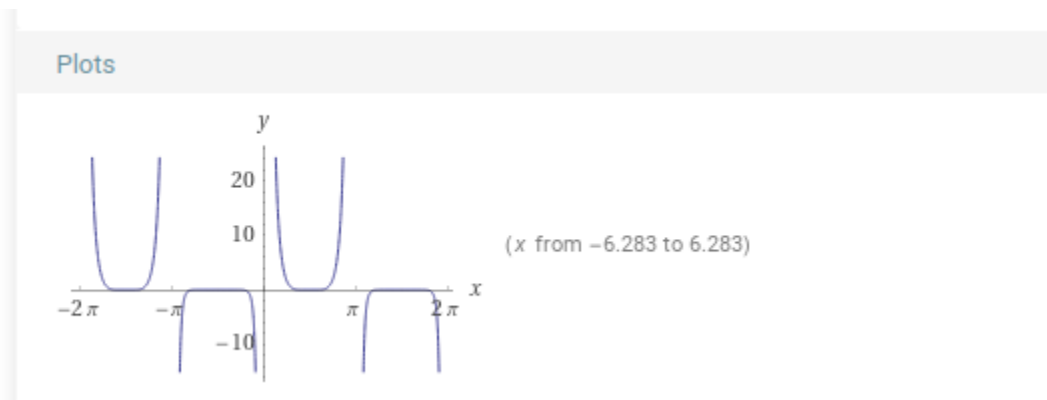
UML - диаграмма классов



Описание тестового покрытия

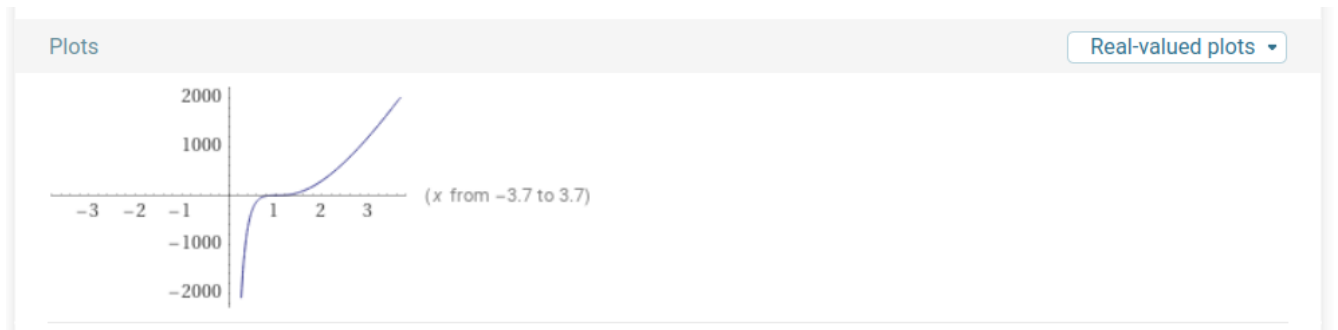


X <= 0:



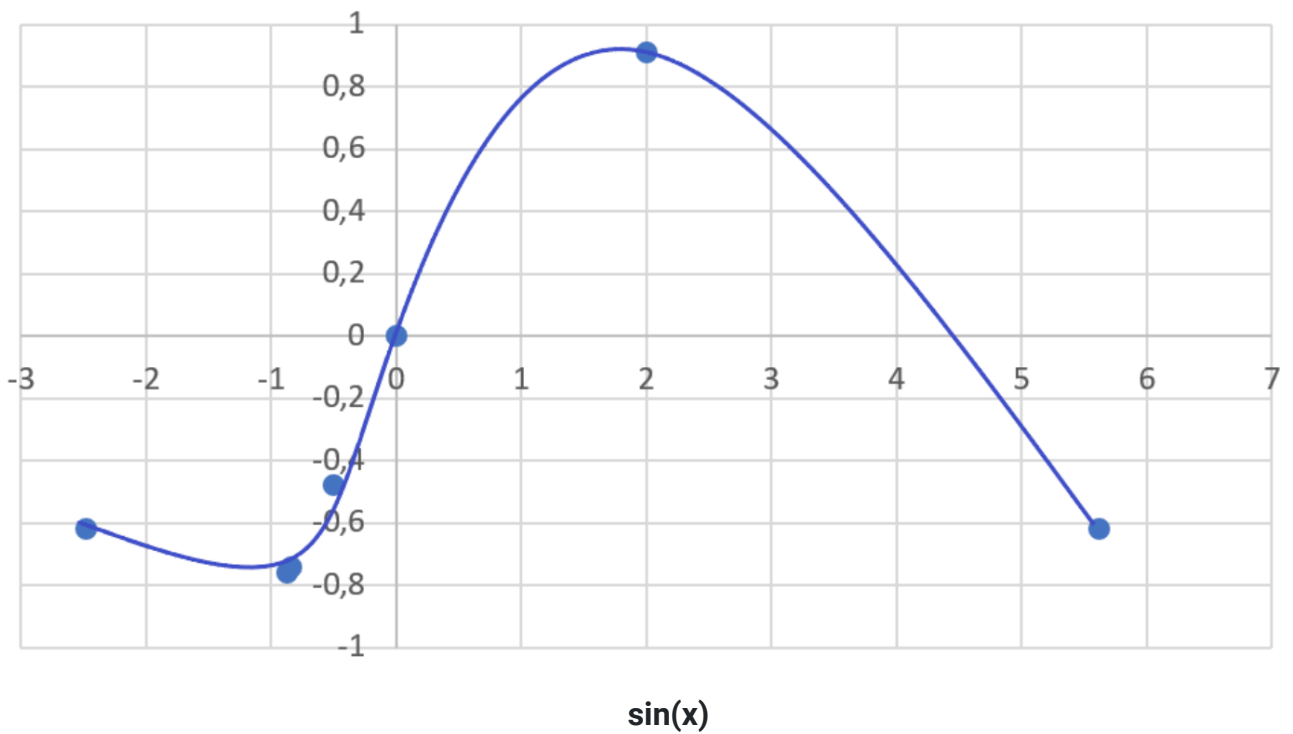
функция ведёт себя как периодическая, есть точки разрыва, рядом с которыми они стремятся к бесконечности. Мы брали точки рядом с точкой, где меняется знак производной. И рядом с точками разрыва функции, в которых функция стремится к бесконечности

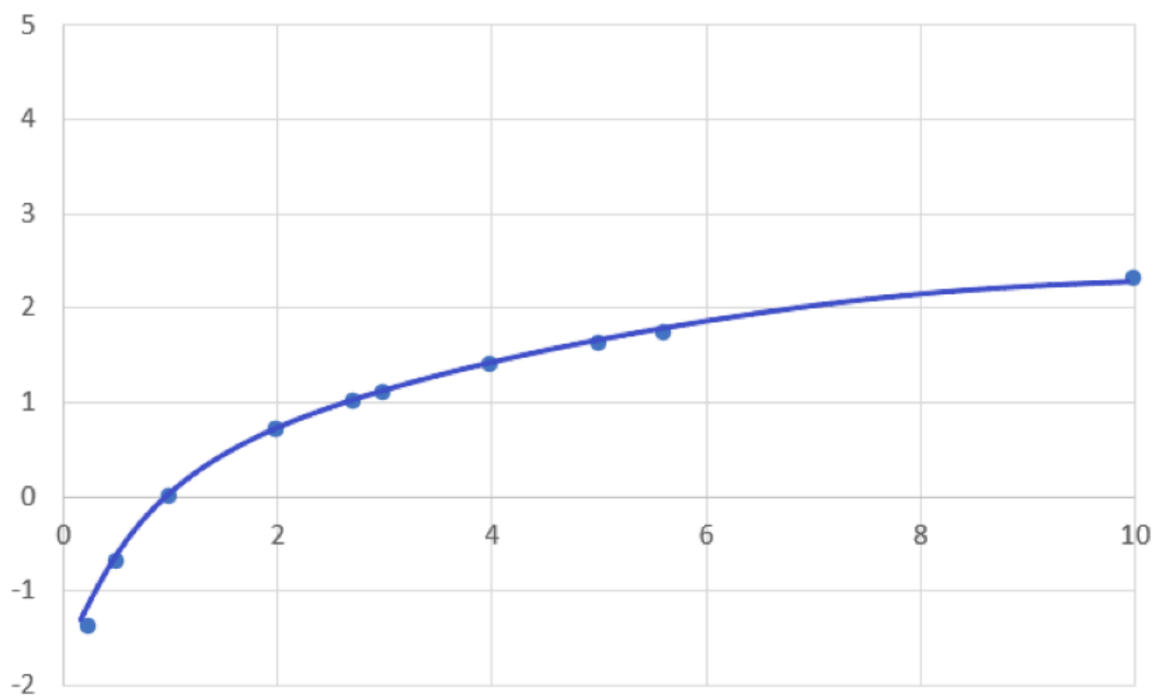
$x > 0$:



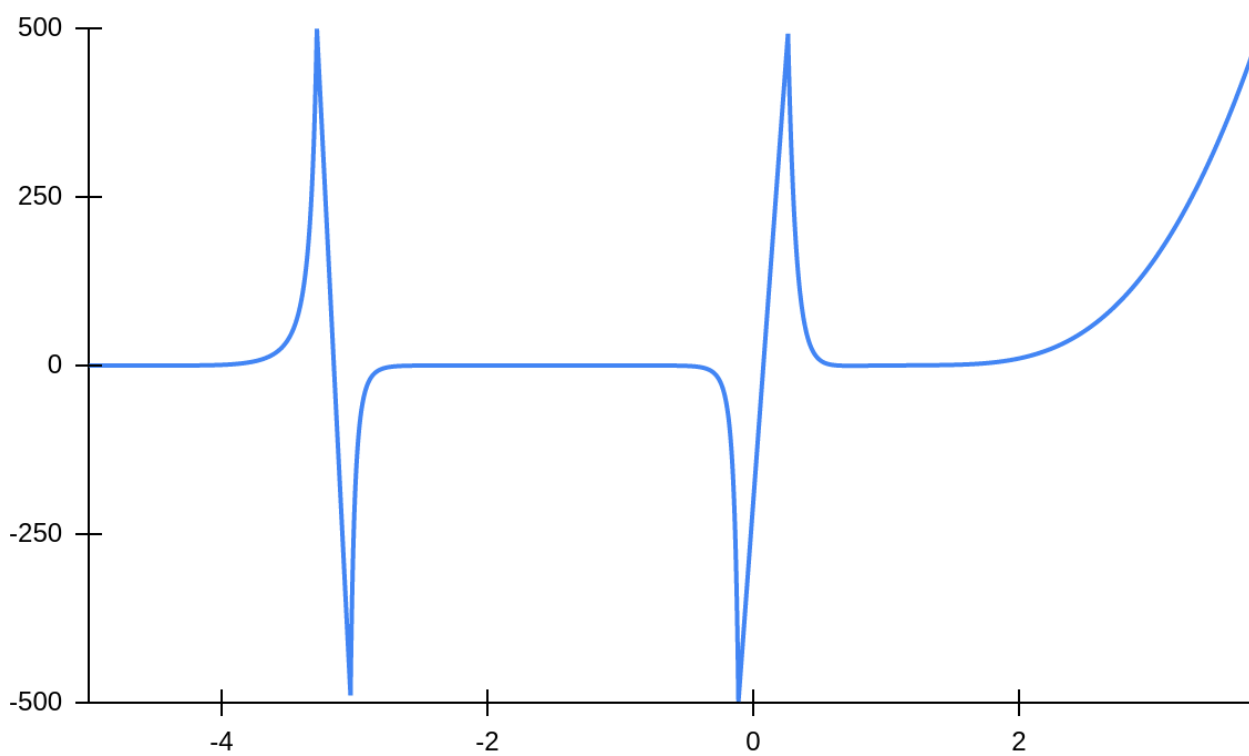
функция уже имеет только одну точку перерыва - в точке ноль. В остальных точках она определена. Мы взяли точки рядом с точкой изменения знака производной

Графики, построенные csv-выгрузкам





$\ln(x)$



$f(x)$

Вывод:

В ходе выполнения лабораторной работы было проведено интеграционное тестирование написанной нами программы, реализованной согласно варианту. Также мы изучили работу классов заглушек на примере библиотеки Mockito и применили ее в ходе тестирования.