

Лабораторная работа #2

Тестирование программного обеспечения

Группа: РЗ3101

Студенты: Бухаров Д. П., Никашкин А. В.

Преподаватель: Машина Е. А.

Вариант №237543

Задание:

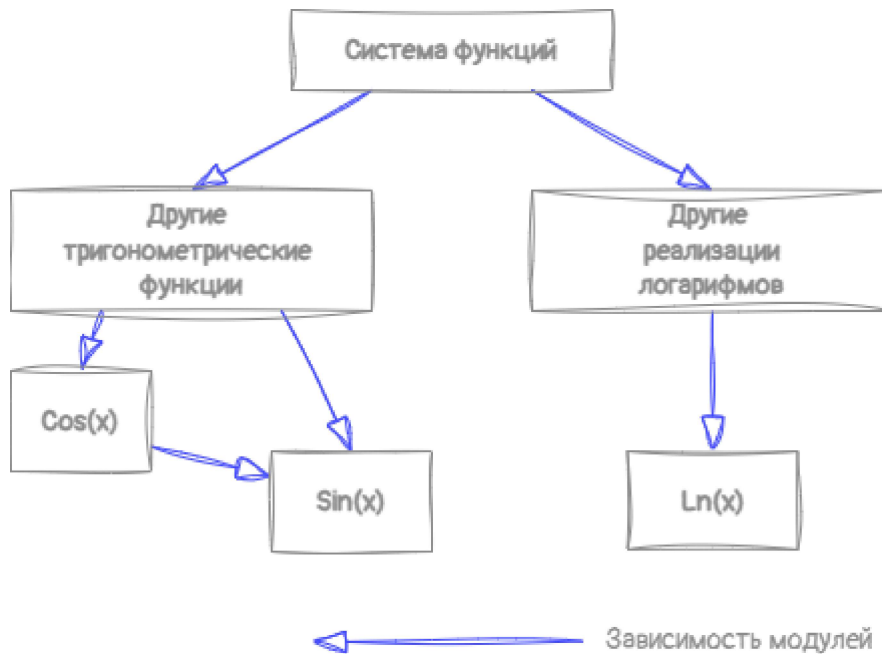
Провести интеграционное тестирование программы, осуществляющей вычисление системы функций (в соответствии с вариантом).

$x \leq 0 : (((((\sec(x) - \sec(x)) + (\sec(x) * \sin(x))) * \cos(x)) - (\sin(x) * \csc(x))) / \tan(x))$

$x > 0 : (((((\log_2(x) - \log_2(x)) + (\log_2(x) * \log_{10}(x))) * \log_3(x)) - (\log_3(x) * \ln(x))) / \log_3(x))$

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.

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.

Отчёт по работе должен содержать

1. Текст задания, систему функций.
2. UML-диаграмму классов разработанного приложения.
3. Описание тестового покрытия с обоснованием его выбора.

4. Графики, построенные csv-выгрузкам, полученным в процессе интеграции приложения.
5. Выводы по работе.

Выполнение:

Было разработано приложения для подсчета заданной функции, реализованы тригонометрические и логарифмические функции, которые считаются через разложение в ряд и зависят друг от друга. Созданы вспомогательные классы, такие как `MathUtil` и `CsvUtil`. Написаны интеграционные тесты, использующие заглушки ("моки").

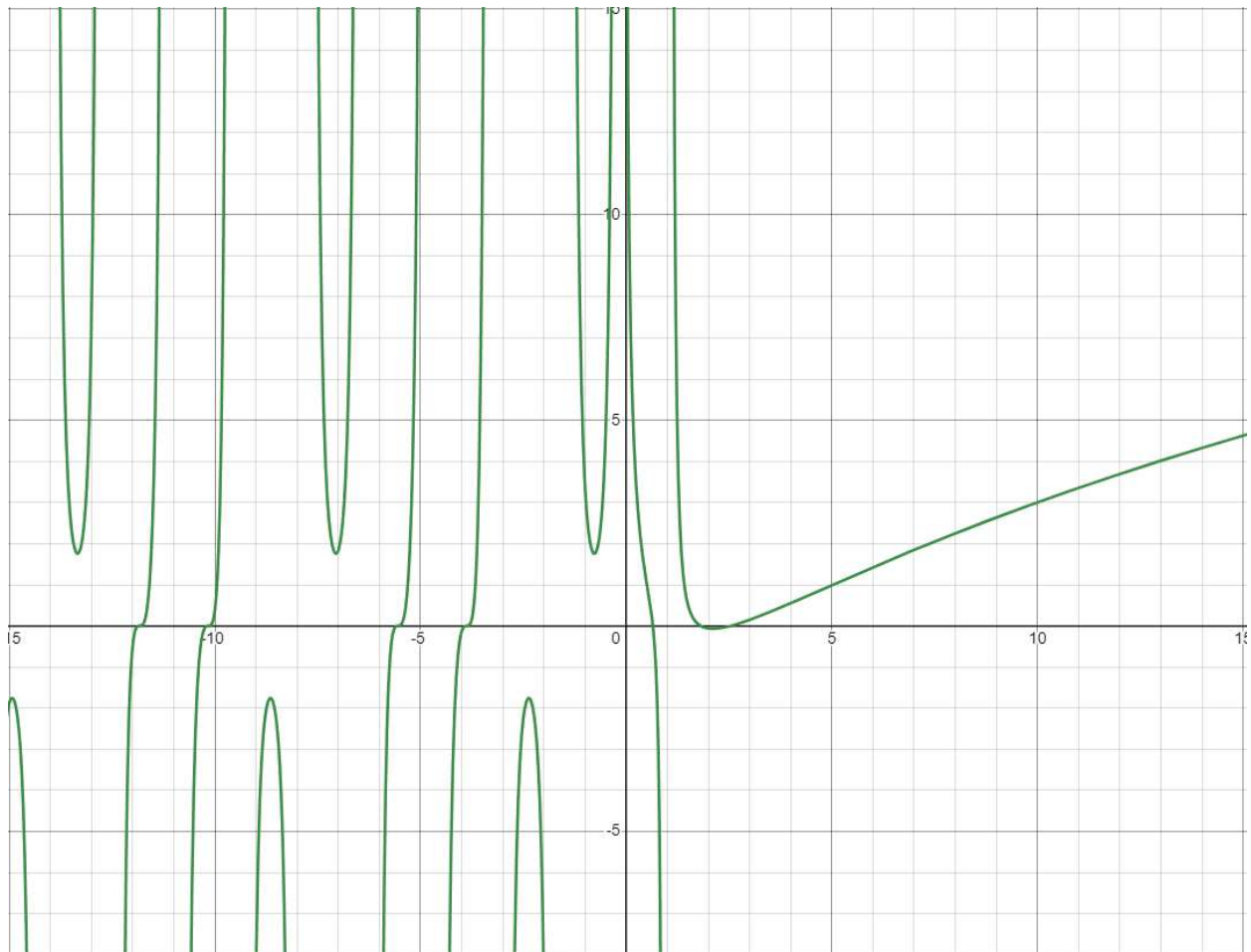
Для тестирования были использованные табличные значения для проверки тригонометрических функций и пограничные значения области определения для логарифмических функций. Кроме того, использовались нечисловые значения, к примеру $+\infty$ и $-\infty$. Также анализировались критические точки на графике и значения подбирались, опираясь на них.

Реализована выгрузка значений в csv файлы.

График функции

$x \leq 0 : (((\sec(x) - \sec(x)) + (\sec(x) * \sin(x))) * \cos(x)) - (\sin(x) * \csc(x))) / \tan(x)$

$x > 0 : (((\log_2(x) - \log_2(x)) + (\log_2(x) * \log_{10}(x))) * \log_3(x)) - (\log_3(x) * \ln(x))) / \log_3(x)$



UML диаграмма классов

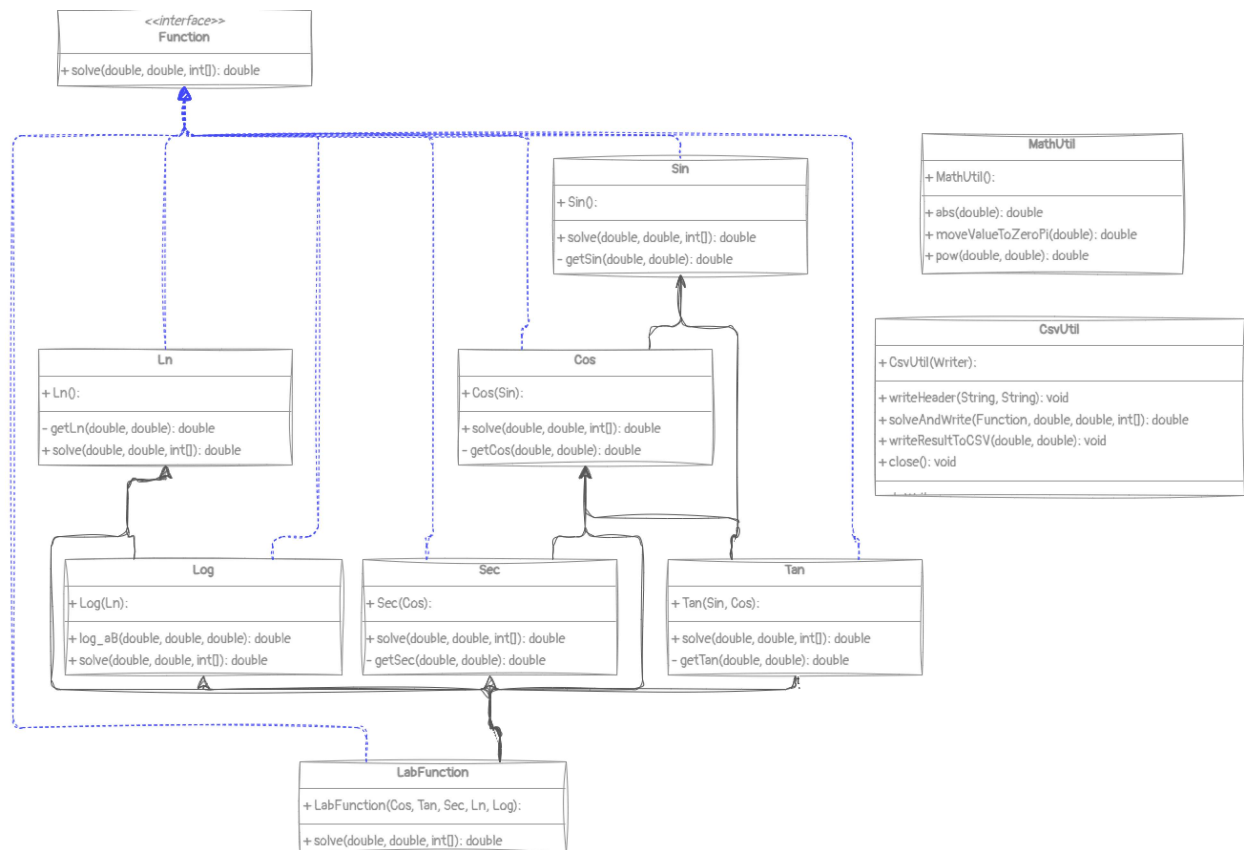
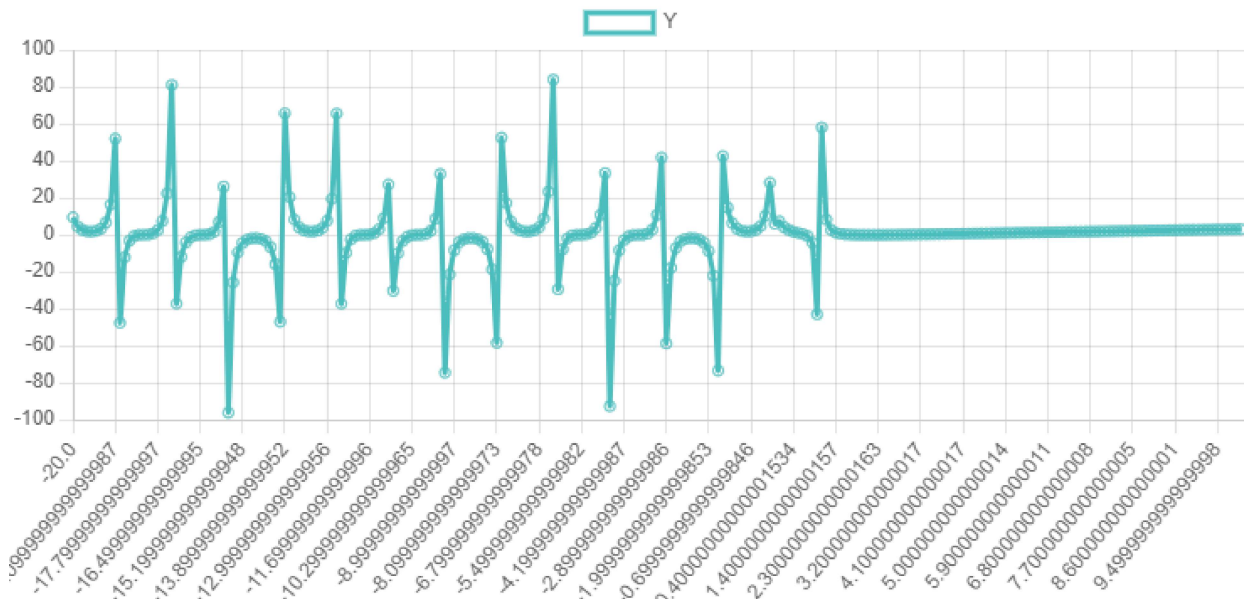


График по CSV выгрузкам



Исходный код

Тригонометрические функции

Логарифмические функции

Функция из варианта

Утилиты

Тесты

Входные csv файлы

Выходные csv файлы

Вывод:

В ходе выполнения данной лабораторной работы была изучена работа заглушек, на примере библиотеки *Mockito*, освоено интеграционное тестирование. Уверены, полученная информация пригодится нам в будущем.