ИТМО

Лабораторная работа №2

по дисциплине: Тестирование программного обеспечения

Вариант: 75123

Выполнил: Неграш Андрей, P33301

Преподаватель: Гаврилов Антон Валерьевич

Санкт-Петербург, 2023

Задание

Лабораторная работа #2

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

Please, enter your variant: 75123

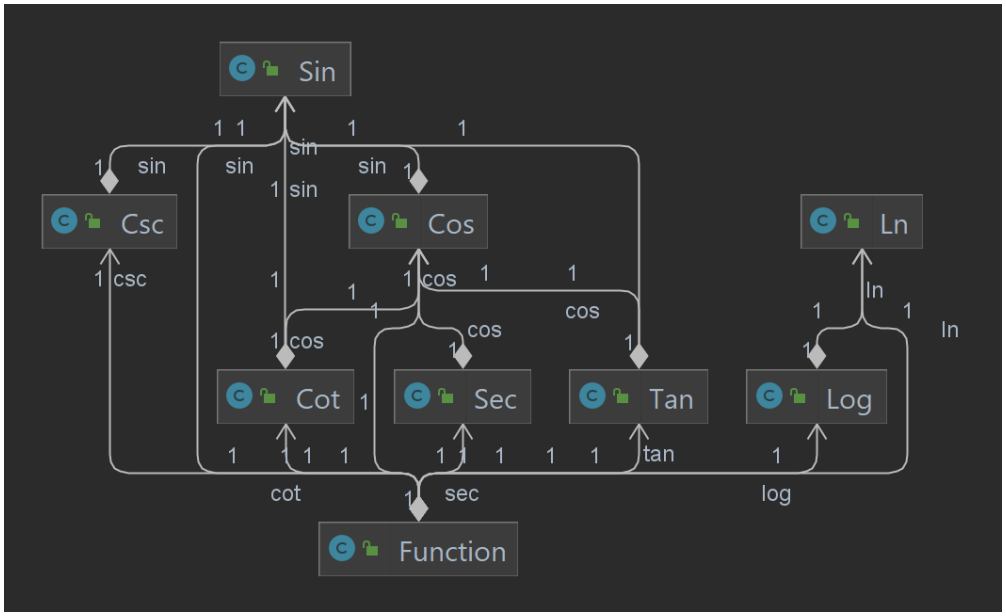
$$\begin{cases} \left(\left(\left(\left(\left(\frac{\sec(x)}{\tan(x)}\right) - (\cos(x) + \cos(x))\right) + \cot(x)\right)^2\right) - (\sin(x) + \csc(x))\right) \cdot \cot(x) & \text{if } x \le 0 \\ \left(\left(\frac{((\log_5(x) \cdot \log_3(x)) \cdot \log_3(x)) - \log_5(x)}{(\log_2(x) \cdot (\ln(x) + \log_{10}(x))) + \log_2(x)}\right) \cdot \ln(x)\right) & \text{if } x > 0 \end{cases}$$

Правила выполнения работы:

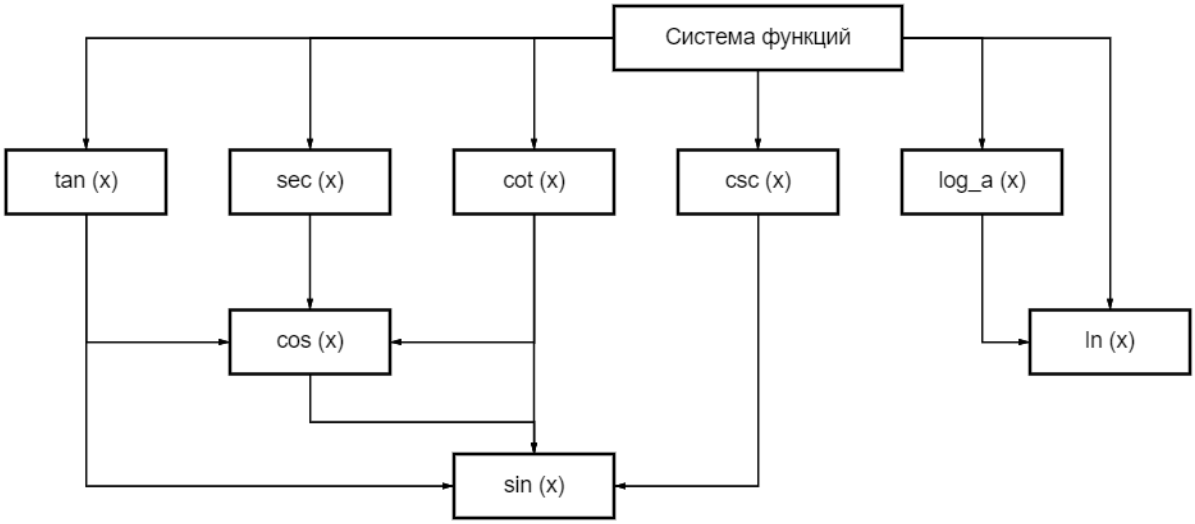
1. Все составляющие систему функции (как тригонометрические, так и логарифмические) должны быть выражены через базовые (тригонометрическая зависит от варианта; логарифмическая - натуральный логарифм).
2. Структура приложения, тестируемого в рамках лабораторной работы, должна выглядеть следующим образом (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.

UML-диаграмма



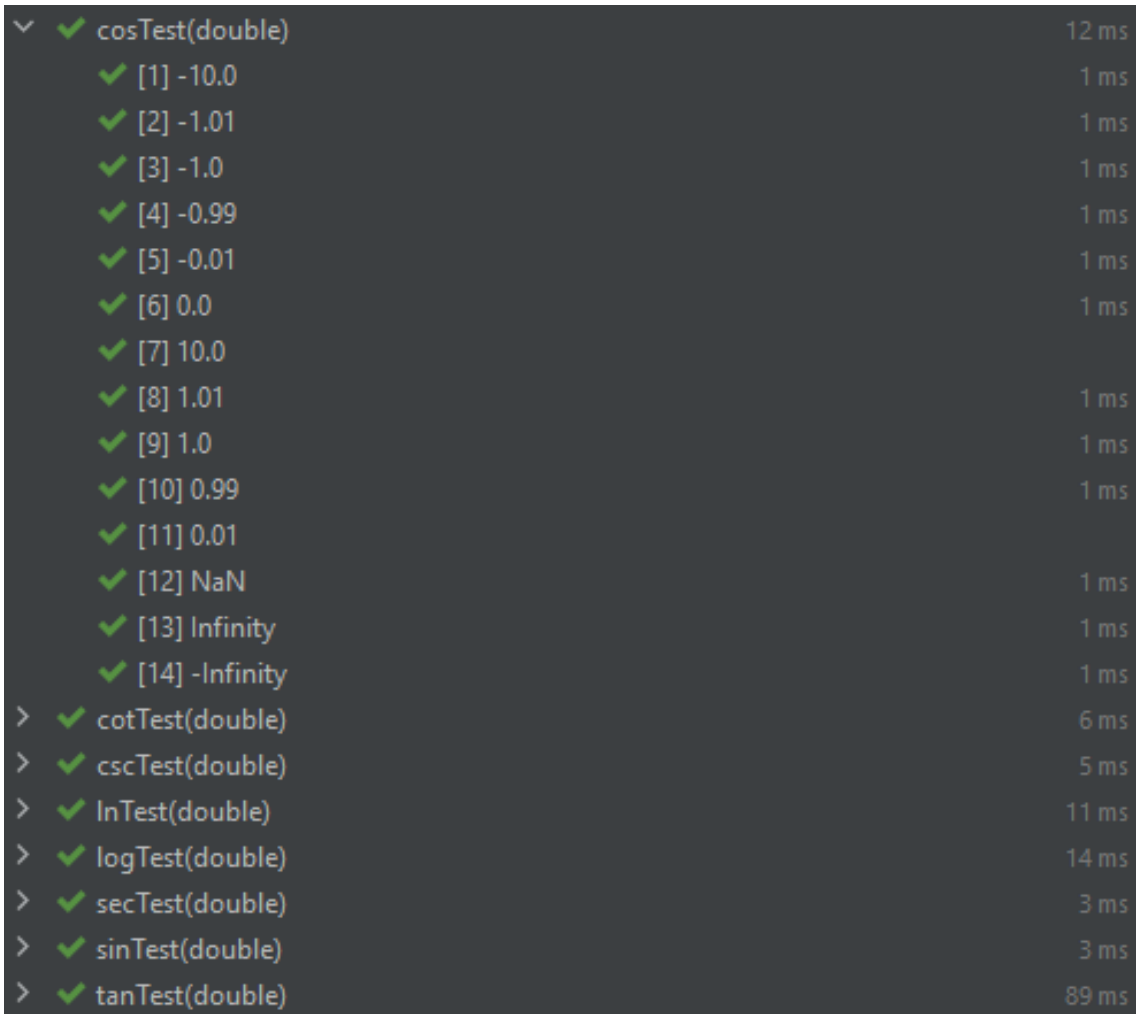
Описание тестового покрытия



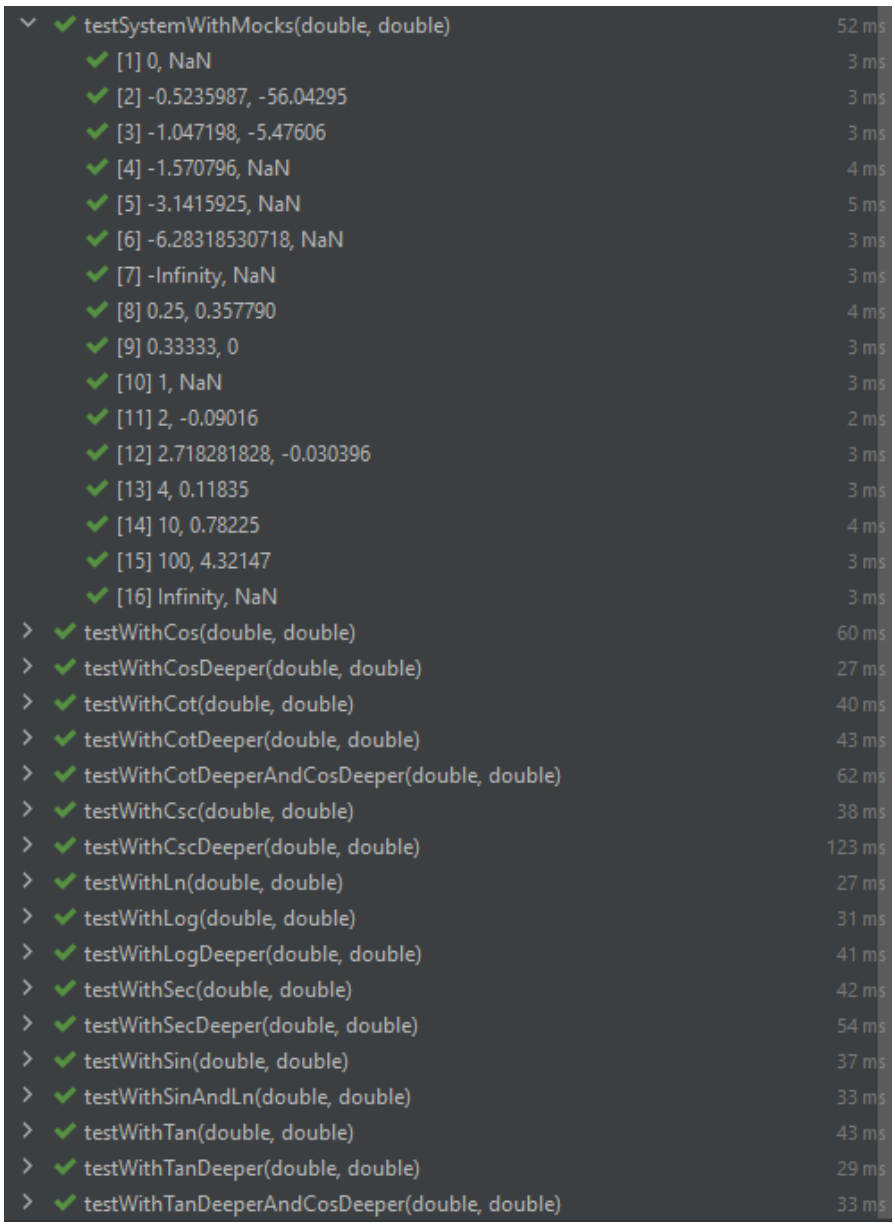
Интеграция приложения проводилась сверху вниз, так как этот метод позволяет использовать заглушки, которые гораздо проще реализовать и в целом данный вид интеграции более прост в реализации.

Тестовое покрытие состоит из 2 частей:

1) Тесты для базовых функций и сравнение с эталоном из библиотеки Math

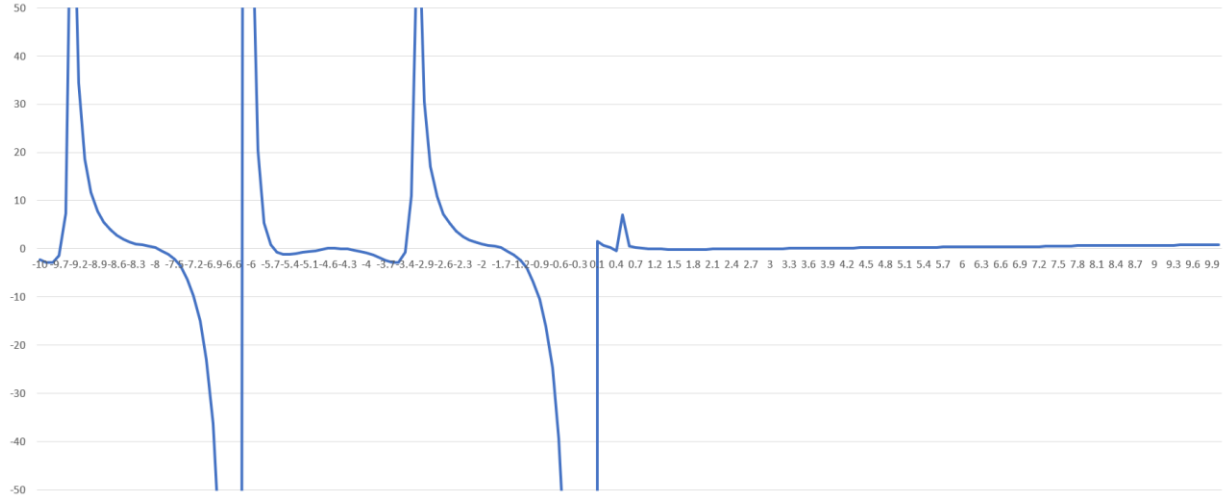


2) Тесты для системы функций согласно заданию

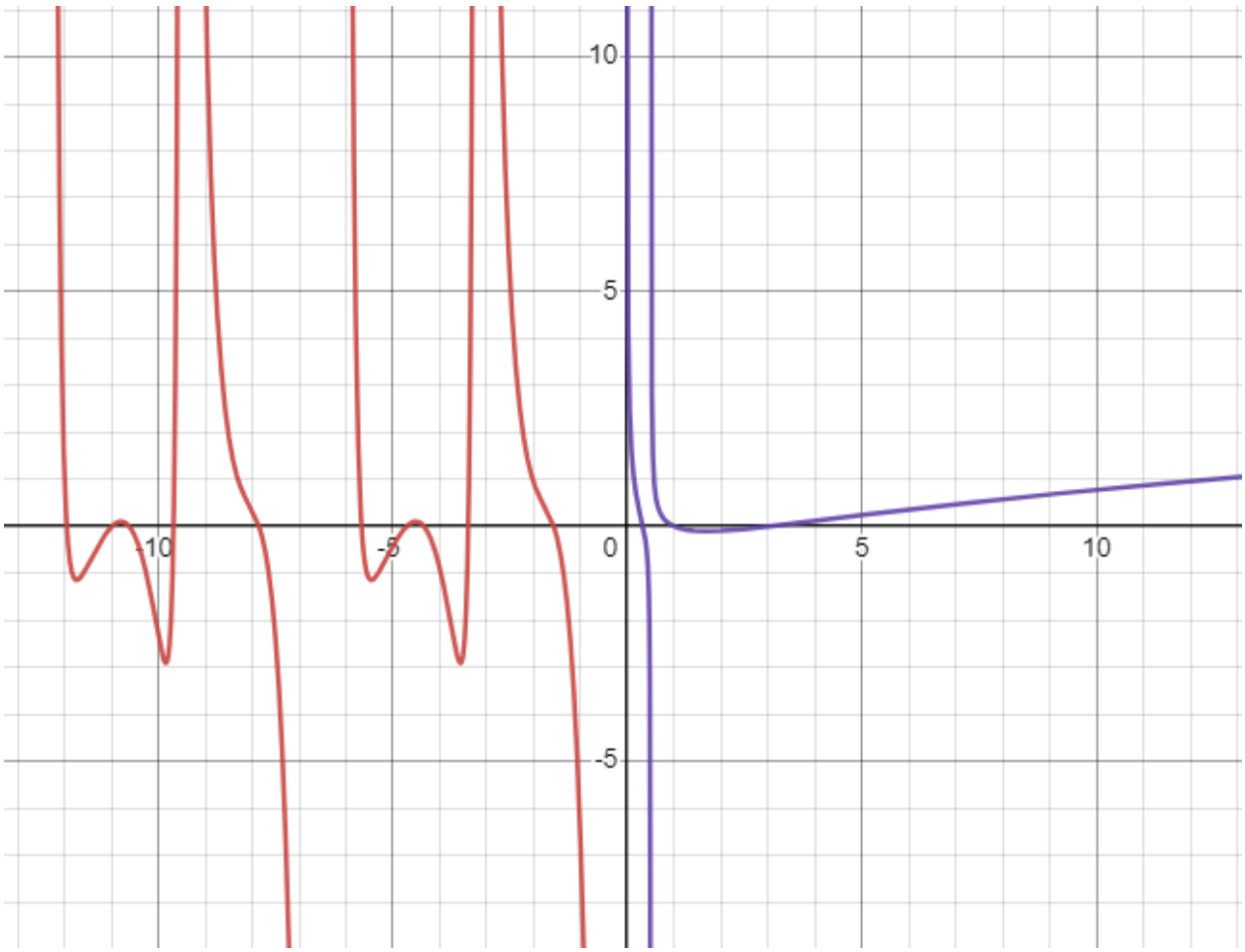


Графики

График, полученный через csv-выгрузки



Эталонный график функции (https://www.desmos.com/calculator/b2umm0us86):



Исходный код

Можно просмотреть на моём GitHub: https://github.com/ANegrash/TPO_lab2

Вывод

Итак, в процессе выполнения данной лабораторной работы мною было выполнено интеграционное тестирование для разработанной программы, а также изучена работа классов заглушек на примере Mockito.