

Федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский университет
ИТМО»

Факультет программной инженерии и компьютерной техники

Лабораторная работа №2

по “Тестирование программного обеспечения”

Вариант №68733

Выполнил:
Студент группы Р3318
Горло Евгений Николаевич

Преподаватель:
Кулинич Ярослав Вадимович

г. Санкт-Петербург

2025 год

Задание

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

Введите вариант:

$$\begin{cases} \left(\left(\left(\left(\sec(x) - \sec(x) \right) + \cot(x) \right) - \cot(x) \right) - \left(\left(\frac{\cos(x) \cdot \sin(x)}{\sin(x)} \right) + \left(\cot(x) \cdot \sec(x)^2 \right) \right) \right) + \left(\left(\left(\frac{\csc(x)}{\cot(x)} \right) \cdot \left(\csc(x)^3 \right) \right)^3 \right) & \text{if } x \leq 0 \\ \left(\left(\left(\left(\frac{\ln(x) \cdot \log_2(x)^2}{\left(\frac{\log_3(x)}{\ln(x)} \right) - \log_5(x)} \right)^3 \right)^2 \right) \right) & \text{if } x > 0 \end{cases}$$

$x \leq 0$: (((((sec(x) - sec(x)) + cot(x)) - cot(x)) - (((cos(x) * sin(x)) / sin(x)) + ((cot(x) * sec(x)) ^ 2))) + (((csc(x) / cot(x)) * (csc(x) ^ 3)) ^ 3))

$x > 0$: (((((ln(x) * log_2(x) ^ 2) / ((log_3(x) / ln(x)) - log_5(x))) ^ 3) ^ 2)

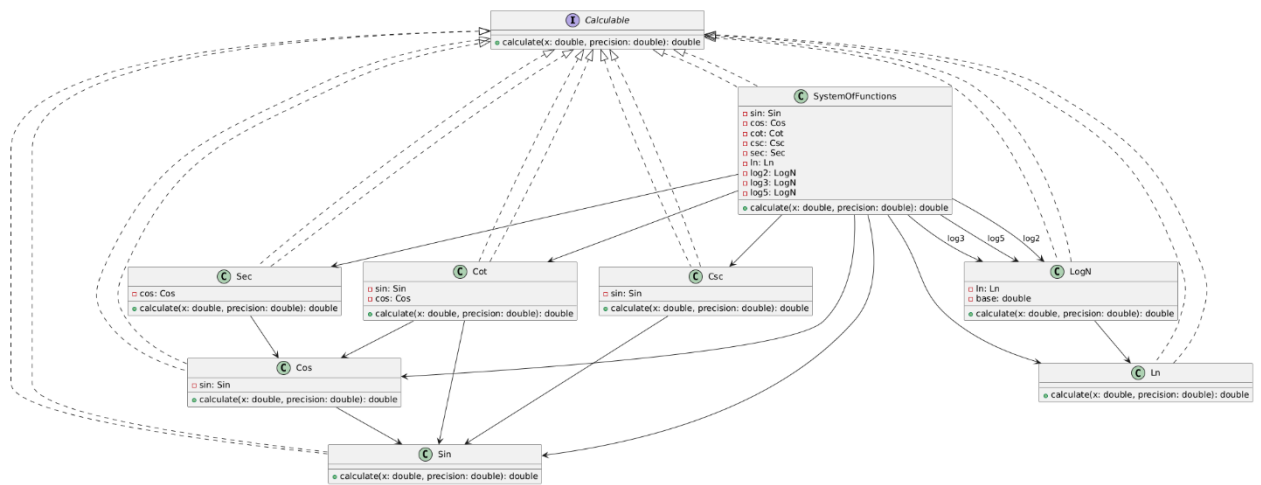
Правила выполнения работы:

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.

UML диаграмма классов приложения



Для тестирования выбраны следующие значения:

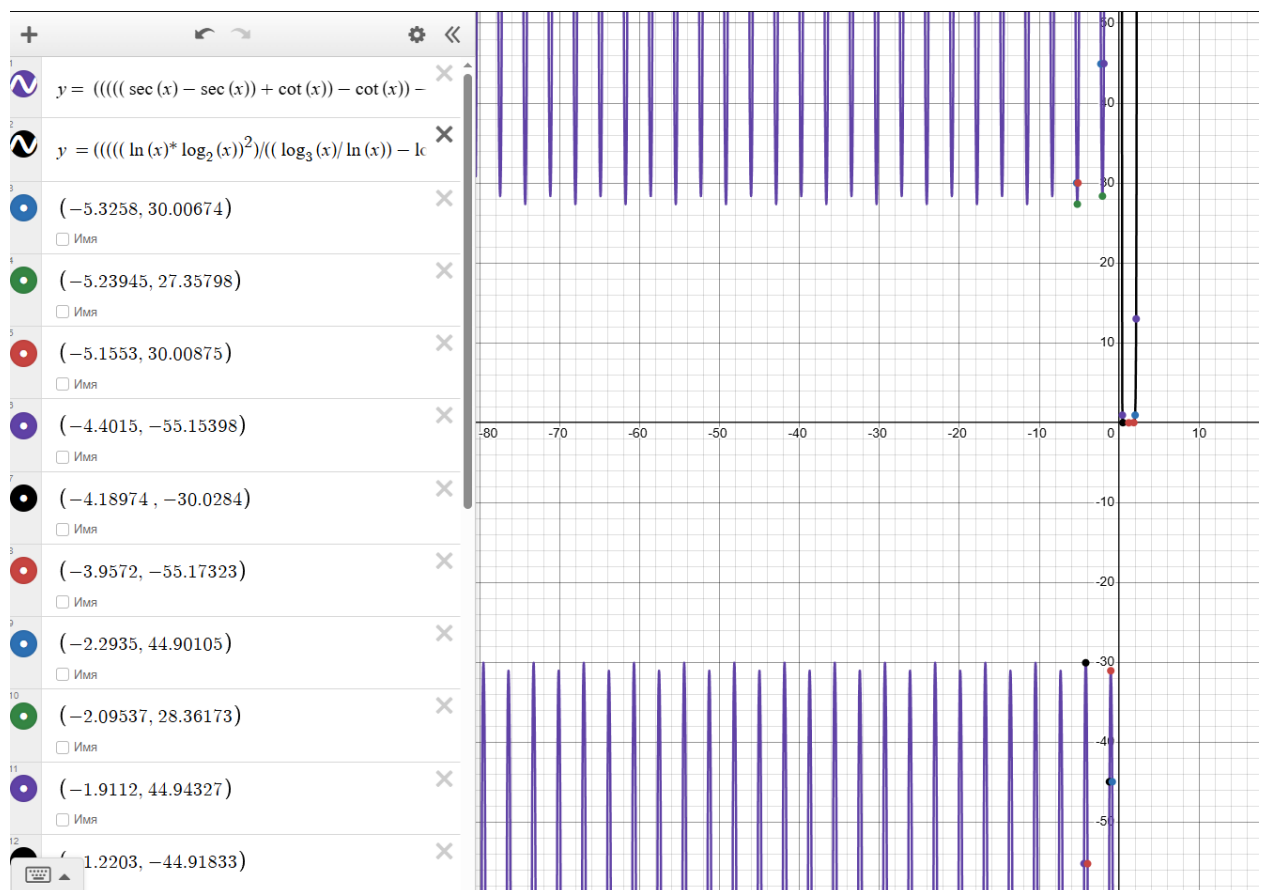
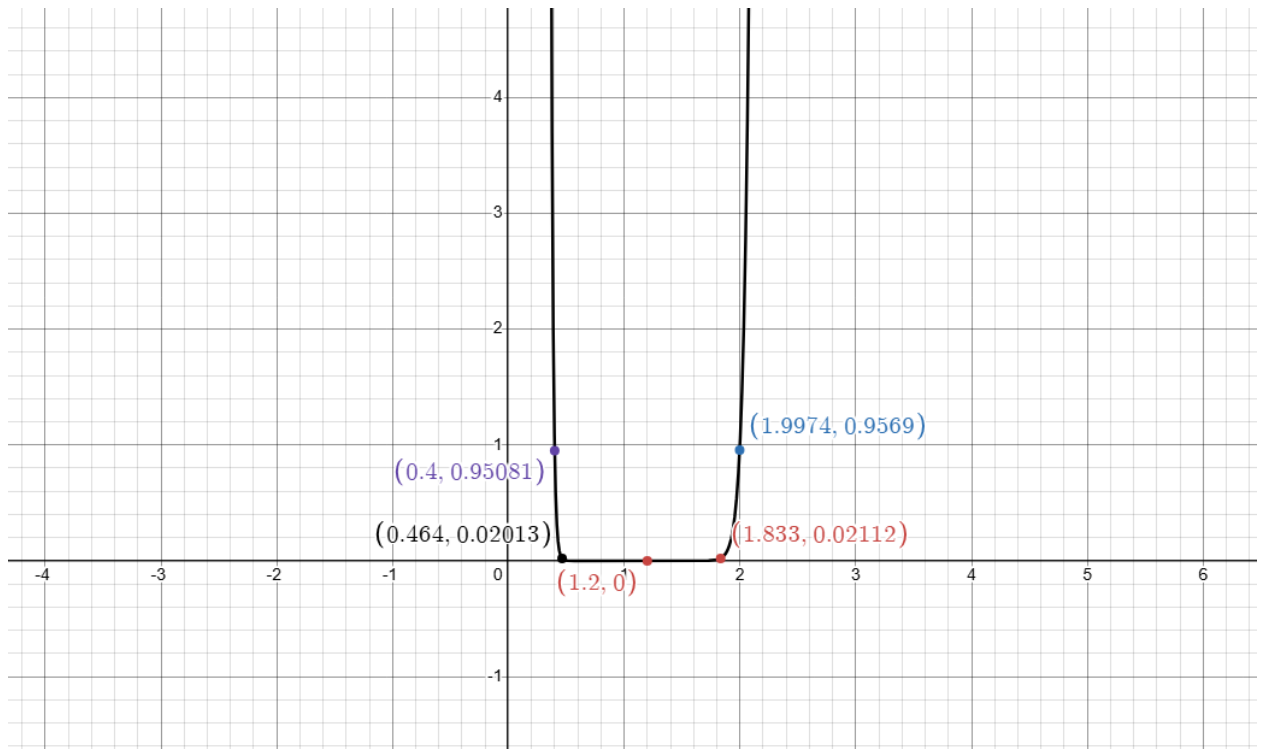
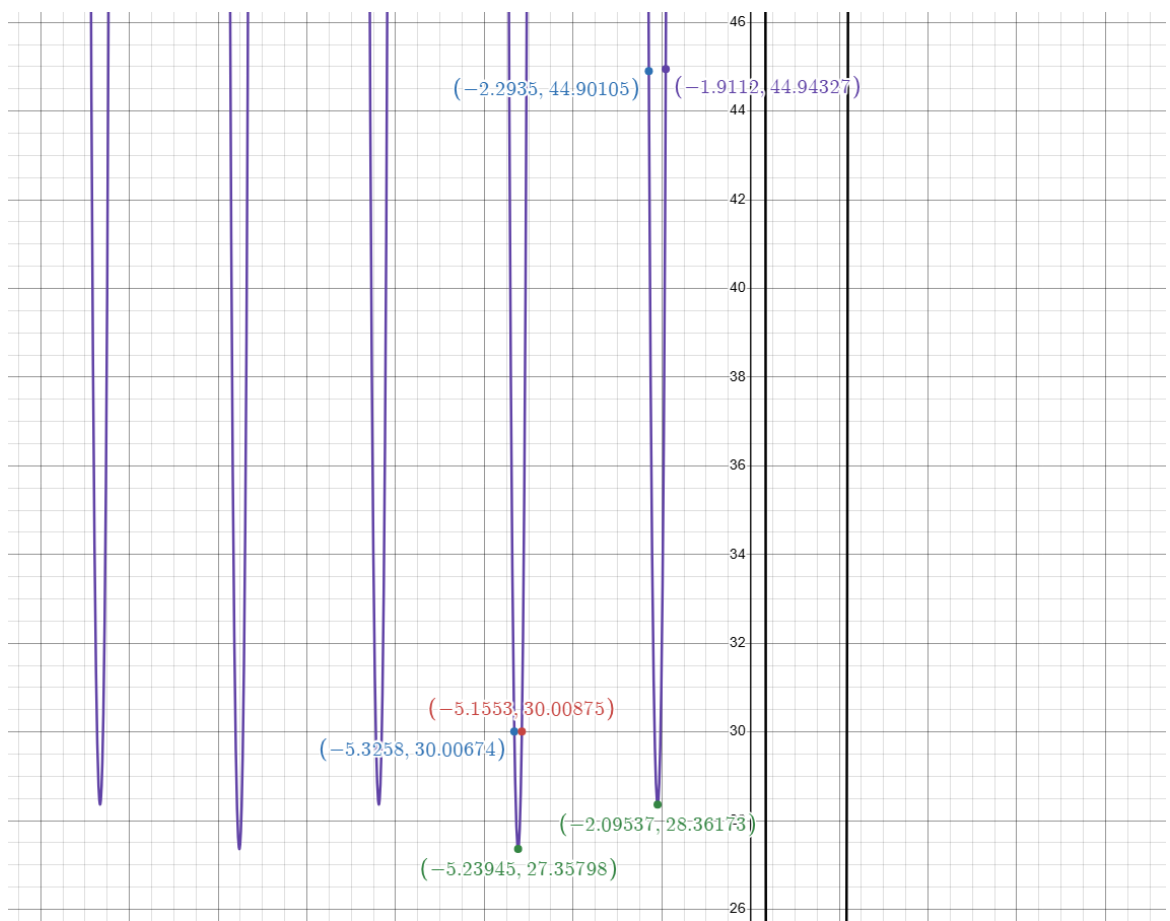
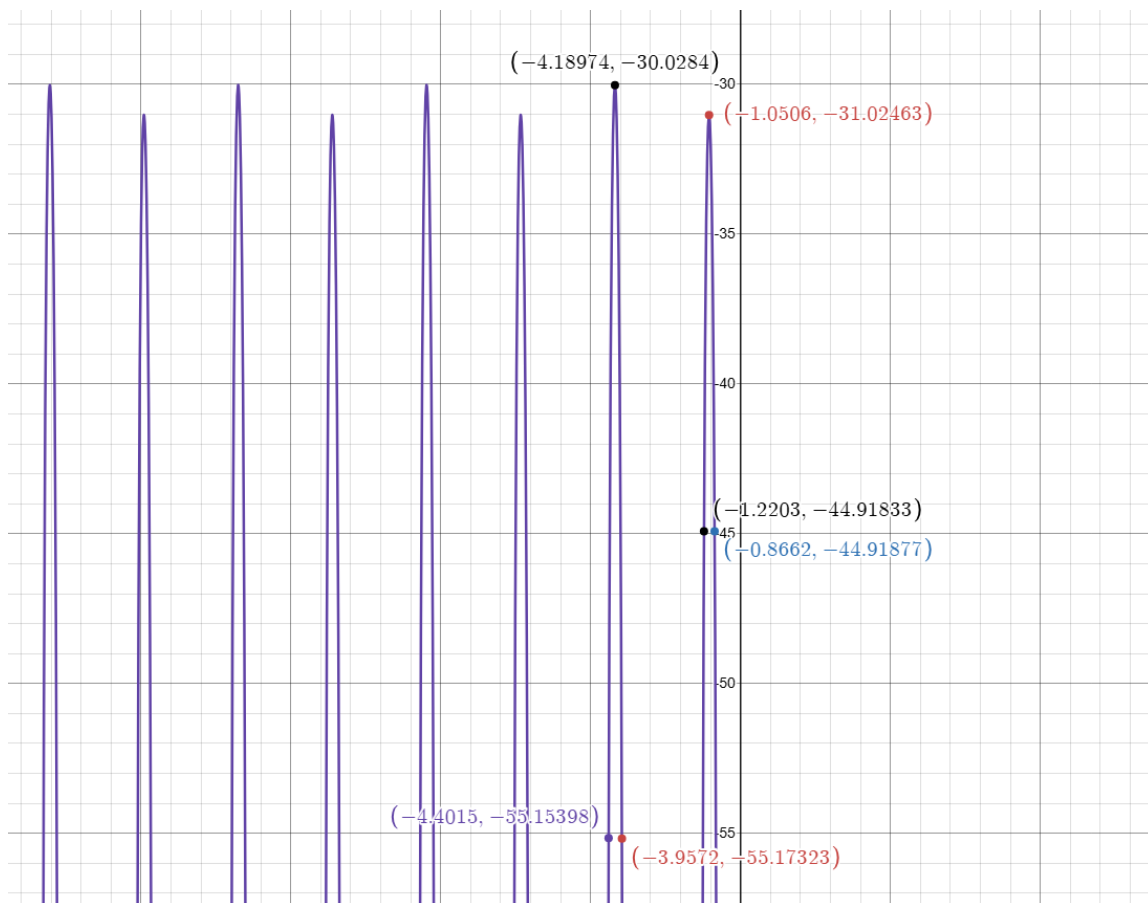


График при $x > 0$:



Графики при $x \leq 0$:





Интеграционное тестирование:

Test Summary

153	0	0	0.278s
tests	failures	ignored	duration

100%
successful

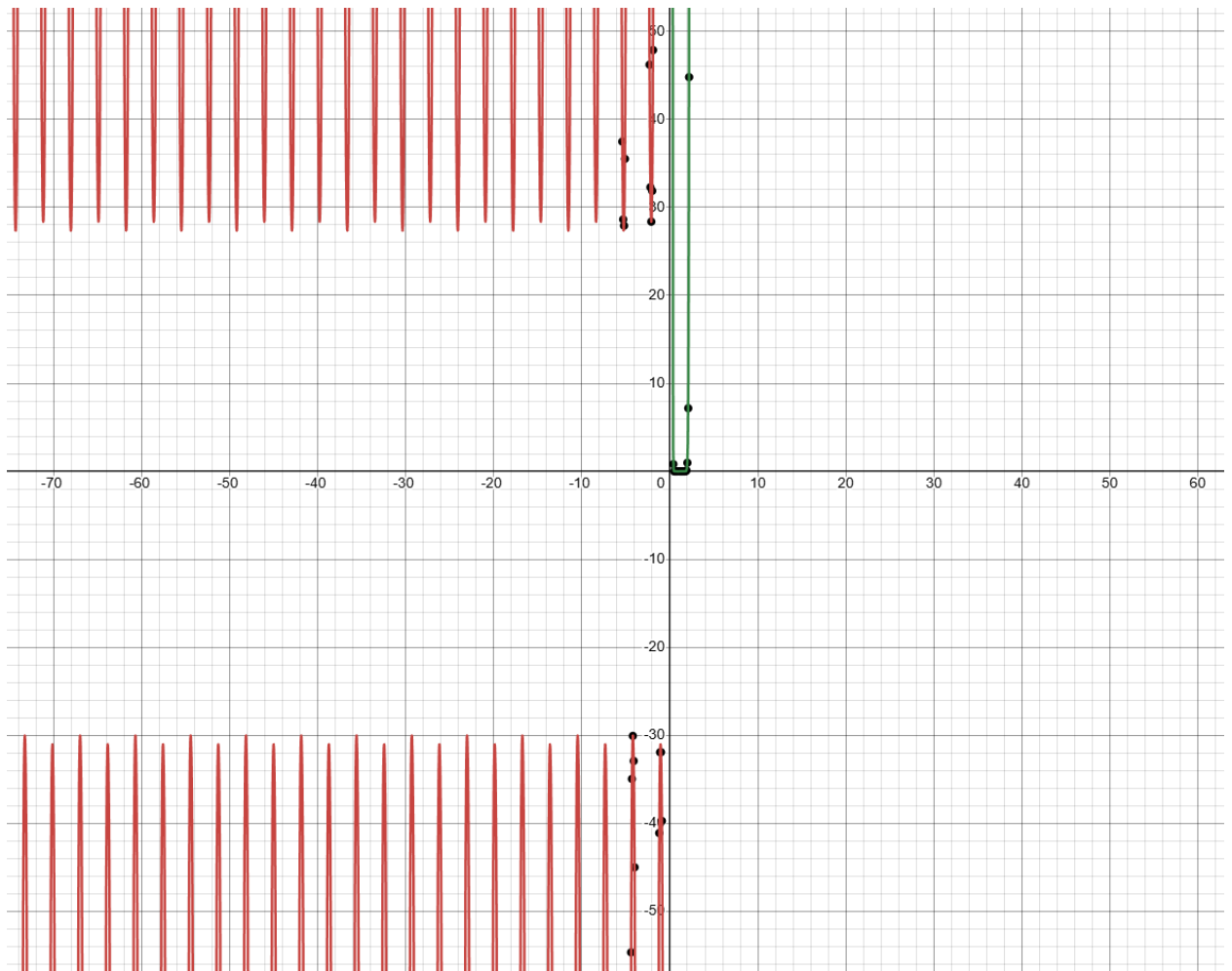
Packages

Classes

Class	Tests	Failures	Ignored	Duration	Success rate
functions.SystemOfFunctionsIntegrationTest	153	0	0	0.278s	100%

Графики, полученные по csv выгрузкам:

От -5.5 до 3 с шагом 0.1



Репозиторий с кодом: <https://github.com/Djerdn/TPO/tree/main/Lab2>

Вывод

В рамках лабораторной работы я познакомился с понятием интеграционного тестирования программы. Реализовал я его согласно подходу «сверху вниз», заменяя все зависимые модули заглушками.