Университет ИТМО

Факультет Программной инженерии и компьютерной техники

ЛАБОРАТОРНАЯ РАБОТА № 2  
ПО ДИСЦИПЛИНЕ «ТЕСТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ»  
ВАРИАНТ 4545

Выполнил студент группы Р3411   
Смирнова Анастасия Александровна

Преподаватель  
Грудина Анна Михайловна

Санкт-Петербург  
2020

Оглавление

[Задание 3](#_Toc53733546)

[Область допустимых значений 4](#_Toc53733547)

[UML-диаграмма классов разработанного приложения 5](#_Toc53733548)

[Порядок тестирования: 5](#_Toc53733549)

[Описание тестового покрытия 8](#_Toc53733550)

[Графики, построенные .csv выгрузками (в сравнении с эталоном) 9](#_Toc53733551)

[Sin(x), cos(x) 9](#_Toc53733552)

[Ctg(x) 9](#_Toc53733553)

[Sec(x), cos(x) 10](#_Toc53733554)

[Sin(x), csc(x) 11](#_Toc53733555)

[Ln(x), log10(x), log2(x) 12](#_Toc53733556)

[Графики функций f(x), x <= 0 и g(x) x > 0 13](#_Toc53733557)

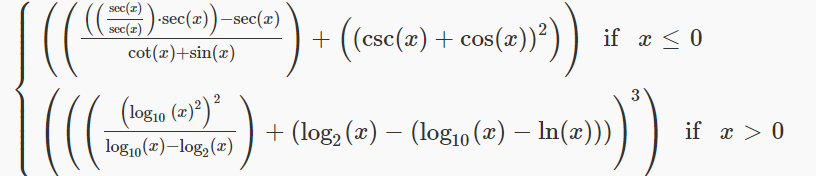
[График функции y(x) - итоговый 14](#_Toc53733558)

[Код приложения и тестов 14](#_Toc53733559)

[Вывод 14](#_Toc53733560)

# Задание

Провести интеграционное тестирование программы, осуществляющей вычисление системы функций:



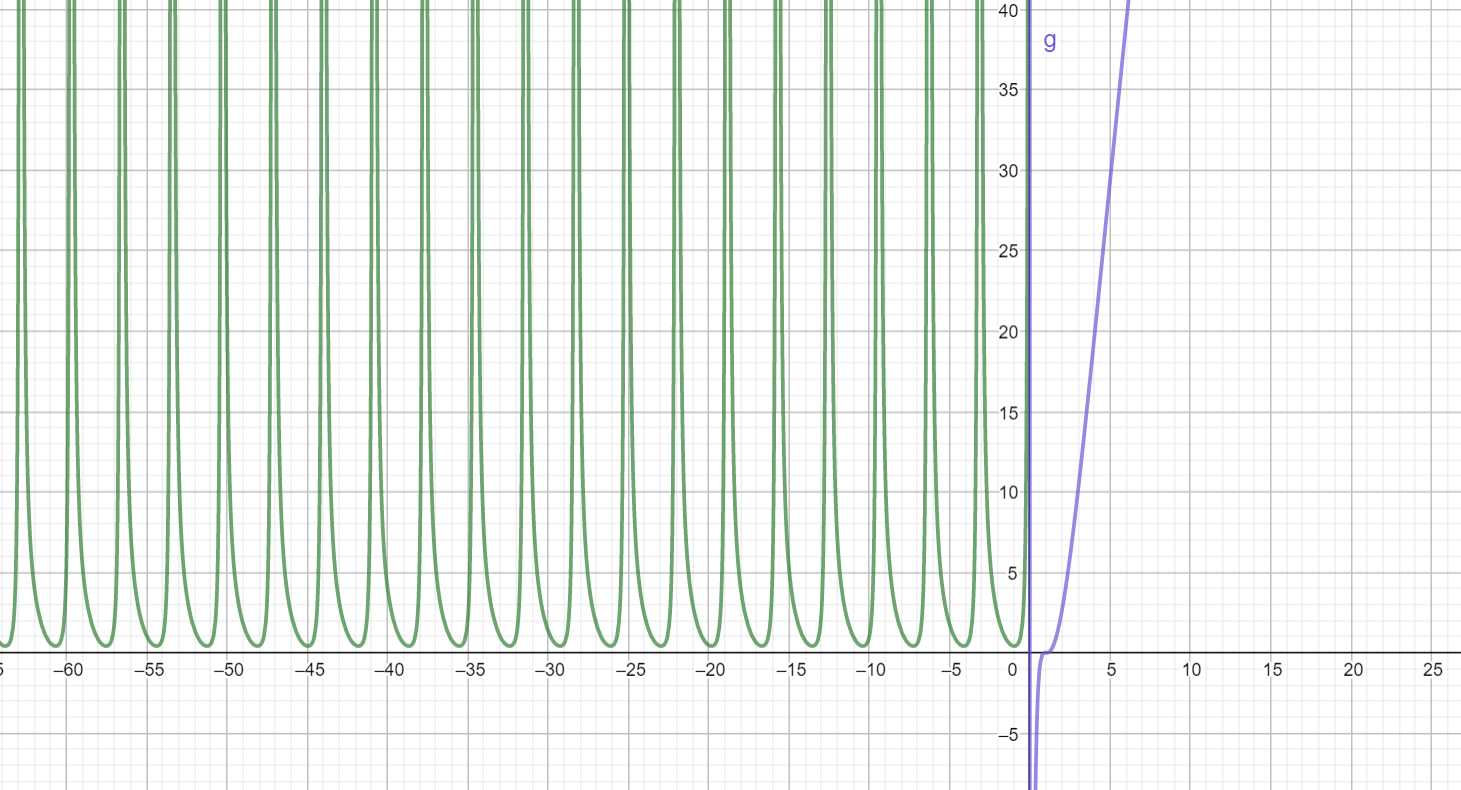


График 1 График функции y(x) – общий вид

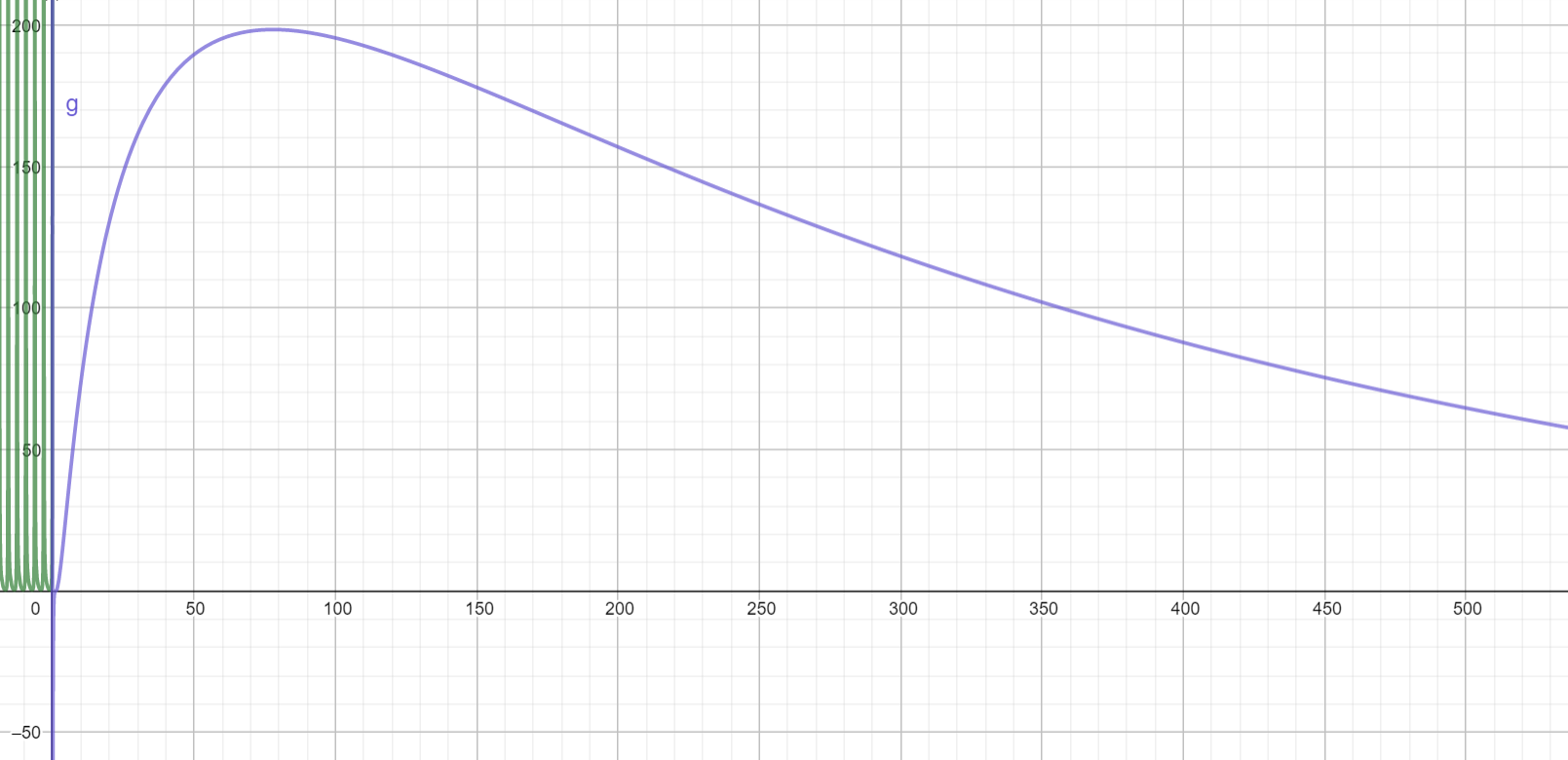


График 2 Общий вид графика функции g(x) - положительный х

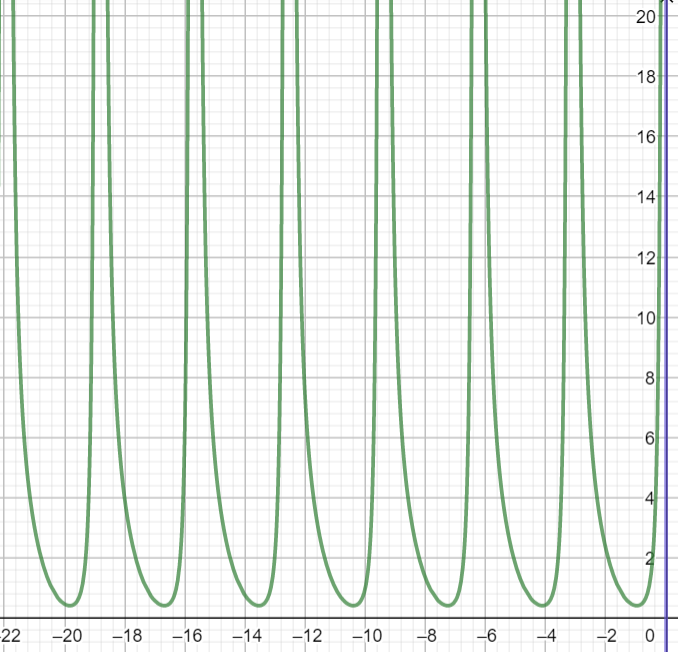


График 3 Приближение графика функции f(x), x <= 0

# Область допустимых значений

1. Рассмотрим первую функцию в системе – f(x) при x<= 0.

Обратим внимание на третье условие

1. Рассмотрим вторую функцию в системе g(x) – положительный х

Первые 2 условия обусловлены областью допустимых значений произвольной логарифмической функции. Подробнее рассмотрим третье условие:

Таким образом, область допустимых значений представляет собой объединение следующих систем:

# UML-диаграмма классов разработанного приложения

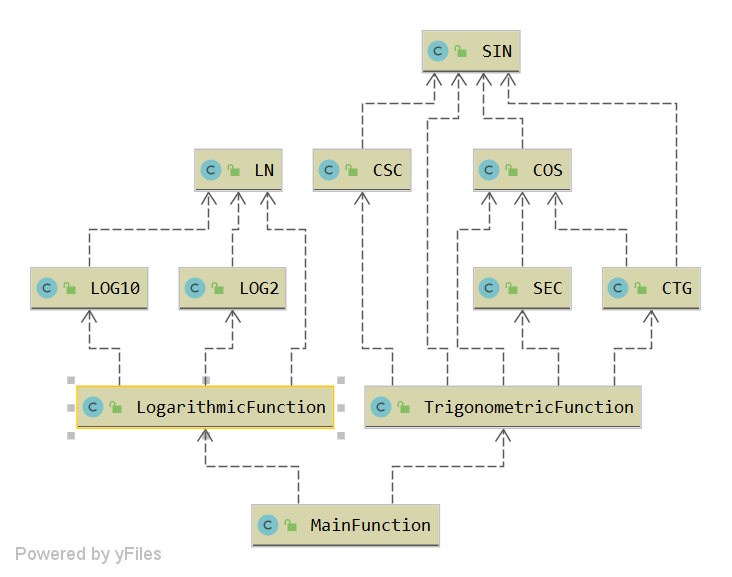
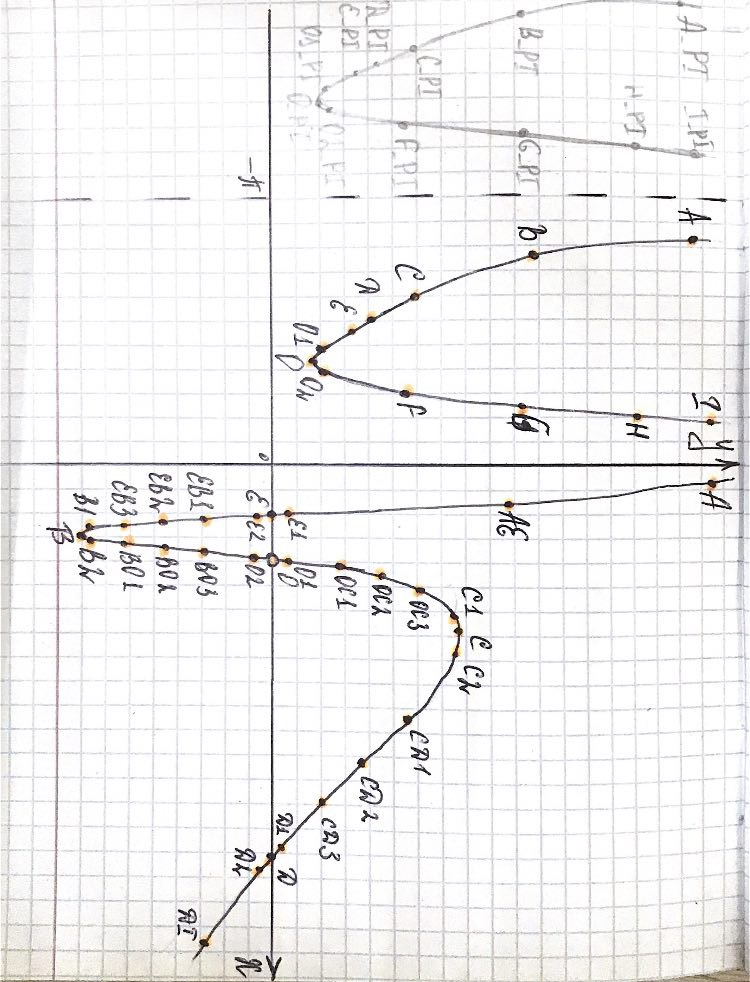


Диаграмма 1 UML-диаграмма классов разработанного приложения

# Порядок тестирования:

|  |  |
| --- | --- |
| Шаг 1 | Шаг 2 |
| Шаг 3 | Шаг 4 |
| Шаг 5 | Шаг 6 |
| Шаг 7 | Шаг 8 |
| Шаг 9 | Шаг 10 |
| Шаг 11 | |

# Описание тестового покрытия



Определяем классы эквивалентности и граничные значения: для каждого класса эквивалентности вычисляем три равноудаленные друг от друга и от концов точки, граничные значения проверяем следующим образом:

1. Непосредственно точка – граница класса
2. Две точки, удаленные от граничной на малую величину

# Графики, построенные .csv выгрузками (в сравнении с эталоном)

## Sin(x), cos(x)

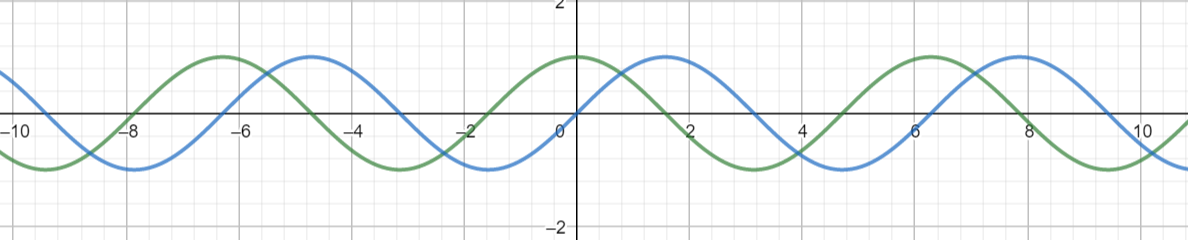


График 4 Эталон

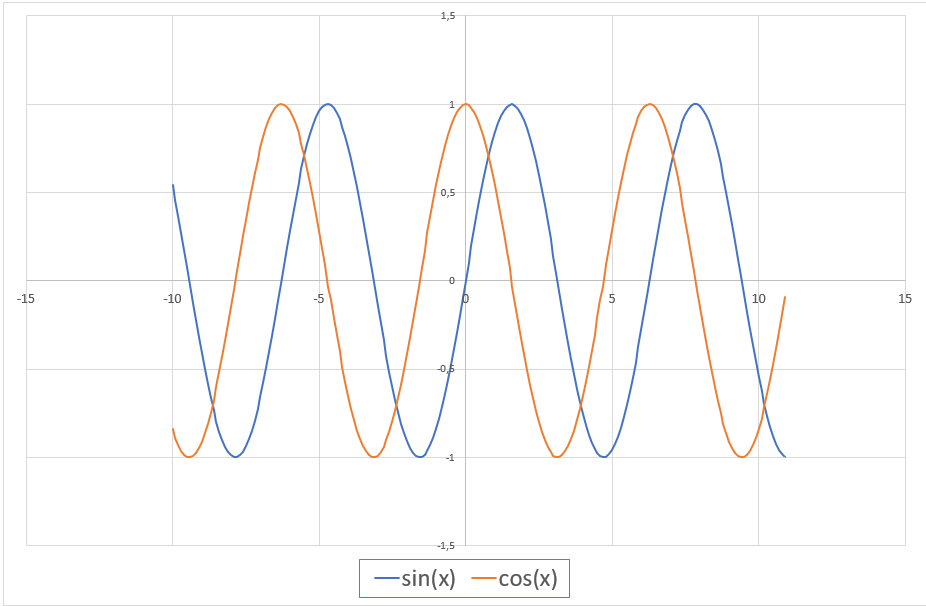


График 5 Графики функций sin(x) и cos(x) из csv

## Ctg(x)

График 6 Из csv

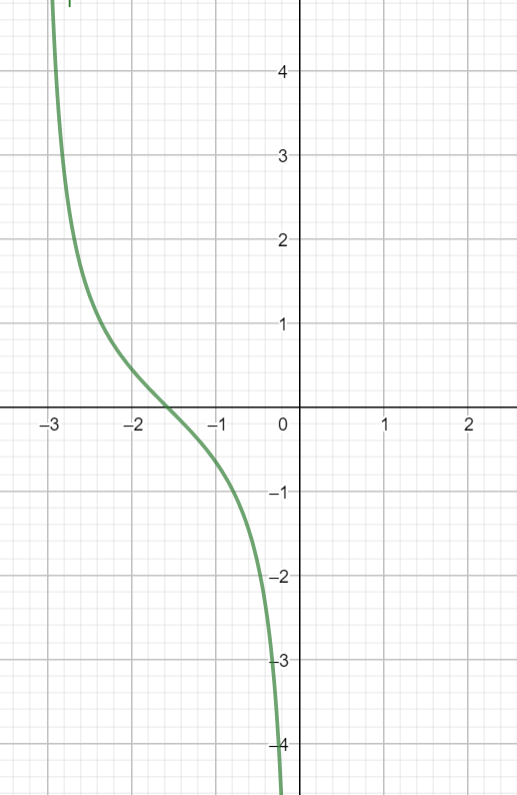


График 7 Эталон

## Sec(x), cos(x)

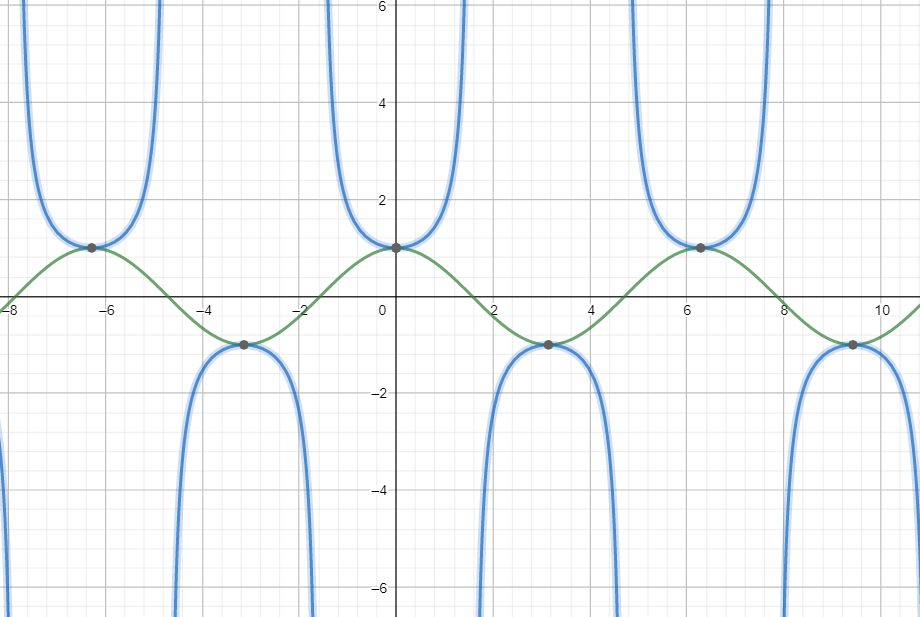


График 8 Эталон

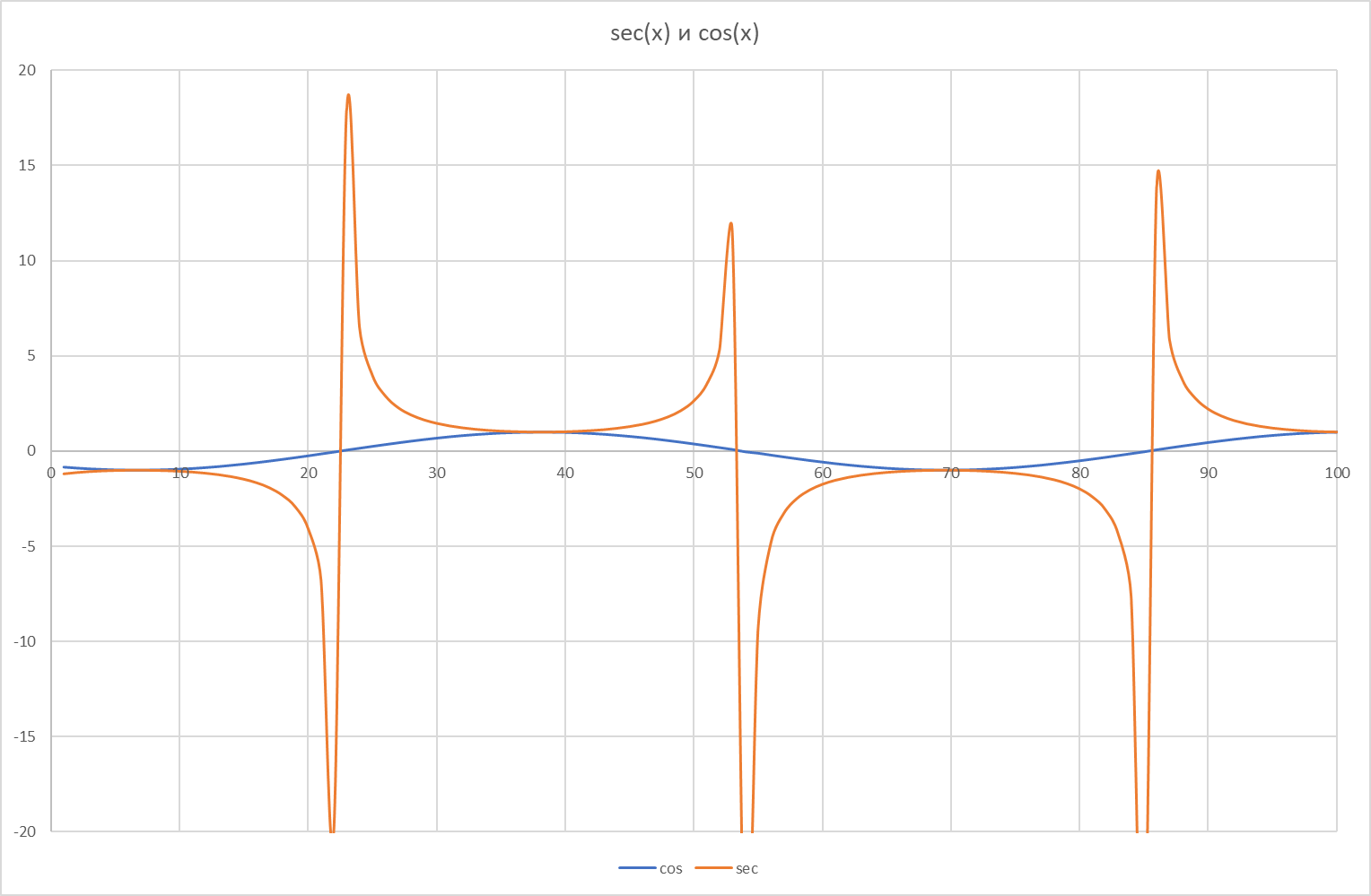


График 9 Из csv

## Sin(x), csc(x)

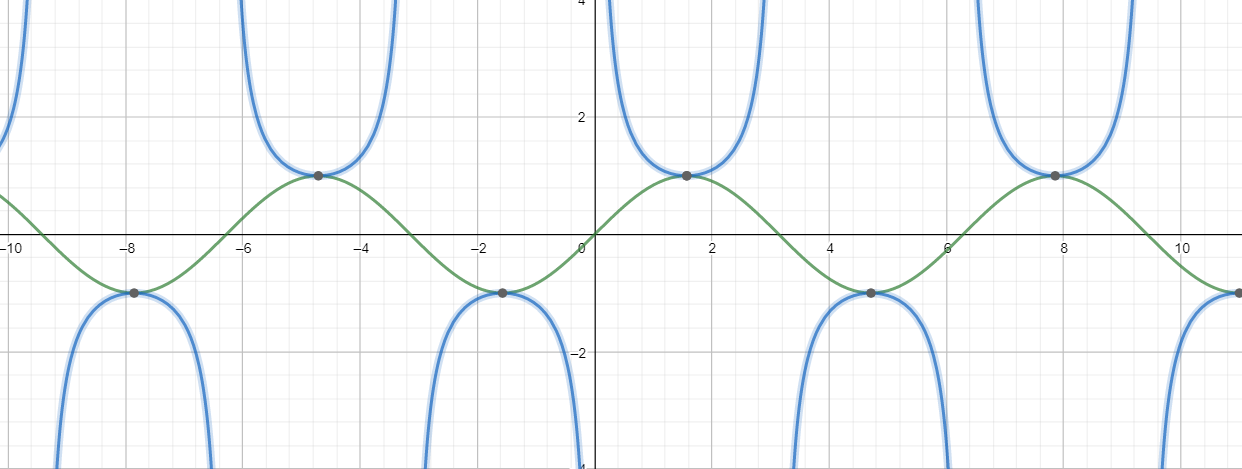


График 10 Эталон

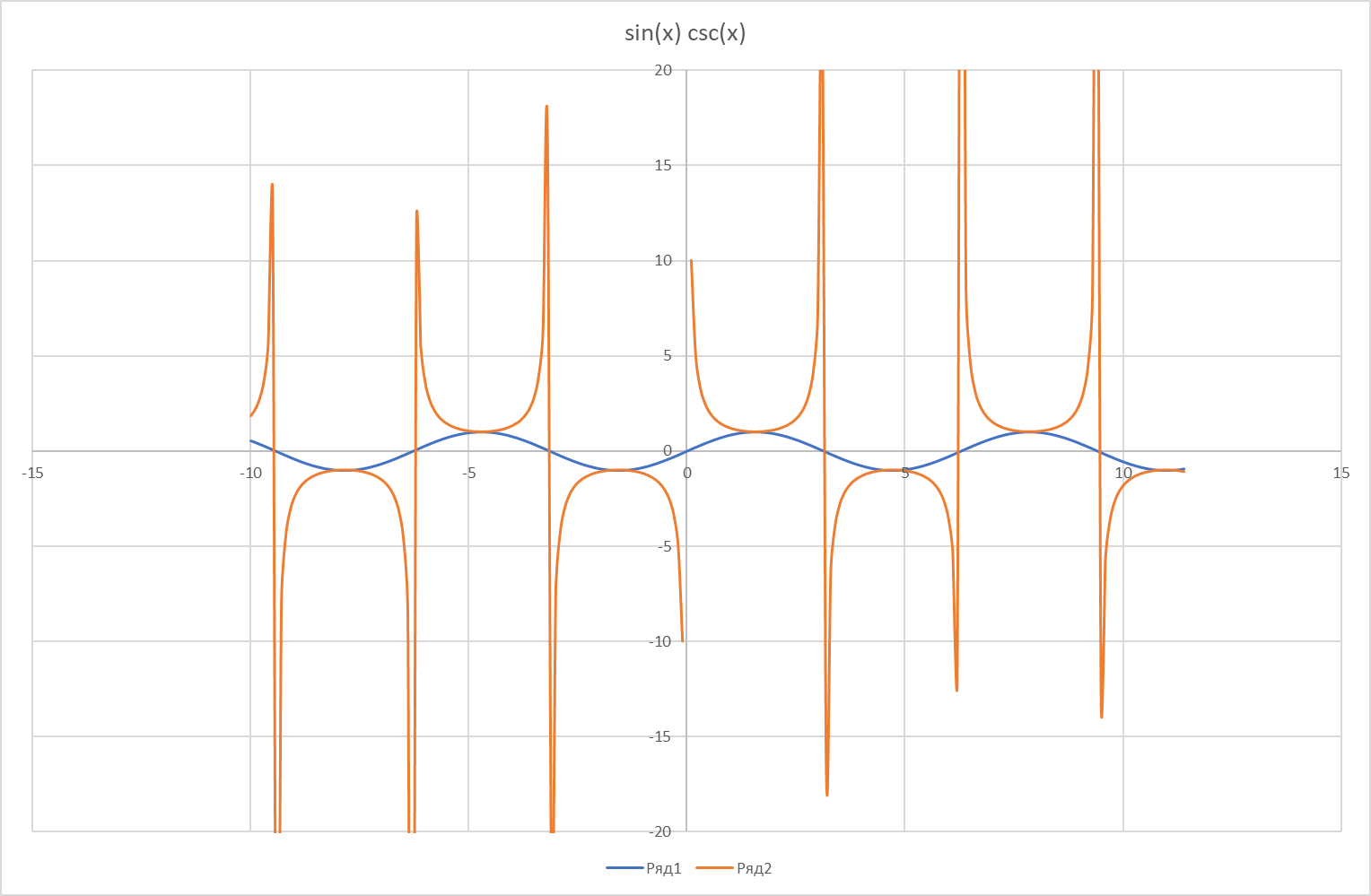


График 11 Из csv

## Ln(x), log10(x), log2(x)

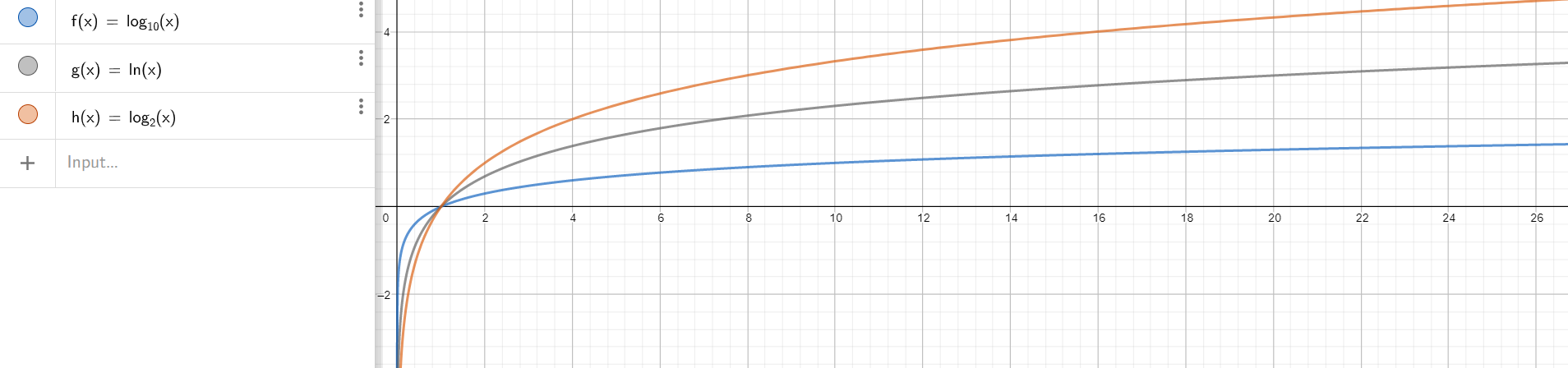


График 12 Эталон

График 13 Из csv

## Графики функций f(x), x <= 0 и g(x) x > 0

## График функции y(x) - итоговый

# Код приложения и тестов

<https://github.com/AnastasiyaSmirnova/TPO_lab2>

# Вывод

В ходе выполнения лабораторной работы были получены базовые навыки использования фреймворка Mockito – работы с заглушками, проведения интеграционного тестирования методом сверху-вниз.