

**Министерство науки и высшего образования Российской Федерации**  
Федеральное государственное автономное образование учреждение  
высшего образования “Национальный исследовательский университет  
ИТМО”

## **ОТЧЁТ**

По лабораторной работе №2  
По дисциплине “Тестирование программного обеспечения”  
Вариант 39574159

**Выполнил:**

Трошкин Александр Евгеньевич

**Факультет:**

ПИиКТ

**Преподаватель:**

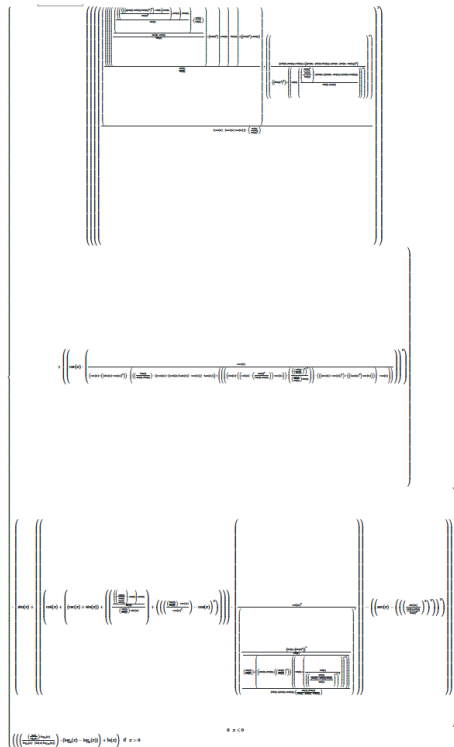
Ермаков Михаил Константинович



Санкт-Петербург, 2025

# Задание

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

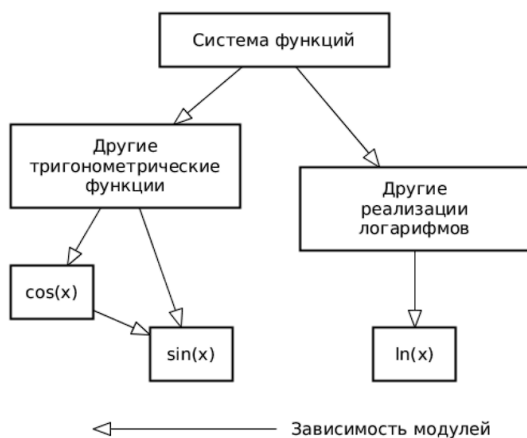


$x \leq 0 : ((((((((((((((((((((((\sin(x) + \cot(x)) * \tan(x)) ^ 2) ^ 3) - \sin(x)) * \csc(x)) / (\sec(x) ^ 2)) + \sec(x)) + \tan(x)) / \cos(x)) - (\csc(x) / \csc(x))) / (\sec(x) - \sec(x))) / \cot(x)) + (\csc(x) ^ 3)) + \sin(x)) - \tan(x)) + ((\csc(x) ^ 2) + \sin(x))) / (\sec(x) / \cot(x))) + (((((\cot(x)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

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

### Правила выполнения работы:

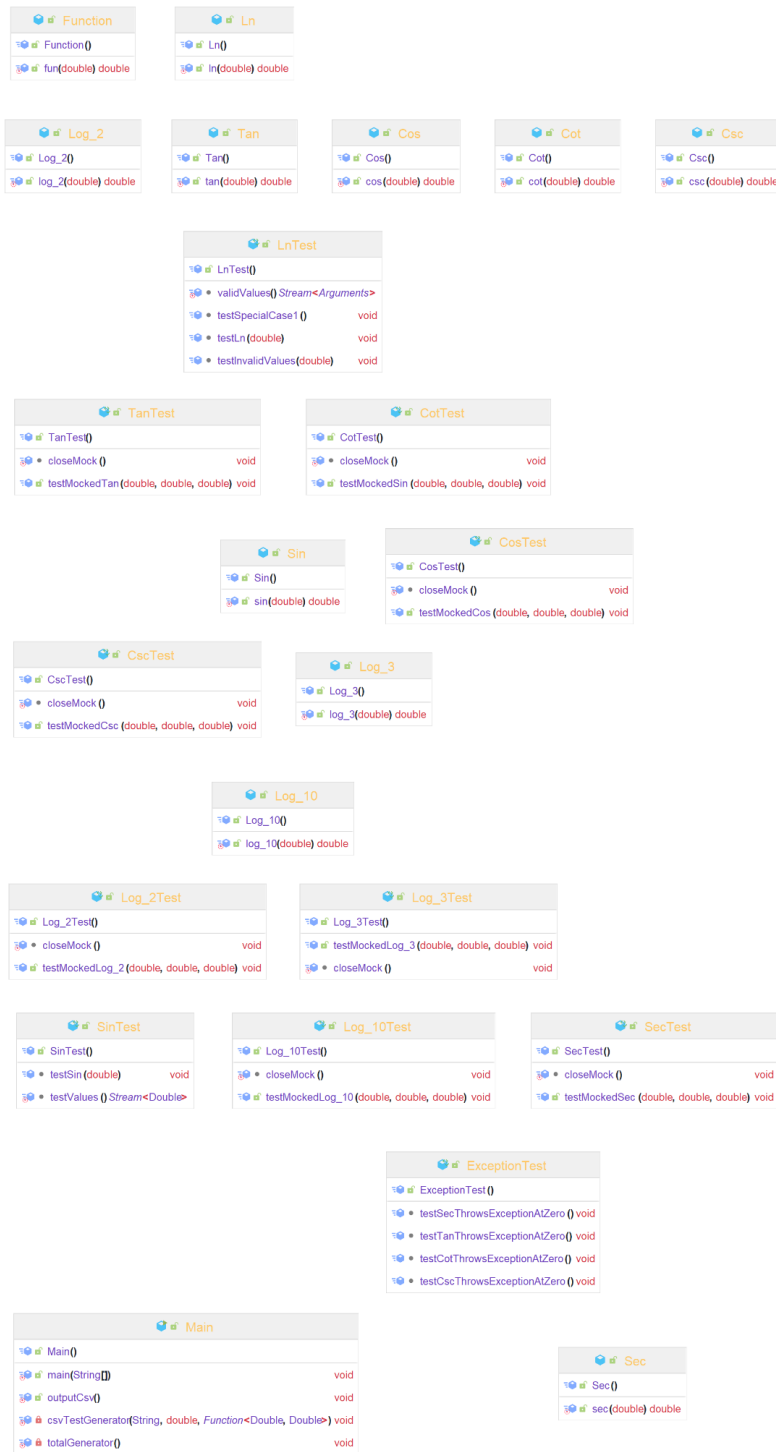
1. Все составляющие систему функции (как тригонометрические, так и логарифмические) должны быть выражены через базовые (тригонометрическая зависит от варианта; логарифмическая - натуральный логарифм).
2. Структура приложения, тестируемого в рамках лабораторной работы, должна выглядеть следующим образом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

# Выполнение

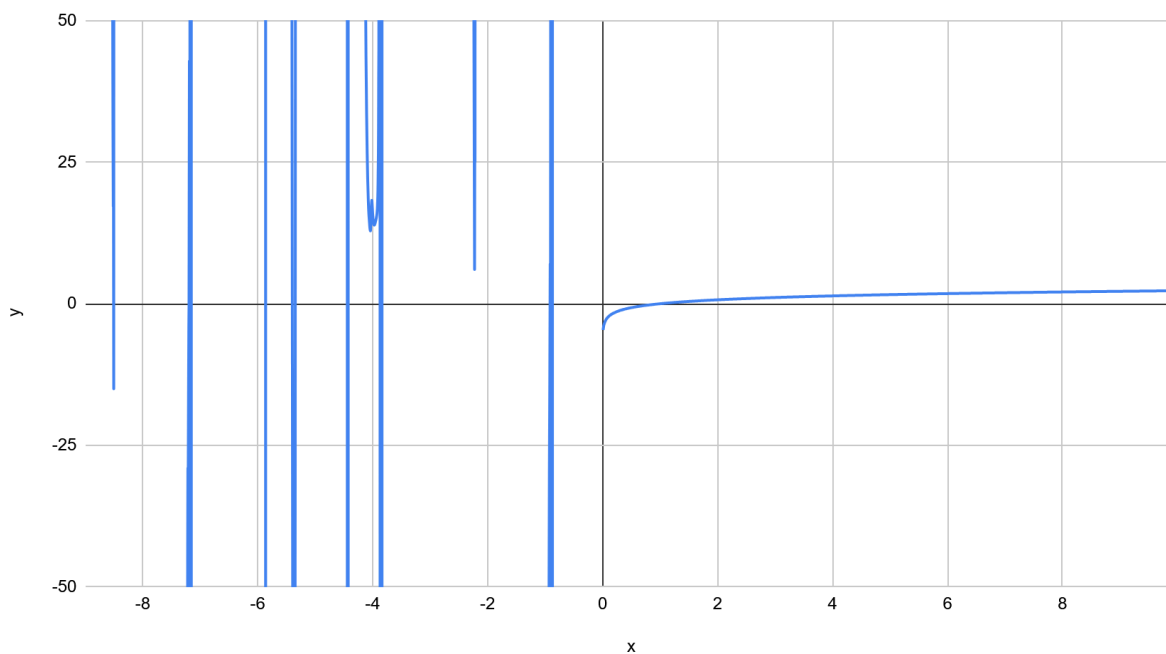
## Диаграмма классов



<https://github.com/k1nd-cat/software-testing/blob/main/lab2/diagram.png>

## График из выгрузки CSV

у относительно параметра "x"



[https://github.com/k1nd-cat/software-testing/blob/main/lab2/testing\\_lab2\\_output2.csv](https://github.com/k1nd-cat/software-testing/blob/main/lab2/testing_lab2_output2.csv)

Более полная выгрузка csv

[https://github.com/k1nd-cat/software-testing/blob/main/lab2/testing\\_lab2\\_output.csv](https://github.com/k1nd-cat/software-testing/blob/main/lab2/testing_lab2_output.csv)

## Исходный код

<https://github.com/k1nd-cat/software-testing/tree/main/lab2/src>

## Вывод

Я понял, что такое интеграционное тестирование, написал свои простейшие функции, интегрировал их в данную по условию задания функцию и провёл интеграционное тестирование, в котором исследовал разные точки, включая точки экстремумов функции.