

«Национальный исследовательский университет ИТМО»

## Лабораторная работа №2 по дисциплину "Тестирование ПО"

Выполнил: Чан Куанг Лок  
Группа: Р33011

Санкт-Петербург  
2021 г

# 1 Описание задания

№ варианта: 661

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

- $x \leq 0$

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

- $x > 0$

$$((((ln(x)^3) + log_2(x)) - log_2(x)) + log_2(x))^2)$$

## 1.1 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.

## 1.2 Порядок выполнения работы

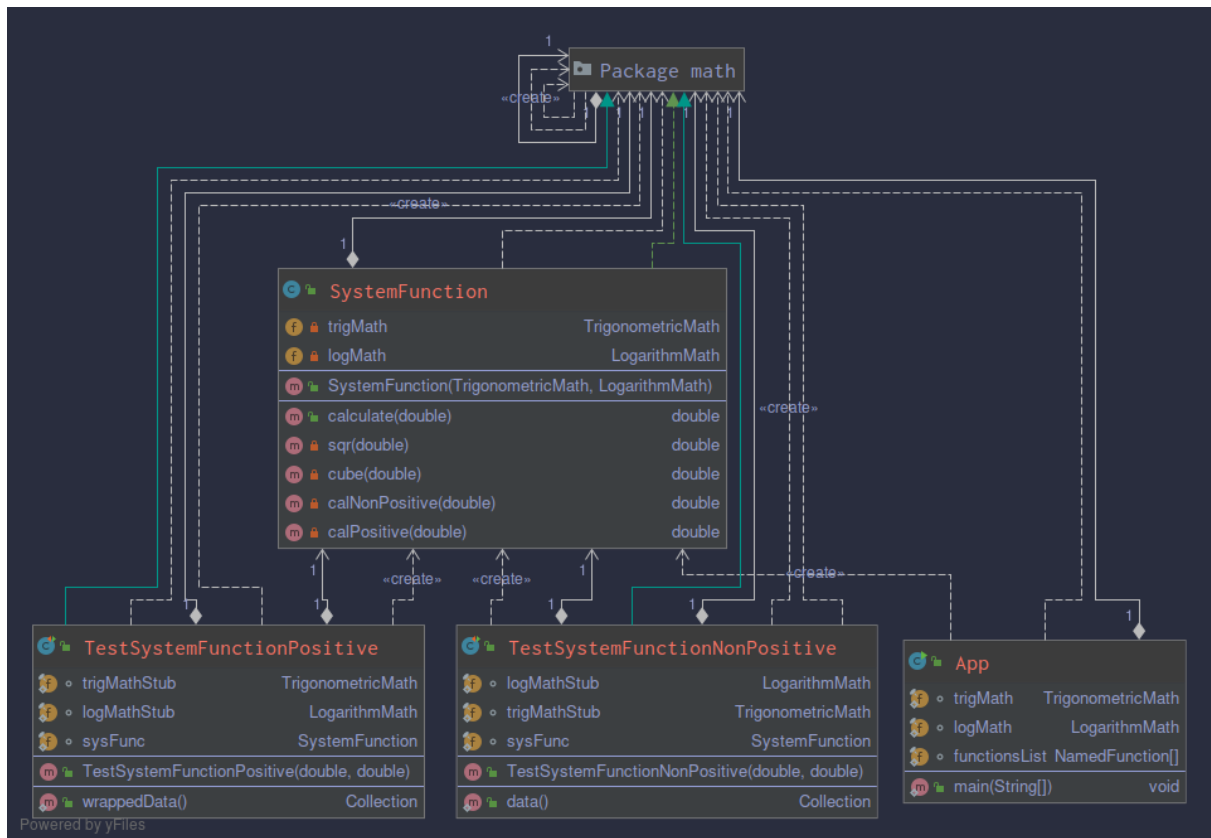
1. Разработать приложение, руководствуясь приведёнными выше правилами.
2. С помощью JUNIT4 разработать тестовое покрытие системы функций, проведя анализ эквивалентности и учитывая особенности системы функций. Для анализа особенностей системы функций и составляющих ее частей можно использовать сайт <https://www.wolframalpha.com/>.
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.

# 2 Выполнение

## 2.1 Ссылка на github

<https://github.com/quangloc99/software-testing-lab2>

## 2.2 UML диаграмм



## 2.3 Описание тестового покрытия

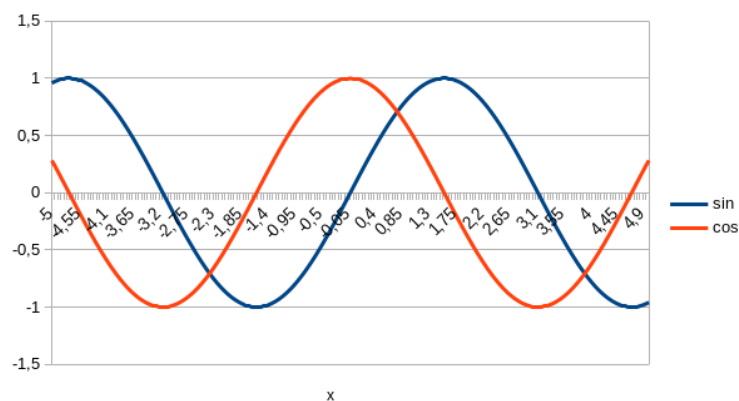
Для всех функции, я выбрал 2 интервалы значений:

- От -10 до 10 с шагом 0.25.
- От -100 до 100 с шагом 5.

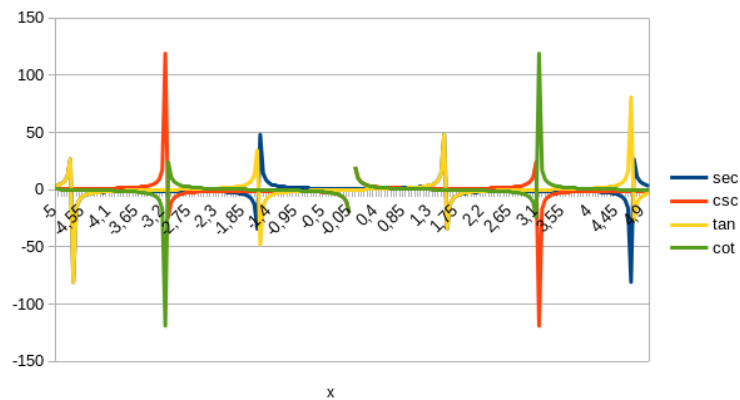
Нет никакой конкретной причины, я просто думаю, что этих интервалов достаточно, и на самом деле в этих диапазонах есть много значений для тестирования.

## 2.4 Графики

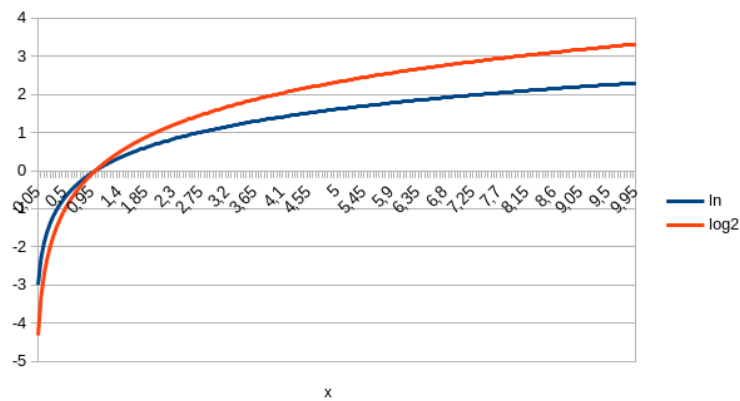
### 2.4.1 sin и cos



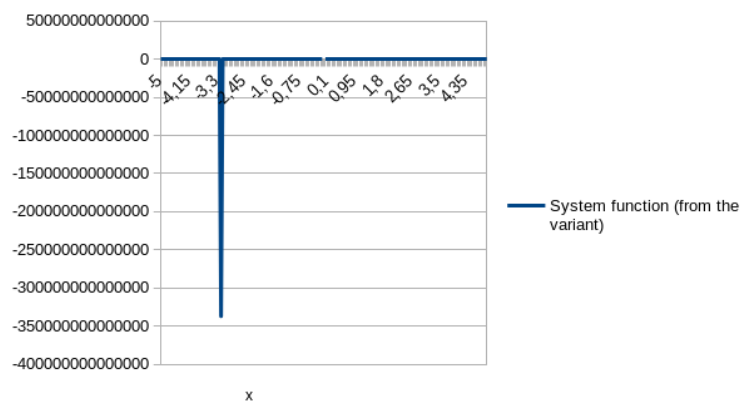
### 2.4.2 sec, csc, tan и cot



### 2.4.3 ln и log2



### 2.4.4 Системная функция



## 3 Вывод

Код для лабораторной работы прост, но количество тестов может очень быстро расти, если мы будем использовать одинаковое количество тестов для каждого модуля. И исправление «тестов», а также заглушек также занимает очень много времени, и в этой лабораторной работе мне нужно исправить тест из соображений точности. В целом лабораторная работа проста, и я получил базовое понимание того, как выполнять интеграционные тесты сверху вниз.