Университет ИТМО

Кафедра вычислительной техники

Лабораторная работа #2

по дисциплине

«Тестирование программного обеспечения»

Вариант 808

Выполнили студенты группы Р3401:

Третьяков Константин Павлович

Тищук Богдана Юрьевна

Преподаватель:

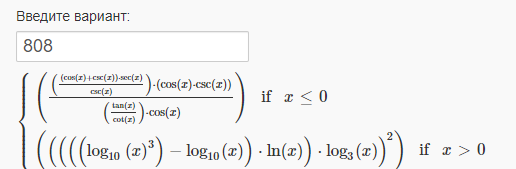
Харитонова Анастасия Евгеньевна

Санкт-Петербург – 2019

# Текст задания:

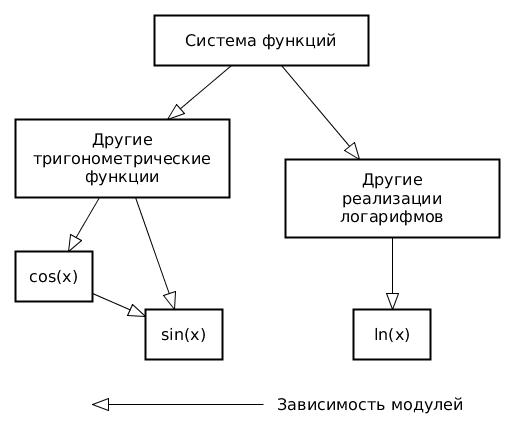
Провести интеграционное тестирование программы, осуществляющей вычисление системы функций (в соответствии с вариантом).

## Система функций:



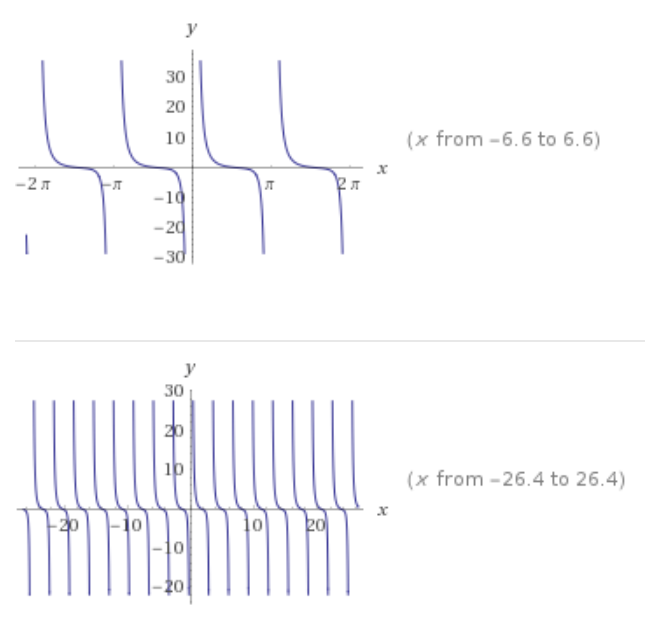
## Правила выполнения работы:

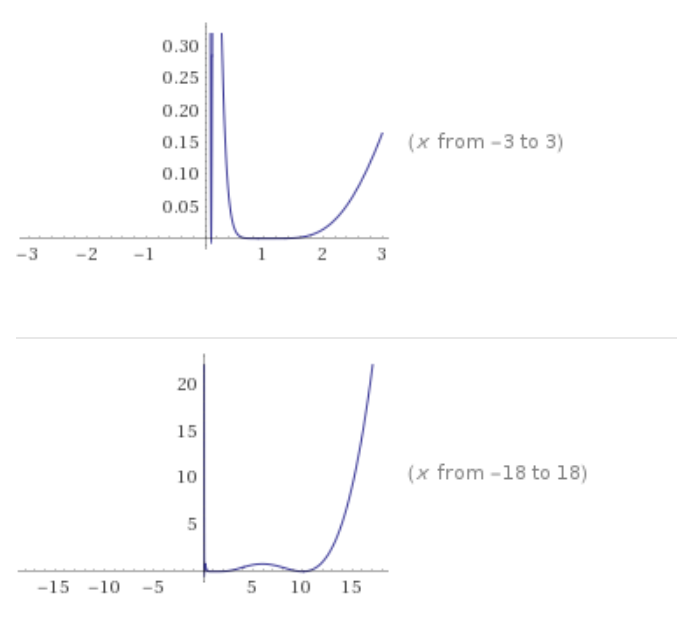
1. Все составляющие систему функции (как тригонометрические, так и логарифмические) должны быть выражены через базовые (тригонометрическая зависит от варианта; логарифмическая - натуральный логарифм).
2. Структура приложения, тестируемого в рамках лабораторной работы, должна выглядеть следующим образом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



1. 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.
2. Для КАЖДОГО модуля должны быть реализованы табличные заглушки. При этом, необходимо найти область допустимых значений функций, и, при необходимости, определить взаимозависимые точки в модулях.
3. Разработанное приложение должно позволять выводить значения, выдаваемое любым модулем системы, в сsv файл вида «X, Результаты модуля (X)», позволяющее произвольно менять шаг наращивания Х. Разделитель в файле csv можно использовать произвольный.

(((((cos(x) + csc(x)) \* sec(x)) / csc(x)) \* (cos(x) \* csc(x))) / ((tan(x) / cot(x)) \* cos(x)))



(((((log\_10(x) ^ 3) - log\_10(x)) \* ln(x)) \* log\_3(x)) ^ 2) 

Области определения функций:

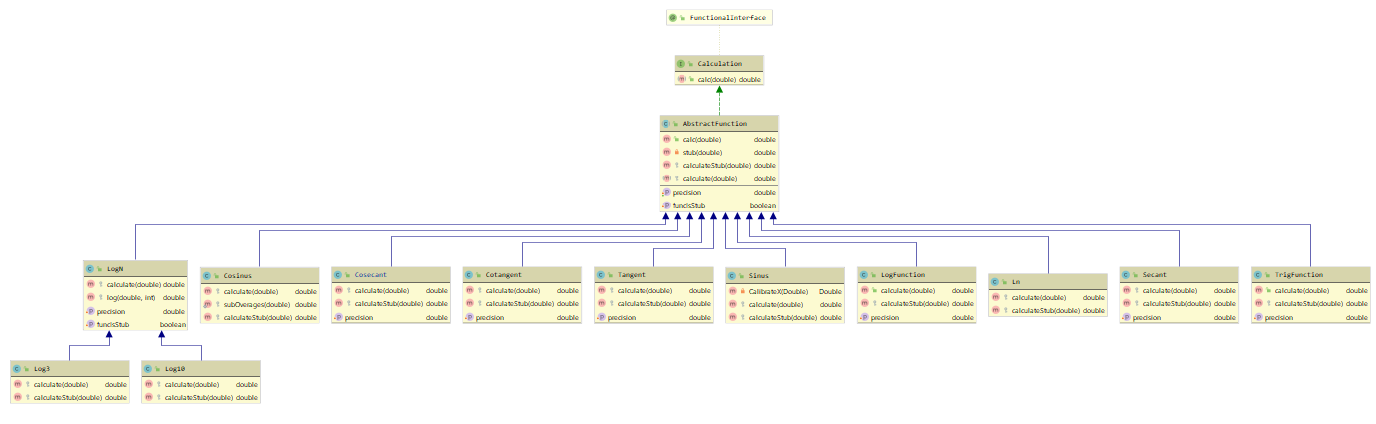
## Описание тестового покрытия с обоснованием его выбора:

(-2PI; 2PI) включая разрывы в -PI, 0, PI для тригонометрической функции ввиду того, что она обладает периодом PI.

[0.5;15],12,14 для логарифмической функции ввиду того, что на этом промежутке она ведет себя всеми возможными для себя способами: убывает, почти не изменяется и возрастает; плюс покрыть основания всех логарифмов.

## Графики, построенные csv-выгрузкам, полученным в процессе интеграции приложения.

# UML Диаграмма классов

****

**Вывод:** в ходе выполнения данной лабораторной работы было проведено интеграционное тестирование разработанных программных модулей с использованием средств JUnit 4.