Университет ИТМО  
Факультет программной инженерии и вычислительной техники

Лабораторная работа №2 по дисциплине  
 «Тестирование программного обеспечения»

Вариант №7999287

Выполнил:  
 Ерехинский Андрей Владимирович  
Студент группы Р33312

Преподаватель:   
[Наумова Надежда Александровна](https://my.itmo.ru/persons/264429)

Санкт-Петербург  
2023

Задание:

**Провести интеграционное тестирование программы, осуществляющей вычисление системы функций: sin ln**

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

x > 0 : (((((log\_2(x) + log\_2(x)) + log\_2(x)) - log\_10(x)) ^ 3) - (log\_3(x) / (ln(x) - log\_5(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)):  
   Изображение выглядит как черный, темнота

   Автоматически созданное описание
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 сsv файл вида «X, Результаты модуля (X)», позволяющее произвольно менять шаг наращивания Х. 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.

**Отчёт по работе должен содержать:**

1. Текст задания, систему функций.
2. UML-диаграмму классов разработанного приложения.
3. Описание тестового покрытия с обоснованием его выбора.
4. Графики, построенные csv-выгрузкам, полученным в процессе интеграции приложения.
5. Выводы по работе.

Выполнение:

**Система функций:**

Изображение выглядит как Параллельный, линия, прямоугольный

Автоматически созданное описание

**Диаграмма классов:**

Изображение выглядит как текст, диаграмма, План, схематичный

Автоматически созданное описание

**Тестовое покрытие:**

Функция на х<=0 периодична, период равен 6,283

Изображение выглядит как текст, Параллельный, линия, прямоугольный

Автоматически созданное описание

Функция на периоде в масштабе от -100000 до 100000, за исключением некоторых участков, функция представляет почти вертикальные линии.

Изображение выглядит как линия, Параллельный, число, прямоугольный

Автоматически созданное описание

Функция на периоде в масштабе, достаточном для опознания не вертикальных линий

Изображение выглядит как линия, прямоугольный, Параллельный, Прямоугольник

Автоматически созданное описание

Тестовое покрытие было выбрано по точкам перегибов и изменениях направлений по их окрестностям. Вертикальные линии выбраны по пересечению с осью абсцисс. Ожидается, что из-за погрешностей вертикальные линии не попадут в необходимое значение ввиду слишком быстрого роста или убывания.

На промежутке от нуля выбраны точки в окрестностях перегиба. Выбирать точки близь оси ординат и после 3 заведомо безрезультатно ввиду погрешности методов приближенного вычисления.

Изображение выглядит как линия, диаграмма, График, текст

Автоматически созданное описание

**Графики:**

Изображение выглядит как текст, линия, диаграмма, Шрифт

Автоматически созданное описаниеИзображение выглядит как текст, линия, диаграмма, График

Автоматически созданное описание

Графики схожи

Изображение выглядит как линия, диаграмма, Параллельный, График

Автоматически созданное описание Изображение выглядит как линия, Параллельный, прямоугольный, диаграмма

Автоматически созданное описание

Изображение выглядит как текст, линия, График, число

Автоматически созданное описание Изображение выглядит как текст, число, График, линия

Автоматически созданное описание

Для проверки выбран промежуток от -6 до -5, практически все линии отображены корректно, за исключением вертикальных линий, например, чуть правее -5.4.

Изображение выглядит как линия, График, диаграмма

Автоматически созданное описаниеИзображение выглядит как текст, линия, График, диаграмма

Автоматически созданное описание

Изображение выглядит как диаграмма, линия, График, Параллельный

Автоматически созданное описание Изображение выглядит как текст, линия, диаграмма, График

Автоматически созданное описание

Здесь выбран промежуток от -4 до -3,6. Отображено корректно

Прочие графики:

Изображение выглядит как линия, График, диаграмма

Автоматически созданное описание

Изображение выглядит как линия, График

Автоматически созданное описание

Изображение выглядит как текст, диаграмма, линия, График

Автоматически созданное описание

Изображение выглядит как текст, диаграмма, линия, число

Автоматически созданное описание

Изображение выглядит как текст, диаграмма, число, снимок экрана

Автоматически созданное описание

Изображение выглядит как текст, число, снимок экрана, Шрифт

Автоматически созданное описание

Изображение выглядит как линия, График, диаграмма, Параллельный

Автоматически созданное описание

Изображение выглядит как линия, диаграмма, текст, График

Автоматически созданное описание

Изображение выглядит как линия, График, диаграмма, Параллельный

Автоматически созданное описание

Изображение выглядит как текст, линия, График, диаграмма

Автоматически созданное описание

Изображение выглядит как текст, линия, График, диаграмма

Автоматически созданное описание

Изображение выглядит как линия, График, диаграмма, Параллельный

Автоматически созданное описаниеИзображение выглядит как линия, График, Параллельный, диаграмма

Автоматически созданное описание

**Исходный код:**

**https://github.com/Erekhinsky/TPO2**

Выводы: Во время выполнения этой лабораторной работы был изучен принцип работы заглушек на примере фреймворка Mockito, а также было проведено интеграционное тестирование приложения, решающего систему уравнений.