

Национальный исследовательский университет ИТМО
Факультет программной инженерии и компьютерной техники

Отчет лабораторной работе № 2
на курсе «Тестирование программного обеспечения»
Вариант 47283

Выполнили:
Батомункева В. Ж.
Туляков Е. В.

Группа:
Р33101

Преподаватели:
Машина Е. А.
Клименков С. В.

Санкт-Петербург
2024

Описание задания

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

Введите вариант:

$$\begin{cases} \left(\left(\frac{\cos(x) + \cot(x)}{\cot(x)} \right) \cdot \cot(x) \right) & \text{if } x \leq 0 \\ \left(\left(\left(\left(\ln(x) - \log_2(x) \right) - \ln(x) \right)^3 \right) - \log_2(x) \right) \cdot \log_3(x) & \text{if } x > 0 \end{cases}$$

$x \leq 0 : (((\cos(x) + \cot(x)) / \cot(x)) * \cot(x))$
 $x > 0 : (((((\ln(x) - \log_2(x)) - \ln(x)) ^ 3) - \log_2(x)) * \log_3(x))$

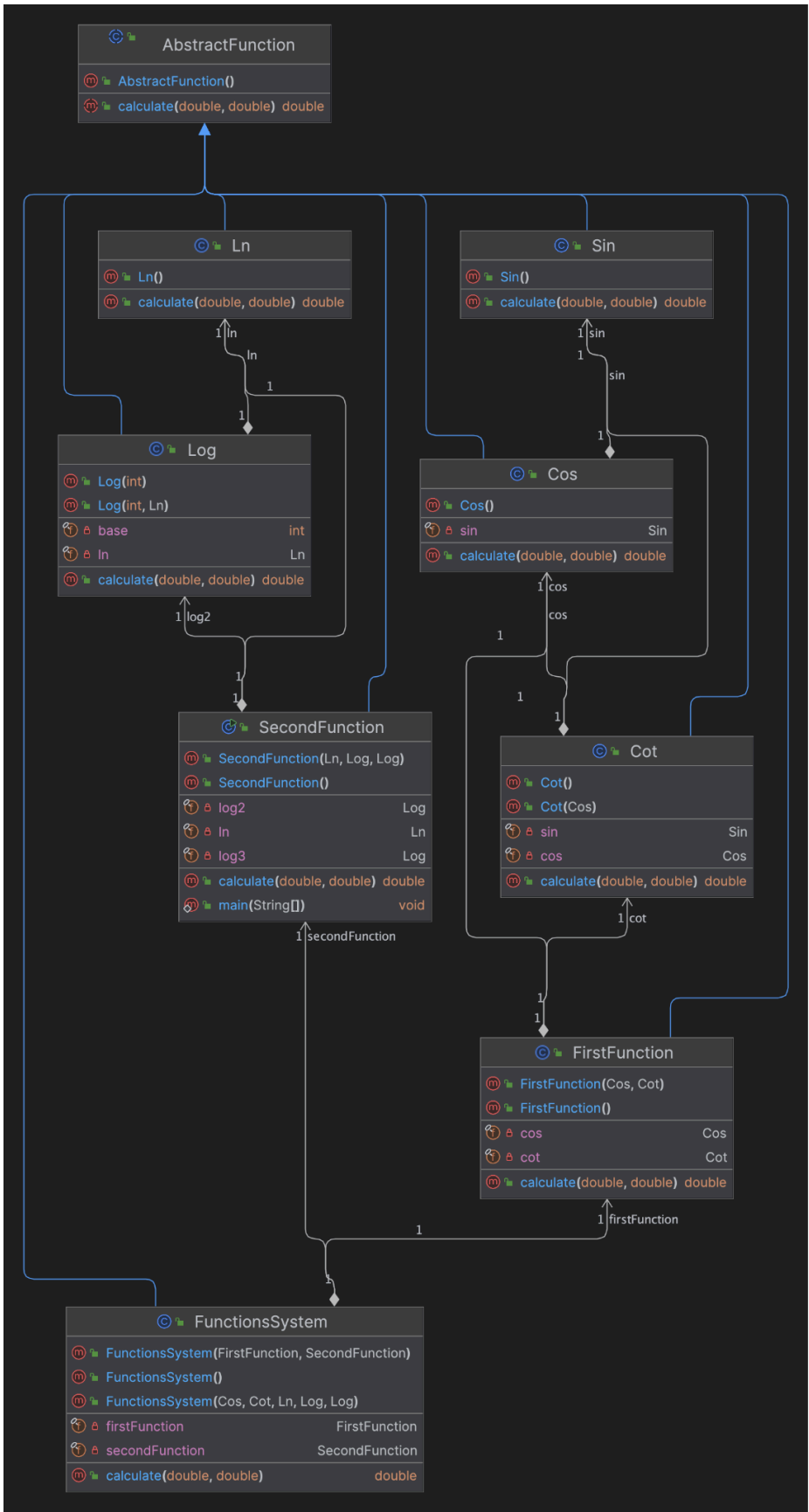
Правила выполнения работы:

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.

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.

UML - диаграмма классов



Исходный код

<https://github.com/youngpopeugene/SoftwareTesting/tree/main/lab2>

Вывод

Во время выполнения лабораторной работы мы изучили работу классов заглушек на примере библиотеки Moskito и применили ее для интеграционного тестирования написанного нами приложения для решения системы уравнений.