Университет ИТМО
Факультет программной инженерии и компьютерной техники

Лабораторная работа №2 (ЛР 2)

По дисциплине «Тестирование программного обеспечения»

Выполнил:
Студент группы P3331
Нодири Хисравхон
Вариант: 32123

Преподаватель:
Гаврилов Антон Валерьевич

г. Санкт-Петербург
2025г.

Содержание

1 Описание задания

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

- **Для** $x \leq 0$**:**

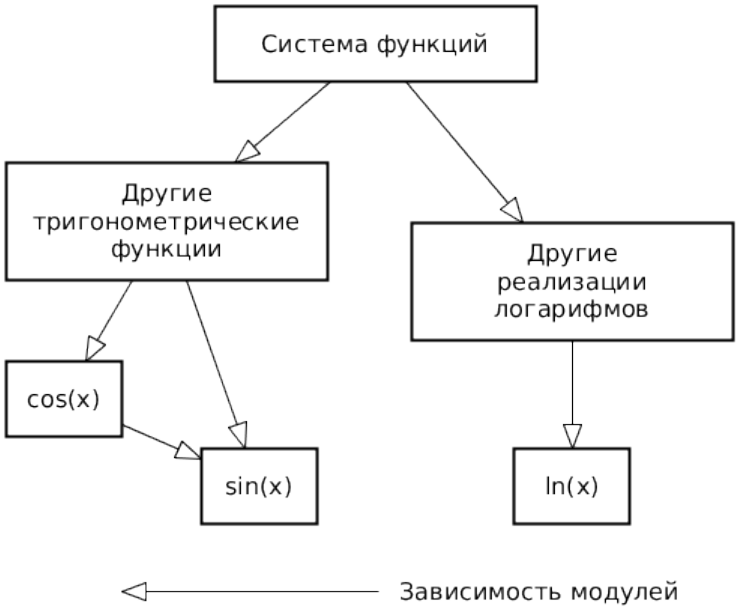
$$f(x) = \Big(\sin(x) - \csc(x)\Big) \cdot \Big(\frac{\csc(x)}{\csc(x)}\Big)$$

- **Для** $x > 0$**:**

$$f(x) = \left(\frac{(\log_2(x) + \log_{10}(x))^3}{\log_{10}(x)}\right)^3 + \ln(x)$$

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.

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.

2 Система функций

Система функций реализована следующим образом:

- **Для $x \leq 0$:**

$$f(x) = \Big(\sin(x) - \csc(x)\Big) \cdot \Big(\frac{\csc(x)}{\csc(x)}\Big)$$

  где:

  – $\sin(x)$ вычисляется через разложение в ряд Маклорена,

  – $\csc(x)$ определяется как обратное значение $\sin(x)$.

- **Для $x > 0$:**

$$f(x) = \left(\frac{(\log_2(x) + \log_{10}(x))^3}{\log_{10}(x)}\right)^3 + \ln(x)$$

  где логарифмические функции $\log_2(x)$ и $\log_{10}(x)$ выражаются через натуральный логарифм $\ln(x)$ посредством разложения в ряд.

В реализации используются две группы функций:

1. **Базовые функции:** функции разложения в ряд — `sin_series(x)` и `ln_series(x)`.
2. **Заглушки:** для каждого модуля (тригонометрического и логарифмического) предусмотрены табличные заглушки (например, `sin_stub(x)`, `ln_stub(x)` и т.д.), которые возвращают заранее заданные значения для критических точек.

3 UML-диаграмма классов

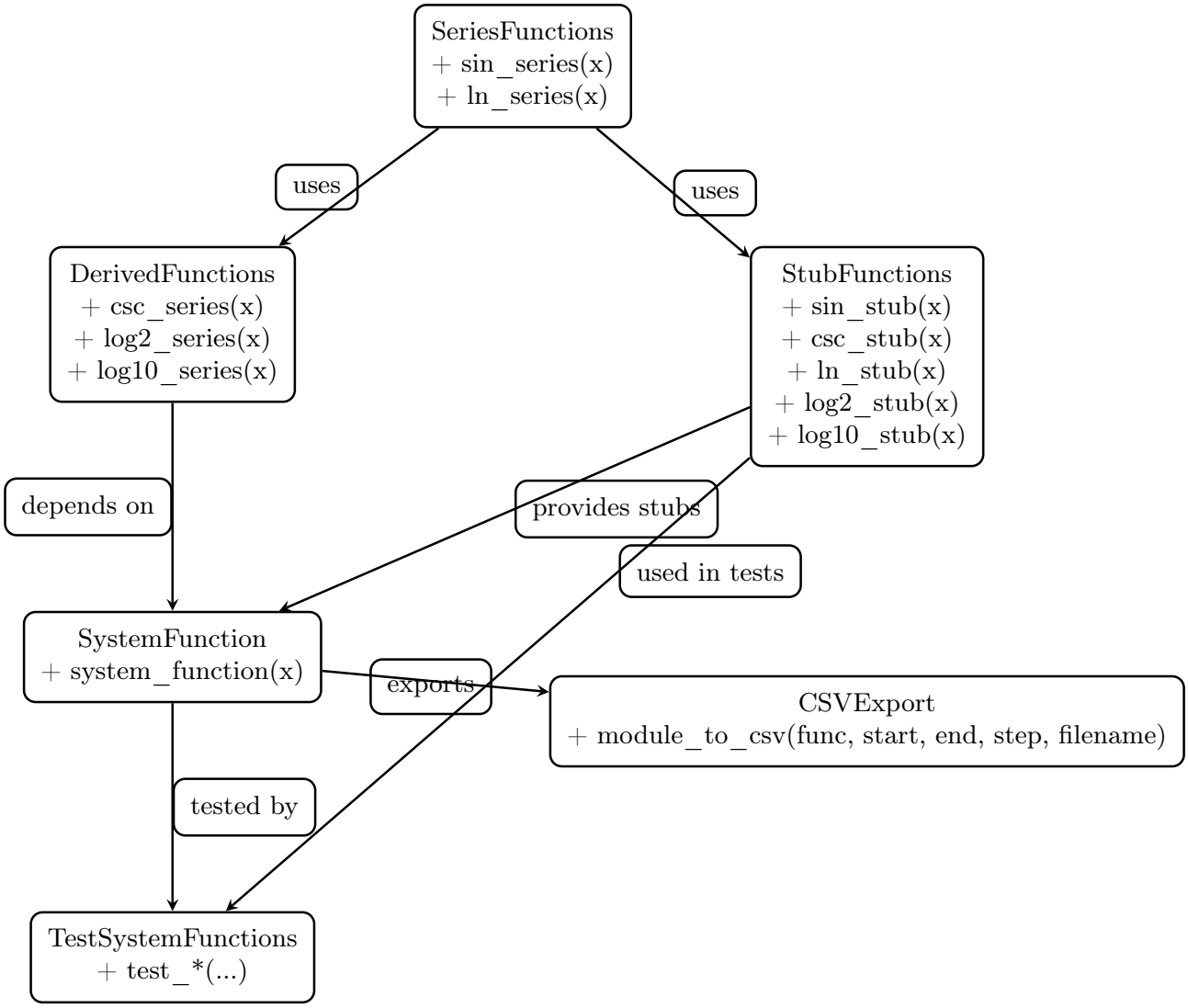


Рис. 1: UML-диаграмма классов приложения

4 Описание тестового покрытия и обоснования его выбора.

Тестовое покрытие реализовано с использованием модуля `unittest` и включает следующие группы тестов:

- **Тестирование базовых функций:** Проверка корректности работы функций `sin_series(x)` и `ln_series(x)` на известных значениях (например, $\sin(0) = 0$, $\sin(\pi/2) = 1$, $\ln(e) = 1$).
- **Тестирование производных функций:** Проверяются функции `log2_series(x)` и `log10_series(x)` для подтверждения правильного вычисления через натуральный логарифм.
- **Тестирование системы функций:** Отдельно тестируются ветки для $x \leq 0$ и $x > 0$. Особое внимание уделяется граничным условиям (например, деление на ноль, недопустимые значения) и дополнительным тестам с конкретными значениями, полученными в процессе интеграции.

Обоснование выбора тестов: Выбранные тесты охватывают все критические точки и ключевые функциональные блоки системы. Использование граничного анализа и эквивалентного разбиения позволяет обеспечить, что система корректно обрабатывает как типичные, так и экстремальные случаи. Дополнительные тесты, основанные на значениях, полученных в интеграционных CSV-выгрузках, позволяют подтвердить соответствие ожидаемым результатам.

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

В процессе интеграции приложения результаты работы системы функций сохранялись в CSV-файлы. Ниже приведены графики, построенные по этим данным.

График для области $x \leq 0$

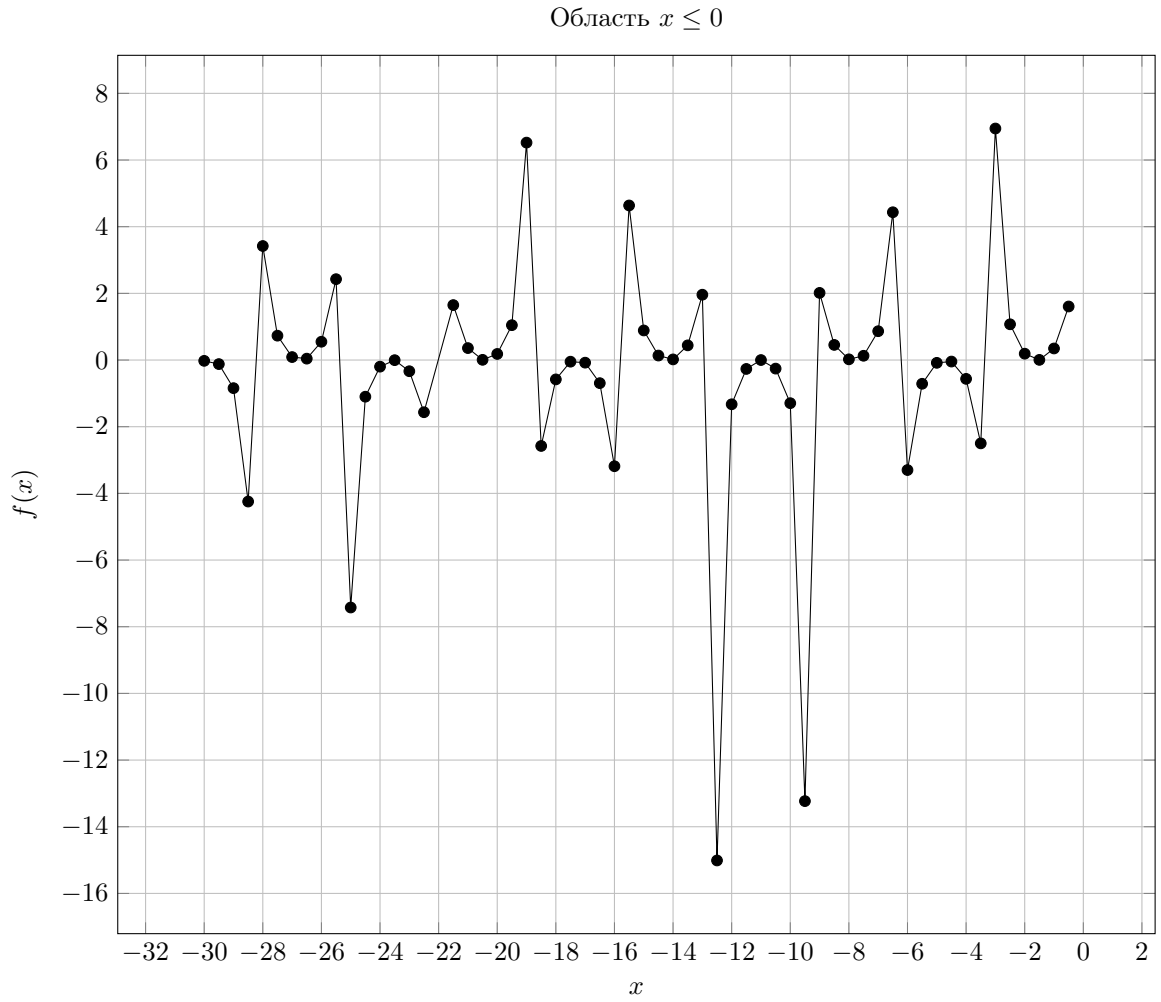


Рис. 2: График системы функций для $x \leq 0$

График для области $x > 0$

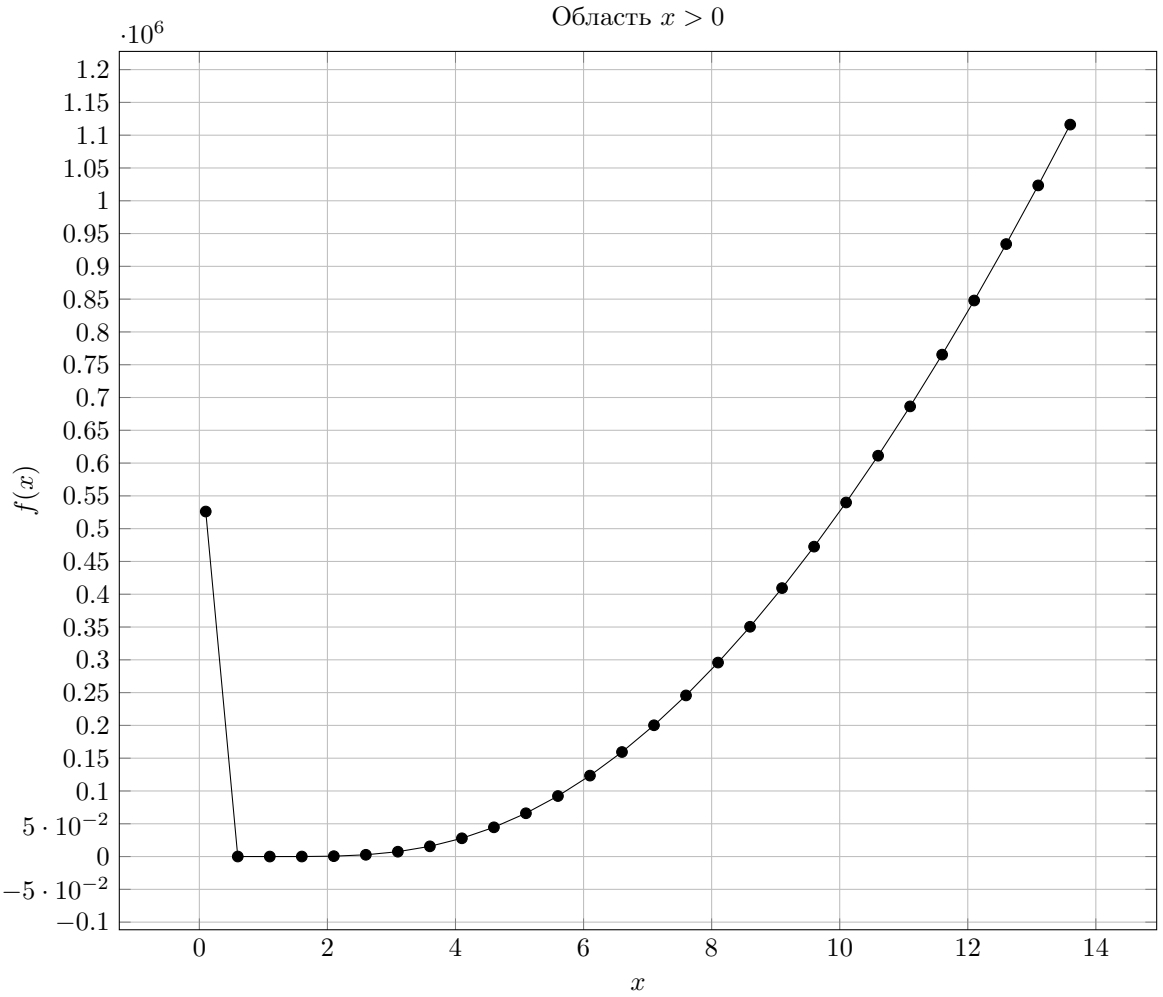


Рис. 3: График системы функций для $x > 0$

6 Выводы по работе

В итоге, интеграционное тестирование продемонстрировало, что система функций, реализованная на основе разложения в ряд, способна стабильно и точно вычислять значения как тригонометрических, так и логарифмических функций в широком диапазоне входных значений. Интересно отметить, что использование табличных заглушек для критических точек позволило избежать неопределённых ситуаций, обеспечив надежность работы приложения даже в граничных условиях. Сравнение полученных результатов с эталонными значениями, выгруженными в CSV-файлы, подтверждает корректность выбранного подхода, а визуальное представление графиков даёт наглядное понимание поведения системы при различных параметрах. Разработанное решение не только удовлетворяет требованиям задания, но и демонстрирует потенциал для дальнейшего расширения и применения в более сложных вычислительных задачах, оставаясь при этом модульным, прозрачным и гибким инструментом для анализа.