Федеральное государственное автономное образовательное учреждение высшего образования

 Национальный исследовательский университет ИТМО

Факультет программной инженерии и компьютерной техники

Направление подготовки 09.03.04 Программная инженерия

Дисциплина «*Тестирование программного обеспечения*»

**Отчёт**

**По лабораторной работе №2**

**Вариант: 5157**

Студент:

*Барсуков М. А.*

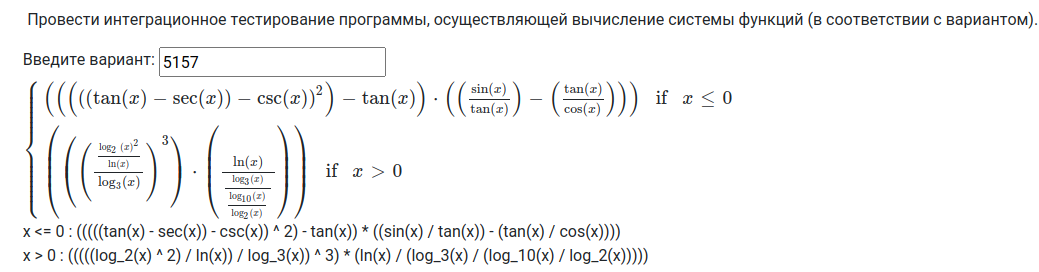
группа *P3315*

Преподаватель:

*Цопа Е. А.*

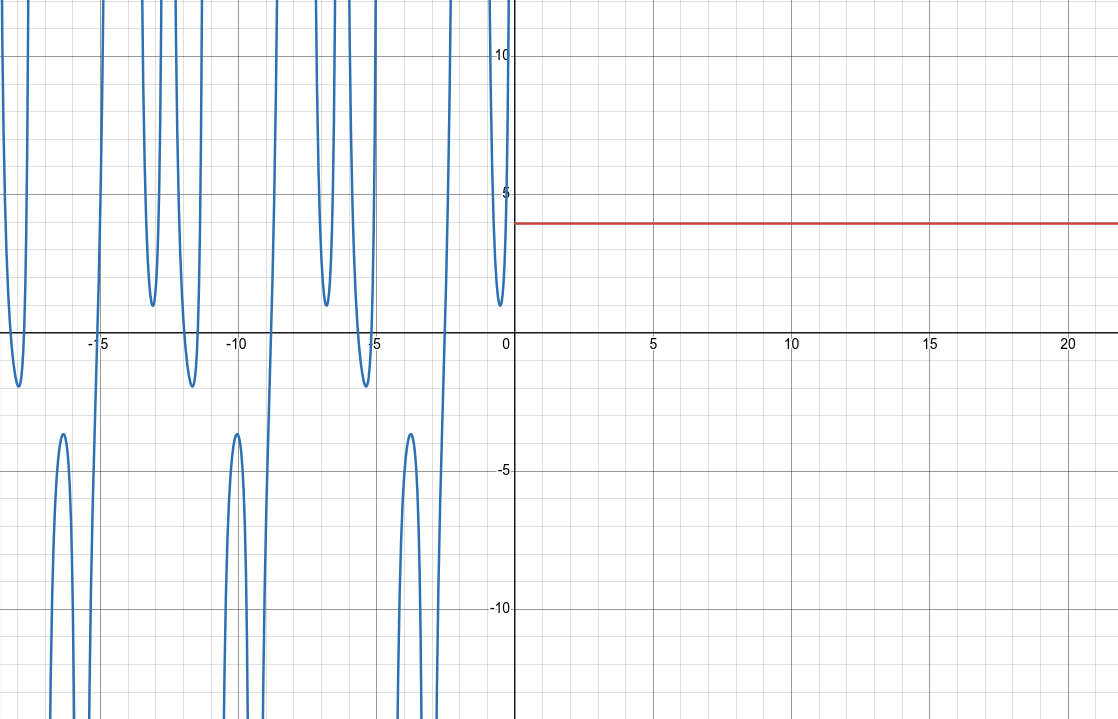
г. Санкт-Петербург, 2025 г.

**Описание задания**



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

x > 0 : (((((log\_2(x) ^ 2) / ln(x)) / log\_3(x)) ^ 3) \* (ln(x) / (log\_3(x) / (log\_10(x) / log\_2(x)))))



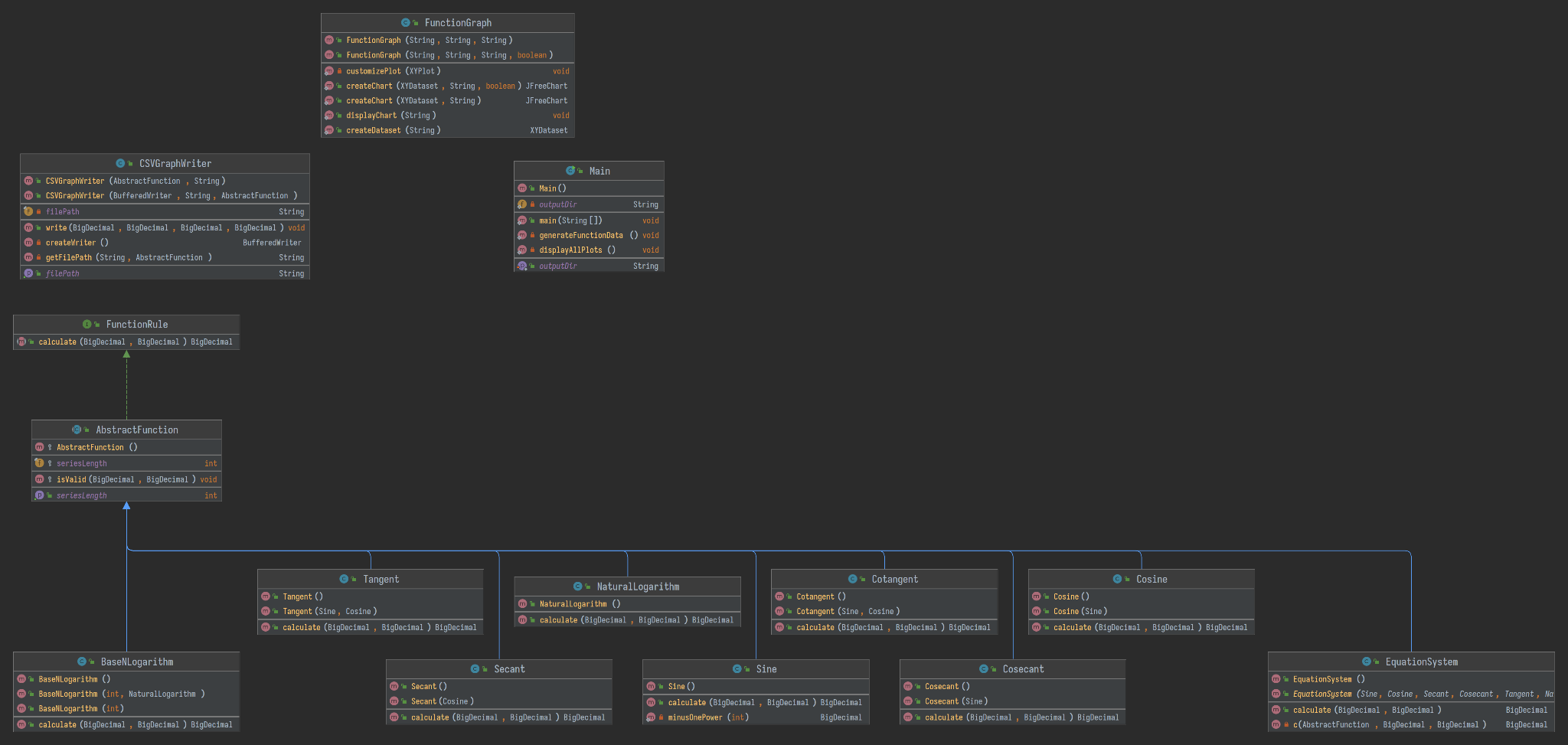
**Выполнение**

**Исходный код**



<https://github.com/maxbarsukov-itmo/tpo-2>

**UML**

****

**Описание тестового покрытия с обоснованием его выбора**

**Тригонометрические функции**

При тестировании тригонометрических функций необходимо учитывать их свойства, периодичность, симметрию и особенности поведения в особых точках. Выделим основные классы входных значений:

* **Обычные значения**: проверка произвольных точек внутри периода, например:

,,,0, π

* **Переодичнось**:

cos x, sin x, csc x, sec x — период 2π;

* **Особые точки (сингулярности):**
  + sec x не определён при x= + kπ;
  + csc x не определён при x= kπ;

**Логарифмические функции**

При тестировании логарифмических функций необходимо учитывать их свойства, область определения, особенности поведения в предельных точках числовую шкалу логарифмирования. Выделим основные классы входных значений:

* Обычные значения: проверка произвольных точек внутри области определения.
* Границы области определения:
  + Логарифмические функции определены для x > 0;
  + Проверяем отрицательные значения или 0.

**Система**

В рамках тестирования функциональности класса EquationSystem были разработаны следующие тестовые сценарии:

* **Обычные значения: проверка** произвольных точек внутри области определения, как позитивные, так и негативные.
* **Особые точки**: функция не определена, например, x = -π, x = 1

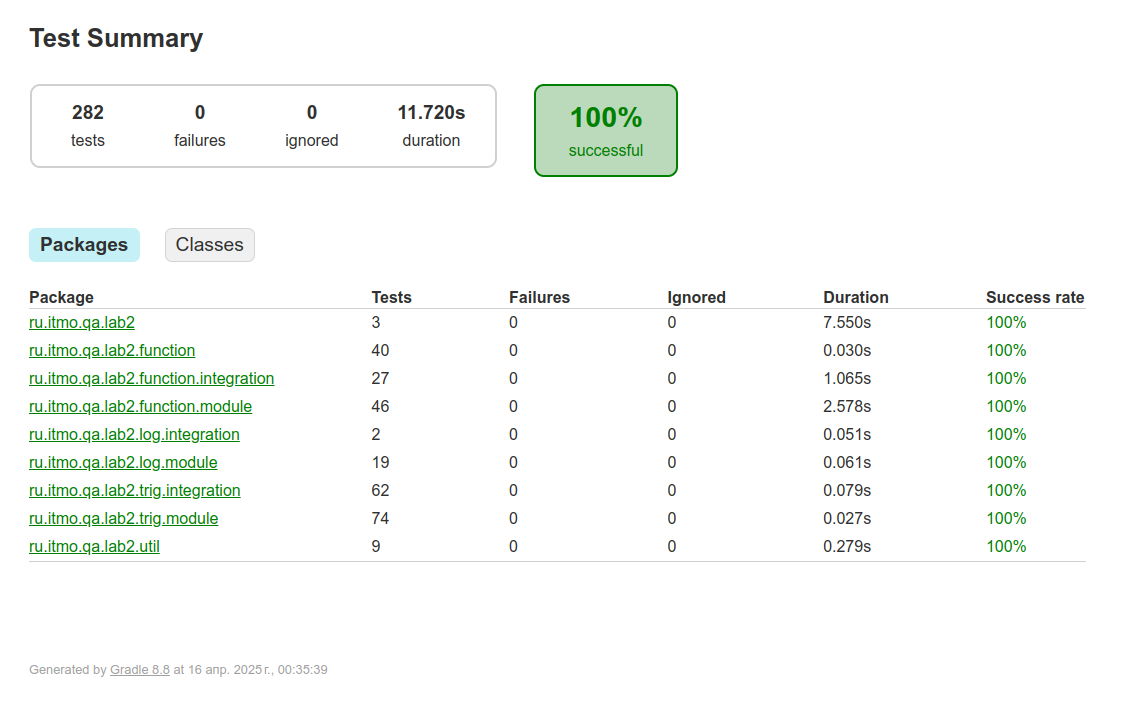
Также анализ функции показал, что часть системы для x > 0 постоянна (упрощается до константного значения), и поэтому может целиком представляться одной областью.

**Общее покрытие**

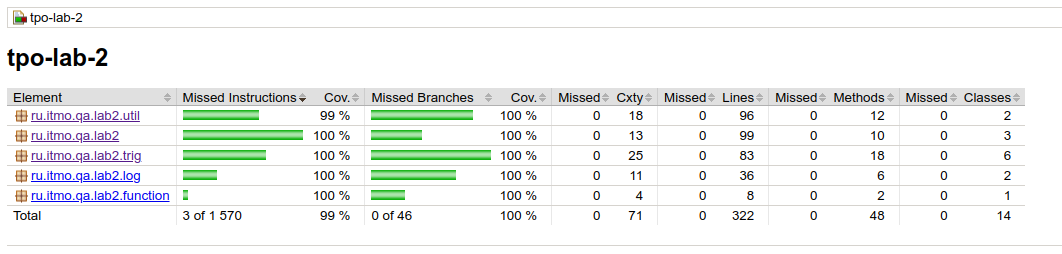
Также были добавлены тесты, проверяющие, некорректные аргументы, точность, и использование модулей (cos использует sin, согласно заданию).

**Тестовое покрытие**

Как мы можем видеть, все работает корректно:

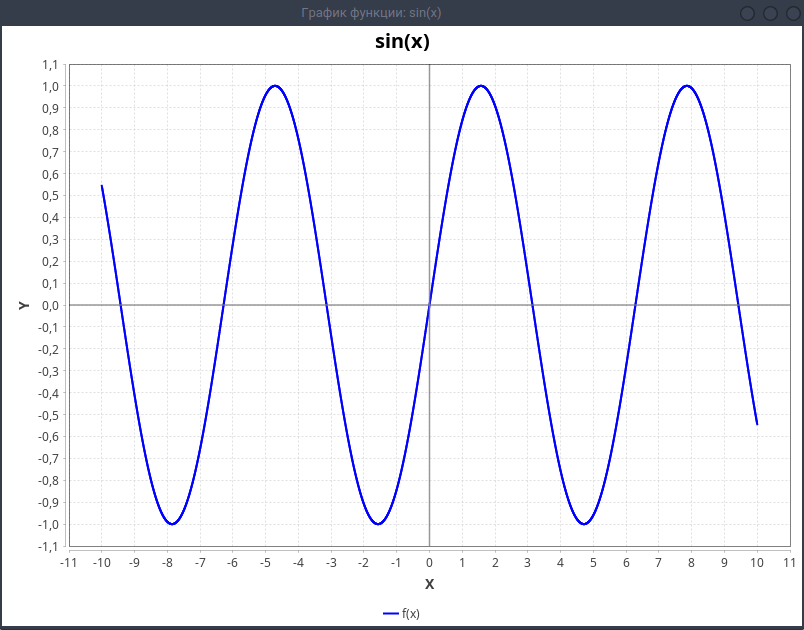
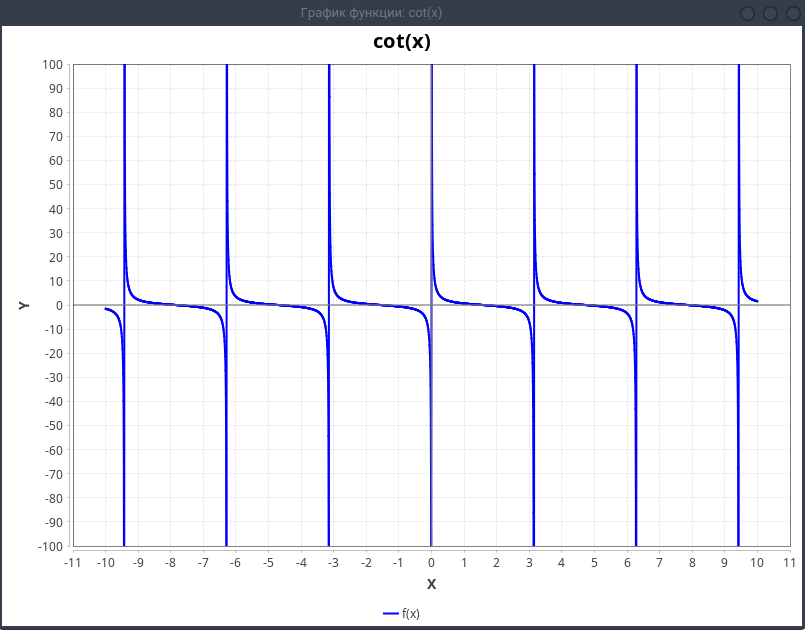
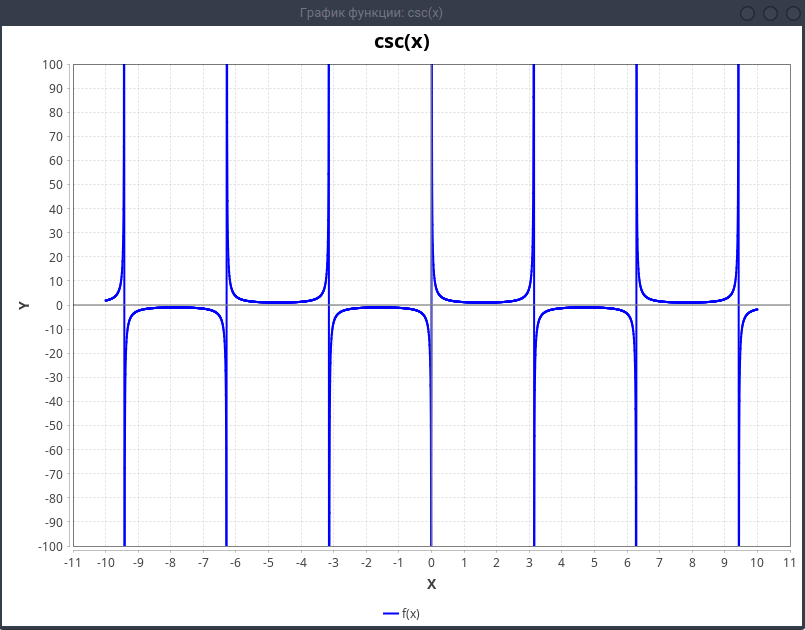
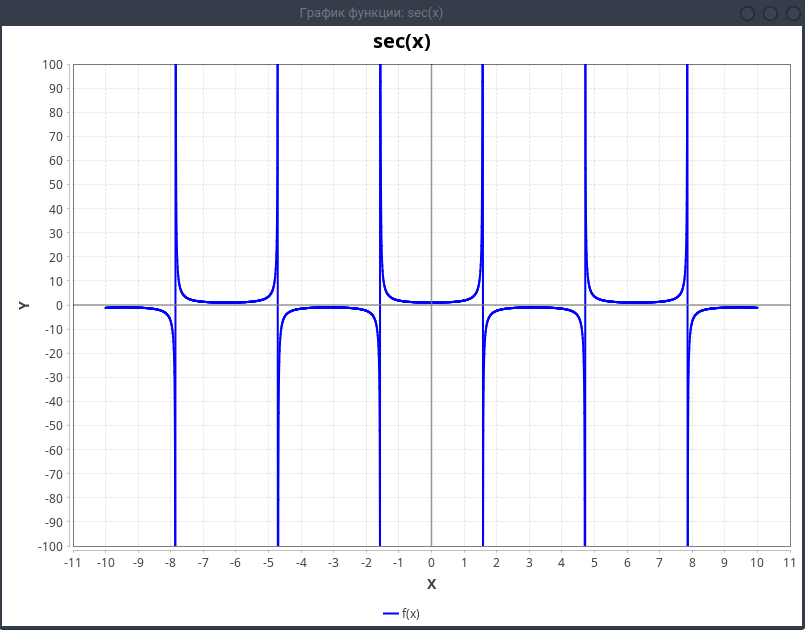
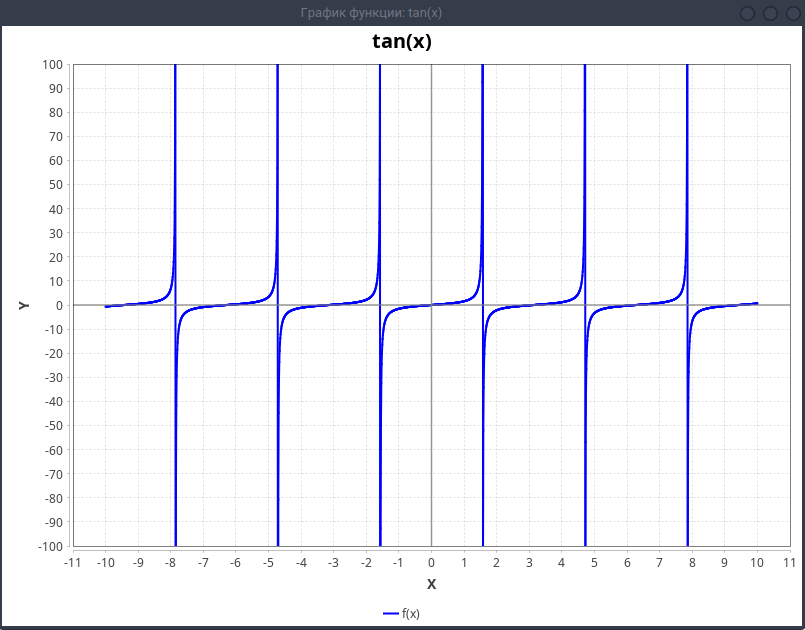
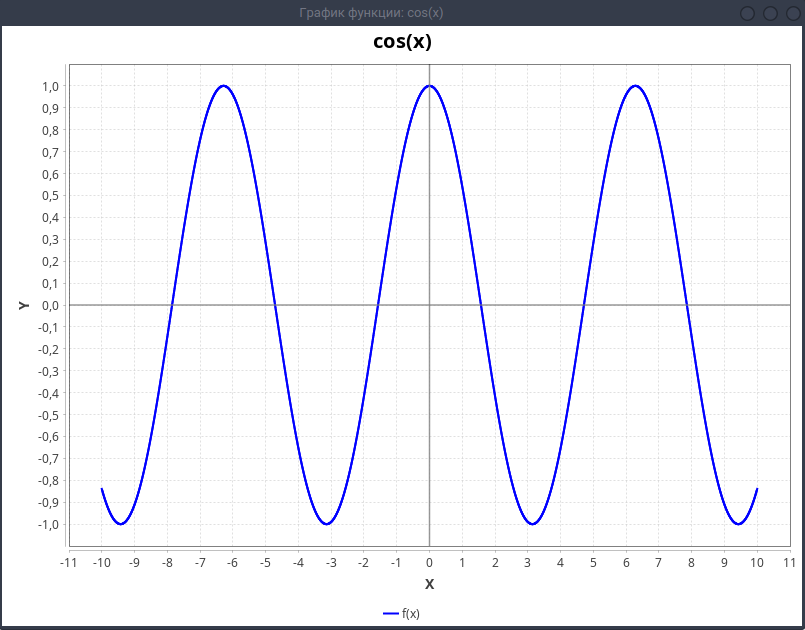


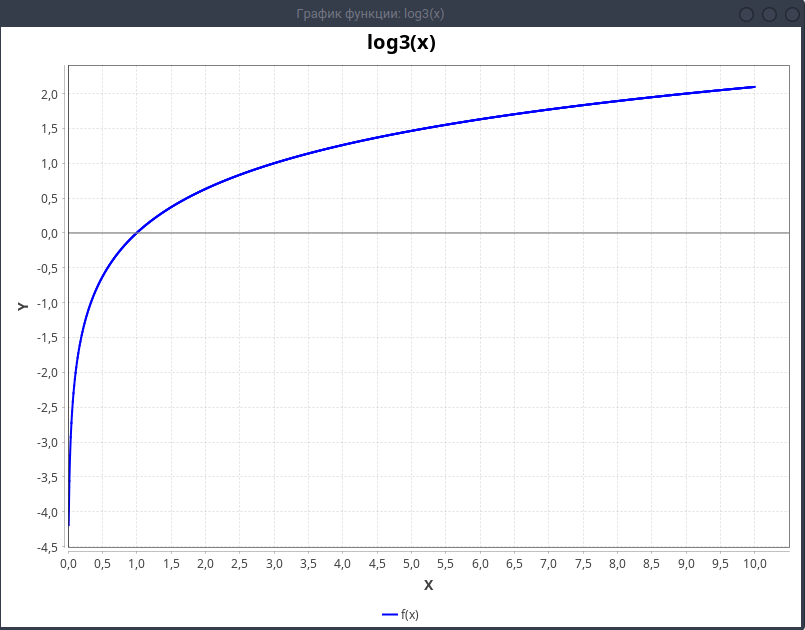
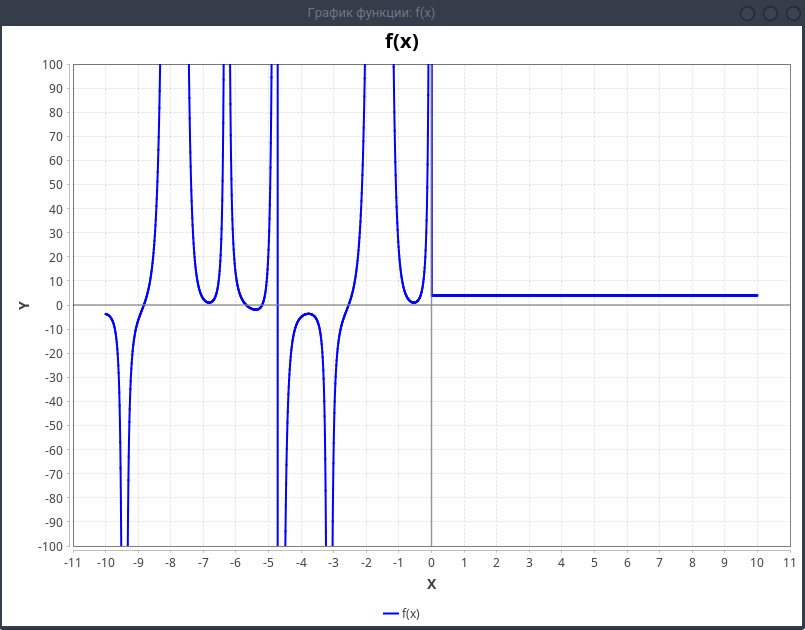
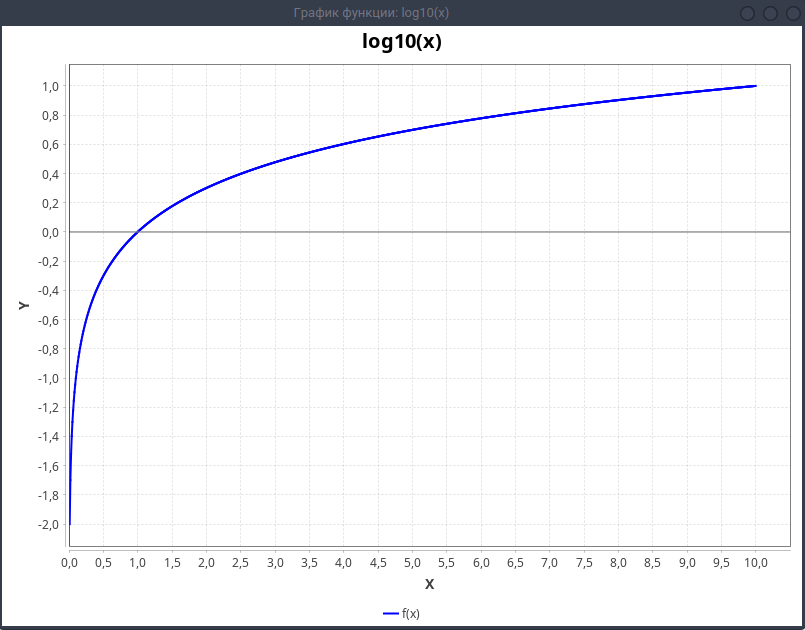
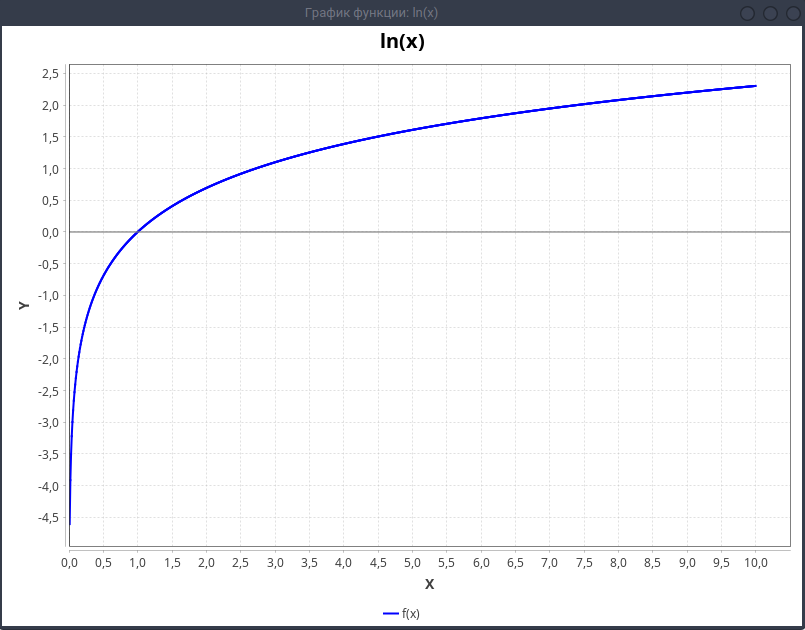
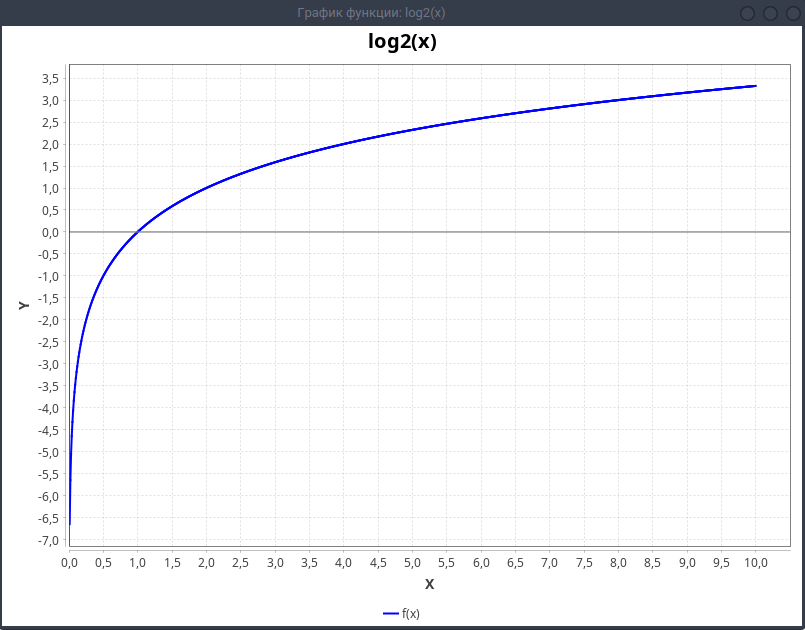
*Тесты успешно проходятся.*



*Итоговое тестовое покрытие.*

**Графики**





**Выводы**

В ходе выполнения лабораторной работы была реализована система функций,основанная на разложении в ряд. Проведено модульное и интеграционное тестирование с использованием JUnit 5 и Mockito, включая анализ граничных значений классов эквивалентности. Кроме того я научился тестировать приложения, работающие с дисплеем, получил базовые навыки CI. Выгрузка данных в CSV построенные графики подтвердили корректность вычислений. Работа позволила закрепить навыки интеграционного тестирования, работы с численными методами и проектирования модульных приложений.