

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики»

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНЫХ
ТЕХНОЛОГИЙ

Лабораторная работа №2
по дисциплине
«Тестирование программного обеспечения»
Вариант 46465456

Выполнил:
Векшин Арсений Иванович Р3316

Преподаватель:
Ермаков Михаил Константинович

Текст задания	3
Система функций	4
Ссылка на репозиторий	4
UML-диаграммы классов	5
Описание тестового покрытия	6
Графики	7
Вывод	13

Текст задания

Лабораторная работа #2

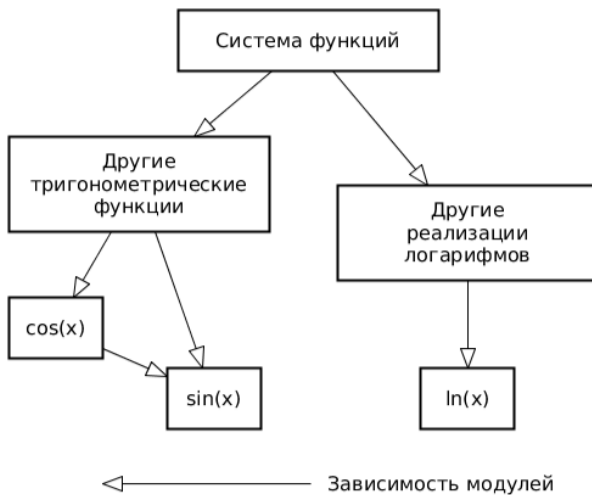
Провести интеграционное тестирование программы, осуществляющей вычисление системы функций (в соответствии с вариантом).

Введите вариант: 46465456

[illegible]

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.

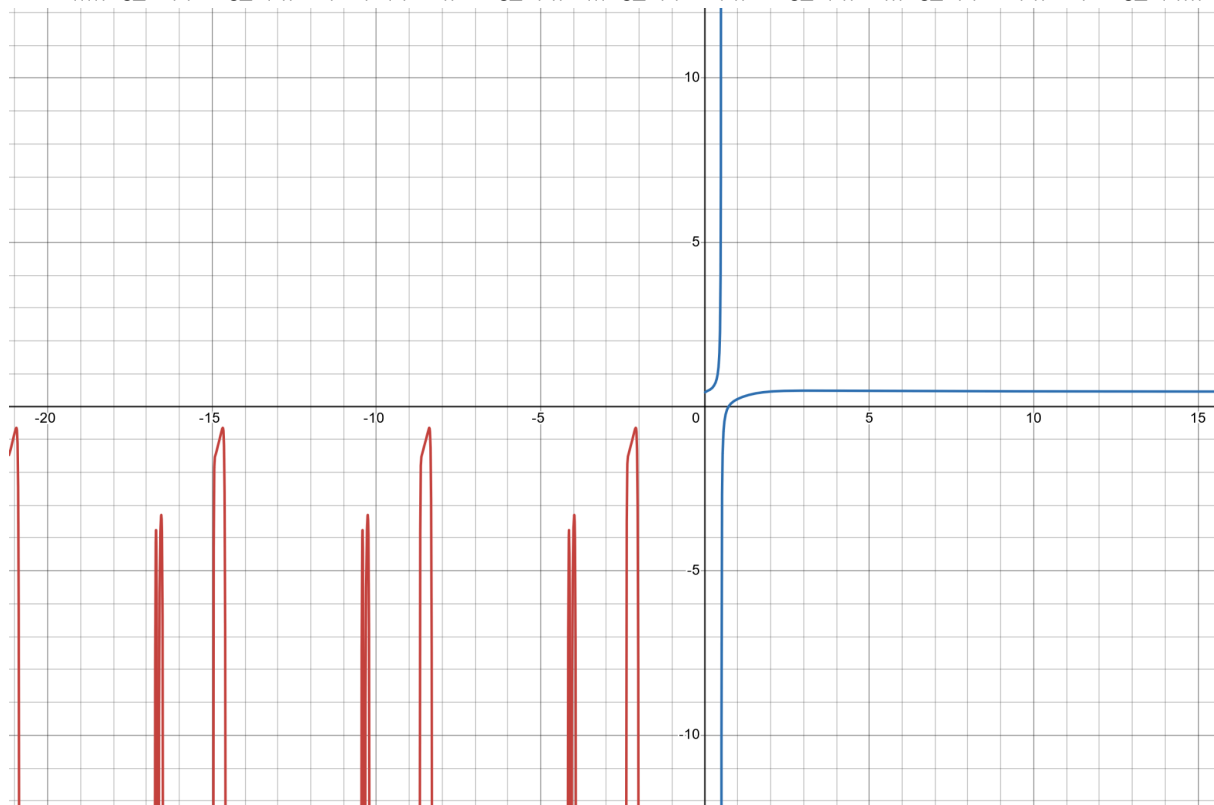
Порядок выполнения работы:

1. Разработать приложение, руководствуясь приведёнными выше правилами.
2. С помощью JUNIT4 разработать тестовое покрытие системы функций, проводя 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.

Система функций

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

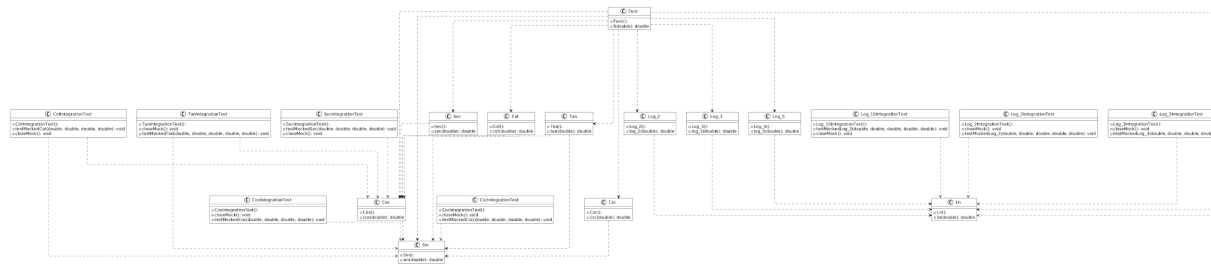
$x > 0 : (((((\log_{10}(x) / \log_5(x)) ^ 3) + (\ln(x) ^ 3)) + \log_3(x)) / (((\log_2(x) / \ln(x)) + \log_3(x)) + (((\log_2(x) * \ln(x)) ^ 2) / \log_3(x))))$



Ссылка на репозиторий

<https://github.com/ArsenyVekshin/ITMO/tree/master/Testing/lab2>

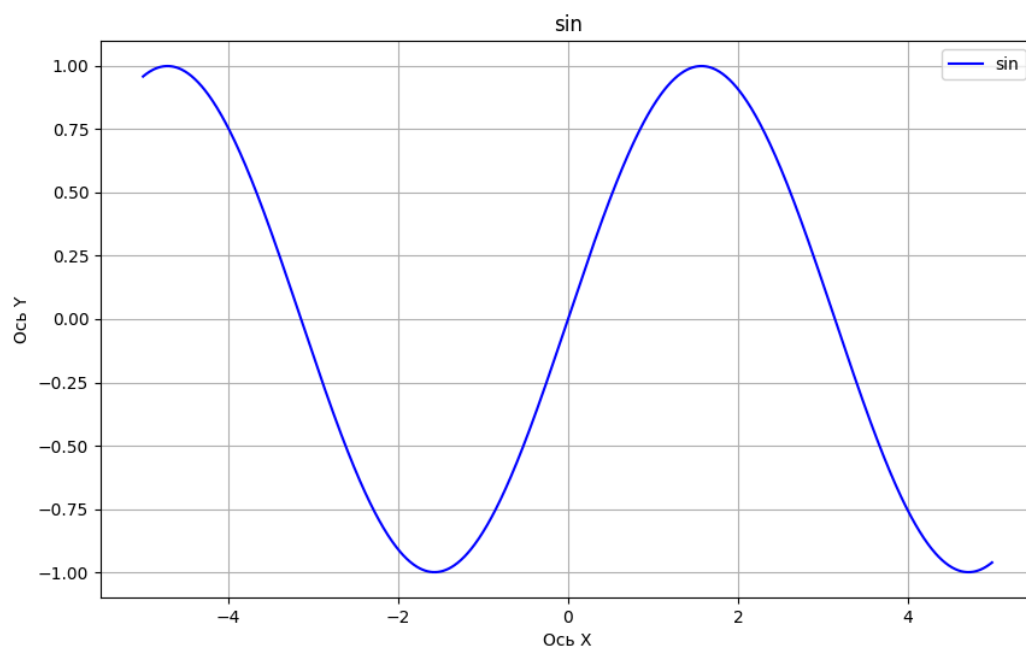
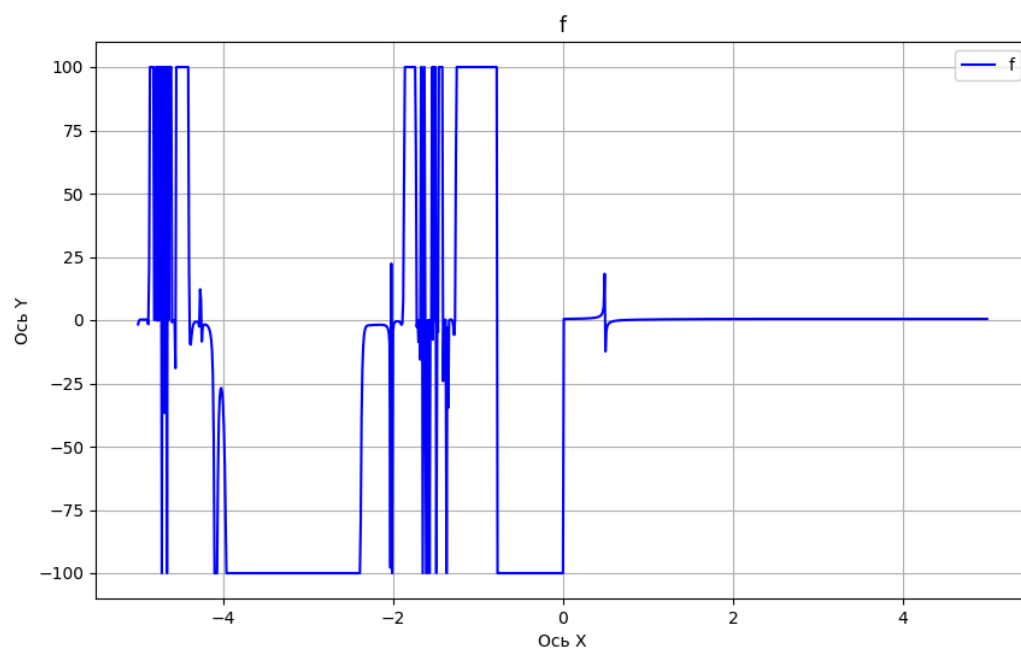
UML-диаграмма классов

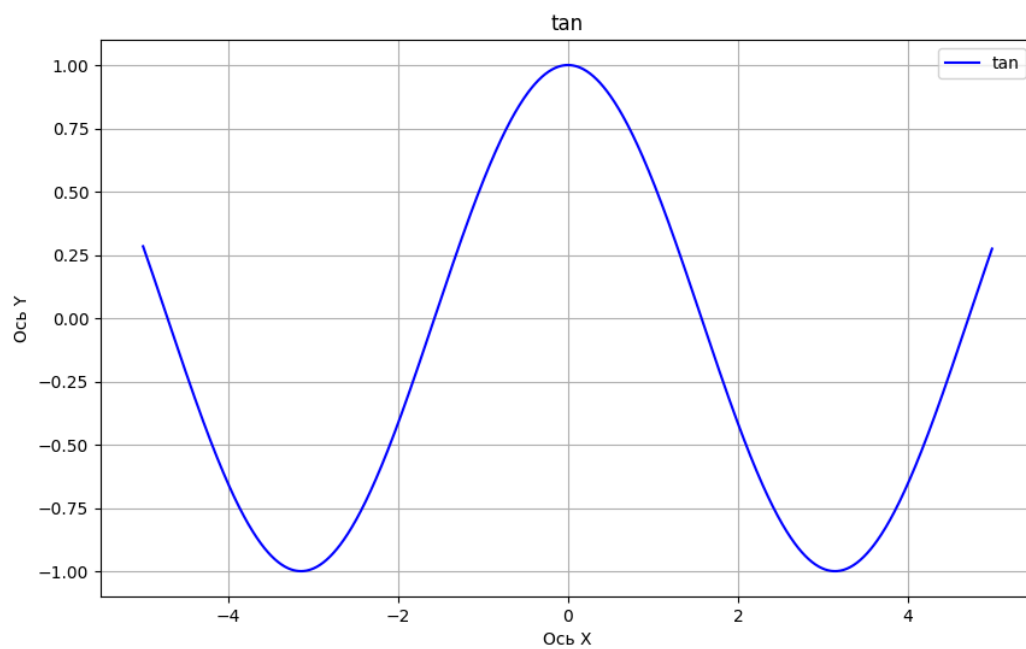
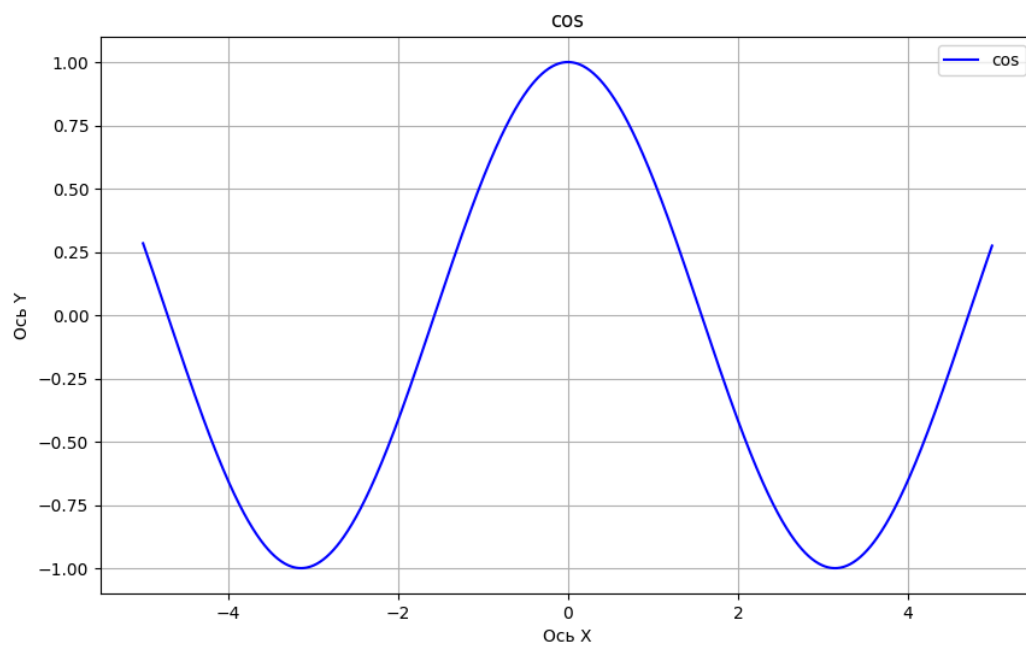


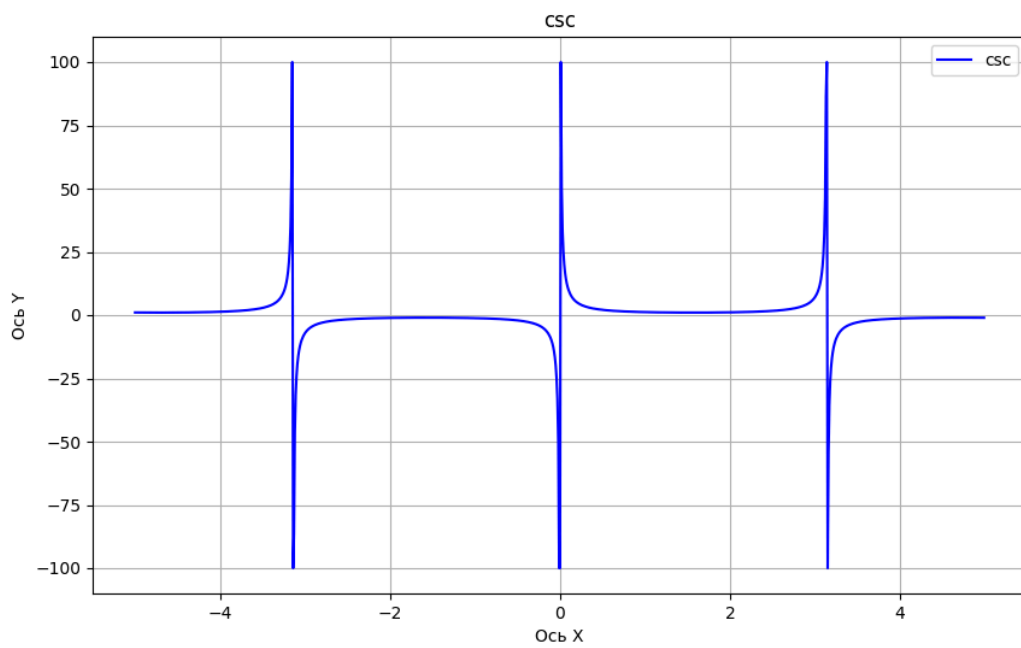
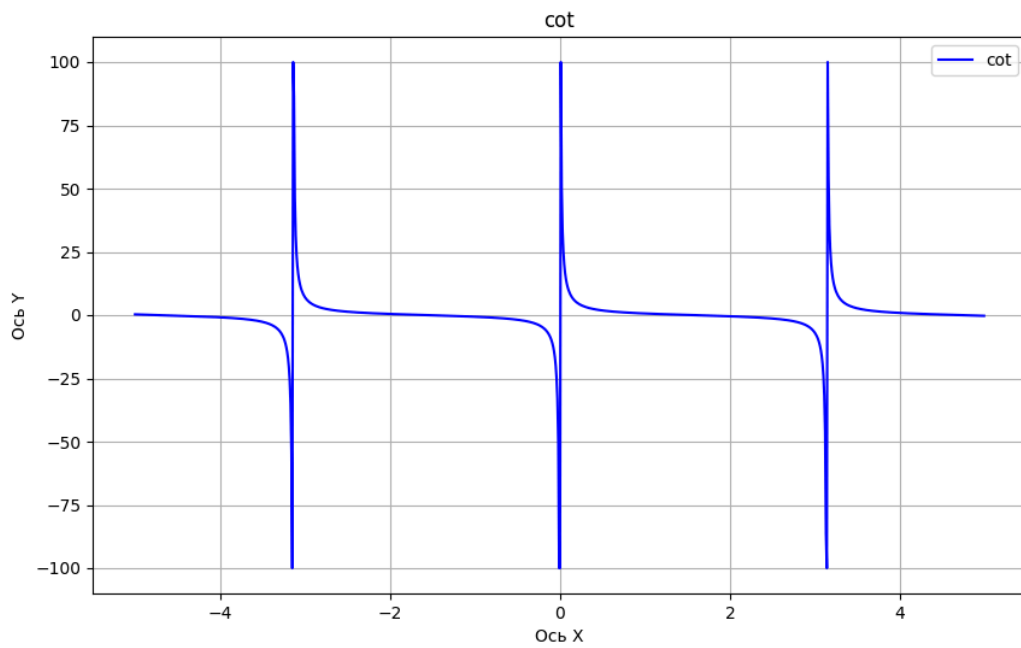
Описание тестового покрытия

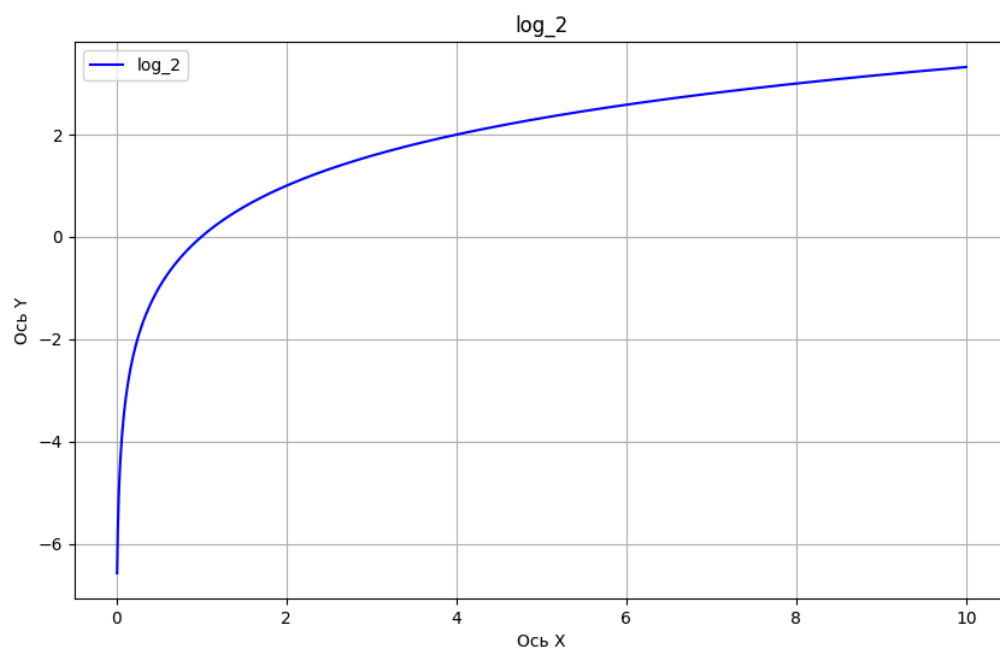
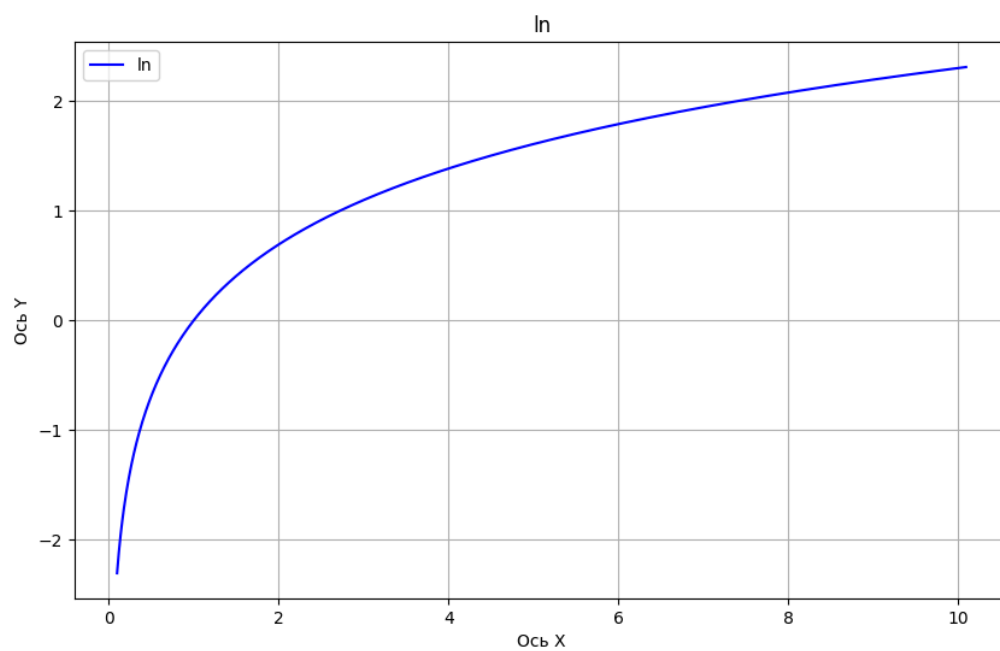
Тестовое покрытие было выбрано так, чтобы в первую очередь проверить проблемные точки в системе функций. Проанализировав их поведения были выявлены критические точки, точки вероятных ошибок, включая ограничения самих примитивных функций.

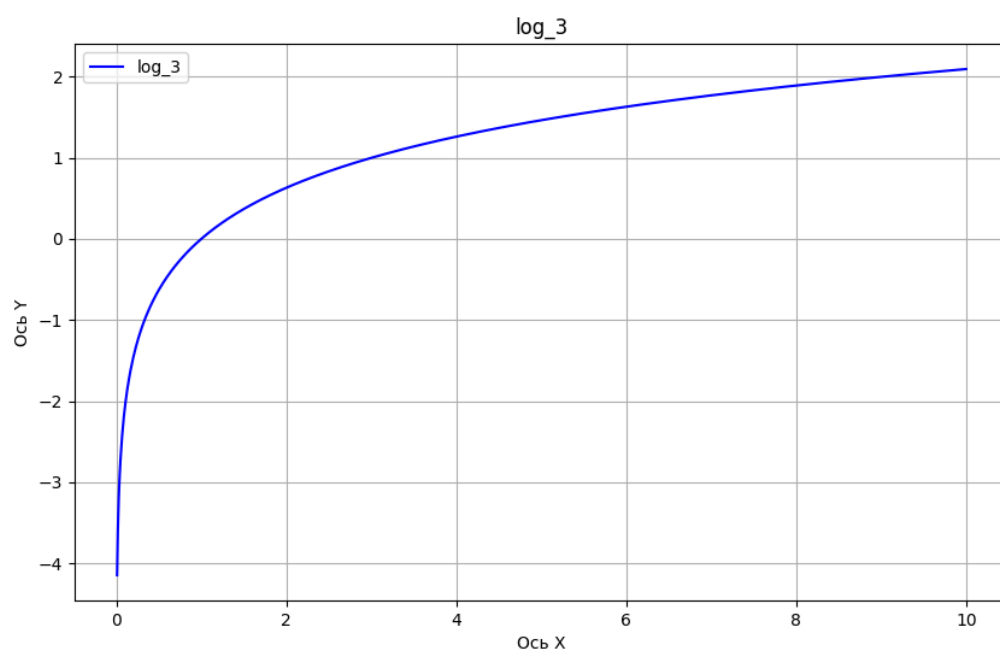
Графики

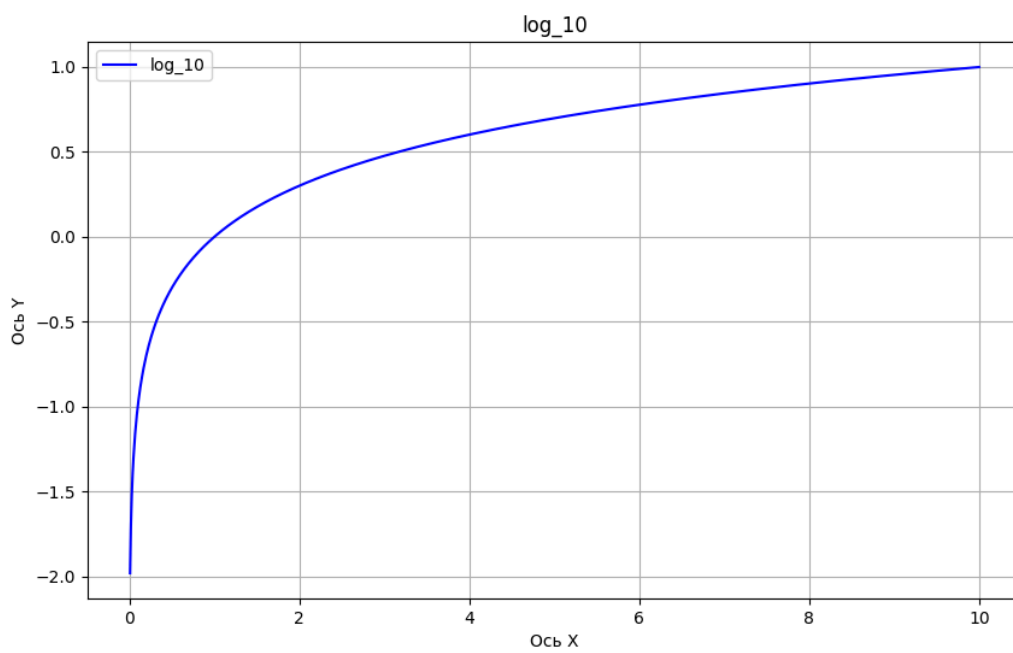
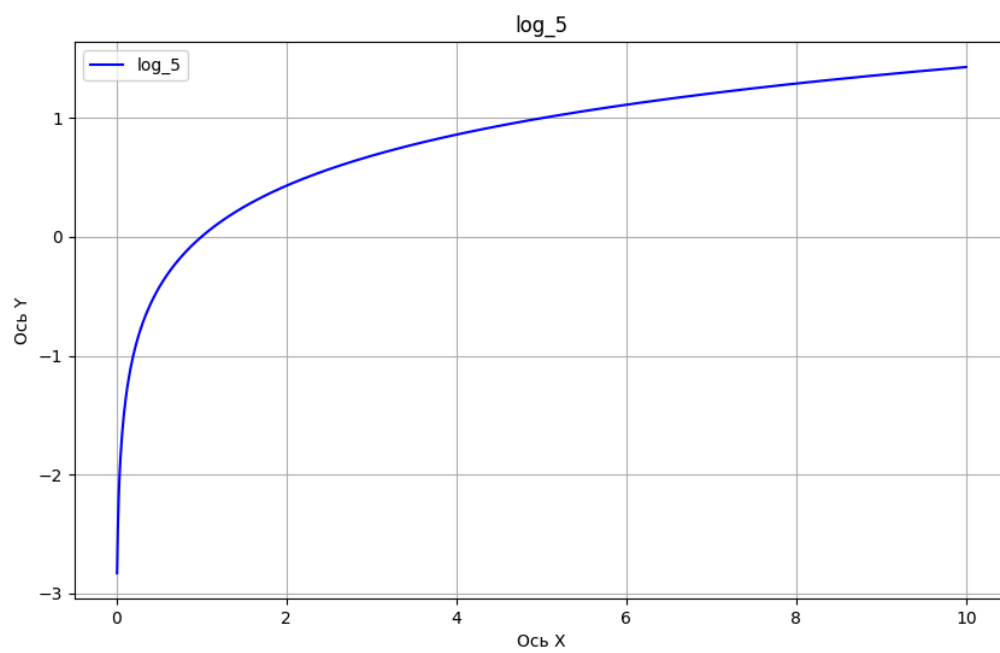












Вывод

