НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

**Мегафакультет:** Компьютерных технологий и управления

**Направление:** 09.03.04 «Программная инженерия»

**Лабораторная работа №2**

По дисциплине:

«Тестирование ПО»

**Вариант 33160**

**Выполнила:**

студентка группы **P33112**

Маньшина Елена

Витальевна

**Преподаватель:**

**Харитонова Анастасия**

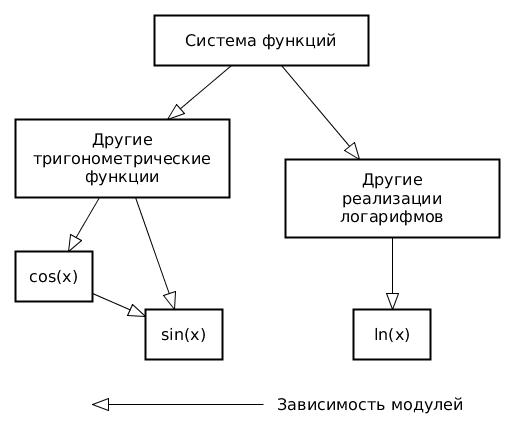
**Евгеньевна**

г. Санкт-Петербург

2021

**Задание**

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

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 сsv файл вида «X, Результаты модуля (X)», позволяющее произвольно менять шаг наращивания Х. Разделитель в файле csv можно использовать произвольный.

Порядок выполнения работы:

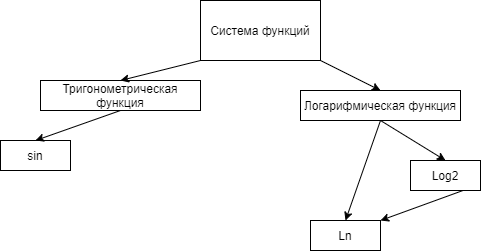
1. Разработать приложение, руководствуясь приведёнными выше правилами.

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/>.

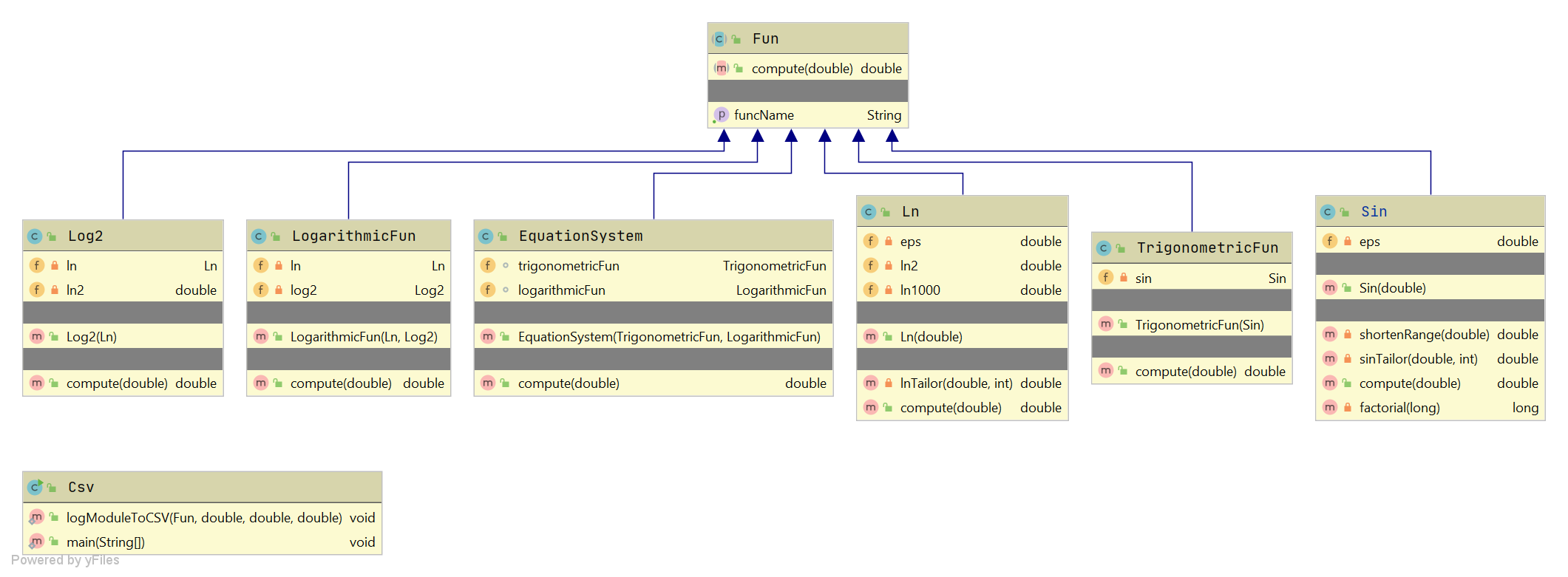
3. Собрать приложение, состоящее из заглушек. Провести интеграцию приложения по 1 модулю, с обоснованием стратегии интеграции, проведением интеграционных тестов и контролем тестового покрытия системы функций.

Заданная система уравнений

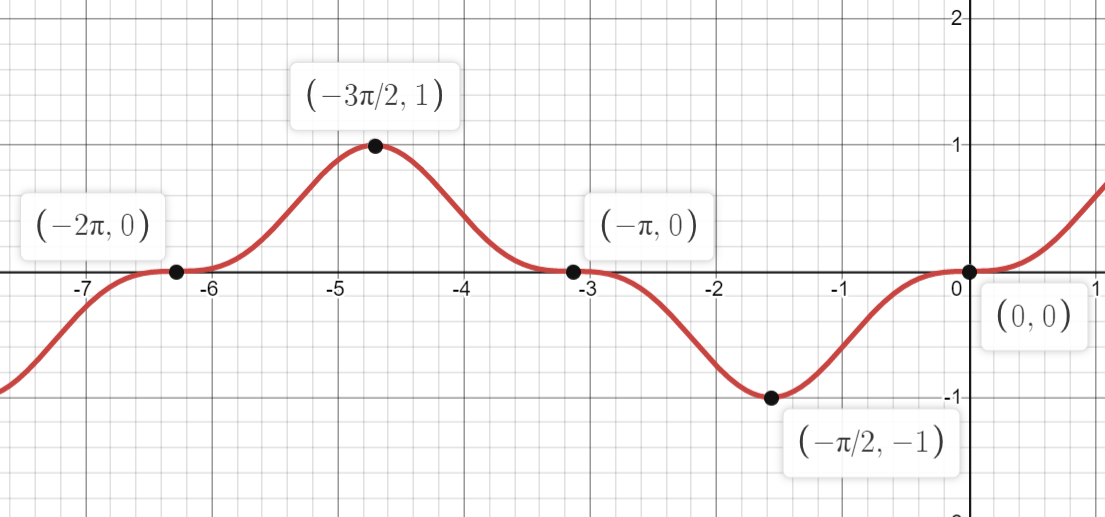
Диаграмма связи модулей

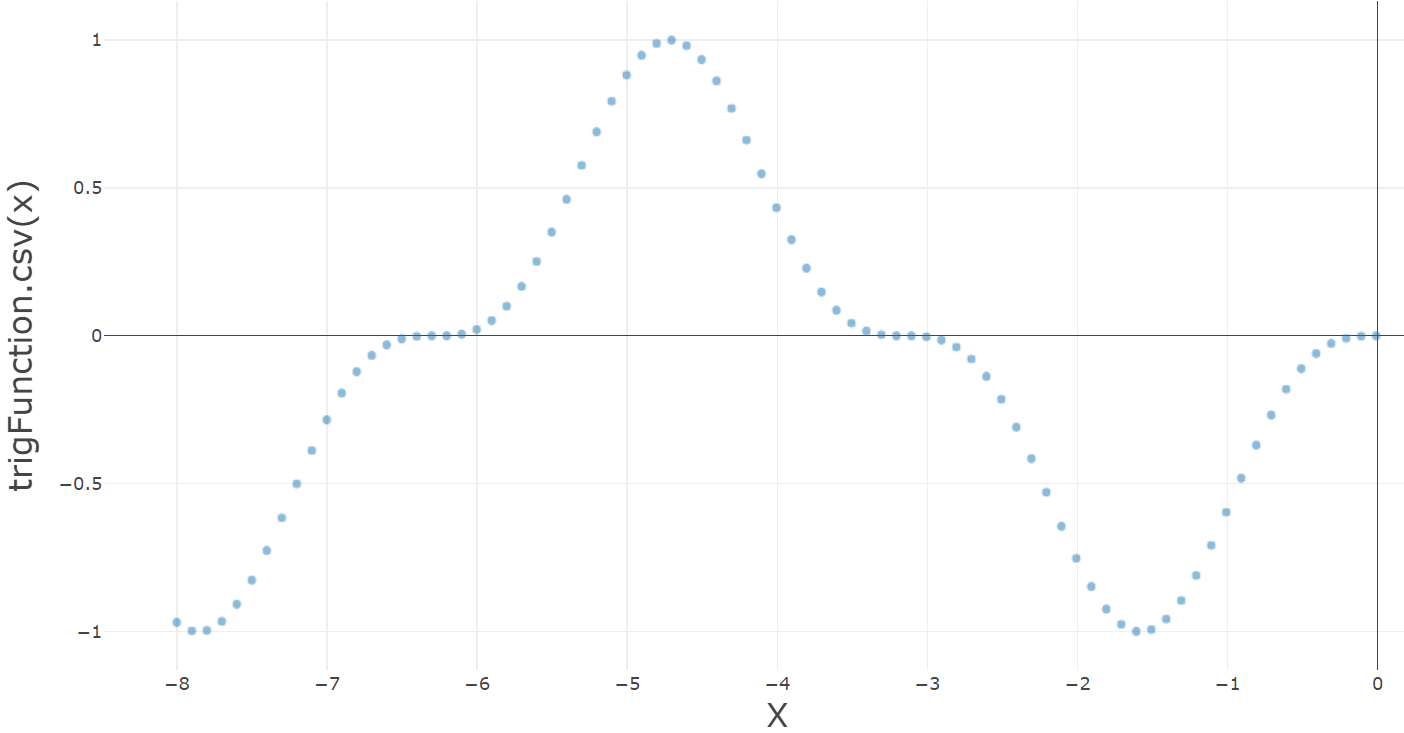


UML-диаграмма классов



**Графики**

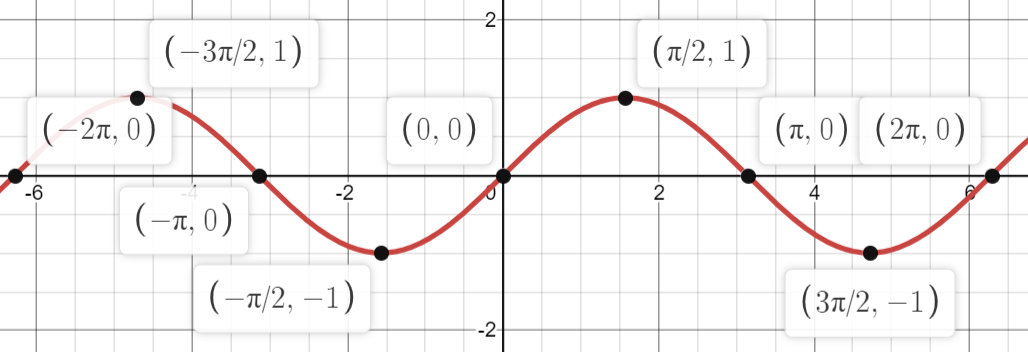


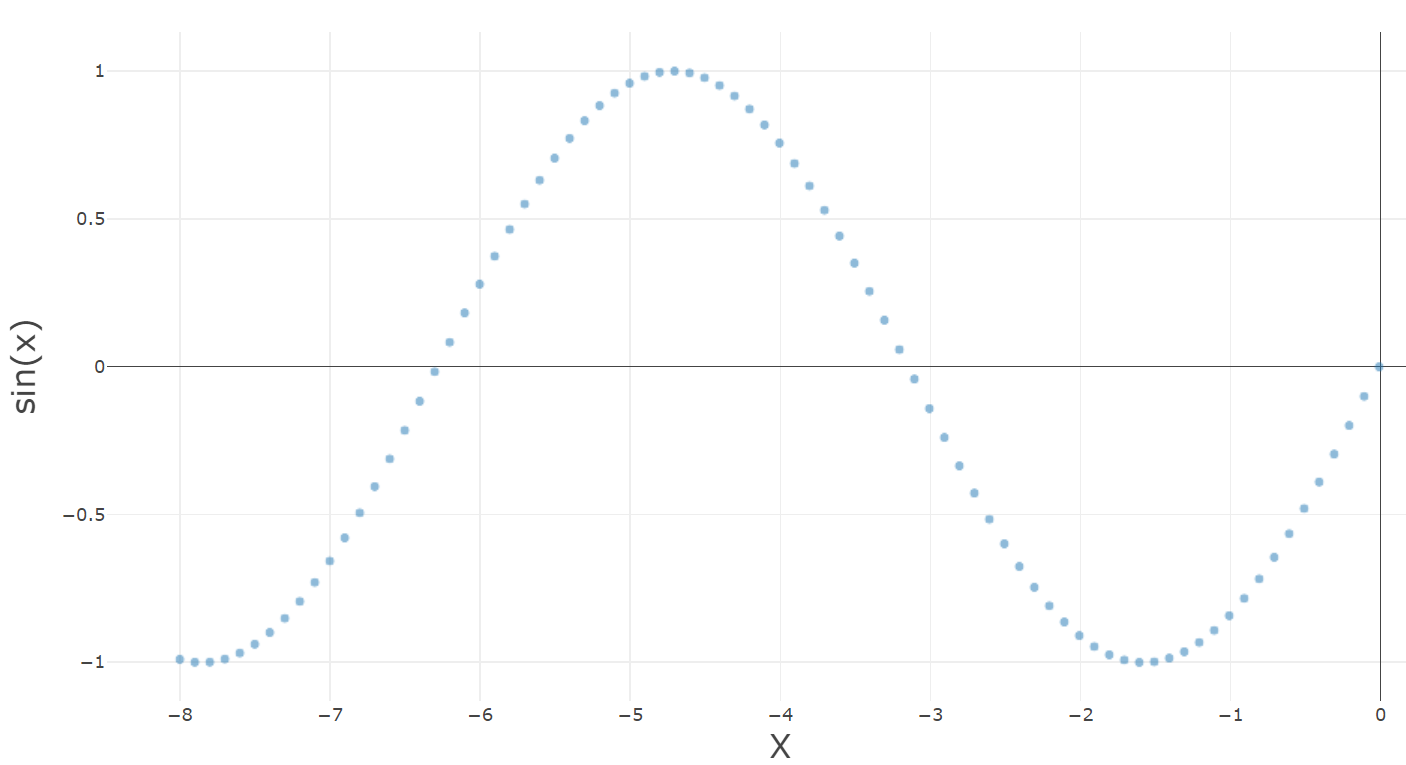


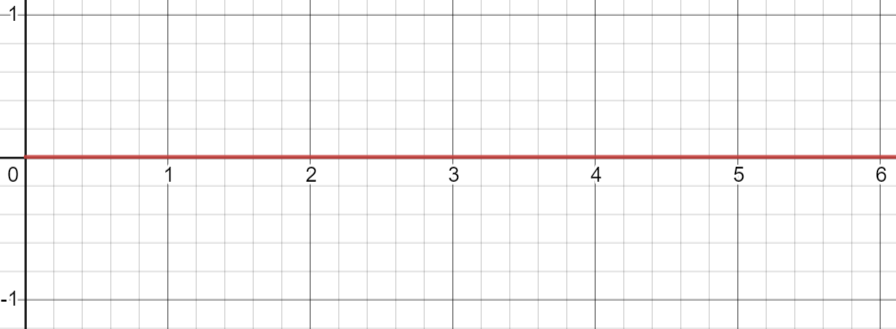
Необходимо выделить классы эквивалентности, а также краевые точки.

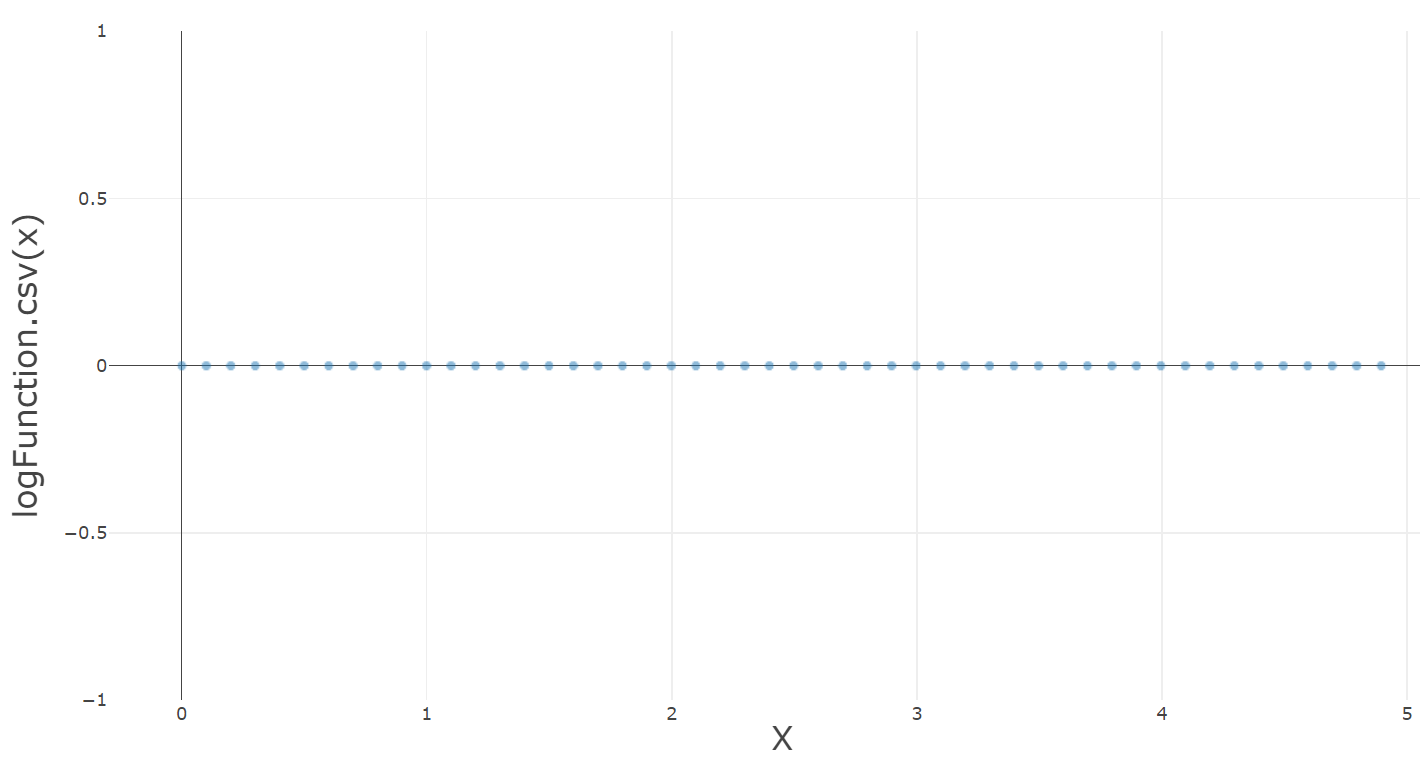
* Функция периодична, т.е 𝑦(𝑥) = 𝑦(𝑥 + 2𝜋𝑛)
* Определена на всем промежутке
* Для простоты возьмем промежуток 𝑥 𝜖 [-2𝜋; 0]

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |



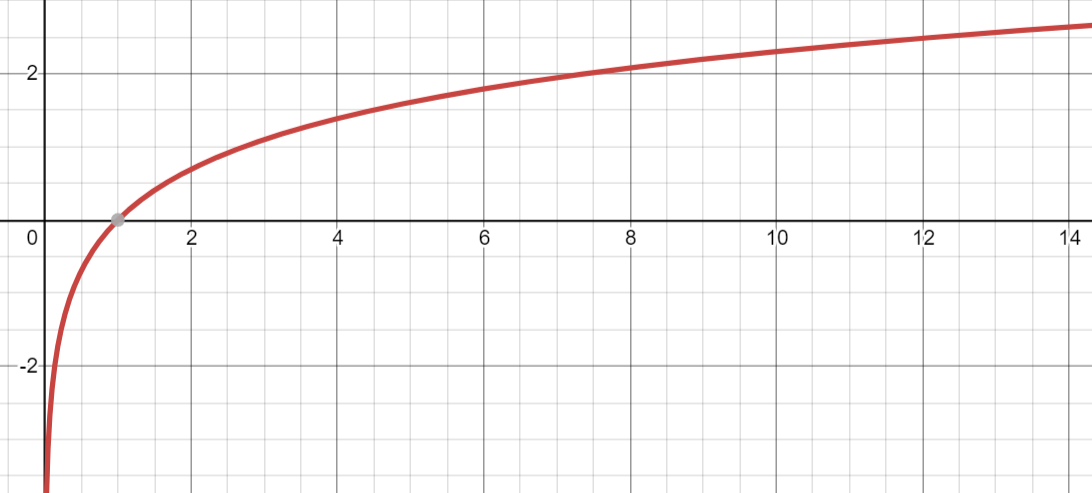


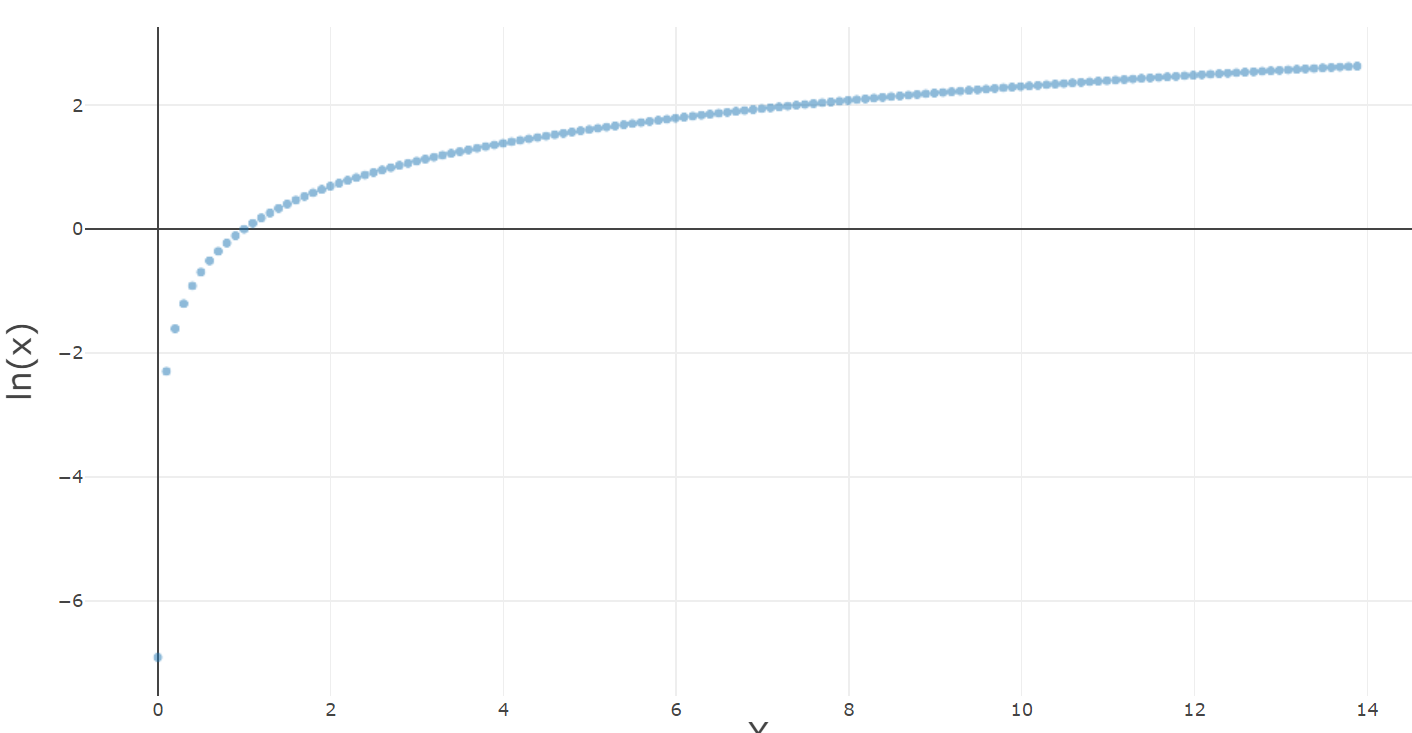




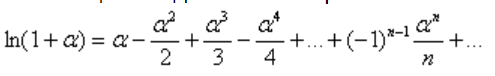
Необходимо выделить классы эквивалентности, а также краевые точки.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  | IllegalArgumentException |
|  |  |  | ArithmeticException  (деление на 0) |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |





Получим разложение lnx из разложения ln(a+1)



Перепишем lnx как ln((x-1)+1) , тогда a = x-1 и получим

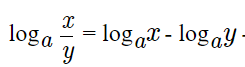
Согласно таблице ряд ln(a+1) сходится при

Тогда и

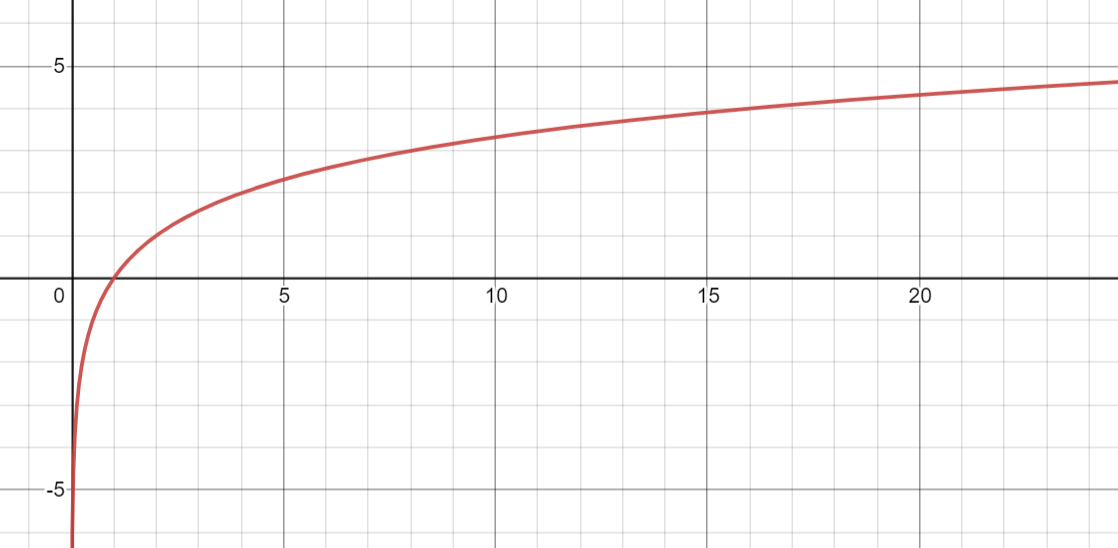
Проверим для граничных значений ln0=

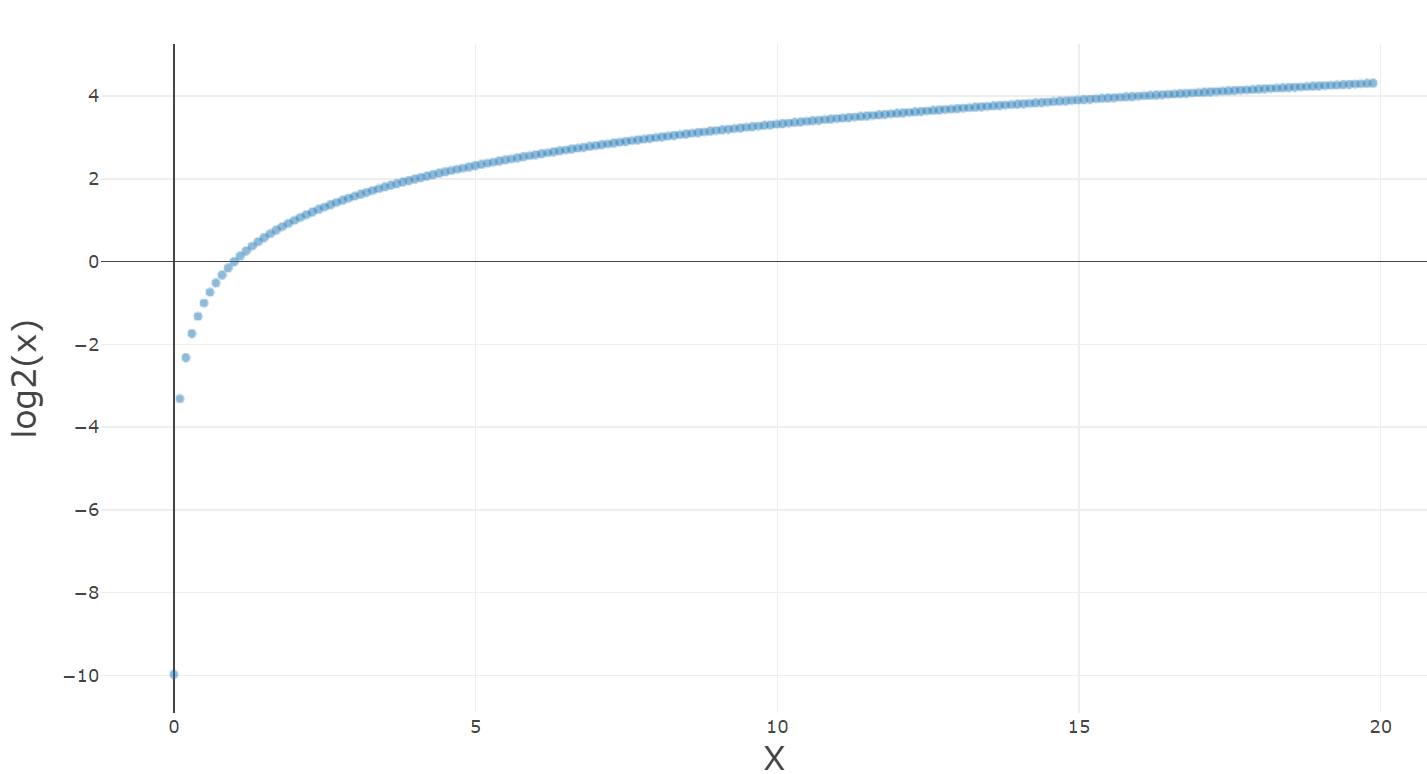
Теперь для х=2. Сумма получившегося знакочередующего ряда в точности равна ln2.





|  |  |
| --- | --- |
|  |  |
|  |  |
|  | |
|  | -2.30258 |
| 0.5 | -0.69315 |
| Пересечение с осью абсцисс | |
| 1 | 0 |
|  | |
| 1.4 | 0.33647 |
| 2 | 0.69314 |
|  | |
| 3 | 1.098612 |
| 100 | 4.60517 |





|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Описание тестового покрытия**

При разработке я действовала на основании стратегии сверху-вниз.

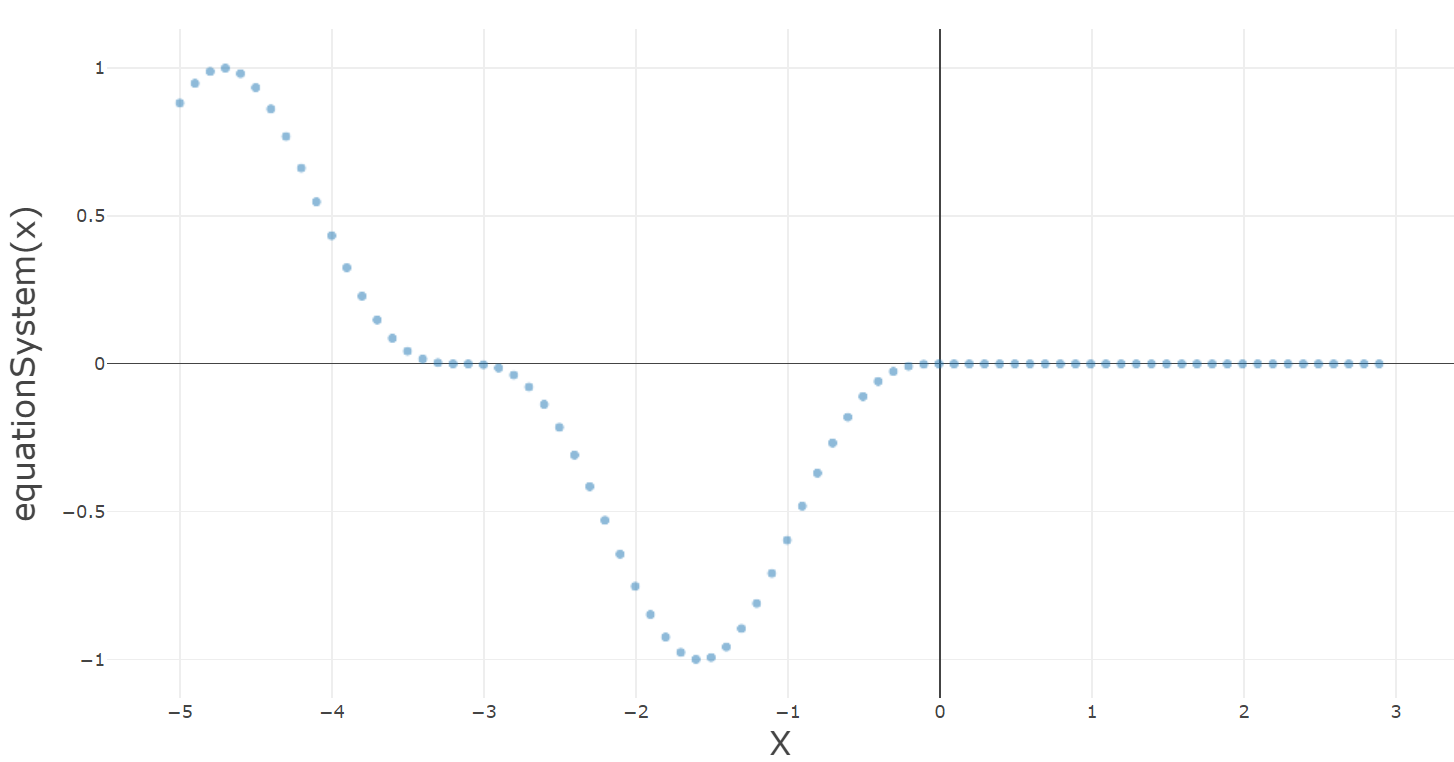
Первым был реализован модуль системы функций, в котором осуществляется выбор тригонометрической или логарифмической функции исходя из значения х (неположительный или положительный). Для него были написаны интеграционные тесты с заглушками для тригонометрической и логарифмической функций.

Также на уровне итоговой системы функций было проведено интеграционное тестирование без использования заглушек, а с применением итоговых реализаций модулей.

Затем были реализованы по отдельности модули самих тригонометрической и логарифмической функций, здесь заглушки стояли для функции синуса, натурального логарифма и логарифма по основанию два.

Затем было проведено модульное тестирование базовых функции синуса, натурального логарифма и логарифма по основанию два.

**График итоговой системы функций**

**

Исходный код: https://github.com/Lenika2000/Testing-Lab-2

Вывод: В ходе лабораторной работы я еще раз проделала модульное тестирование и познакомилась с Moсkito, облегчающем интеграционное тестирование.