

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Мегафакультет: Компьютерных технологий и управления

Направление: 09.03.04 «Программная инженерия»

Лабораторная работа №2

По дисциплине:

«Тестирование программного обеспечения»

Вариант 34159

Выполнила:

студентка группы **Р33112**

Корнишова Евгения
Александровна

Преподаватель:

Харитоновна Анастасия
Евгеньевна

г. Санкт-Петербург

2021 г.

Текст задания:

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

Система функций:

$$\begin{cases} ((\cot(x) - \sin(x)) \cdot \cos(x)) & \text{if } x \leq 0 \\ \left(\left(\frac{\frac{\log_5(x) + \log_3(x)}{\log_2(x)^2}}{\frac{\ln(x) - \log_2(x)}{\log_3(x)}} \right) + \left(\frac{\ln(x)}{\log_{10}(x) + \ln(x)} \right) \right) & \text{if } x > 0 \end{cases}$$

$x \leq 0: ((\cot(x) - \sin(x)) * \cos(x))$

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

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 $\sin(x)$):



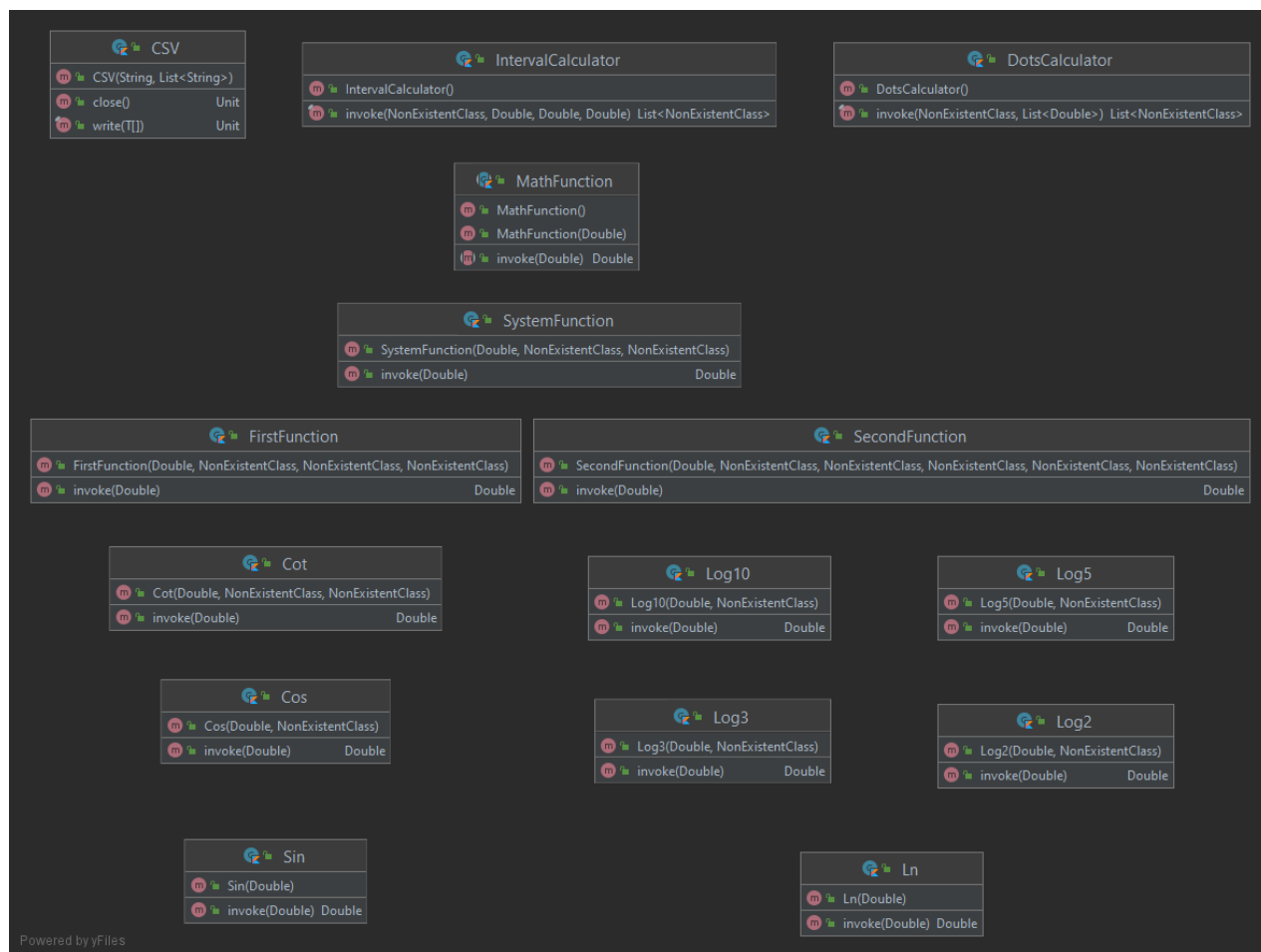
3. Обе "базовые" функции (в примере выше - $\sin(x)$ и $\ln(x)$) должны быть реализованы при помощи разложения в ряд с задаваемой погрешностью. Использовать тригонометрические/логарифмические преобразования для упрощения функций ЗАПРЕЩЕНО.

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.

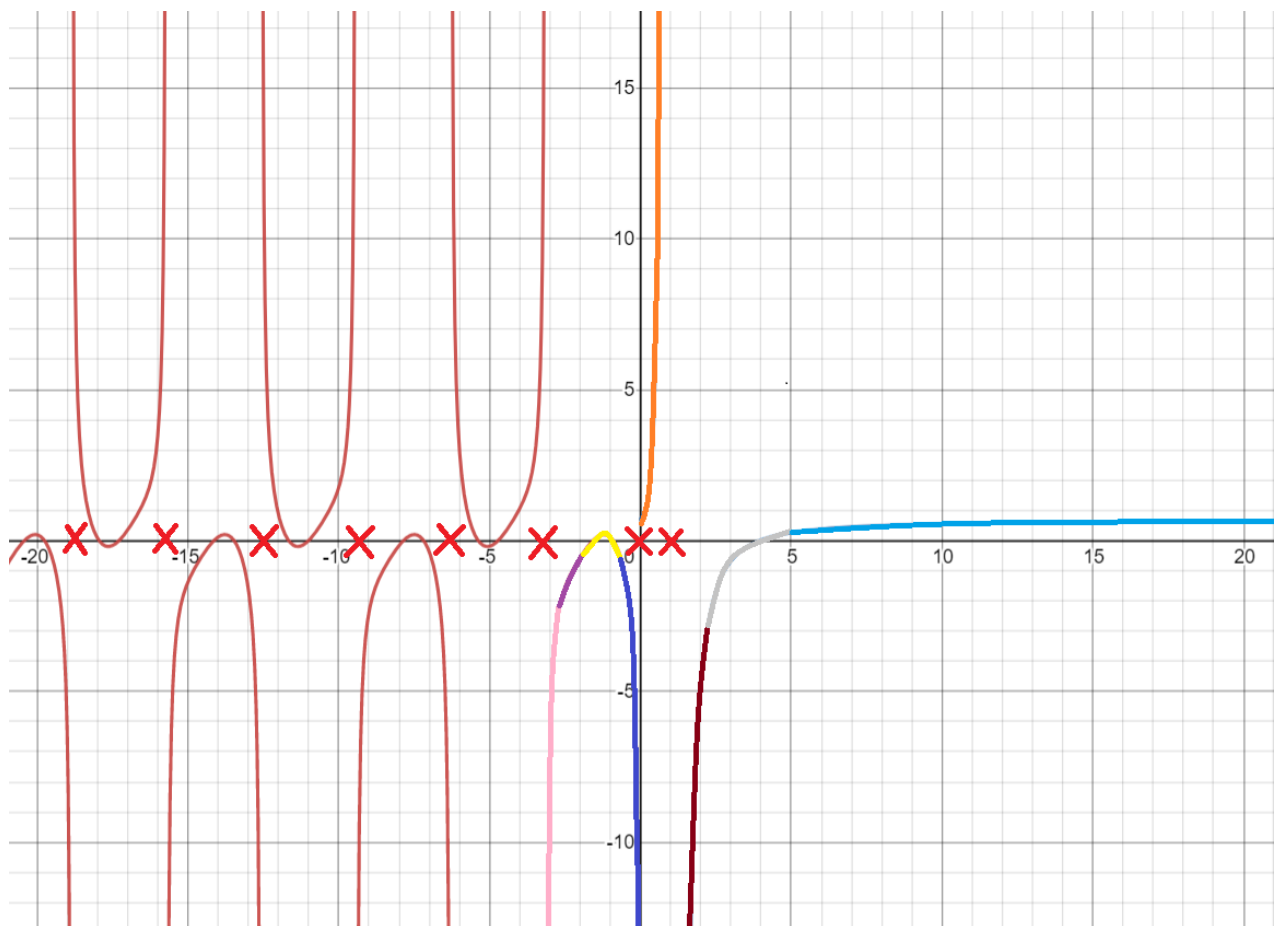
UML-диаграмма классов разработанного приложения:



Описание тестового покрытия с обоснованием его выбора:

Правая функция ($x > 0$) была разбита на четыре партии эквивалентности, внутри каждой из которых было выбрано по две точки для тестирования, еще одной точкой для тестирования стала 1, не входящая в ОДЗ функции.

Левая функция ($x \leq 0$) периодическая, причем ведет себя подобным образом относительно центра периода, который также был разбит на четыре партии эквивалентности (по одной точке для тестирования в каждой). Дополнительные точки для тестирования: 0, $-\pi$, -2π (не входят в ОДЗ) и $-\frac{5\pi}{3}$ (точка за пределами первого полупериода, на котором производятся все расчеты тригонометрических функций при помощи приведения к нему всех остальных).



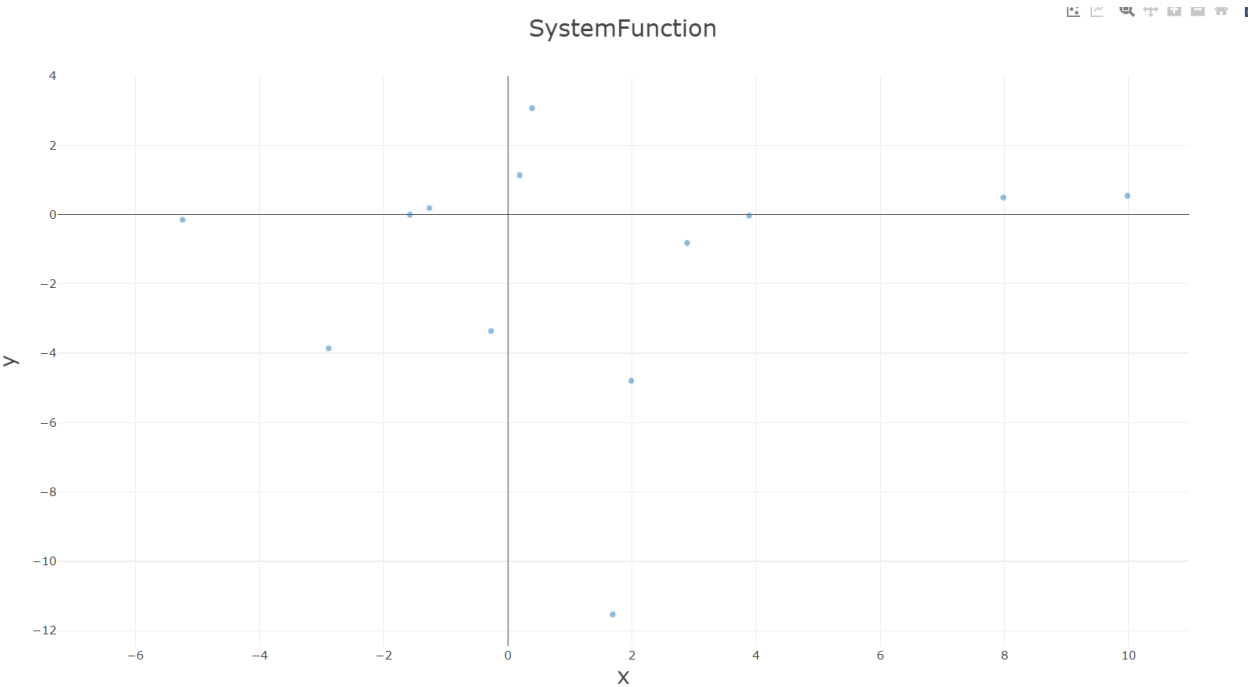
Графики, построенные csv-выгрузкам, полученным в процессе интеграции приложения:

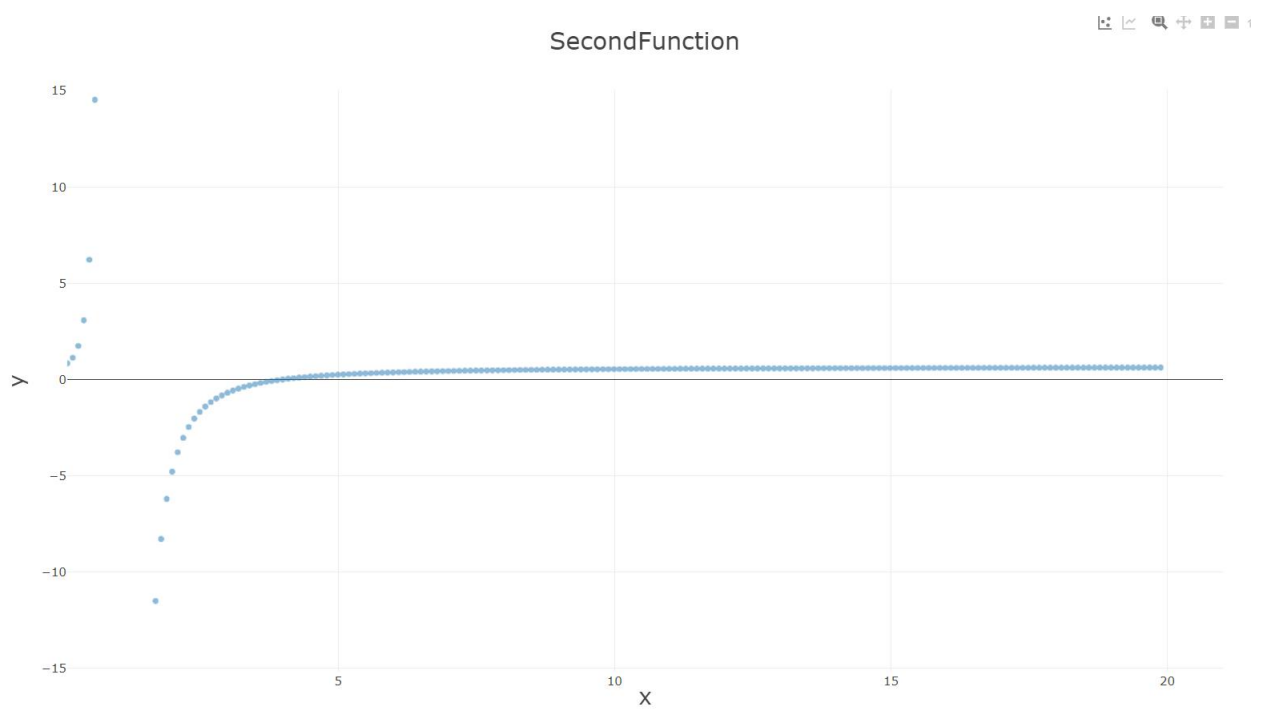
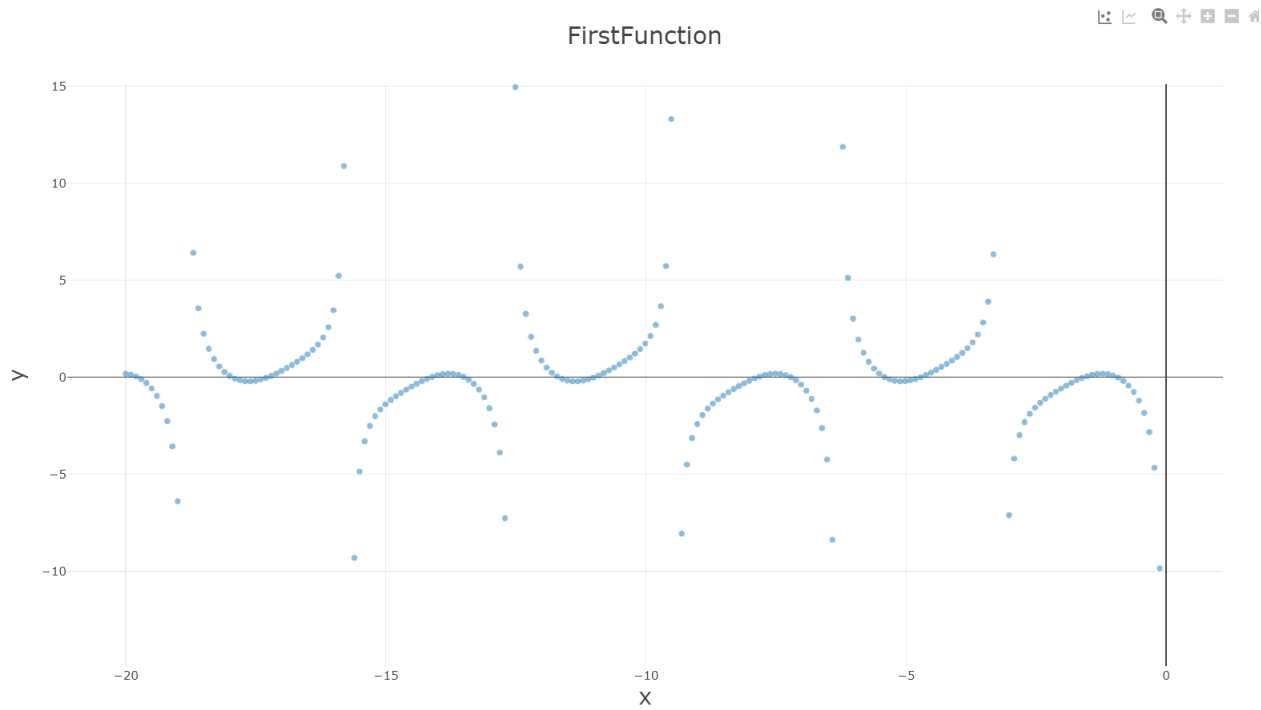
Используемая стратегия интеграции — сверху вниз.

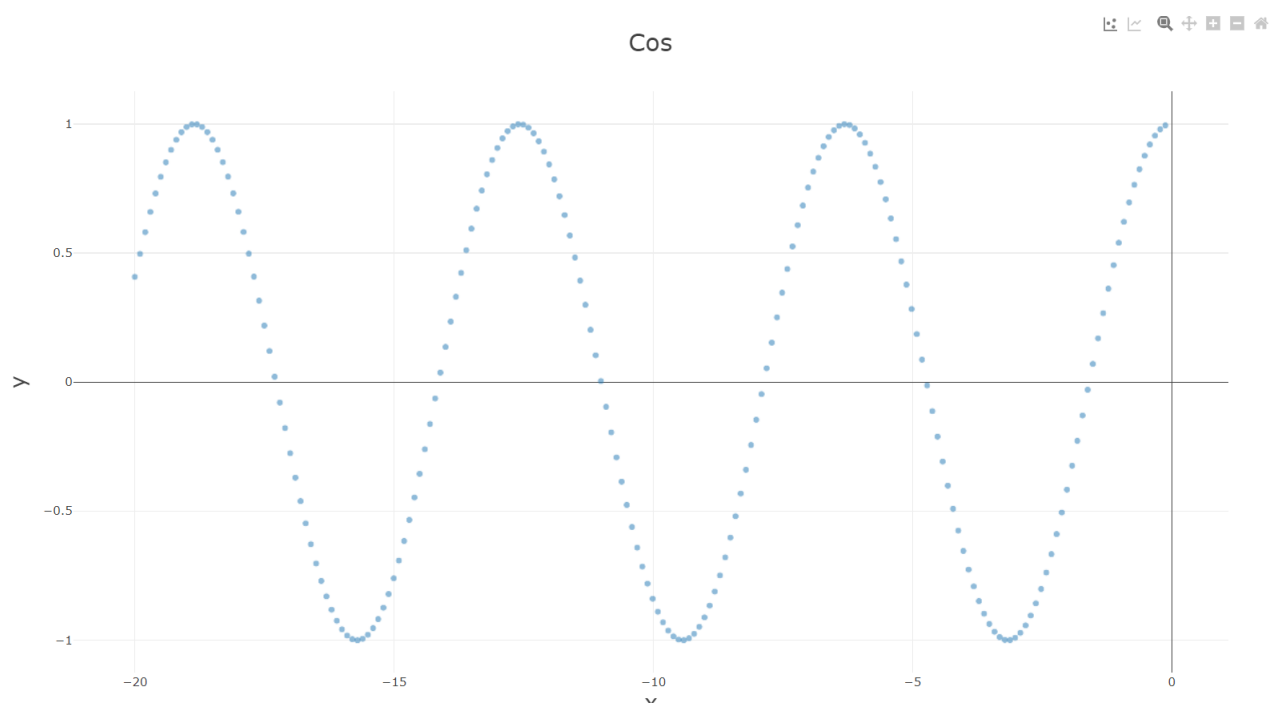
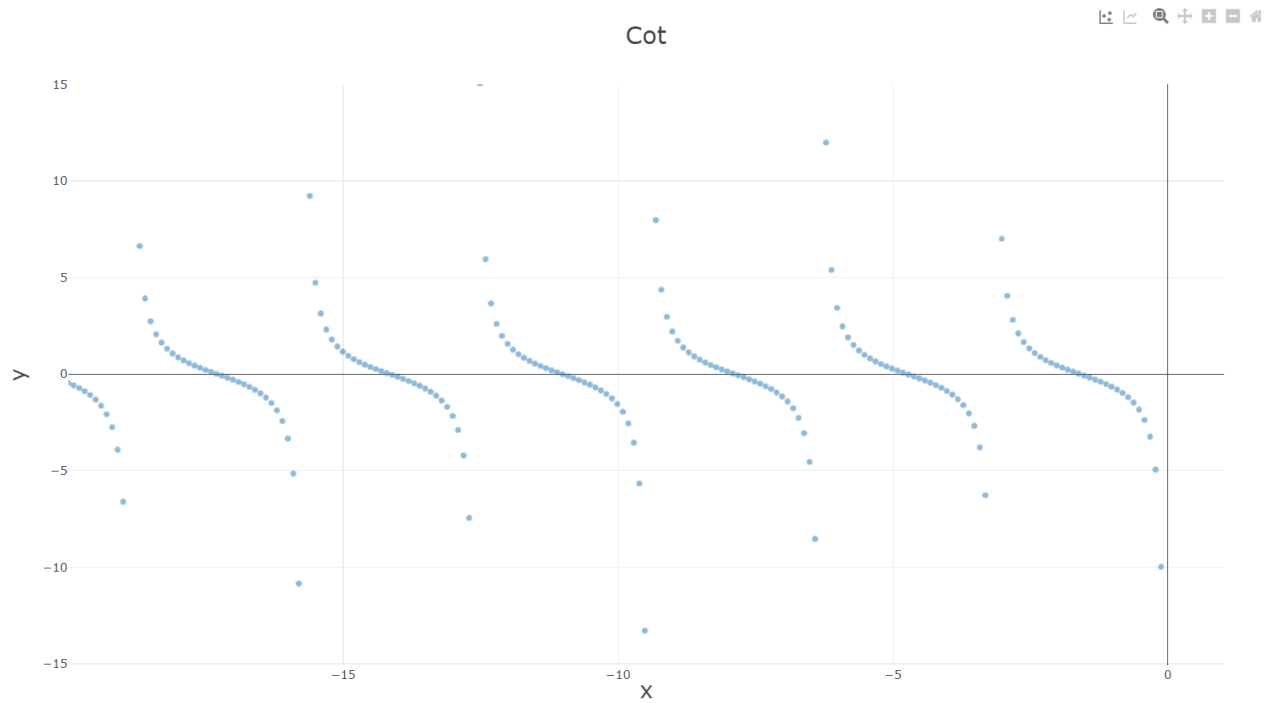
Используется напрямую

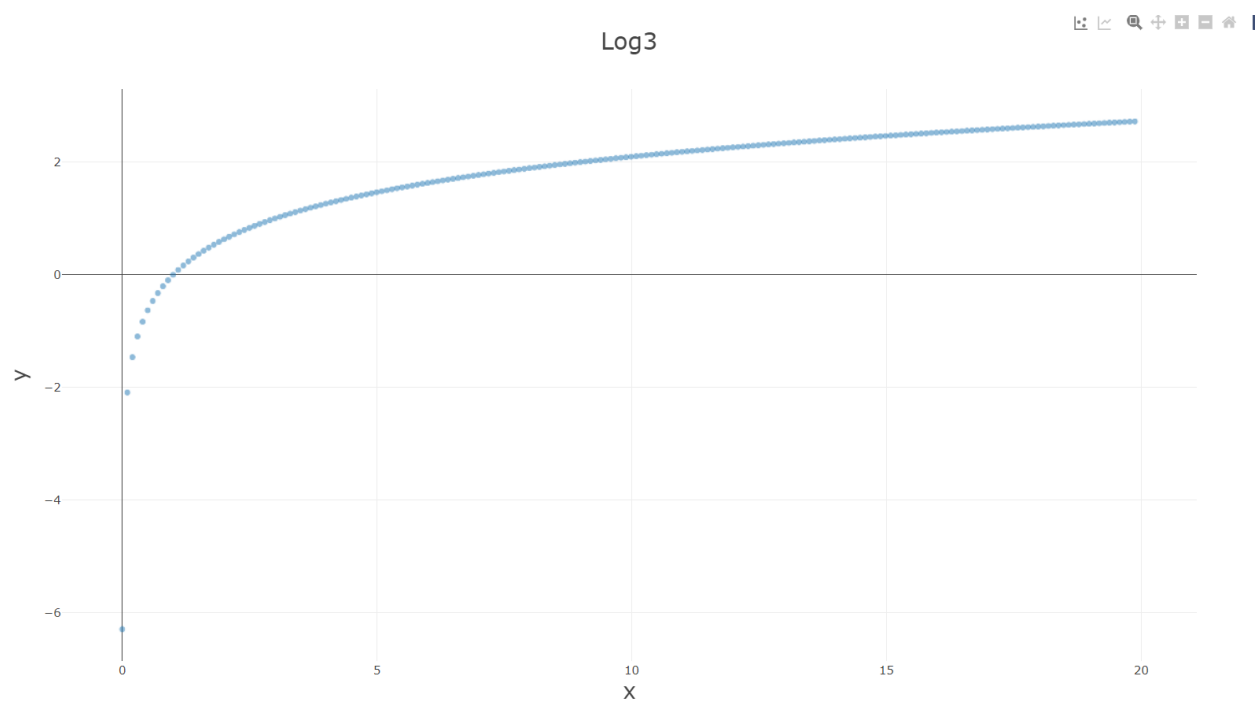
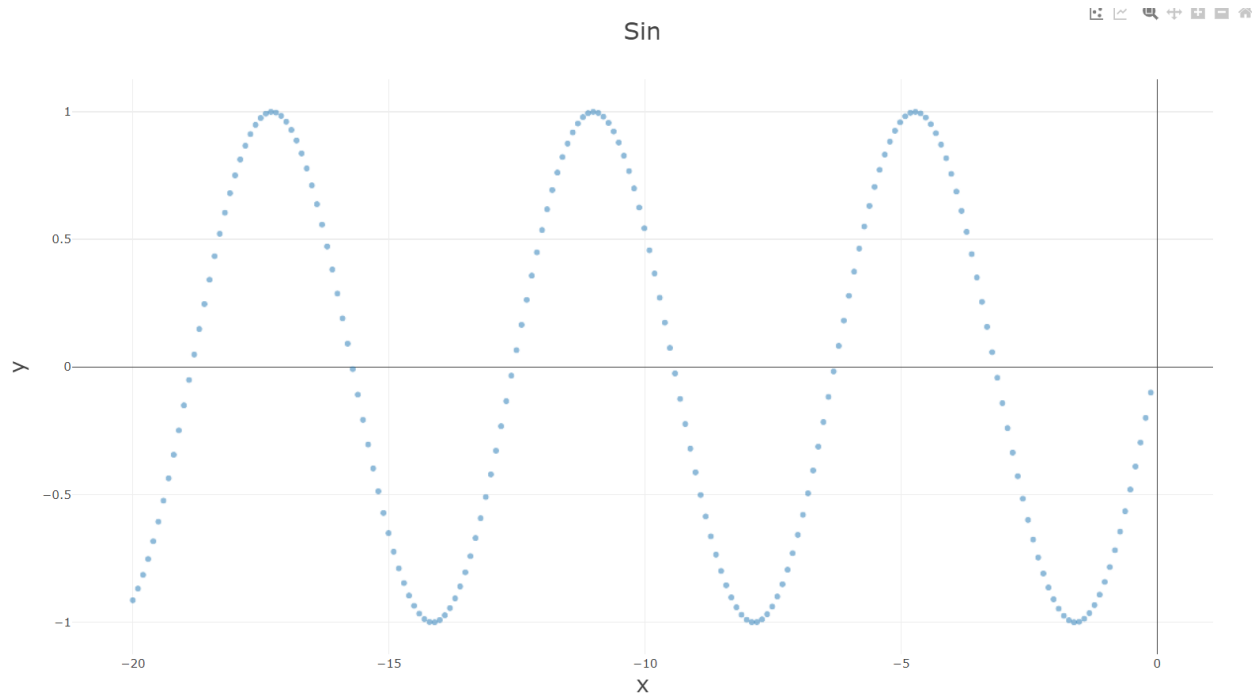
Используется в виде mock'a

Тест / задействованные модули	SystemFunction	FirstFunction	SecondFunction	Cot	Cos	Sin	Log10	Log5	Log3	Log2	Ln
SystemFunctionIntegrationTest											
FirstFunctionIntegrationTest											
SecondFunctionIntegrationTest											
CotFunctionIntegrationTest											
CosFunctionIntegrationTest											
SinFunctionUnitTest											
Log10FunctionIntegrationTest											
Log5FunctionIntegrationTest											
Log3FunctionIntegrationTest											
Log2FunctionIntegrationTest											
LnFunctionUnitTest											
FullSystemIntegrationTest											

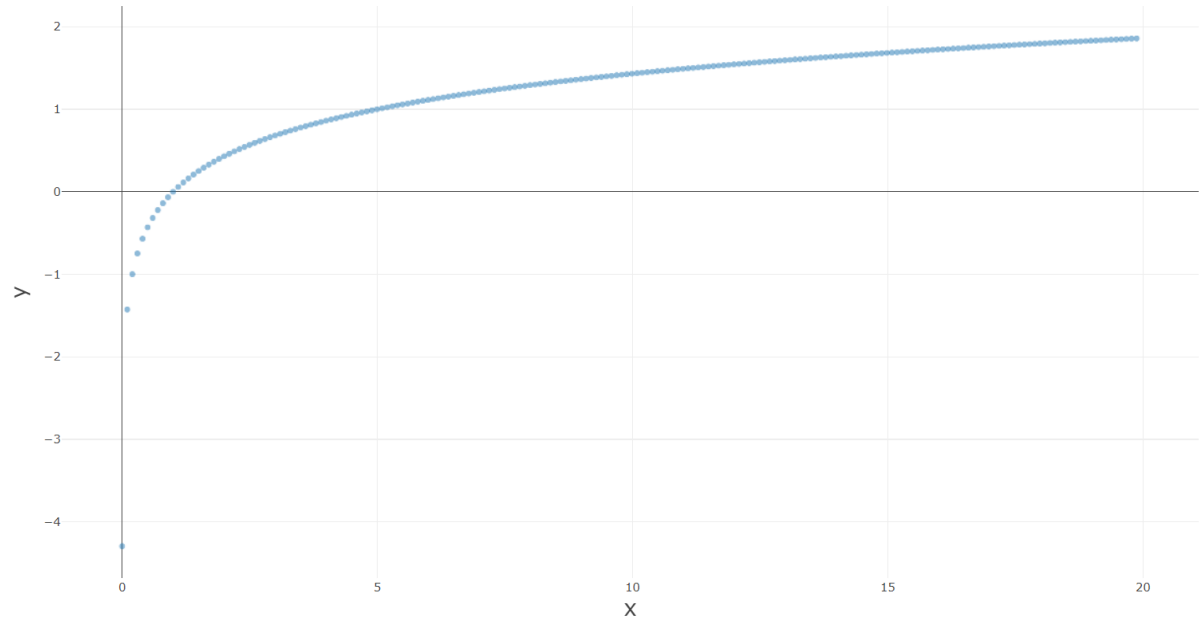




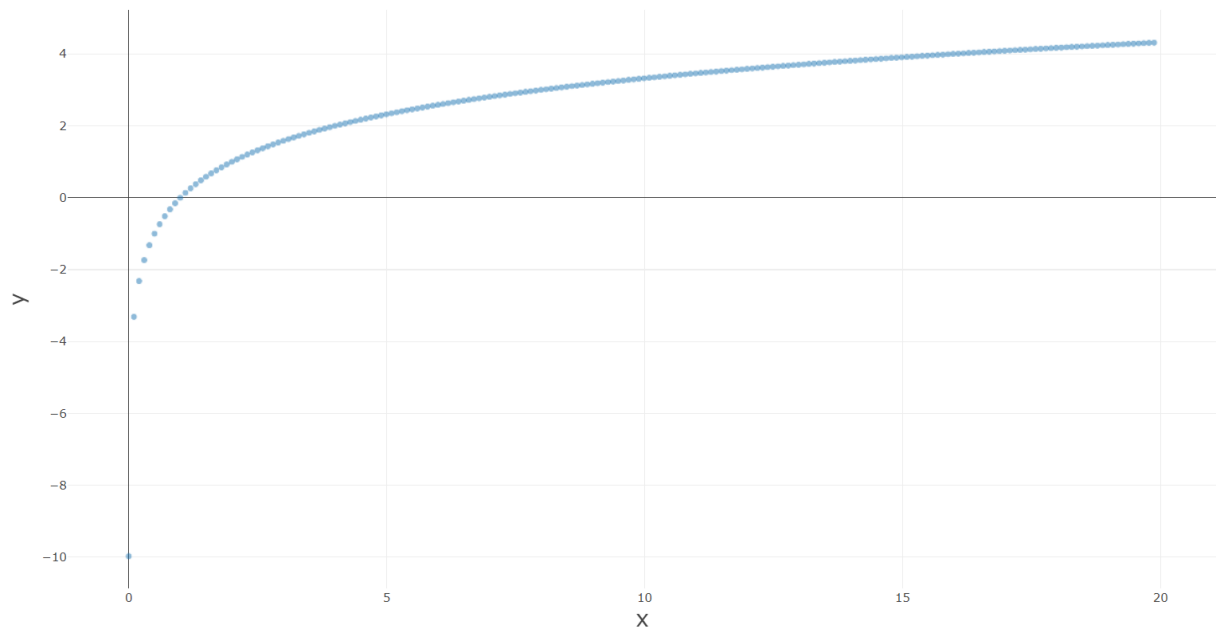


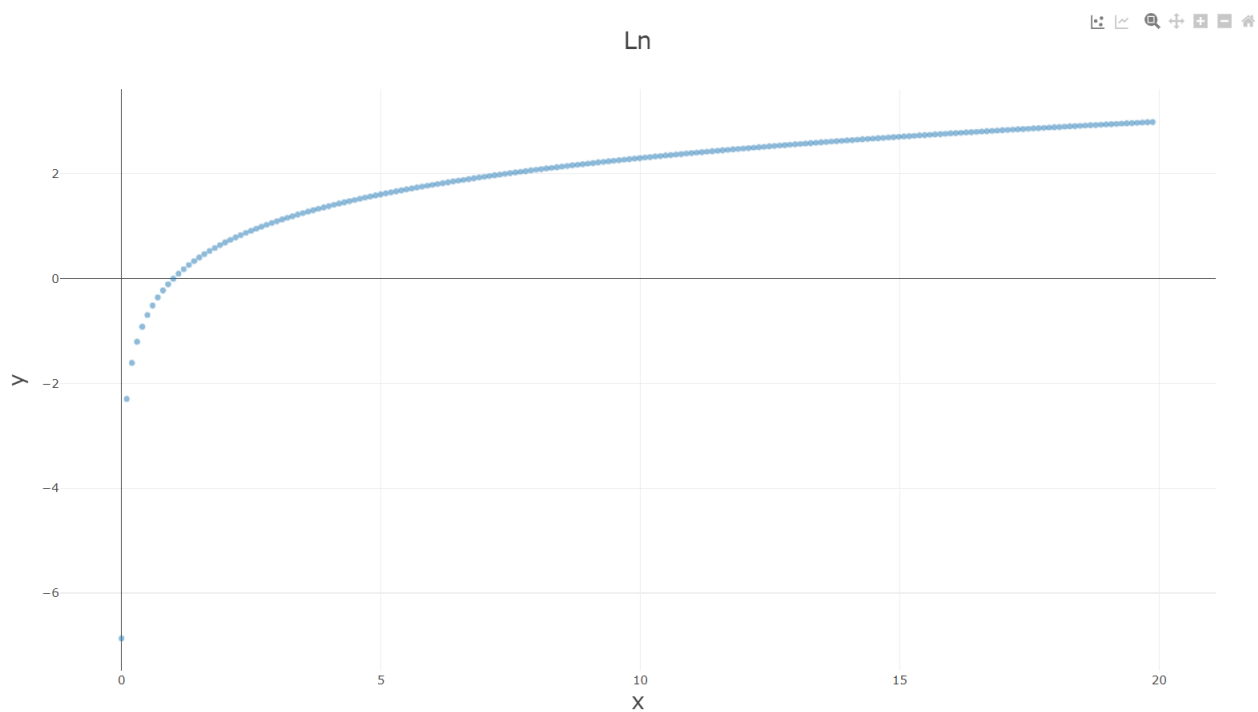
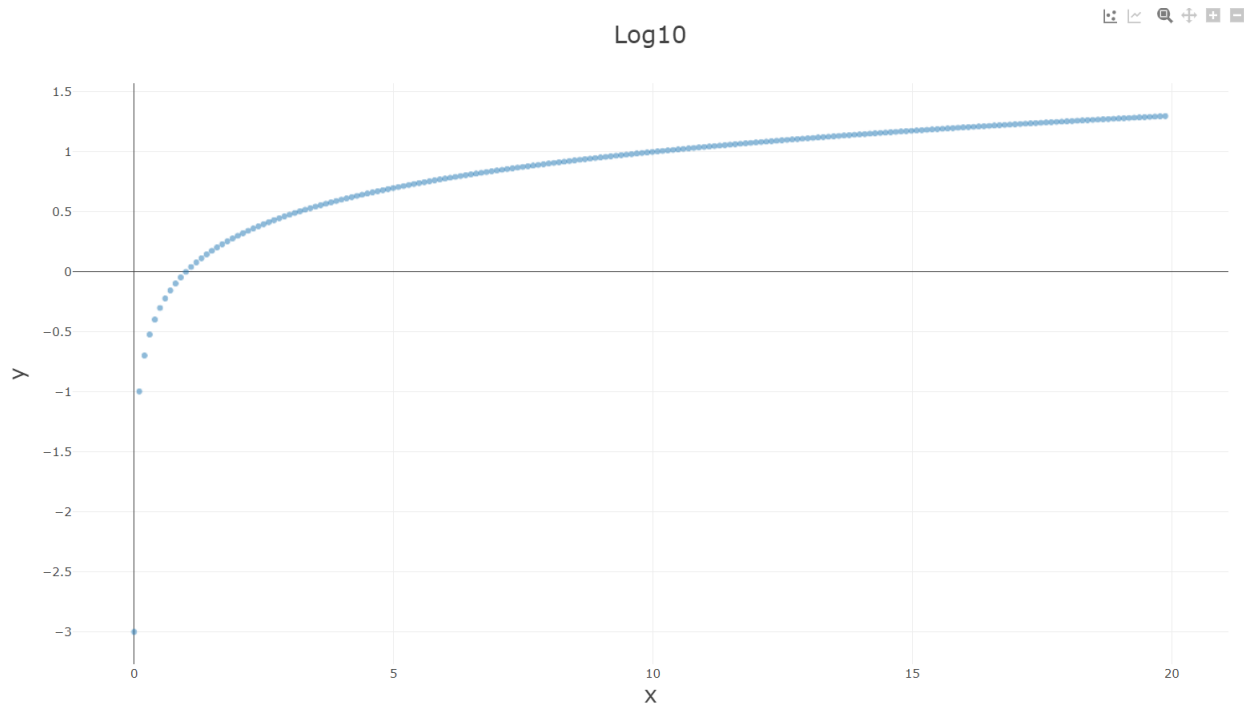


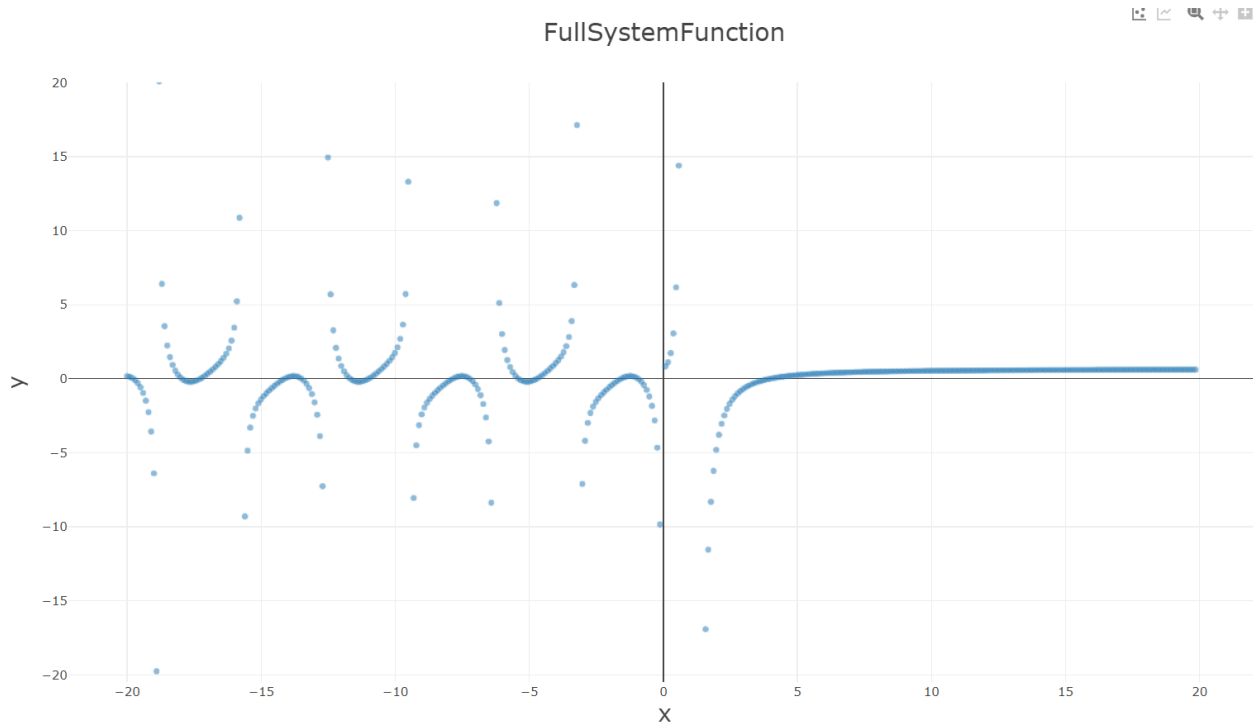
Log5



Log2







Вывод: в ходе выполнения лабораторной работы я написала программу, вычисляющую значения заданной системы функций, выбрала для нее тестовое покрытие и провела интеграционное тестирование.