

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерных технологий

Тестирование программного обеспечения

Лабораторная работа № 2

Вариант 55535

Выполнил студент

Чайка Алексей

Группа № Р33122

Преподаватель: Харитоновна Анастасия Евгеньевна

г. Санкт-Петербург

2021

Задание

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.

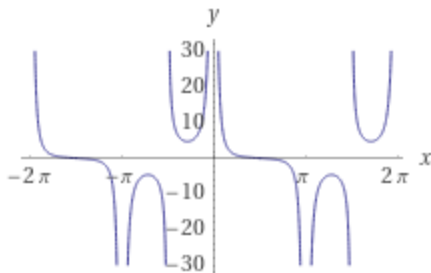
По варианту

$$x \leq 0 : (((\sin(x) * \csc(x)) - \sin(x)) / \sin(x)) * (\sec(x) / \sin(x)))$$

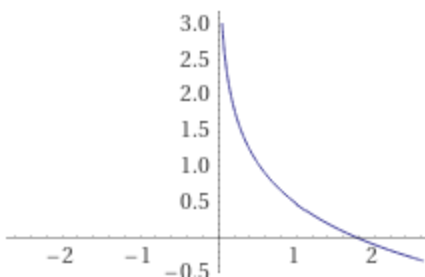
$$x > 0 : (((\log_5(x) * \log_2(x)) + \log_{10}(x)) / \log_3(x)) - (\log_2(x) + \ln(x))) + \log_5(x)$$

Графики уравнений

Equation 1



Equation 2





Код



CustomFunctionImplementation

Companion

Functions

<code>sin(double, double)</code>	double
<code>cos(double, double)</code>	double
<code>sec(double, double)</code>	double
<code>csc(double, double)</code>	double
<code>log(double, double, double)</code>	double
<code>log_2(double, double)</code>	double
<code>log_3(double, double)</code>	double
<code>log_5(double, double)</code>	double
<code>log_10(double, double)</code>	double
<code>ln(double, double)</code>	double

ActualFunctionImplementation

Companion

Companion

<code>sin(double, double)</code>	double
<code>cos(double, double)</code>	double
<code>sec(double, double)</code>	double
<code>csc(double, double)</code>	double
<code>ln(double, double)</code>	double
<code>log(double, double, double)</code>	double
<code>log_2(double, double)</code>	double
<code>log_3(double, double)</code>	double
<code>log_5(double, double)</code>	double
<code>log_10(double, double)</code>	double
<code>factorial(int)</code>	int

Companion

<code>sin(double, double)</code>	double
<code>cos(double, double)</code>	double
<code>sec(double, double)</code>	double
<code>csc(double, double)</code>	double
<code>ln(double, double)</code>	double
<code>log(double, double, double)</code>	double
<code>log_2(double, double)</code>	double
<code>log_3(double, double)</code>	double
<code>log_5(double, double)</code>	double
<code>log_10(double, double)</code>	double

IntegrationTest

<code>prepareCSV()</code>	void
<code>actualFunctionValue(double)</code>	double
<code>appendRow(ArrayList<ArrayList<Double>>, ArrayList<Double>)</code>	ArrayList<ArrayList<Double>>
<code>applyToCSV(ArrayList<ArrayList<Double>>)</code>	void
<code>custom sin with actual function(double, double)</code>	void
<code>custom ln with actual function(double, double)</code>	void
<code>Integration test for function with gradually increasing number of custom functions implementations(double, double)</code>	void

Utility

Companion Companion

MainKt

`main(String[])` void

Companion

<code>f(double, HashMap<String, Function0<Double>>)</code>	double
<code>replaceWith(HashMap<String, Function0<Double>>, String, Function0<Double>)</code>	HashMap<String, Function0<Double>>

Вывод

Выполнив лабораторную работу, познакомился с JUnit на Kotlin, с ParametrizedTests и укрепил свои познания в тестировании в целом.