МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«Национальный исследовательский университет ИТМО»

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ

**ЛАБОРАТОРНАЯ РАБОТА №2**

по дисциплине

«ТЕСТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ»

***Выполнил:***

Студент группы P3318

Рамеев Тимур

Ильгизович

***Преподаватель:***

Кулинич Ярослав

Вадимович

# Задание

Провести интеграционное тестирование программы, осуществляющей вычисление системы функций.

Изображение выглядит как текст, Шрифт, рукописный текст, белый

Контент, сгенерированный ИИ, может содержать ошибки.

Ход работы

## Реализация базовой функции Sin

|  |
| --- |
| public class Sin implements Computable {   @Override  public BigDecimal compute(double x){  BigDecimal result = BigDecimal.*valueOf*(0), term;  for (int n = 0; n < *N\_SIZE*; n++) {  int prefix = n % 2 == 0 ? 1 : -1;  term = BigDecimal.*valueOf*(x).pow(2 \* n + 1)  .divide(*factorials*.get(2 \* n + 1), *mc*)  .multiply(BigDecimal.*valueOf*(prefix));  result = result.add(term);  }  return result.round(*mc*);  } } |

## Реализация базовой функции Ln

|  |
| --- |
| public class Ln implements Computable {   @Override  public BigDecimal compute(double x) {  if (x <= 0) {  throw new ArithmeticException("В данной точке функция ln(x) не определена");  }  double top = x - 1, bot = x + 1;  BigDecimal result = BigDecimal.*valueOf*(0), term;  for (int n = 0; n < *N\_SIZE*; n++) {  term = BigDecimal.*valueOf*(top)  .divide(  BigDecimal.*valueOf*(bot),  *mc*)  .pow(2 \* n + 1, *mc*)  .divide(BigDecimal.*valueOf*(2 \* n + 1), *mc*);  result = result.add(term);  }  return result.multiply(BigDecimal.*TWO*, *mc*).round(*mc*);  } } |

## Реализация функции Log

|  |
| --- |
| class Log implements Computable {  private final int base;  private final Ln ln;  public Log(int base, Ln ln) {  this.base = base;  this.ln = ln;  boolean flag = false;  for (int b : logBases) {  if (b == this.base) {  flag = true;  break;  }  }  if (!flag) {  throw new ArithmeticException("Я не могу создать такой логарифм");  }  }  @Override  public BigDecimal compute(double x) {  return ln.compute(x).divide(ln.compute(base), mc);  } } |

## Реализация функции Cos

|  |
| --- |
| class Cos implements Computable {   private final Sin sin;   public Cos(Sin sin) {  this.sin = sin;  }   @Override  public BigDecimal compute(double x) {  int m = Math.*floor*((Math.*abs*(x) - Math.*PI* / 2 ) / Math.*PI*) % 2 == 0 ? -1 : 1;  return sin.compute(x).pow(2, *mc*)  .multiply(BigDecimal.*valueOf*(-1), *mc*)  .add(BigDecimal.*valueOf*(1), *mc*)  .sqrt(*mc*)  .multiply(BigDecimal.*valueOf*(m), *mc*);  } } |

## Реализация функции Cot

|  |
| --- |
| class Cot implements Computable {   private final Sin sin;  private final Cos cos;   public Cot(Sin sin, Cos cos) {  this.sin = sin;  this.cos = cos;  }   @Override  public BigDecimal compute(double x) {  if (Math.*abs*(x) % Math.*PI* <= *DELTA*)  throw new ArithmeticException("В данной точке функция cot(x) не определена");  return cos.compute(x)  .divide(sin.compute(x), *mc*);  } } |

## Реализация функции Csc

|  |
| --- |
| class Csc implements Computable {   private final Sin sin;  public Csc(Sin sin) {  this.sin = sin;  }   @Override  public BigDecimal compute(double x) {  if (Math.*abs*(x) % Math.*PI* <= *DELTA*)  throw new ArithmeticException("В данной точке функция csc(x) не определена");  return BigDecimal.*valueOf*(1)  .divide(sin.compute(x), *mc*);  } } |

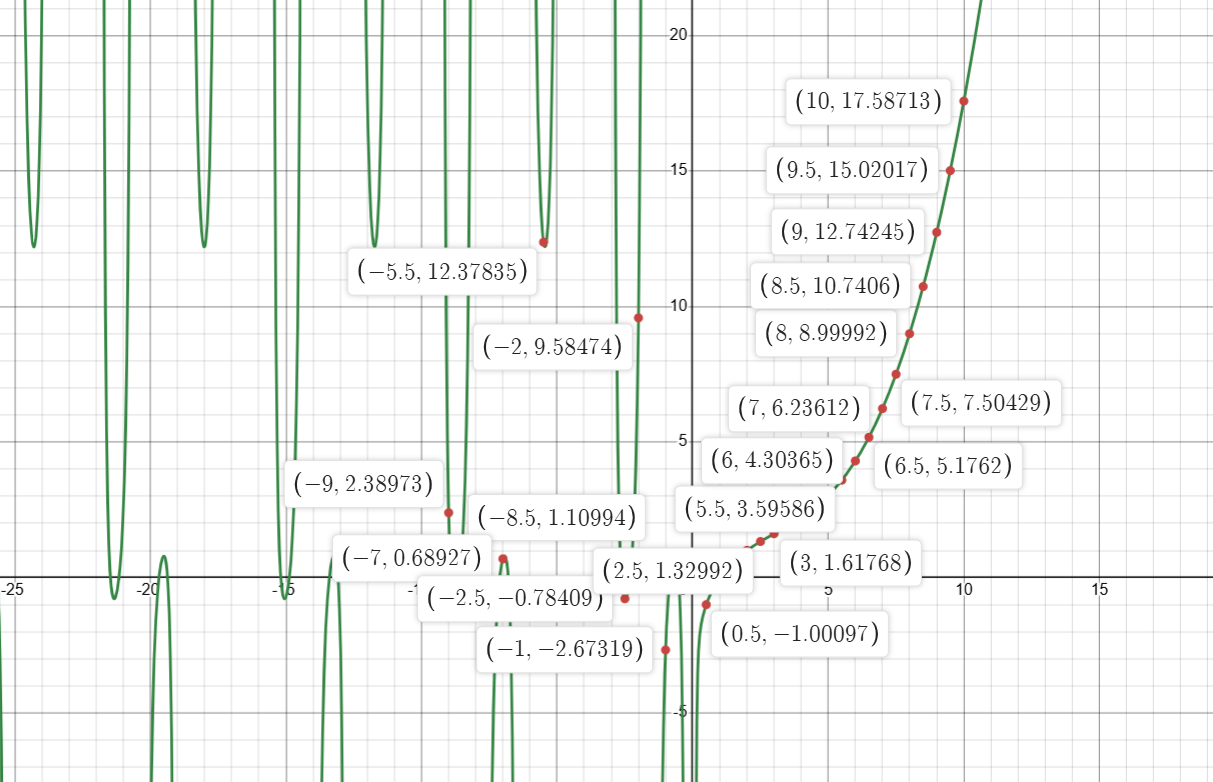
## Реализация функции Sec

|  |
| --- |
| class Sec implements Computable {   private final Cos cos;   public Sec(Cos cos) {  this.cos = cos;  }   @Override  public BigDecimal compute(double x) {  if (Math.*abs*(x) % (Math.*PI* / 2) <= *DELTA* && Math.*floor*(Math.*abs*(x) / (Math.*PI* / 2)) % 2 == 1)  throw new ArithmeticException("В данной точке функция sec(x) не определена");  return BigDecimal.*valueOf*(1)  .divide(cos.compute(x), *mc*);  } } |

## Реализация системы

|  |
| --- |
| package org.example;  import java.math.BigDecimal;  public class SysFunc implements Computable {  private final Sin sinF;  private final Cos cosF;  private final Csc cscF;  private final Sec secF;  private final Cot cotF;   private final Ln lnF;  private final Log ln2F, ln5F, ln10F;   public SysFunc(Sin sinF, Cos cosF, Csc cscF, Sec secF, Cot cotF, Ln lnF, Log ln2F, Log ln5F, Log ln10F) {  this.sinF = sinF;  this.cosF = cosF;  this.cscF = cscF;  this.secF = secF;  this.cotF = cotF;  this.lnF = lnF;  this.ln2F = ln2F;  this.ln5F = ln5F;  this.ln10F = ln10F;  }   @Override  public BigDecimal compute(double x) {  if (x <= 0.0){  var sin = sinF.compute(x).doubleValue();  var cos = cosF.compute(x).doubleValue();  var csc = cscF.compute(x).doubleValue();  var sec = secF.compute(x).doubleValue();  var cot = cotF.compute(x).doubleValue();  return BigDecimal.*valueOf*((Math.*pow*((sec + cot), 2) \* sec + sin \* cos) \* csc);  } else {  var ln = lnF.compute(x).doubleValue();  var ln2 = ln2F.compute(x).doubleValue();  var ln5 = ln5F.compute(x).doubleValue();  var ln10 = ln10F.compute(x).doubleValue();  return BigDecimal.*valueOf*((ln5 \* ln2 \* ln + ln2) \* Math.*pow*(ln10, 6) + ln2);  }  } } |

График системы по csv выгрузке



Вывод

Модульные и интеграционные тесты прошли успешно. Вычисленные точки совпадают с графиком функции.