“Санкт-Петербургский политехнический университет Петра Великого”

Институт компьютерных наук и технологий

ОТЧЕТ

По лабораторной работе «Задание для технологии программирования»

по дисциплине «Теория и технология программирования»

**Выполнил:**

Студент группы з3530902/90001                        \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.С. Кондрин

подпись, дата

**Проверил**

Доцент, кандидат технических наук                  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ С.В. Хлопин

подпись, дата

Санкт-Петербург 2022г.

Содержание

[Задание 3](#_Toc92593231)

[Код программы 4](#_Toc92593232)

[Блок-схема 5](#_Toc92593233)

[Пример выполнения программы 6](#_Toc92593234)

[Вывод 7](#_Toc92593235)

# Задание

На языке программирования создать программу, которая будет просить у пользователя ввести начальные и конечные значения для диапазона расчета X, шаг изменения переменной deltaX. Программа должна вывести на экран таблицу (которая корректно выводит значения для разного набора исходных данных – столбцы «не едут», правая граница таблицы постоянна) с номером строки, значению X, значению полученного выражения. В случае невозможности вычисления выражения для конкретного случая X, num (деление на ноль, логарифм из отрицательного числа, значение синуса или косинуса равно 0), в строке таблицы необходимо вывести сообщение об ошибке.

В программе должны быть введены две функции пользователя, которые возвращают значения для сравнения в функции min или max.

Для четных номеров:

Для нечетных номеров:

# **Код программы**

package com.ask0n;  
  
import java.util.Arrays;  
import java.util.List;  
import java.util.Scanner;  
  
public class Main {  
 private static final Table *table* = new Table();  
  
 private static double *startValue*;  
 private static double *endValue*;  
 private static double *deltaX*;  
 private static int *num*;  
  
 public static void main(String[] args) {  
 *inputData*();  
 *table*.create();  
  
 while (*deltaX* < 0 ? *endValue* <= *startValue* : *startValue* <= *endValue*) {  
 final List<Double> value;  
 if (*num* % 2 == 0) {  
 value = *funMax*(*startValue*, *num*);  
 } else {  
 value = *funMin*(*startValue*, *num*);  
 }  
  
 *table*.insert(*startValue*, value.get(0), value.get(1), Double.*isNaN*(value.get(2)) ? "Error" : value.get(2));  
 *startValue* += *deltaX*;  
 }  
  
 *table*.delimiter();  
 }  
  
 public static void inputData() {  
 while (true) {  
 *num* = *intInput*("Введите номер: ");  
 if (*num* > 0) break;  
 else System.*out*.println("Номер должен быть больше 0");  
 }  
 while (true) {  
 *deltaX* = *doubleInput*("Введите deltaX (шаг): ");  
 if (*deltaX* != 0) break;  
 else System.*out*.println("DeltaX не должен быть равен 0. Введите заного");  
 }  
 while (true) {  
 *startValue* = *doubleInput*("Введите стартовое значение: ");  
 *endValue* = *doubleInput*("Введите конечное значение: ");  
 if (*deltaX* < 0 ? *endValue* <= *startValue* : *startValue* <= *endValue*) break;  
 else System.*out*.println("Ошибка ввода. Проверьте значения.");  
 }  
 }

public static double doubleInput(String message) {  
 System.*out*.print(message);  
 Scanner in = new Scanner(System.*in*);  
 while (true) {  
 String input = in.nextLine();  
 if (input == null) throw new NumberFormatException("Неверный формат числа, повторите ввод: ");  
 try {  
 return Double.*parseDouble*(input.replace(",", "."));  
 } catch (NumberFormatException e) {  
 System.*out*.print("Неверный формат числа, повторите ввод: ");  
 }  
 }  
  
 }  
  
 public static int intInput(String message) {  
 System.*out*.print(message);  
 Scanner in = new Scanner(System.*in*);  
 while (true) {  
 String input = in.nextLine();  
 if (input == null) throw new NumberFormatException("Неверный формат числа, повторите ввод: ");  
 try {  
 return Integer.*parseInt*(input);  
 } catch (NumberFormatException e) {  
 System.*out*.print("Неверный формат числа, повторите ввод: ");  
 }  
 }  
  
 }  
  
 public static List<Double> funMax(double xDelta, int num) {  
 final double first = *customLog*((1 - num) / Math.*sin*(xDelta + num), 21);  
 final double second = Math.*abs*(Math.*cos*(xDelta) / num);  
 return Arrays.*asList*(first, second, Math.*max*(first, second));  
 }  
  
 public static List<Double> funMin(double xDelta, int num) {  
 final double first = *customLog*((1 - num) / Math.*cos*(xDelta - num), 21);  
 final double second = Math.*sin*(xDelta) / num;  
 return Arrays.*asList*(first, second, Math.*max*(first, second));  
 }  
  
 private static double customLog(double base, double logNumber) {  
 return Math.*log*(logNumber) / Math.*log*(base);  
 }  
}

package com.ask0n;  
  
public class Table {  
 private static final String *OUTPUT\_FORMAT* = "| %-5s | %-20s | %-25s | %-25s | %-25s |%n";  
 private static final String *DELIMITER* = "+-------+----------------------+---------------------------+---------------------------+---------------------------+";  
 private int rowNum;  
  
 public Table() {  
 this.rowNum = 1;  
 }  
  
 public void delimiter() {  
 System.*out*.println(*DELIMITER*);  
 }  
  
 public void create() {  
 delimiter();  
 System.*out*.format(*OUTPUT\_FORMAT*, "#", "DeltaX", "Function 1", "Function 2", "Result");  
 delimiter();  
 }  
  
 public void insert(Object deltaX, Object f1, Object f2, Object result) {  
 System.*out*.format(*OUTPUT\_FORMAT*, rowNum++, deltaX, f1, f2, result);  
 }  
}

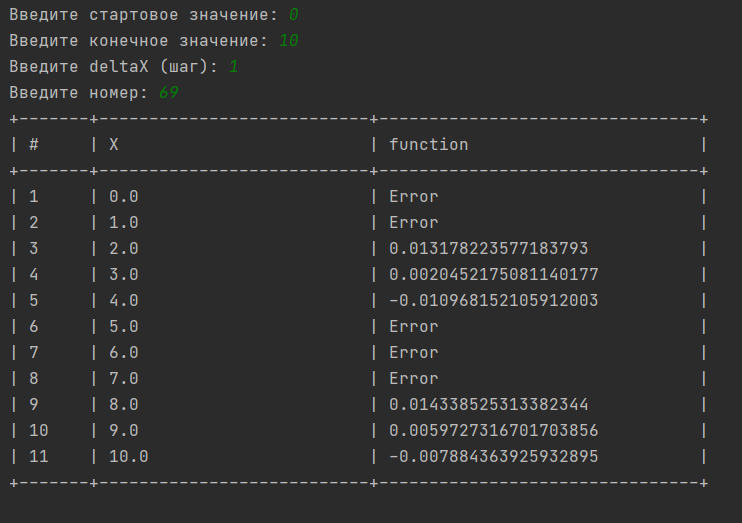
# Блок-схема







# Пример выполнения программы



# Вывод

В ходе выполнения данной работы, были проведены основные тесты программы на выявление ошибок. Так как ни одной ошибки в процессе тестирования не было, можем заключить, что программа успешно функционирует.