

Федеральное государственное автономное образовательное учреждение  
высшего образования  
«Санкт-Петербургский политехнический университет Петра Великого»  
Институт Компьютерных Наук и Технологий

РГР

по дисциплине: «Теория и технологии программирования»

Выполнил студент  
группы з3532703/00001

К.И. Телегин

Проверил

С.В. Хлопин

Санкт-Петербург  
2020 г.

## Задание

На языке программирования создать программу, которая будет просить у пользователя ввести начальные и конечные значения для диапазона расчета  $X$ , шаг изменения переменной  $\Delta X$ . 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:

$$\max \left( \ln \left( \frac{1 - \text{num}}{\sin(x + \text{num})} \right), \left| \frac{\text{ctg}(x)}{\text{num}} \right| \right)$$

Для нечетных номеров:

$$\min \left( \ln \left( \frac{1 - \text{num}}{\cos(x - \text{num})} \right), \frac{\text{tg}(x)}{\text{num}} \right)$$

## Листинг программы

Класс Application — основной класс приложения, содержащий функцию main.

```
import java.util.Map;
import java.util.Scanner;
public class Application {
    public static void main(String[] args) {
        if(args.length > 0){
            Application.start(Integer.parseInt(args[0]));
        }else{
            Application.start(15);
        }
    }
    private static void start(int num){
        try{
            InputService in = new InputService(new Scanner(System.in));
            PrintService printService = new PrintService();
            DataAccessObject data = in.createData();
            EquationService equationService = new EquationService(data, num);
            Map<Double, Double> result = equationService.equation();
            printService.print(result);
        }catch (Exception e){
            System.out.println(e.getMessage());
        }
    }
}
```

Класс DataAccessObject для хранения данных, передаваемых в ходе выполнения программы.

```
final public class DataAccessObject {
    private final Double varA, varB, delta;
    public DataAccessObject(Double varA, Double varB, Double delta) {
        this.varA = varA;
        this.varB = varB;
        this.delta = delta;
    }
}
```

```

public Double getVarA() {
    return varA;
}
public Double getVarB() {
    return varB;
}
public Double getDelta() {
    return delta;
}
}

```

Класс InputService, отвечающий за обработку пользовательского ввода.

```

import java.util.Scanner;
final public class InputService {
    private final Scanner in;
    public InputService(Scanner in) {
        this.in = in;
    }
    public DataAccessObject createData() throws Exception{
        Double _a, _b, _d;
        while(true){
            _a = inputDoubleValue("Please enter the left border of the interval X");
            _b = inputDoubleValue("Please enter the right border of the interval X");
            if(_a.equals(_b)){
                System.out.println("The interval X should not be empty");
            }else{
                break;
            }
        }
        while(true){
            _d = inputDoubleValue("Please enter step for interval");
            if(_d.equals((double)0)){
                System.out.println("The step must not be zero");
            }else{
                break;
            }
        }
    }
}

```

```

        this.in.close();
        return new DataAccessObject(_a, _b, _d);
    }
    private Double inputDoubleValue(String message){
        System.out.println(message);
        while (!in.hasNextDouble()) {
            System.out.println("That not a number");
            in.next();
        }
        return in.nextDouble();
    }
}

```

Класс EquationService — класс, содержащий основную логику программы.

```

import java.util.HashMap;
import java.util.Map;
final public class EquationService {
    private final DataAccessObject data;
    private final int num;
    public EquationService(DataAccessObject data, int num) {
        this.data = data;
        this.num = num;
    }
    public Map<Double, Double> equation(){
        Map<Double, Double> result = new HashMap<>();
        Double x, stop, step;
        if(this.data.getVarA() > this.data.getVarB()){
            x = data.getVarB();
            stop = data.getVarA();
        }else{
            stop = data.getVarB();
            x = data.getVarA();
        }
        if(this.data.getDelta() > 0){
            step = this.data.getDelta();
        }else{
            step = -this.data.getDelta();
        }
    }
}

```

```

    }
    result.put(stop, calculateFunction(stop));
    while(x < stop){
        result.put(x, calculateFunction(x));
        x += step;
    }
    return result;
}

private Double calculateFunction(Double x){
    if(this.num % 2 == 0){
        return max(leftFunction(x), rightFunction(x));
    }else{
        return min(leftFunction(x), rightFunction(x));
    }
}

private Double leftFunction(Double x){
    if(this.num % 2 == 0){
        return Math.log((1-num)/(Math.sin(x+num)));
    }else{
        return Math.log((1-num)/(Math.cos(x-num)));
    }
}

private Double rightFunction(Double x){
    if(this.num % 2 == 0){
        return Math.abs((1/Math.tan(x))/num);
    }else{
        return Math.tan(x)/num;
    }
}

private Double min(Double x1, Double x2){
    return x1 > x2 ? x2 : x1;
}

private Double max(Double x1, Double x2){
    return x1 > x2 ? x1 : x2;
}
}

```

Класс PrintService, отвечающий за отображение результата выполнения программы.

```
import dnl.utils.text.table.TextTable;
import java.util.Arrays;
import java.util.Map;
final public class PrintService {
    public void print(Map<Double, Double> result){
        String[] columnNames = {"X", "Y"};
        Object[][] data = new Object[result.size()][2];
        Object[] keys = result.keySet().toArray();
        Arrays.sort(keys);
        int i = 0;
        for (Object key : keys) {
            data[i] = new Object[]{key, (result.get(key).isNaN() ? "error" : result.get(key))};
            i++;
        }
        TextTable tt = new TextTable(columnNames, data);
        tt.printTable();
    }
}
```

## Блок схема программы

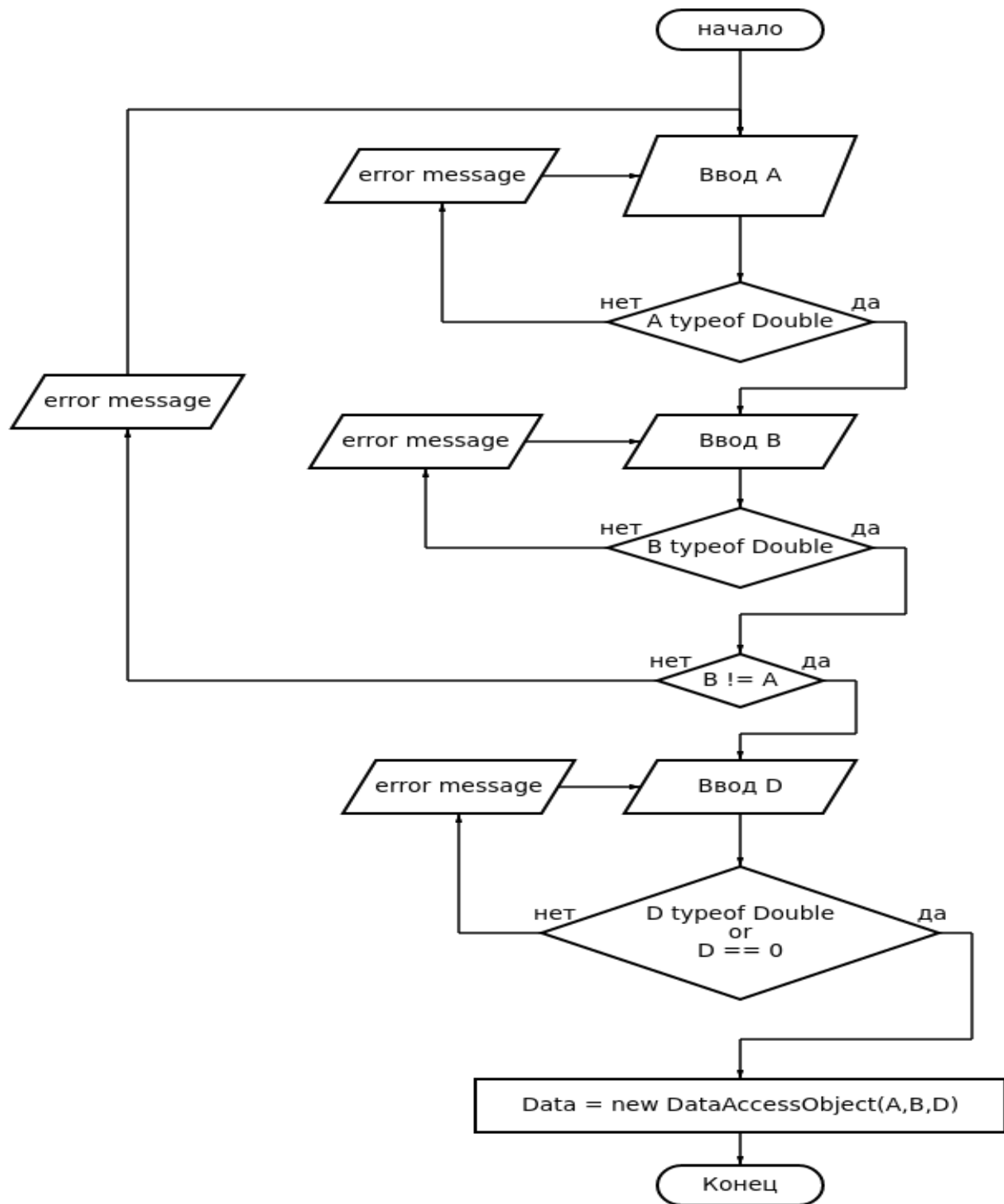


Рисунок 1. Блок схема подпрограммы `InputService`



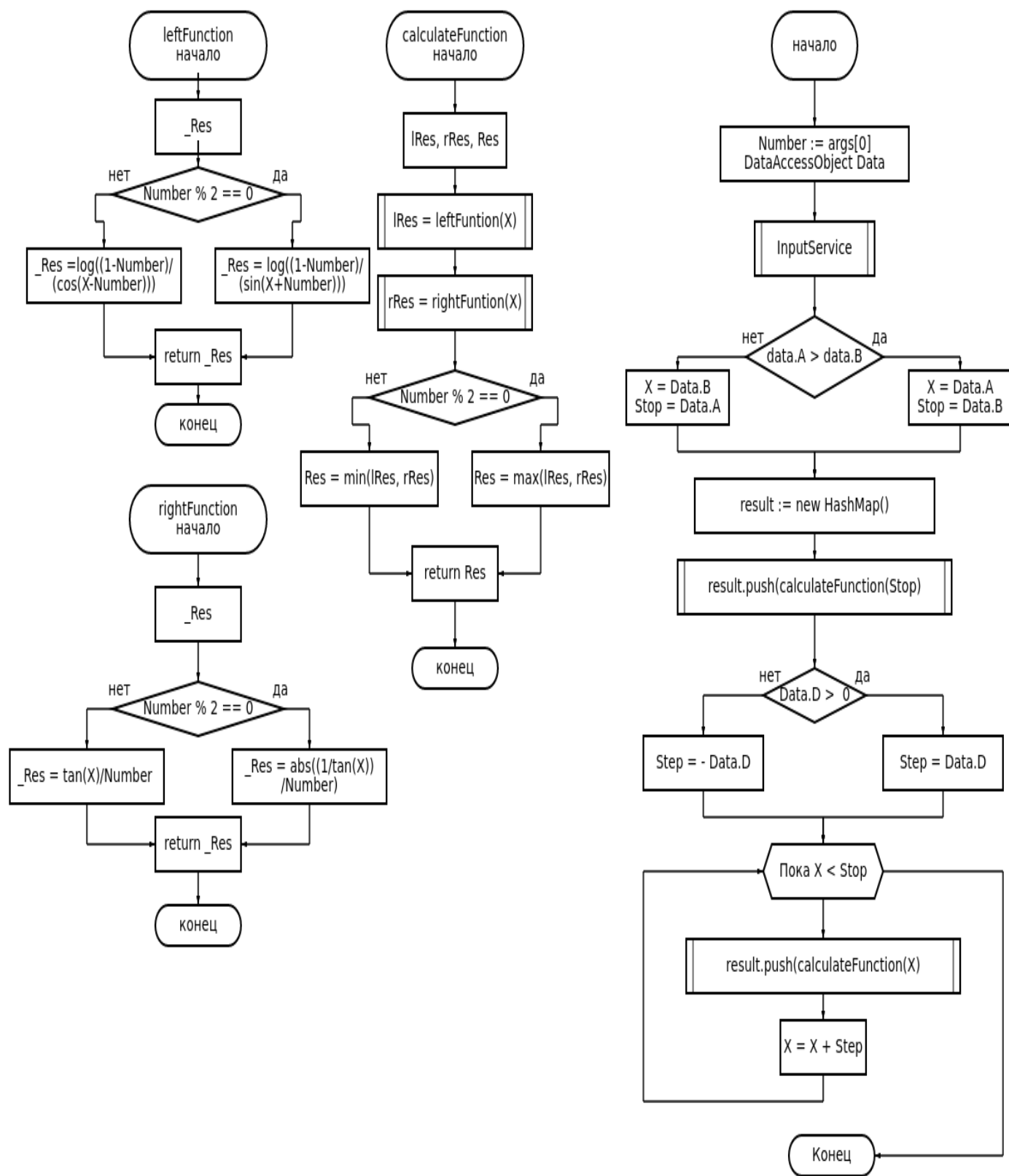


Рисунок 2. Блок-схема программы

## Программа испытаний

Пункт программы тестовых испытаний	Вид испытаний (проверок)	Номинальное значение	Пункт методики
1.	Запуск программы. (корректность запуска)	учитывается	1
2.	Ввод данных	учитывается	2
3.	Корректная работа для набора X как целого, так и числа с плавающей точкой	учитывается	2
4.	Отсутствие «повисания» при вводе шага равного 0	учитывается	2
5.	Отсутствие «повисания» при вводе отрицательного шага; либо корректность работы	учитывается	2
6.	При некорректных данных вывод сообщения об ошибки в поле таблицы	учитывается	2
7.	При корректных данных вывод значений в поле таблицы для случая максимума левого и правого выражений	учитывается	2
8.	Корректность отрисовки таблицы при получении «больших» чисел	учитывается	2

### Методики

1. Запуск скомпилированной в jar-файл программы
2. Запуск из IDE в режиме Debug

## Результаты выполнения программы

```
mendax@HAL-9000:~/Документы/education/spbpu/programTechnology/build/libs$ java -jar programTechnology-1.0-SNAPSHOT.jar 15
Please enter the left border of the interval X
1
Please enter the right border of the interval X
20
Please enter step for interval
1
```

X	Y
1.0	error
2.0	error
3.0	error
4.0	error
5.0	-0.2253676670831057
6.0	-0.019400412758983278
7.0	0.05809653218162125
8.0	error
9.0	error
10.0	error
11.0	-15.063389763613008
12.0	-0.04239066191077205
13.0	0.030868075529099308
14.0	error
15.0	error
16.0	error
17.0	0.2329277096983227
18.0	-0.07582091415584578
19.0	0.010105964707493338
20.0	error

Рисунок 3. Целочисленный интервал

```
mendax@HAL-9000:~/Документы/education/spbpu/programTechnology/build/libs$ java -jar programTechnology-1.0-SNAPSHOT.jar 15
Please enter the left border of the interval X
-10
Please enter the right border of the interval X
-100
Please enter step for interval
-10
```

X	Y
-100.0	0.03914759434379527
-90.0	0.1330133608138828
-80.0	error
-70.0	-0.08146399454246289
-60.0	error
-50.0	0.01812670746650872
-40.0	error
-30.0	error
-20.0	-0.14914406294831614
-10.0	error

Рисунок 4. Интервал с отрицательным шагом

```

mendax@HAL-9000:~/Документы/education/spbpu/programTechnology/build/libs$ java -jar programTechnology-1.0-SNAPSHOT.jar 15
Please enter the left border of the interval X
1
Please enter the right border of the interval X
12
Please enter step for interval
0
The step must not be zero
Please enter step for interval
2

```

X	Y
1.0	error
3.0	error
5.0	-0.2253676670831057
7.0	0.05809653218162125
9.0	error
11.0	-15.063389763613008
12.0	-0.04239066191077205

Рисунок 5. Обработка нулевого значения deltaX

```

mendax@HAL-9000:~/Документы/education/spbpu/programTechnology/build/libs$ java -jar programTechnology-1.0-SNAPSHOT.jar 15
Please enter the left border of the interval X
A
That not a number
1,3
Please enter the right border of the interval X
22,456
Please enter step for interval
2,3

```

X	Y
1.3	error
3.5999999999999996	error
5.8999999999999995	-0.02687405999154848
8.2	error
10.5	0.12332666622812849
12.8	0.015864999676524792
15.100000000000001	error
17.400000000000002	-0.5471748159514909
19.700000000000003	0.07595684965790951
22.000000000000004	error
22.456	error

Рисунок 6. Интервал с нецелым шагом

## Вывод

В ходе выполнения расчетно-графической работы была написана программа на языке Java, которая принимает аргумент командной строки (номер варианта), значения пользовательского ввода для границ интервала и шага интервала и рассчитывает по этим параметрам значение функции. Для четных вариантов это:

$$\max \left( \ln \left( \frac{1 - num}{\sin(x + num)} \right), \left| \frac{ctg(x)}{num} \right| \right)$$

Для нечетных:

$$\min \left( \ln \left( \frac{1 - num}{\cos(x - num)} \right), \frac{tg(x)}{num} \right)$$

Программа была разбита на классы, каждый из которых имеет свою зону ответственности: Application — главный класс программы с функцией main, InputService — класс, отвечающий за прием и валидацию пользовательского ввода, PrintService — класс, отвечающий за вывод результатов выполнения программы, и класс EquationService, который выполняет расчет функции. Класс InputService способен обработать следующие случаи некорректного ввода:

1. Нулевой интервал (выведет сообщение об ошибке и попросит ввести границы интервала еще раз).
2. Нулевой шаг интервала (укажет, что шаг не может быть нулевым, и попросит ввести шаг еще раз).
3. Нечисловые значения (укажет, что введенные символы не являются числом, и попросит ввести значение еще раз).

Было проведено тестирование программы из 8 пунктов по двум методикам — запуск в режиме отладки и запуск скомпилированной в jar-файл программы. Каждый пункт из плана тестирования был выполнен успешно.

Таким образом, результатом работы стало понимание технологического процесса разработки простейших программ — от изучения ТЗ до выполнения тестирования и написания отчета.

**Приложение 1**  
к Программе и методике  
тестовых испытаний программы

Перечень средств измерений и испытательного оборудования, необходимых для проведения испытаний.

Наименование, тип и марка	Кол-во	ГОСТ, ТУ или обозначение	Основные характеристики
Персональный компьютер на базе AMD Ryzen 5 3600X	1	ПК-1	AMD Ryzen 5 3600X, 16 ГБ ОЗУ, NVIDIA GeForce 2060, openjdk 11.0.9.1, Kubuntu