**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ**

**И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»**

(СПбГУТ)

ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И ПРОГРАММНОЙ ИНЖЕНЕРИИ (**ИТПИ**)

КАФЕДРА ПРОГРАММНОЙ ИНЖЕНЕРИИ И ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ (ПИ И ВТ)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Лабораторная работа №5**

по дисциплине «Разработка фронтенд-приложений управления телекоммуникациями»

Тема «Параллельное вычисление логарифмов»

Вариант 24

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент гр. | ИКПИ-23 |  | 09.12.2024 | Харлова А.А. |
| Преподаватель |  |  |  | Белая Т.И. |

г. Санкт-Петербург

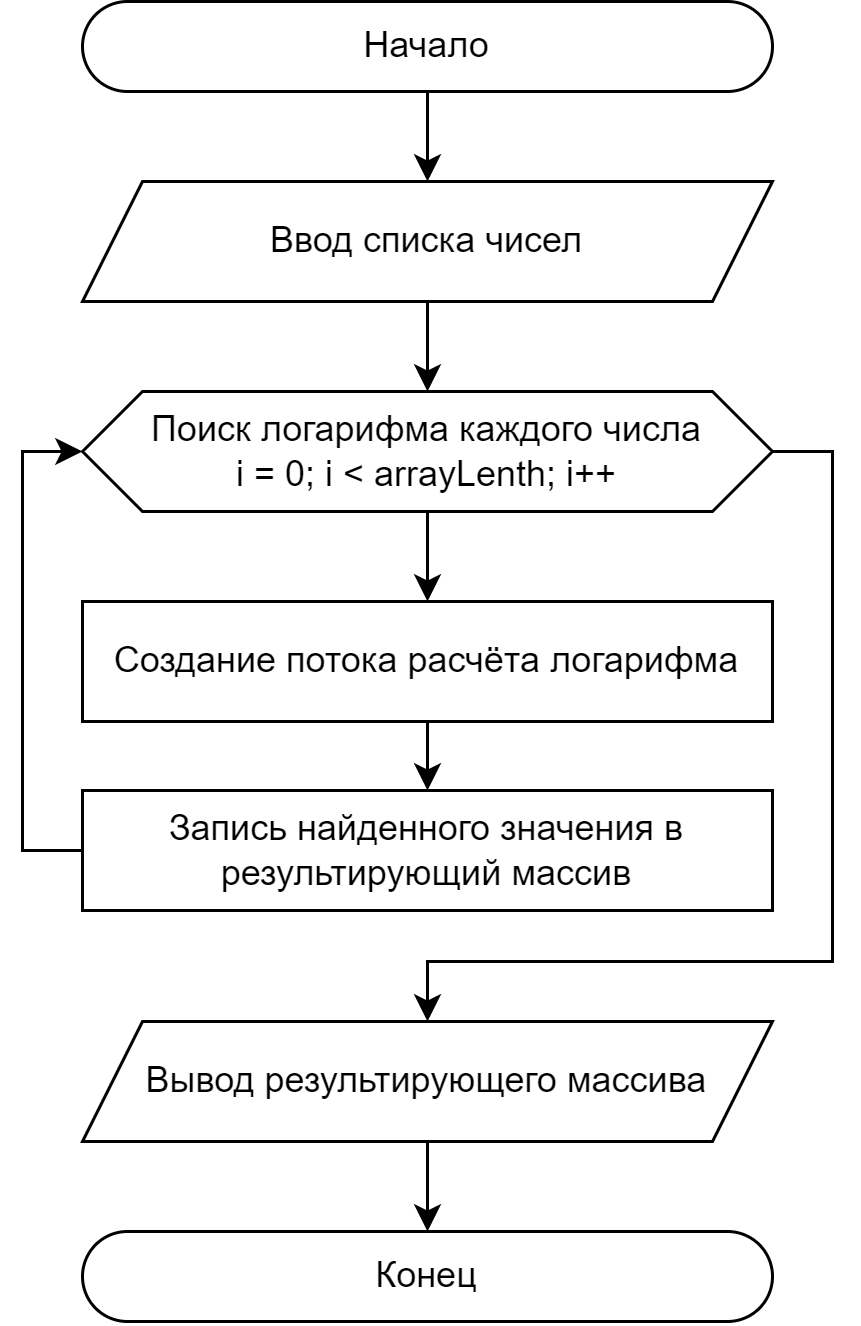
2024 г.

# ПОСТАНОВКА ЗАДАЧИ

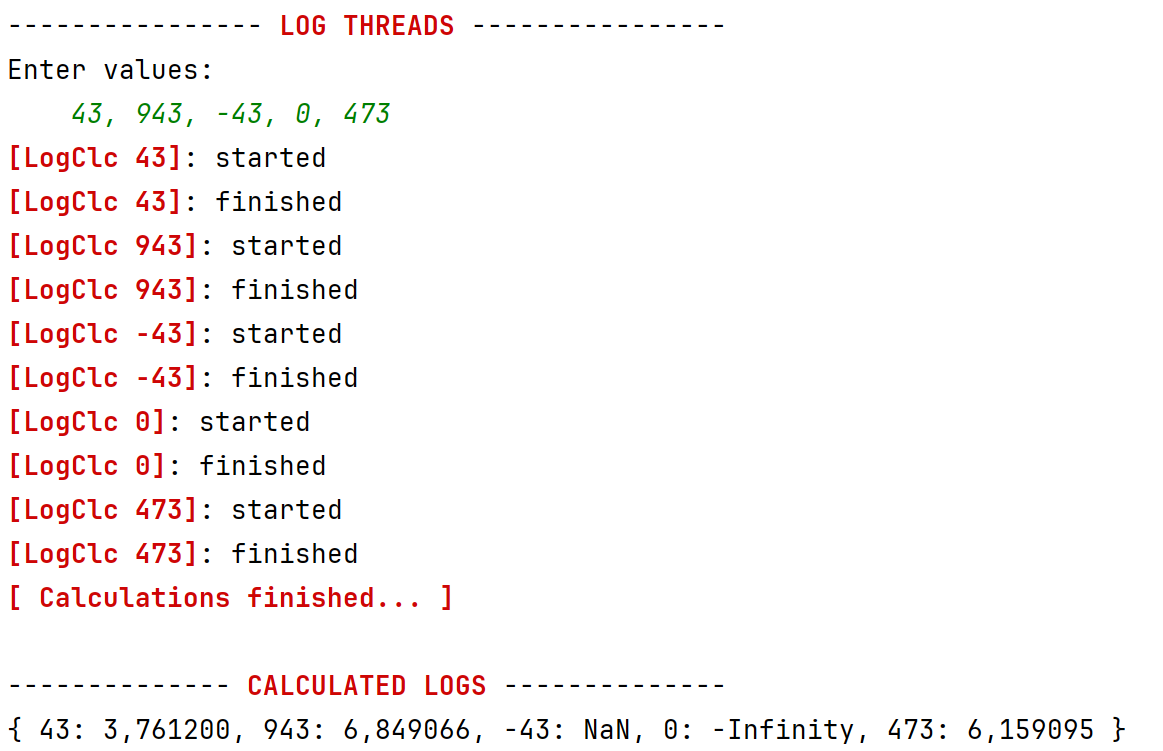
В настоящей лабораторной работе следует разработать программу, которая будет вычислять логарифмы для нескольких чисел параллельно. Каждый поток должен обрабатывать одно число.

**РАЗРАБОТКА АЛГОРИТМА**

1. Создание точки входа в программу. Разработка класса "lab\_3" с функцией "main", создающего экземпляры класса логарифмов и вызывающего методы для работы с введенными данными.
   1. Написание функций ввода и вывода данных.
   2. Определение цикла, в котором создается и запускается поток для каждого введенного значения.
2. Создание класса "LogClc", наследуемого от "Thread" содержащего поток для вычисления логарифма числа.



*Рисунок 1 - Алгоритм программы*



*Рисунок 2 - Пример работы программы*

**ЛИСТИНГ ПРОГРАММЫ**

**«lab\_3.java»**

package lab\_3;  
  
import java.util.Scanner;  
  
public class lab\_3 {  
 public static void main(String[] args) throws InterruptedException {  
 System.*out*.print("---------------- \033[1;31mLOG THREADS\033[0m ----------------\n");  
 System.*out*.print("Enter values:\n\t");  
 *// считывание данных* Scanner in = new Scanner(System.*in*);  
 String[] inp\_data = (in.nextLine()).split("\\s\*,\\s\*");  
 in.close();  
  
 *// обработка введенных чисел* double[] result = new double[inp\_data.length];  
 LogClc new\_thread;  
 int i = 0;  
 do {  
 *// создание потока* new\_thread = new LogClc("LogClc " + inp\_data[i], Integer.*parseInt*(inp\_data[i]));  
 *// запуск потока* new\_thread.start();  
 *// обавление потока в очередь исполнения (по порядку)* new\_thread.join();  
 *// запись вычисленного значения в результирующий массив* result[i] = new\_thread.log;  
 i++;  
 } while (i < inp\_data.length);  
 System.*out*.println("\033[1;31m[ Calculations finished... ]\033[0m");  
  
 *// вывод результата* System.*out*.print("\n-------------- \033[1;31mCALCULATED LOGS\033[0m --------------\n");  
 System.*out*.print("{ ");  
 for (i = 0; i < inp\_data.length - 1; i++) {  
 System.*out*.printf("%s: %f, ", inp\_data[i], result[i]);  
 }  
 System.*out*.printf("%s: %f", inp\_data[inp\_data.length - 1], result[inp\_data.length - 1]);  
 System.*out*.print(" }");  
 }  
}

**«LogClc.java»**

package lab\_3;  
  
*// класс-обработчик потока*public class LogClc extends Thread {  
 float value;  
 double log;  
  
 *// конструктор класса  
 // задание имени потока и значения для вычисления* LogClc(String name, float value) {  
 super(name);  
 this.value = value;  
 }  
  
 *// получение вычисленного значения* public double getLog() {  
 return this.log;  
 }  
  
 *// функция вычисления логарифма  
 // исполняется при запуске потока* @Override  
 public void run() {  
 System.*out*.printf("\033[1;31m[%s]\033[0m: started\n", Thread.*currentThread*().getName());  
  
 *// блок для обработки прерываний потока* try {  
 this.log = Math.*log*(this.value);  
 Thread.*sleep*(0);  
 } catch (InterruptedException e) {  
 System.*out*.printf("\033[1;31m[ %s has been interrupted ]\033[0m\n", Thread.*currentThread*().getName());  
 }  
  
 System.*out*.printf("\033[1;31m[%s]\033[0m: finished\n", Thread.*currentThread*().getName());  
 }  
}