Министерство образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №4

по курсу «Программирование на языке Java»

на тему «Многопоточность в Java»

Вариант №3

Выполнили:

студенты группы 19ВВ2

Ходакова Д. А.

Ходакова М. А.

Принял:

Юрова О.В.

Карамышева Н.С.

Пенза 2022

**Цель работы:** научиться создавать многопоточные приложения c использованием стандартных средств языка Java.

**Задание на лабораторную работу**

Модифицировать приложение из предыдущей лабораторной работы, реализовав вычисление определенного интеграла в нескольких дополнительных потоках (число потоков определяется номером варианта), снимая нагрузку с основного потока и предотвращая "подвисание" графического интерфейса. Варианты с номерами до 5 включительно реализуют многопоточность путем наследования от класса Thread, остальные реализуют интерфейс Runnable. Оформление лабораторной работы должно быть выполнено в соответствии с требованиями, приведенными в Приложении 2.

**Ход лабораторной работы:**

1. Создали новый класс MyThread.java, где прописали функцию run() для Thread, в которой расчитывается логарифм функции.

package mypackage;

import javax.swing.\*;

public class MyThread extends Thread {

private JTable jTable1;

private int i;

public MyThread(JTable jTable, int i) {

this.jTable1 = jTable;

this.i = i;

setName("Поток рассчета " + i);

}

@Override

public void run() {

super.run();

if (jTable1.getValueAt(i, 0) != null && jTable1.getValueAt(i, 1) != null && jTable1.getValueAt(i, 2) != null) {

Double result = 0.0;

Double bottom;

Double high;

Double step;

if (jTable1.getValueAt(i, 0) instanceof String) {

bottom = Double.parseDouble((String) jTable1.getValueAt(i, 0));

high = Double.parseDouble((String) jTable1.getValueAt(i, 1));

step = Double.parseDouble((String) jTable1.getValueAt(i, 2));

} else {

bottom = (Double) jTable1.getValueAt(i, 0);

high = (Double) jTable1.getValueAt(i, 1);

step = (Double) jTable1.getValueAt(i, 2);

}

while (bottom < high) {

double temp;

temp = Math.cos(bottom) / 2;

bottom += step;

temp += Math.cos(bottom) / 2;

result += temp \* step;

}

jTable1.setValueAt(result, i, 3);

}

}

}

1. Поправили код кнопки «Вычислить», реализовав вычисление логарифма в потоке:

int countRows = jTable1.getRowCount();

for (int i = 0; i < countRows; i++) {

MyThread myThread = new MyThread(jTable1, i);

myThread.start();

}

**Решение:**

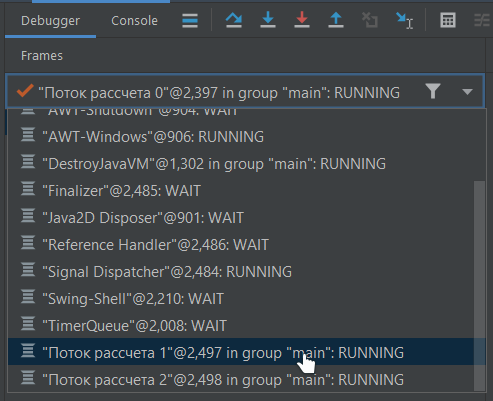


Рисунок 1 - Потоки 0, 1, 2.

**Вывод**

Научились создавать многопоточные приложения c использованием стандартных средств языка Java.