Министерство науки и высшего образования РФ

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №5

по дисциплине «разработка кроссплатформенных приложений»

на тему «Многопоточность в Java»

Вариант №2

Выполнили: студенты группы 22ВВВ1

Дунюшкин В.А.

НиколаевА.А.

Приняла:

Доцент Юрова О.В.

Пенза 2025

**Тема работы:**

изучить механизм многопоточности в Java.

**Ход работы:**

Модифицировать приложение из предыдущей лабораторной работы, реализовав вычисление определенного интеграла в нескольких дополнительных потоках, снимая нагрузку с основного потока и предотвращая "подвисание" графического интерфейса

**Описаниe:**

1. Создаём вложенный ParallelIntegral для организации параллельных вычислений. Интервал интегрирования делится на две части, каждая вычисляется в отдельном потоке.

private class ParallelIntegral {

private double low;

private double high;

private double step;

private int selectedRow;

private DefaultTableModel model;

public ParallelIntegral(double low, double high, double step,

int selectedRow, DefaultTableModel model) {

this.low = low;

this.high = high;

this.step = step;

this.selectedRow = selectedRow;

this.model = model;

}

public void calculate() {

// Разделяем интервал на две части

double mid = low + (high - low) / 2.0;

ExecutorService executor = Executors.newFixedThreadPool(2);

// Задачи для каждой половины интервала

Future<Double> futureFirstHalf = executor.submit(() ->

computeIntegral(low, mid, step/2));

Future<Double> futureSecondHalf = executor.submit(() ->

computeIntegral(mid, high, step/2));

try {

double firstResult = futureFirstHalf.get();

double secondResult = futureSecondHalf.get();

double totalResult = firstResult + secondResult;

SwingUtilities.invokeLater(() -> {

model.setValueAt(totalResult, selectedRow, 3);

listR.set(selectedRow, new Rect\_integral(high, low, step, totalResult));

});

} catch (InterruptedException | ExecutionException e) {

SwingUtilities.invokeLater(() ->

JOptionPane.showMessageDialog(null,

"Ошибка вычисления: " + e.getMessage(),

"Ошибка",

JOptionPane.ERROR\_MESSAGE));

} finally {

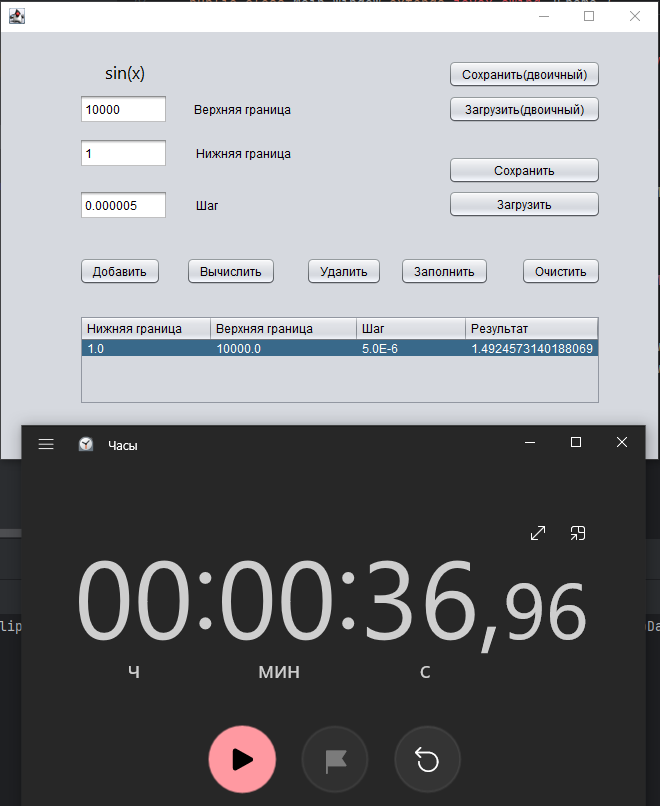
executor.shutdown();

}

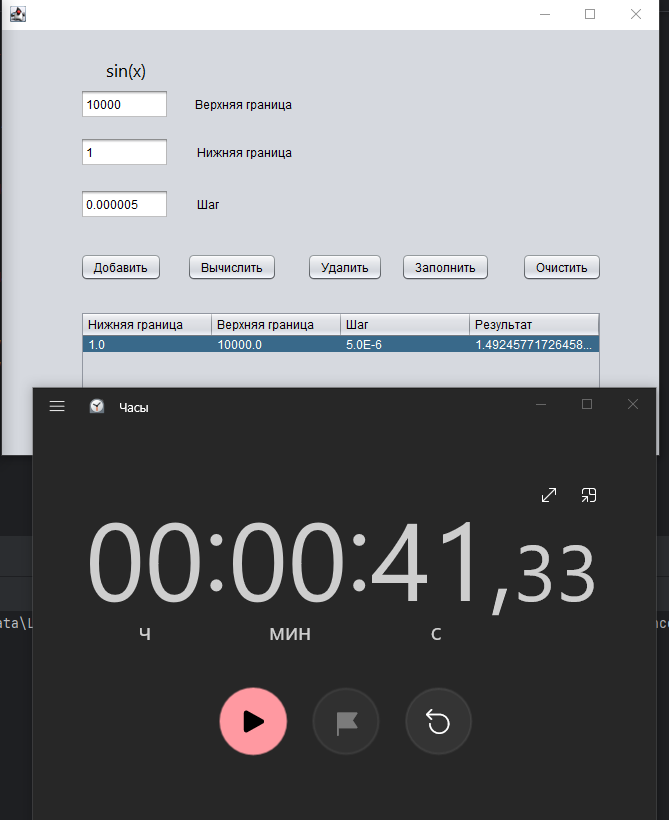
}

}

**Результат работы программы:**



Сравнение с предыдущей версией



**Вывод:**

Научились писать многопоточные приложения