Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №5

по курсу «Программирование на языке Java»

на тему: «Многопоточность в Java»

Вариант 9



Пенза 2024

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

**Задание:** 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. Реализовать многопоточность с помощью интерфейса Runnable.

**Листинг:**

JButton calculateButton = new JButton("Вычислить");

calculateButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

int subsetSize = dataTable.size() / 9; // Делим dataTable на 9 подмножеств

List<Thread> threads = new ArrayList<>(); // Создаем список для хранения потоков

for (int i = 0; i < 9; i++) {

int start = i \* subsetSize; // Вычисляем начало подмножества

int end = (i == 8) ? dataTable.size() : (i + 1) \* subsetSize; // Вычисляем конец подмножества

List<RecIntegral> subset = dataTable.subList(start, end); // Получаем подмножество данных

Thread thread = new Thread(new IntegrationCalculator(subset)); // Создаем поток для вычисления интегралов

threads.add(thread); // Добавляем поток в список

thread.start(); // Запускаем поток

}

// Ждем завершения всех потоков

for (Thread thread : threads) {

try {

thread.join(); // Ждем завершения каждого потока

} catch (InterruptedException ex) {

Thread.currentThread().interrupt();

}

}

tableModel.setRowCount(0);

for (RecIntegral recIntegral : dataTable)

tableModel.addRow(recIntegral.getDataAsStringArray());

}

});

// Создаем класс, реализующий интерфейс Runnable

class IntegrationCalculator implements Runnable {

private List<RecIntegral> dataSubset; // Список данных для вычисления интегралов

public IntegrationCalculator(List<RecIntegral> dataSubset) {

this.dataSubset = dataSubset;

}

@Override

public void run() {

// Выполняем вычисление интеграла для подмножества

for (RecIntegral recIntegral : dataSubset) {

// Выполняем вычисление интеграла для каждого recIntegral

double result = calculateIntegration(recIntegral.getLowerBound(), recIntegral.getUpperBound(), recIntegral.getIntervalLength());

recIntegral.setResult(result);

}

}

}

**Результаты работы программы:**

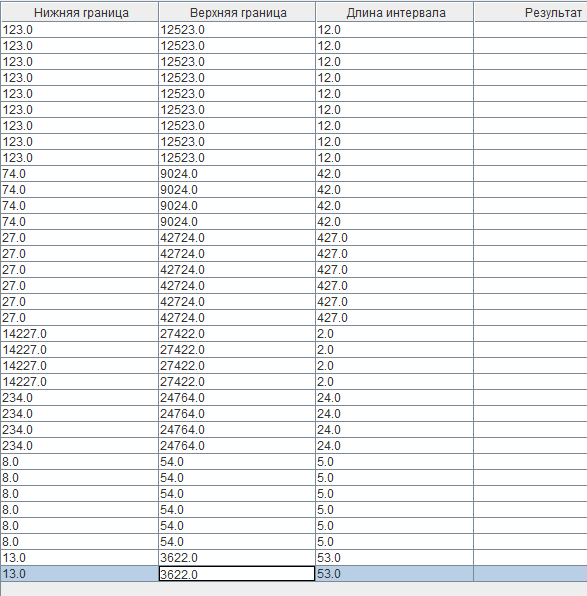


Рис 1 - Заполнили таблицу

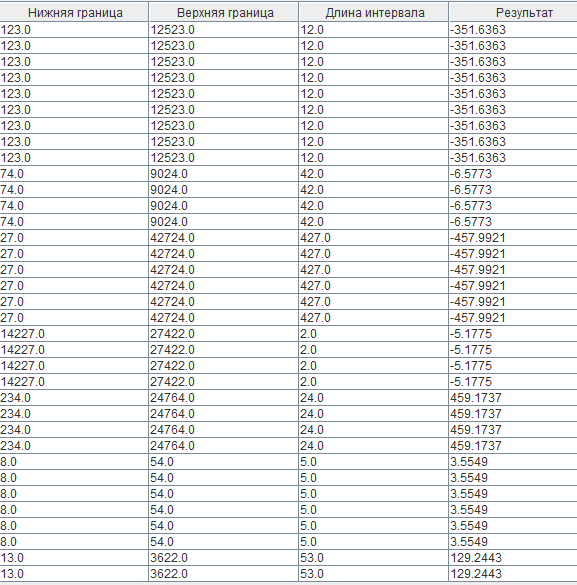


Рис 2 - Вычислили интегралы

**Вывод:** мы научились создавать многопоточные приложения c использованием стандартных средств языка Java.