Министерство науки и высшего образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №6

по курсу “Программирование на языке Java”

на тему: “ Сетевое взаимодействие в Java”

Вариант 4

Выполнили студенты группы 21ВВП2:

Тюкалов В.Е.

Чинов Д.Д.

Рузляев Д.А.

Приняли:

к.т.н., доцент Юрова О.В.,

к.т.н., доцент Карамышева Н.С.

Пенза 2024

**Цель работы**

Научиться создавать многопоточные приложения c использованием стандартных средств языка Java.

**Задание на лабораторную работу**

Модифицировать приложение из предыдущей лабораторной работы, реализовав клиент-серверную архитектуру, обеспечивающую распределенное вычисление определенного интеграла на нескольких вычислительных узлах (клиентах) при этом каждый узел использует несколько нитей, как в предыдущей работе. Сервер не занимается вычислениями, а лишь реализует взаимодействие с пользователем и агрегацию результатов вычислений от клиентов. Нечетные варианты используют протокол UDP, а четные TCP. Оформление лабораторной работы должно быть выполнено в соответствии с требованиями, приведенными в Приложении 2.

**Ход работы**

Разработали клиент-серверную архитектуру программы, которая представлена на рисунке 1.

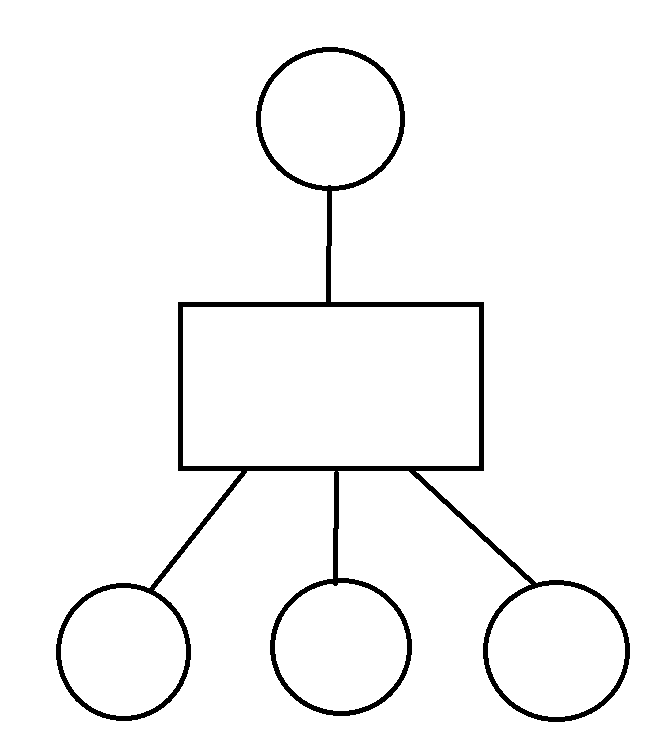


Рисунок 1 – клиент-серверная архитектура

**Листинг программы:**

/\*

\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

\* Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template

\*/

package com.mycompany.java\_course;

import java.io.\*;

import java.net.Socket;

import java.util.ArrayList;

import java.util.concurrent.ExecutorService;

import java.util.concurrent.Executors;

import java.util.concurrent.ThreadPoolExecutor;

import java.util.concurrent.TimeUnit;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JFileChooser;

import javax.swing.table.DefaultTableModel;

/\*\*

\*

\* @author Василий

\*/

public class Form extends javax.swing.JFrame {

/\*\*

\* Creates new form Lb1\_form

\*/

public Form() {

initComponents();

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

stepInt = new javax.swing.JTextField();

startPoint = new javax.swing.JTextField();

finishPoint = new javax.swing.JTextField();

jScrollPane1 = new javax.swing.JScrollPane();

table = new javax.swing.JTable();

addBtn = new javax.swing.JButton();

clearBtn = new javax.swing.JButton();

resBtn = new javax.swing.JButton();

jLabel1 = new javax.swing.JLabel();

jLabel2 = new javax.swing.JLabel();

jLabel3 = new javax.swing.JLabel();

printData = new javax.swing.JButton();

clearTable = new javax.swing.JButton();

save = new javax.swing.JButton();

load = new javax.swing.JButton();

save2 = new javax.swing.JButton();

load2 = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

addWindowListener(new java.awt.event.WindowAdapter() {

public void windowClosing(java.awt.event.WindowEvent evt) {

formWindowClosing(evt);

}

public void windowOpened(java.awt.event.WindowEvent evt) {

formWindowOpened(evt);

}

});

stepInt.setName("step"); // NOI18N

table.setModel(new javax.swing.table.DefaultTableModel(

new Object [][] {

{null, null, null, null},

{null, null, null, null},

{null, null, null, null},

{null, null, null, null}

},

new String [] {

"Title 1", "Title 2", "Title 3", "Title 4"

}

));

jScrollPane1.setViewportView(table);

addBtn.setText("Add");

addBtn.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

addBtnActionPerformed(evt);

}

});

clearBtn.setText("Delete");

clearBtn.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

clearBtnActionPerformed(evt);

}

});

resBtn.setText("Result");

resBtn.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

resBtnActionPerformed(evt);

}

});

jLabel1.setText("Step");

jLabel2.setText("Start point");

jLabel3.setText("Finish point");

printData.setText("Print data");

printData.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

printDataActionPerformed(evt);

}

});

clearTable.setText("Clear data");

clearTable.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

clearTableActionPerformed(evt);

}

});

save.setText("Save");

save.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

saveActionPerformed(evt);

}

});

load.setText("Load");

load.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

loadActionPerformed(evt);

}

});

save2.setText("Save (2)");

save2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

save2ActionPerformed(evt);

}

});

load2.setText("Load (2)");

load2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

load2ActionPerformed(evt);

}

});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(76, 76, 76)

.addComponent(printData, javax.swing.GroupLayout.PREFERRED\_SIZE, 100, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(37, 37, 37)

.addComponent(clearTable, javax.swing.GroupLayout.PREFERRED\_SIZE, 100, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addGroup(layout.createSequentialGroup()

.addGap(17, 17, 17)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(stepInt, javax.swing.GroupLayout.DEFAULT\_SIZE, 100, Short.MAX\_VALUE)

.addComponent(jLabel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addGap(34, 34, 34)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(startPoint, javax.swing.GroupLayout.PREFERRED\_SIZE, 100, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED\_SIZE, 100, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(finishPoint, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.PREFERRED\_SIZE, 100, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel3, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.PREFERRED\_SIZE, 100, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)

.addGroup(layout.createSequentialGroup()

.addComponent(addBtn, javax.swing.GroupLayout.PREFERRED\_SIZE, 100, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(31, 31, 31)

.addComponent(clearBtn, javax.swing.GroupLayout.PREFERRED\_SIZE, 100, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(resBtn, javax.swing.GroupLayout.PREFERRED\_SIZE, 100, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addComponent(jScrollPane1, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.PREFERRED\_SIZE, 0, Short.MAX\_VALUE)

.addGroup(layout.createSequentialGroup()

.addComponent(save, javax.swing.GroupLayout.PREFERRED\_SIZE, 80, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addComponent(load, javax.swing.GroupLayout.PREFERRED\_SIZE, 80, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addComponent(save2, javax.swing.GroupLayout.PREFERRED\_SIZE, 80, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(load2, javax.swing.GroupLayout.PREFERRED\_SIZE, 80, javax.swing.GroupLayout.PREFERRED\_SIZE))))

.addGap(19, 19, 19))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(24, 24, 24)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel1)

.addComponent(jLabel2)

.addComponent(jLabel3))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(stepInt, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(startPoint, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(finishPoint, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 18, 18)

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 279, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(addBtn)

.addComponent(clearBtn)

.addComponent(resBtn))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(printData)

.addComponent(clearTable))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(save2)

.addComponent(load)

.addComponent(save)

.addComponent(load2))

.addContainerGap(24, Short.MAX\_VALUE))

);

stepInt.getAccessibleContext().setAccessibleName("step");

pack();

}// </editor-fold>

private int countElemInTable = 0;

private static boolean windowClose = false;

private static ServerProcess Server;

private static ExecutorService executeIt = Executors.newFixedThreadPool(1);

private void addBtnActionPerformed(java.awt.event.ActionEvent evt) {

// Проверка наличия данных в полях заполнения

if (startPoint.getText() == null || "".equals(startPoint.getText()))

return;

if (finishPoint.getText() == null || "".equals(finishPoint.getText()))

return;

if (stepInt.getText() == null || "".equals(stepInt.getText())

|| "0".equals(stepInt.getText()))

return;

int start = Integer.parseInt(startPoint.getText());

int finish = Integer.parseInt(finishPoint.getText());

double step = Double.parseDouble(stepInt.getText());

// Ввод данных в таблицу

DefaultTableModel tbModel = (DefaultTableModel)table.getModel();

tbModel.insertRow(countElemInTable, new Object[] {step, start, finish});

stepInt.setText("");

startPoint.setText("");

finishPoint.setText("");

RecIntegral.WritingRecordFromATable((DefaultTableModel)table.getModel(), countElemInTable);

countElemInTable++;

}

private void clearBtnActionPerformed(java.awt.event.ActionEvent evt) {

int index = table.getSelectedRow();

if (table.getSelectedRow() == -1)

return;

DefaultTableModel tbModel = (DefaultTableModel)table.getModel();

RecIntegral.deleteRecord(tbModel, index);

tbModel.removeRow(index);

countElemInTable--;

}

// Распределение вычеслений на разные потоки

private void dataCalculation(int num)

{

double step = Double.parseDouble(table.getValueAt(num, 0).toString());

double start = Double.parseDouble(table.getValueAt(num, 1).toString());

double finish = Double.parseDouble(table.getValueAt(num, 2).toString());

double res = 0;

int NumStep = 4;

// Интегрирование в отдельных потоках

ArrayList<IntRes> resInt = new ArrayList<IntRes>();

for (int j = 0; j < NumStep; j++)

resInt.add(new IntRes(step,

start+(finish-start)\*j\*((double)1 / (double)NumStep),

start+(finish-start)\*(j+1)\*((double)1 / (double)NumStep)));

for (int j = 0; j < NumStep; j++)

resInt.get(j).start();

try

{

for (int j = 0; j < NumStep; j++)

resInt.get(j).join();

}

catch (InterruptedException ex) {

ex.getMessage();

}

for (int j = 0; j < NumStep; j++)

res += resInt.get(j).getRes();

table.getModel().setValueAt(res, num, 3);

RecIntegral.setResult((DefaultTableModel)table.getModel(), num, res);

}

private static int NumStep = 5;

private void resBtnActionPerformed(java.awt.event.ActionEvent evt) {

// Считывание данных из таблицы, рассчет результата

// и ввод результата в таблицу

for (int i = 0; i < table.getRowCount(); i++)

{

if ("".equals(table.getValueAt(i, 0).toString())

|| table.getValueAt(i, 0) == null)

continue;

// Считывание данных из таблицы

double step = Double.parseDouble(table.getValueAt(i, 0).toString());

double start = Double.parseDouble(table.getValueAt(i, 1).toString());

double finish = Double.parseDouble(table.getValueAt(i, 2).toString());

// Строка для 5 лаб. работы

//dataCalculation(i);

// Создание клиентов

for (int j = 0; j < NumStep; j++)

clients.add(new Client());

// Запуск клиентов

for (var item : clients)

item.start();

// Передача данных серверу

Server.setValues(step, start, finish);

// Ожидадание завершения работы клиентов

try {

for (var item : clients)

item.join();

} catch (InterruptedException ex) {

Logger.getLogger(Form.class.getName()).log(Level.SEVERE, null, ex);

}

// Вывод результатов

table.setValueAt(Server.getRes(), i, 3);

// Очистка массива клиентов

clients.clear();

}

}

private void formWindowOpened(java.awt.event.WindowEvent evt) {

// Изменение названия заголовков

table.getColumn("Title 1").setHeaderValue("Step");

table.getColumn("Title 2").setHeaderValue("Start");

table.getColumn("Title 3").setHeaderValue("Finish");

table.getColumn("Title 4").setHeaderValue("Result");

table.repaint();

DefaultTableModel tbModel = (DefaultTableModel)table.getModel();

tbModel.setRowCount(0);

}

private void printDataActionPerformed(java.awt.event.ActionEvent evt) {

DefaultTableModel tbModel = (DefaultTableModel)table.getModel();

tbModel.setRowCount(0);

countElemInTable = 0;

countElemInTable = RecIntegral.PrintDataInTable((DefaultTableModel)table.getModel(), countElemInTable);

}

private void clearTableActionPerformed(java.awt.event.ActionEvent evt) {

DefaultTableModel tbModel = (DefaultTableModel)table.getModel();

tbModel.setRowCount(0);

countElemInTable = 0;

// try

// {

// RecIntegral.WritingRecordFromATable((DefaultTableModel)table.getModel());

// countElemInTable = 0;

// }

// catch (MyException ex)

// {

// JOptionPane.showMessageDialog(null, ex);

// }

}

private JFileChooser fileChooser()

{

JFileChooser fileChooser = new JFileChooser();

fileChooser.setDialogTitle("Выбор файла");

// Определение режима - только каталог

fileChooser.setFileSelectionMode(JFileChooser.FILES\_ONLY);

int result = fileChooser.showOpenDialog(null);

return fileChooser;

}

private void saveActionPerformed(java.awt.event.ActionEvent evt) {

JFileChooser fileChooser = fileChooser();

// Открыл файл для записи

try (FileWriter file = new FileWriter(fileChooser.getSelectedFile().getAbsolutePath()))

{

// Записал данные в файл

DefaultTableModel tb = (DefaultTableModel)table.getModel();

for (int i = 0; i < tb.getRowCount(); i++)

{

for (int j = 0; j < tb.getColumnCount(); j++)

file.write(tb.getValueAt(i, j).toString() + " ");

file.write("\n");

}

}

catch (IOException ex)

{

ex.printStackTrace();

}

}

private void loadActionPerformed(java.awt.event.ActionEvent evt) {

JFileChooser fileChooser = fileChooser();

// Открыл файл для записи

try (FileReader file = new FileReader(fileChooser.getSelectedFile().getAbsolutePath()))

{

BufferedReader br = new BufferedReader(file);

// Записал данные из файла

DefaultTableModel tb = (DefaultTableModel)table.getModel();

String line;

int count = 0;

while ((line = br.readLine()) != null)

{

String[] strArr = line.split(" ");

tb.insertRow(count, new Object[] {strArr[0], strArr[1],

strArr[2], strArr[3]});

count++;

}

}

catch (IOException ex)

{

ex.printStackTrace();

}

}

private void save2ActionPerformed(java.awt.event.ActionEvent evt) {

JFileChooser fileChooser = fileChooser();

// Открыл файл для записи

var path = fileChooser.getSelectedFile().getAbsolutePath();

try (ObjectOutputStream file = new ObjectOutputStream(new FileOutputStream(path)))

{

// Записал данные в файл

DefaultTableModel tb = (DefaultTableModel)table.getModel();

for (int i = 0; i < tb.getRowCount(); i++)

{

file.writeObject(new Result(tb.getValueAt(i, 0).toString(),

tb.getValueAt(i, 1).toString(),

tb.getValueAt(i, 2).toString(),

tb.getValueAt(i, 3).toString()));

}

}

catch (IOException ex)

{

ex.printStackTrace();

}

}

private void load2ActionPerformed(java.awt.event.ActionEvent evt) {

JFileChooser fileChooser = fileChooser();

// Открыл файл для записи

var path = fileChooser.getSelectedFile().getAbsolutePath();

try(ObjectInputStream file = new ObjectInputStream(new FileInputStream(path)))

{

Result obj;

while ((obj = (Result)file.readObject()) != null)

{

DefaultTableModel tb = (DefaultTableModel)table.getModel();

int count = 0;

tb.insertRow(count, new Object[] {obj.getS1(),

obj.getS2(),

obj.getS3(),

obj.getS4()});

}

}

catch (IOException ex)

{

ex.printStackTrace();

}

catch (ClassNotFoundException ex)

{

ex.getMessage();

}

}

private void formWindowClosing(java.awt.event.WindowEvent evt) {

windowClose = true;

//executeIt.shutdown();

}

//static ThreadPoolExecutor executeClients = (ThreadPoolExecutor)Executors.newFixedThreadPool(5);

static ArrayList<Thread> clients = new ArrayList<Thread>();

private ArrayList<Socket> arrSocket = new ArrayList<Socket>();

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[])

{

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Form.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Form.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Form.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Form.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

//</editor-fold>

//</editor-fold>

//</editor-fold>

/\* Create and display the Form \*/

java.awt.EventQueue.invokeLater(new Runnable()

{

public void run()

{

new Form().setVisible(true);

Server = new ServerProcess();

executeIt.execute(Server);

}

});

}

// Variables declaration - do not modify

private javax.swing.JButton addBtn;

private javax.swing.JButton clearBtn;

private javax.swing.JButton clearTable;

private javax.swing.JTextField finishPoint;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JButton load;

private javax.swing.JButton load2;

private javax.swing.JButton printData;

private javax.swing.JButton resBtn;

private javax.swing.JButton save;

private javax.swing.JButton save2;

private javax.swing.JTextField startPoint;

private javax.swing.JTextField stepInt;

private javax.swing.JTable table;

// End of variables declaration

}

**Результат работы программы**

Результат работы программы приведён на рисунке 2.

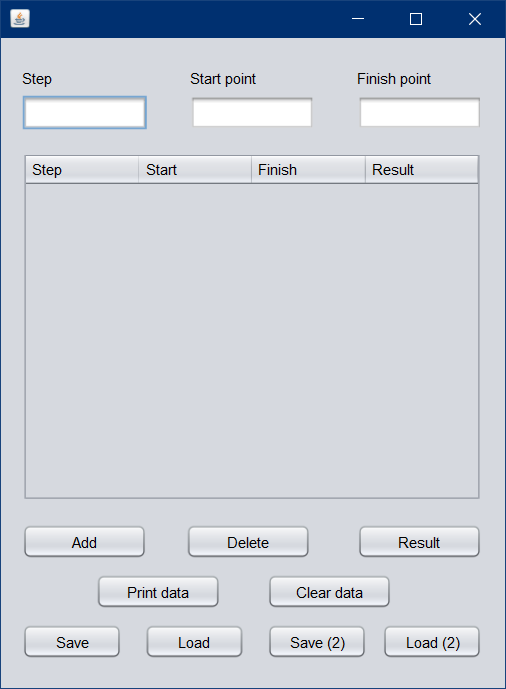


Рисунок 2 – Результат работы программы

**Вывод:** в ходе выполнения данной лабораторной работы мы ознакомились с сетевым программированнием на языке JAVA, созданием приложений с графическим интерфейсом с помощью библиотеки Swing.