Выполнили:

студенты группы 19ВВ2

Махнаев Н.С.

Урядов В.Д.

Трошкин В.К.

Приняли:

Карамышева Н.С

Юрова О.В.

Пенза 2022

ОТЧЕТ

по лабораторной работе №6

по курсу «Программирование на языке Java»

на тему «Сетевое взаимодействие в Java»

Министерство образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**Цель работы**: научиться создавать клиент-серверные приложения c использованием стандартных классов 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. Протокол TCP.

Листинг:

Client.java:

package Var8;

import java.io.\*;

import java.net.\*;

public class Client {

String result;

public String getRes() {

return this.result;

}

public void cln(String l, String h, String s) {

try (Socket socket = new Socket("192.168.56.1", 80);

BufferedWriter writer = new BufferedWriter(new OutputStreamWriter(socket.getOutputStream()));

BufferedReader reader = new BufferedReader(new InputStreamReader(socket.getInputStream()))) {

System.out.println("Соеденение с сервером");

String request = l + " " + h + " " + s;

System.out.println("Запрос:" + request);

writer.write(request);

writer.newLine();

writer.flush();

result = reader.readLine();

System.out.println("Ответ:" + result);

} catch (Exception e) {

e.printStackTrace();

}

}

}

Server.java:

package Var8;

import java.io.\*;

import java.net.\*;

import java.util.logging.Level;

import java.util.logging.Logger;

public class Server {

public static void main(String[] args) {

try ( ServerSocket server = new ServerSocket(80)) {

System.out.println("Сервер запущен");

while (true) {

try ( Socket socket = server.accept(); BufferedWriter writer = new BufferedWriter(new OutputStreamWriter(socket.getOutputStream())); BufferedReader reader = new BufferedReader(new InputStreamReader(socket.getInputStream()))) {

String result;

String[] values;

String request = reader.readLine();

values = request.split(" ");

System.out.println("Запрос:" + values[0] + values[1] + values[2]);

String l = values[0];

String h = values[1];

String s = values[2];

ThreadRunable threadRunable = new ThreadRunable(l, h, s);

threadRunable.run();

result = threadRunable.getResult();

System.out.println("Ответ:" + result);

String buf = result;

writer.write(buf);

writer.newLine();

writer.flush();

} catch (NullPointerException e) {

e.printStackTrace();

}

}

} catch (IOException e) {

throw new RuntimeException(e);

}

}

}

ThreadRunnable.java:

package Var8;

import java.io.\*;

import java.net.\*;

import java.util.LinkedList;

public class ThreadRunable implements Runnable {

private double start;

private double end;

private double step;

private String resultIntegral;

public ThreadRunable(String l, String h, String s) {

start = Double.valueOf(l);

end = Double.valueOf(h);

step = Double.valueOf(s);

}

@Override

public void run() {

double n = (end - start) / step;

double result = 0;

for (int i = 1; i < n - 1; i++) {

double index = start + i \* step;

result += (Math.sin(Math.pow(index, 2)));

}

result += (Math.sin(Math.pow(start, 2)) + Math.sin(Math.pow(end, 2))) / 2;

result = step \* result;

resultIntegral = Double.toString(result);

}

public String getResult() {

return this.resultIntegral;

}

}

NumExcetion.java:

public class NumException extends Exception{

String message;

NumException(String except\_message)

{

this.message = except\_message;

}

@Override

public String toString()

{

return ("NumException trigger: " + this.message);

}

}

RecIntegral.java:

public class RecIntegral implements Serializable {

private String lowerIntegral;

private String upperIntegral;

private String integralStep;

private String integralResult;

public RecIntegral(String lowerIntegral, String upperIntegral, String integralStep) throws NumException {

if (Double.valueOf(lowerIntegral) < 0.000001

|| Double.valueOf(lowerIntegral) > 1000000

|| Double.valueOf(upperIntegral) < 0.000001

|| Double.valueOf(upperIntegral) > 1000000

|| Double.valueOf(integralStep) < 0.000001

|| Double.valueOf(integralStep) > 1000000) {

throw new NumException("Numbers must be between 0.000001 and 1000000");

}

this.lowerIntegral = lowerIntegral;

this.upperIntegral = upperIntegral;

this.integralStep = integralStep;

this.integralResult = "0";

}

public RecIntegral(String lowerIntegral, String upperIntegral, String integralStep, String integralResult) throws NumException {

if (Double.valueOf(lowerIntegral) < 0.000001

|| Double.valueOf(lowerIntegral) > 1000000

|| Double.valueOf(upperIntegral) < 0.000001

|| Double.valueOf(upperIntegral) > 1000000

|| Double.valueOf(integralStep) < 0.000001

|| Double.valueOf(integralStep) > 1000000) {

throw new NumException("Numbers must be between 0.000001 and 1000000");

}

this.lowerIntegral = lowerIntegral;

this.upperIntegral = upperIntegral;

this.integralStep = integralStep;

this.integralResult = integralResult;

}

public void setResult(String integralResult) {

this.integralResult = integralResult;

}

public String getLowerIntegral() {

return this.lowerIntegral;

}

public String getUpperIntegral() {

return this.upperIntegral;

}

public String getIntegralStep() {

return this.integralStep;

}

public String getIntegralResult() {

return this.integralResult;

}

public double calculateIntegral() {

double l = Double.valueOf(this.lowerIntegral);

double h = Double.valueOf(this.upperIntegral);

double s = Double.valueOf(this.integralStep);

double n = (h - l) / s;

double result = 0;

for (int i = 1; i < n - 1; i++) {

double index = l + i \* s;

result += (Math.sin(Math.pow(index, 2)));

}

result += (Math.sin(Math.pow(l, 2)) + Math.sin(Math.pow(h, 2))) / 2;

result = s \* result;

this.integralResult = Double.toString(result);

return result;

}

}

JFrame.java:

package Var8;

import java.util.\*;

import javax.swing.JFileChooser;

import javax.swing.JOptionPane;

import javax.swing.table.DefaultTableModel;

import java.io.\*;

public class JFrame extends javax.swing.JFrame {

/\*\*

\* Creates new form JFrame

\*/

public JFrame() {

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() {

jTextField1 = new javax.swing.JTextField();

jTextField2 = new javax.swing.JTextField();

jTextField3 = new javax.swing.JTextField();

jScrollPane1 = new javax.swing.JScrollPane();

jTable1 = new javax.swing.JTable();

jButton1 = new javax.swing.JButton();

jButton2 = new javax.swing.JButton();

jButton3 = new javax.swing.JButton();

jButton4 = new javax.swing.JButton();

label1 = new java.awt.Label();

label2 = new java.awt.Label();

label3 = new java.awt.Label();

jButton5 = new javax.swing.JButton();

jButton6 = new javax.swing.JButton();

jButton7 = new javax.swing.JButton();

jButton8 = new javax.swing.JButton();

jButton9 = new javax.swing.JButton();

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

jTextField1.setText("0");

jTextField2.setText("0");

jTextField3.setText("0");

jTable1.setModel(new javax.swing.table.DefaultTableModel(

new Object [][] {

},

new String [] {

"Lower", "Upper", "Step", "Result"

}

));

jScrollPane1.setViewportView(jTable1);

jButton1.setText("Add");

jButton1.addActionListener(new java.awt.event.ActionListener() {

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

jButton1ActionPerformed(evt);

}

});

jButton2.setText("Delete");

jButton2.addActionListener(new java.awt.event.ActionListener() {

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

jButton2ActionPerformed(evt);

}

});

jButton3.setText("Calculate");

jButton3.addActionListener(new java.awt.event.ActionListener() {

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

jButton3ActionPerformed(evt);

}

});

jButton4.setText("Load list");

jButton4.addActionListener(new java.awt.event.ActionListener() {

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

jButton4ActionPerformed(evt);

}

});

label1.setText("Step:");

label2.setText("Lower:");

label3.setText("Upper:");

jButton5.setText("Clear");

jButton5.addActionListener(new java.awt.event.ActionListener() {

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

jButton5ActionPerformed(evt);

}

});

jButton6.setText("Save bin");

jButton6.addActionListener(new java.awt.event.ActionListener() {

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

jButton6ActionPerformed(evt);

}

});

jButton7.setText("Load bin");

jButton7.addActionListener(new java.awt.event.ActionListener() {

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

jButton7ActionPerformed(evt);

}

});

jButton8.setText("Save text");

jButton8.addActionListener(new java.awt.event.ActionListener() {

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

jButton8ActionPerformed(evt);

}

});

jButton9.setText("Load text");

jButton9.addActionListener(new java.awt.event.ActionListener() {

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

jButton9ActionPerformed(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(33, 33, 33)

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

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 498, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGroup(layout.createSequentialGroup()

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

.addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED\_SIZE, 90, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(label2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 18, 18)

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

.addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED\_SIZE, 90, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(label3, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 18, 18)

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

.addComponent(label1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jTextField3, javax.swing.GroupLayout.PREFERRED\_SIZE, 90, javax.swing.GroupLayout.PREFERRED\_SIZE))))

.addGap(18, 18, 18)

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

.addComponent(jButton9, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

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

.addComponent(jButton3, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jButton2, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jButton1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jButton4, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jButton5, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jButton6, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jButton7, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jButton8, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)))

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

);

layout.setVerticalGroup(

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

.addGroup(layout.createSequentialGroup()

.addContainerGap()

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

.addGroup(layout.createSequentialGroup()

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 257, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

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

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

.addComponent(label2, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.DEFAULT\_SIZE, 22, Short.MAX\_VALUE)

.addComponent(label3, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addComponent(label1, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.PREFERRED\_SIZE, 20, javax.swing.GroupLayout.PREFERRED\_SIZE))

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

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

.addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jTextField2)

.addComponent(jTextField3, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addContainerGap(53, Short.MAX\_VALUE))

.addGroup(layout.createSequentialGroup()

.addComponent(jButton1)

.addGap(18, 18, 18)

.addComponent(jButton2)

.addGap(18, 18, 18)

.addComponent(jButton3)

.addGap(18, 18, 18)

.addComponent(jButton4)

.addGap(18, 18, 18)

.addComponent(jButton5)

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

.addComponent(jButton6)

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

.addComponent(jButton7)

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

.addComponent(jButton8)

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

.addComponent(jButton9)

.addContainerGap(25, Short.MAX\_VALUE))))

);

pack();

}// </editor-fold>

LinkedList<RecIntegral> recIntegral = new LinkedList<>();

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

try {

DefaultTableModel dt = (DefaultTableModel) jTable1.getModel();

recIntegral.add(0, new RecIntegral(jTextField1.getText(), jTextField2.getText(), jTextField3.getText()));

dt.insertRow(0, new Object[]{jTextField1.getText(), jTextField2.getText(), jTextField3.getText(), 0});

} catch (NumException ex) {

JOptionPane.showMessageDialog(null, ex);

}

}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

DefaultTableModel dt = (DefaultTableModel) jTable1.getModel();

int row = jTable1.getSelectedRow();

if (row != -1) {

dt.removeRow(jTable1.getSelectedRow());

recIntegral.remove(row);

}

}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

Client client = new Client();

DefaultTableModel dt = (DefaultTableModel) jTable1.getModel();

int row = jTable1.getSelectedRow();

String res;

if (row != -1) {

String h = recIntegral.get(row).getUpperIntegral();

String l = recIntegral.get(row).getLowerIntegral();

String s = recIntegral.get(row).getIntegralStep();

client.cln(l, h, s);

res = client.getRes();

recIntegral.get(row).setResult(res);

dt.setValueAt(recIntegral.get(row).getIntegralResult(), row, 3);

}

}

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {

DefaultTableModel dt = (DefaultTableModel) jTable1.getModel();

dt.setRowCount(0);

for (RecIntegral recInt : recIntegral) {

dt.addRow(new Object[]{recInt.getLowerIntegral(), recInt.getUpperIntegral(), recInt.getIntegralStep(), recInt.getIntegralResult()});

}

}

private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {

DefaultTableModel dt = (DefaultTableModel) jTable1.getModel();

dt.setRowCount(0);

}

private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {

JFileChooser fileChooser = new JFileChooser();

fileChooser.setDialogTitle("Save file binary");

int res = fileChooser.showSaveDialog(null);

if (res == JFileChooser.APPROVE\_OPTION) {

File fopen = fileChooser.getSelectedFile();

ObjectOutputStream saveArray = null;

try {

saveArray = new ObjectOutputStream(new BufferedOutputStream(new FileOutputStream(fopen)));

saveArray.writeObject(recIntegral);

} catch (IOException e) {

e.printStackTrace();

} finally {

try {

saveArray.close();

} catch (IOException e) {

e.printStackTrace();

}

}

}

}

private void jButton7ActionPerformed(java.awt.event.ActionEvent evt) {

DefaultTableModel dt = (DefaultTableModel) jTable1.getModel();

dt.setRowCount(0);

JFileChooser fileChooser = new JFileChooser();

fileChooser.setDialogTitle("Load file binary");

int res = fileChooser.showOpenDialog(null);

if (res == JFileChooser.APPROVE\_OPTION) {

File fopen = fileChooser.getSelectedFile();

ObjectInputStream loadArray = null;

try {

loadArray = new ObjectInputStream(new BufferedInputStream(new FileInputStream(fopen)));

recIntegral = (LinkedList) loadArray.readObject();

} catch (IOException e) {

e.printStackTrace();

} catch (ClassNotFoundException classErr) {

JOptionPane.showMessageDialog(null, classErr.getMessage());

} finally {

try {

loadArray.close();

} catch (IOException e) {

e.printStackTrace();

}

}

for (RecIntegral recInt : recIntegral) {

dt.addRow(new Object[]{recInt.getLowerIntegral(), recInt.getUpperIntegral(), recInt.getIntegralStep(), recInt.getIntegralResult()});

}

}

}

private void jButton8ActionPerformed(java.awt.event.ActionEvent evt) {

JFileChooser fileChooser = new JFileChooser();

fileChooser.setDialogTitle("Save file text");

int res = fileChooser.showSaveDialog(null);

if (res == JFileChooser.APPROVE\_OPTION) {

File fopen = fileChooser.getSelectedFile();

FileWriter fwriter = null;

try {

fwriter = new FileWriter(fopen);

for (RecIntegral recInt : recIntegral) {

fwriter.write(recInt.getLowerIntegral() + " " + recInt.getUpperIntegral() + " " + recInt.getIntegralStep() + " " + recInt.getIntegralResult() + "\r\n");

}

fwriter.close();

} catch (IOException e) {

e.printStackTrace();

} finally {

if (fwriter != null) {

try {

fwriter.close();

} catch (IOException e) {

e.printStackTrace();

}

}

}

}

}

private void jButton9ActionPerformed(java.awt.event.ActionEvent evt) {

DefaultTableModel dt = (DefaultTableModel) jTable1.getModel();

JFileChooser fileChooser = new JFileChooser();

fileChooser.setDialogTitle("Load file text");

int res = fileChooser.showOpenDialog(null);

if (res == JFileChooser.APPROVE\_OPTION) {

File fopen = fileChooser.getSelectedFile();

BufferedReader bufread = null;

FileReader fread = null;

String line;

String[] values;

try {

fread = new FileReader(fopen);

bufread = new BufferedReader(fread);

while (true) {

line = bufread.readLine();

if (line == null) {

break;

}

values = line.split(" ");

try {

recIntegral.add(0, new RecIntegral(values[0], values[1], values[2], values[3]));

dt.addRow(new Object[]{values[0], values[1], values[2], values[3]});

} catch (NumException e) {

e.printStackTrace();

}

}

} catch (IOException e) {

e.printStackTrace();

} finally {

try {

fread.close();

bufread.close();

} catch (IOException e) {

e.printStackTrace();

}

}

}

}

/\*\*

\* @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(JFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

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

} catch (IllegalAccessException ex) {

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

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

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

}

//</editor-fold>

/\* Create and display the form \*/

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

public void run() {

new JFrame().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JButton jButton1;

private javax.swing.JButton jButton2;

private javax.swing.JButton jButton3;

private javax.swing.JButton jButton4;

private javax.swing.JButton jButton5;

private javax.swing.JButton jButton6;

private javax.swing.JButton jButton7;

private javax.swing.JButton jButton8;

private javax.swing.JButton jButton9;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JTable jTable1;

private javax.swing.JTextField jTextField1;

private javax.swing.JTextField jTextField2;

private javax.swing.JTextField jTextField3;

private java.awt.Label label1;

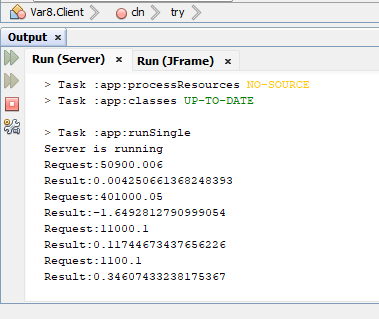
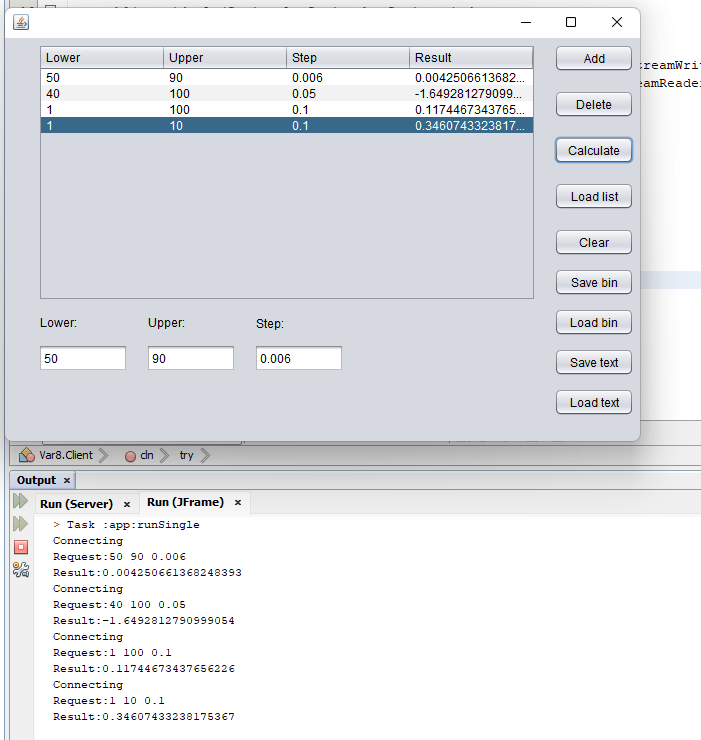
private java.awt.Label label2;

private java.awt.Label label3;

// End of variables declaration

}

Результат работы:



Вывод: В ходе лабораторной работы научились создавать клиент-серверные приложения c использованием стандартных классов Java.