6Пензенский Государственный университет

Кафедра «Вычислительная техника»

**Отчёт**

По лабораторной работе №6

по дисциплине: «Программирование на языке Java»

на тему: «Сетевое взаимодействие в Java»

Вариант №6

Выполнили студенты группы 20ВВП2:

Китаев А.А.

Новиков И.М.

Приняли:

Юрова О.В.

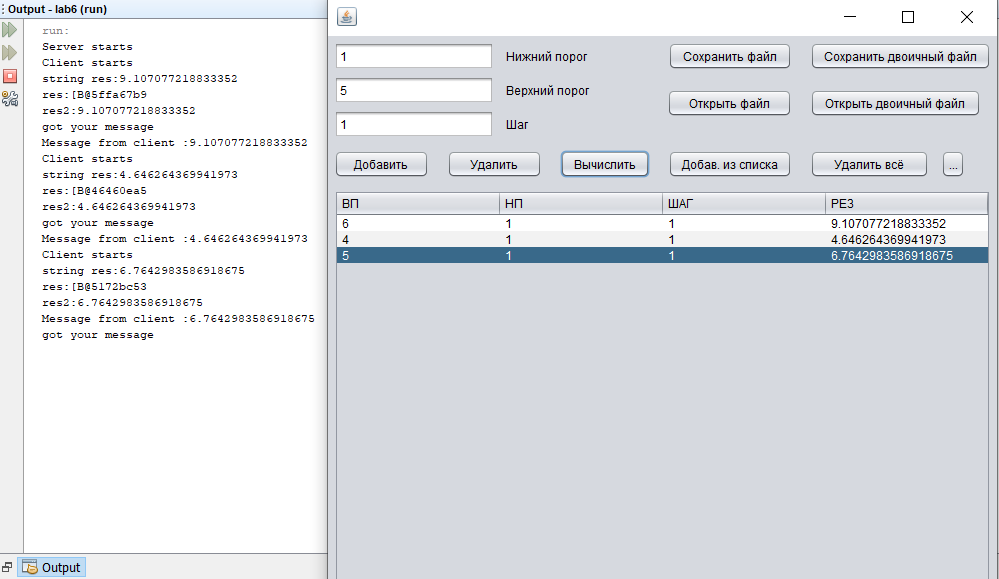
Карамышева Н.С.

Пенза 2023

**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. Нечетные варианты используют протокол UDP, а четные TCP.

**Решение:**



*Рисунок 1.*

**Вывод:** в ходе выполнения данной лабораторной работы были изучены методы сетевого взаимодействия в java.

**Листинг:**

**TestClient.java**

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

import java.util.logging.Level;

import java.util.logging.Logger;

import java.io.IOException;

import java.net.DatagramPacket;

import java.net.DatagramSocket;

import java.net.InetAddress;

import java.util.concurrent.BlockingQueue;

import java.util.concurrent.LinkedBlockingQueue;

import java.nio.ByteBuffer;

import java.nio.DoubleBuffer;

/\*\*

\*

\* @author user

\*/

public class TestClient implements Runnable {

private DatagramSocket socket;

private InetAddress address; // localhost

private int port; //server port - 8080

private BlockingQueue<byte[]> queue = new LinkedBlockingQueue<>();

private int maxPacketSize = 512;

private byte[] buffer = new byte[maxPacketSize];

private int[] timeouts = {11, 29, 73, 277, 997};

public TestClient(String ip, int port) throws IOException {

this.address = InetAddress.getByName(ip);

this.port = port;

this.socket = new DatagramSocket();

}

@Override

public void run() {

System.out.println("Client starts");

while(true) {

try {

byte[] message = queue.take();

String messageStr = new String(message); // преобразуем массив байтов в строку

String[] numbersStr = messageStr.split(","); // разбиваем строку на массив строк по разделителю ","

double[] numbers = new double[numbersStr.length]; // создаем массив чисел формата double

for (int i = 0; i < numbersStr.length; i++) {

numbers[i] = Double.parseDouble(numbersStr[i]);

//System.out.println("test1:" + numbers[i]);// преобразуем каждую строку в число формата double

}

//===================================================================

//double result = 0;

MyThread Thread1[] = new MyThread[5];

Thread A[] = new Thread[5];

for(int i = 0; i < 5; i ++){

double start = numbers[0];

double end = numbers[1];

double step = numbers[2];

Thread1[i] = new MyThread(end,start,step);

//System.out.println(Thread1[i].Tend);

A[i] = new Thread(Thread1[i]);

A[i].start();

try {

A[i].join();

} catch (InterruptedException ex) {

Logger.getLogger(frame1.class.getName()).log(Level.SEVERE, null, ex);

}

//System.out.println(Thread1[i].Tresult);

//model.setValueAt(Thread1[i].Tresult, i, 3);

}

String strResult = Double.toString(Thread1[1].Tresult);

System.out.println("string res:" + strResult);

//byte[] resultMessage = ByteBuffer.allocate(8).putDouble(Thread1[1].Tresult).array();

byte[] resultMessage = strResult.getBytes();

System.out.println("res:" + resultMessage);

System.out.println("res2:" + new String(resultMessage));

//===============================================================================================

DatagramPacket packetToServer = new DatagramPacket(resultMessage, resultMessage.length,

address, port);

for (int i = 0; i < timeouts.length ; i++) {

try {

socket.setSoTimeout(timeouts[i]);

socket.send(packetToServer);

DatagramPacket packetFromServer = new DatagramPacket(buffer, buffer.length);

socket.receive(packetFromServer);

System.out.println(new String(packetFromServer.getData()));

break;

} catch (IOException e) {

System.out.println("Fail: too long waiting");

}

}

} catch (Exception e) {

}

}

}

public boolean push(byte[] message) {

if (message.length < maxPacketSize) {

queue.add(message);

return true;

}

return false;

}

public void stop() {

socket.close();

Thread.currentThread().interrupt();

}

}

**TestServer.java**

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

import javax.swing.table.DefaultTableModel;

import java.io.IOException;

import java.net.DatagramPacket;

import java.net.DatagramSocket;

/\*\*

\*

\* @author user

\*/

public class TestServer implements Runnable {

private int maxPacketSize = 512;

private byte[] buffer = new byte[maxPacketSize];

private DatagramSocket socket;

public TestServer(int port) throws IOException {

this.socket = new DatagramSocket(port);

}

@Override

public void run() {

System.out.println("Server starts");

DefaultTableModel model = (DefaultTableModel)frame1.jTable2.getModel();

while (true) {

try {

DatagramPacket packetFromClient = new DatagramPacket(buffer, buffer.length);

socket.receive(packetFromClient); // blocking operation like Socket.accept()

byte[] response = "got your message".getBytes();

DatagramPacket packetToClient = new DatagramPacket(

response, response.length,

packetFromClient.getAddress(), packetFromClient.getPort());

socket.send(packetToClient);

model.setValueAt(new String(packetFromClient.getData()), frame1.jTable2.getSelectedRow(), 3);

System.out.println("Message from client :" + new String(packetFromClient.getData()));

} catch (IOException e) {

}

}

}

public void shutdown() {

this.socket.close();

Thread.currentThread().interrupt();

}

}

**valueException.java**

import javax.swing.JOptionPane;

public class valueException extends Exception{

public valueException() {

}

public valueException(String message) {

super(message);

}

public valueException(String message, Throwable cause) {

super(message, cause);

}

public valueException(Throwable cause) {

super(cause);

}

public valueException(String message, Throwable cause, booleanenableSuppression, booleanwritableStackTrace) {

super(message, cause, enableSuppression, writableStackTrace);

}

}

**RecIntegral.java**

public class RecIntegral implements Serializable{

public double start;

public double end;

public double step;

public double result ;

public RecIntegral(){

}

public RecIntegral(double \_start, double \_end, double \_step) throws valueException{

if(CheckMaxMin(\_start, 0.000001, 1000000)==false ||

CheckMaxMin(\_end, 0.000001, 1000000)==false ||

CheckMaxMin(\_step, 0.000001, 1000000)==false)

{

JOptionPane.showMessageDialog(null, "missing range", "Error", JOptionPane.ERROR\_MESSAGE);

throw new valueException("missing range");

}

else if((\_step > \_start - \_end)){

JOptionPane.showMessageDialog(null, "step can't be higher than interval", "Error", JOptionPane.ERROR\_MESSAGE);

throw new valueException("step can't be higher than interval");

}

else if (\_start < \_end){

JOptionPane.showMessageDialog(null, "end can't be higher than start", "Error", JOptionPane.ERROR\_MESSAGE);

throw new valueException("end can't be higher than start");

}

else{

start = \_start;

end = \_end;

step = \_step;

}

}

private booleanCheckMaxMin(double number, double min, double max){

if(number > min && number < max){

return true;

}

return false;

}

}

**Frame1.java**

import java.util.Set;

import javax.swing.table.DefaultTableModel;

import java.util.LinkedList;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JOptionPane;

import java.lang.Math;

import javax.swing.filechooser.FileNameExtensionFilter;

import java.io.\*;

import java.awt.\*;

import java.util.Vector;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import static java.lang.Thread.sleep;

import java.nio.file.Files;

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

/\*\*

\*

\* @author Vanya

\*/

public class frame1 extends javax.swing.JFrame {

/\*\*

\* Creates new form frame1

\*/

public frame1() {

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() {

jFileChooser1 = new javax.swing.JFileChooser();

addButton = new javax.swing.JButton();

deleteButton = new javax.swing.JButton();

jButton3 = new javax.swing.JButton();

jTextField1 = new javax.swing.JTextField();

jTextField2 = new javax.swing.JTextField();

jTextField3 = new javax.swing.JTextField();

jScrollPane2 = new javax.swing.JScrollPane();

jTable2 = new javax.swing.JTable();

addToTableFromList = new javax.swing.JButton();

deleteALLButton = new javax.swing.JButton();

jLabel1 = new javax.swing.JLabel();

jLabel2 = new javax.swing.JLabel();

jLabel3 = new javax.swing.JLabel();

saveFile = new javax.swing.JButton();

saveBinFile = new javax.swing.JButton();

openBinFile = new javax.swing.JButton();

openFile = new javax.swing.JButton();

startServer = new javax.swing.JButton();

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

addButton.setText("Добавить");

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

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

addButtonActionPerformed(evt);

}

});

deleteButton.setText("Удалить");

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

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

deleteButtonActionPerformed(evt);

}

});

jButton3.setText("Вычислить");

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

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

jButton3ActionPerformed(evt);

}

});

jTextField1.addActionListener(new java.awt.event.ActionListener() {

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

jTextField1ActionPerformed(evt);

}

});

jTextField2.addActionListener(new java.awt.event.ActionListener() {

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

jTextField2ActionPerformed(evt);

}

});

jTable2.setModel(new javax.swing.table.DefaultTableModel(

new Object [][] {

},

new String [] {

"ВП", "НП", "ШАГ", "РЕЗ"

}

));

jTable2.getTableHeader().setReorderingAllowed(false);

jTable2.addMouseListener(new java.awt.event.MouseAdapter() {

public void mouseClicked(java.awt.event.MouseEvent evt) {

jTable2MouseClicked(evt);

}

});

jScrollPane2.setViewportView(jTable2);

addToTableFromList.setText("Добав. из списка");

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

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

addToTableFromListActionPerformed(evt);

}

});

deleteALLButton.setText("Удалить всё");

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

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

deleteALLButtonActionPerformed(evt);

}

});

jLabel1.setText("Нижний порог");

jLabel2.setText("Верхний порог");

jLabel3.setText("Шаг");

saveFile.setText("Сохранить файл");

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

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

saveFileActionPerformed(evt);

}

});

saveBinFile.setText("Сохранить двоичный файл");

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

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

saveBinFileActionPerformed(evt);

}

});

openBinFile.setText("Открыть двоичный файл");

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

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

openBinFileActionPerformed(evt);

}

});

openFile.setText("Открыть файл");

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

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

openFileActionPerformed(evt);

}

});

startServer.setText("jButton1");

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

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

startServerActionPerformed(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()

.addContainerGap()

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

.addGroup(layout.createSequentialGroup()

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

.addGroup(layout.createSequentialGroup()

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

.addGroup(layout.createSequentialGroup()

.addComponent(addButton, javax.swing.GroupLayout.PREFERRED\_SIZE, 95, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addComponent(deleteButton, javax.swing.GroupLayout.PREFERRED\_SIZE, 95, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addComponent(jButton3))

.addGroup(layout.createSequentialGroup()

.addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED\_SIZE, 160, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED\_SIZE, 97, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addGap(18, 18, 18)

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

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

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

.addGroup(layout.createSequentialGroup()

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

.addGroup(layout.createSequentialGroup()

.addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED\_SIZE, 160, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

.addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED\_SIZE, 97, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(layout.createSequentialGroup()

.addComponent(jTextField3, javax.swing.GroupLayout.PREFERRED\_SIZE, 160, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

.addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED\_SIZE, 97, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addGap(64, 64, 64)

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

.addGap(18, 18, 18)

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

.addComponent(openBinFile, javax.swing.GroupLayout.PREFERRED\_SIZE, 171, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(saveBinFile)

.addGroup(layout.createSequentialGroup()

.addComponent(deleteALLButton, javax.swing.GroupLayout.PREFERRED\_SIZE, 119, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

.addComponent(startServer, javax.swing.GroupLayout.PREFERRED\_SIZE, 24, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addGap(0, 0, Short.MAX\_VALUE))

.addComponent(jScrollPane2))

.addContainerGap())

);

layout.setVerticalGroup(

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

.addGroup(layout.createSequentialGroup()

.addContainerGap()

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

.addComponent(jTextField2, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

.addComponent(jLabel1)

.addComponent(saveFile)

.addComponent(saveBinFile)))

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

.addGroup(layout.createSequentialGroup()

.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(jLabel2, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

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

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

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

.addComponent(jLabel3, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

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

.addGroup(layout.createSequentialGroup()

.addGap(19, 19, 19)

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

.addComponent(openBinFile)

.addComponent(openFile))

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

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

.addComponent(addButton)

.addComponent(deleteButton)

.addComponent(jButton3)

.addComponent(deleteALLButton)

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

.addComponent(startServer))

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

.addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap())

);

pack();

}// </editor-fold>

double vp,np,interval,result;

private LinkedList<RecIntegral> list = new LinkedList<>();

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

try {

DefaultTableModel model = (DefaultTableModel)jTable2.getModel();

list.add(new RecIntegral(Double.valueOf((String)jTextField1.getText()),

Double.valueOf((String)jTextField2.getText()),

Double.valueOf((String)jTextField3.getText()))

);

model.addRow(new Object[]{jTextField1.getText(),jTextField2.getText(),jTextField3.getText()});

} catch (valueException ex) {

Logger.getLogger(frame1.class.getName()).log(Level.SEVERE, null, ex);

//ex.printStackTrace();

return;

}

}

private void jTextField1ActionPerformed(java.awt.event.ActionEvent evt) {

}

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

DefaultTableModel model = (DefaultTableModel)jTable2.getModel();

list.remove(jTable2.getSelectedRow());

model.removeRow(jTable2.getSelectedRow());

}

private void jTable2MouseClicked(java.awt.event.MouseEvent evt) {

DefaultTableModel model = (DefaultTableModel)jTable2.getModel();

vp = Double.valueOf((String)model.getValueAt(jTable2.getSelectedRow(), 0));

np = Double.valueOf((String)model.getValueAt(jTable2.getSelectedRow(), 1));

interval = Double.valueOf((String)model.getValueAt(jTable2.getSelectedRow(), 2));

//System.out.print(vp+":"+np+":"+interval+"\n");

}

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

int port = 8080;

TestClient clientUDP = null;

try {

clientUDP = new TestClient("localhost", port);

} catch (IOException ex) {

Logger.getLogger(frame1.class.getName()).log(Level.SEVERE, null, ex);

}

//String mes = jTextField1.getText();

new Thread(clientUDP).start();

//clientUDP.push(mes.getBytes());

result = 0;

DefaultTableModel model = (DefaultTableModel)jTable2.getModel();

//MyThread Thread1[] = new MyThread[model.getRowCount()];

//Thread A[] = new Thread[model.getRowCount()];

String mes = "";

//for(int i = 0; i < model.getRowCount(); i ++){

// double start = Double.valueOf((String)jTextField1.getText());

// double end = Double.valueOf((String)jTextField2.getText());

// double step = Double.valueOf((String)jTextField3.getText());

String value0 = (String)model.getValueAt(jTable2.getSelectedRow(), 0);

String value1 = (String)model.getValueAt(jTable2.getSelectedRow(), 1);

String value2 = (String)model.getValueAt(jTable2.getSelectedRow(), 2);

mes = mes.concat(value0 + "," + value1 + "," + value2 + ",");

clientUDP.push(mes.getBytes());

mes = "";

}

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

DefaultTableModel model = (DefaultTableModel)jTable2.getModel();

for(int row = 0; row<list.size(); row++){

model.addRow(new Object[]{list.get(row).start,list.get(row).end,list.get(row).step});

};

}

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

DefaultTableModel model = (DefaultTableModel)jTable2.getModel();

list.clear();

model.setRowCount(0);

}

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

DefaultTableModel model = (DefaultTableModel)jTable2.getModel();

jFileChooser1.setDialogTitle("Сохранение файла");

if(jFileChooser1.showSaveDialog(null) == jFileChooser1.APPROVE\_OPTION){

File file = new File(jFileChooser1.getSelectedFile() + ".txt");

try{

if(!file.exists()){

file.createNewFile();

}

PrintWriter out = new PrintWriter(file);

try {

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

for(int j = 0; j < 3; j++){

out.println(model.getValueAt(i, j));

}

}

}finally {

out.close();

}

}catch (IOException ex){

throw new RuntimeException(ex);

}

}

}

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

try{

int ret = jFileChooser1.showSaveDialog(null);

File file = null;

if(ret == jFileChooser1.APPROVE\_OPTION){

file = jFileChooser1.getSelectedFile();

} else{

file = new File(jFileChooser1.getSelectedFile() + ".ser");

}

FileOutputStream fileStream = new FileOutputStream(file);

ObjectOutputStream os = new ObjectOutputStream(fileStream);

for(int row = 0; row<list.size(); row++){

os.writeObject(list);

}

}catch (Exception ex){

ex.printStackTrace();

}

}

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

DefaultTableModel model = (DefaultTableModel)jTable2.getModel();

jFileChooser1.setDialogTitle("Выберите файл");

int result = jFileChooser1.showOpenDialog(frame1.this);

if(result == jFileChooser1.APPROVE\_OPTION){

if(model.getRowCount() > 0){

JOptionPane.showMessageDialog(null, "Таблица перезаписана", "message",0);

list.clear();

model.setRowCount(0);

}

File fileReader = jFileChooser1.getSelectedFile();

ObjectInputStream obj\_in = null;

try{

obj\_in = new ObjectInputStream(new BufferedInputStream(new FileInputStream( fileReader.getAbsolutePath())));

list = (LinkedList<RecIntegral>) obj\_in.readObject();

}catch (IOException ex){

ex.printStackTrace();

}catch (ClassNotFoundException exception){

}

// clear.doClick();

// fill.doClick();

}

for(int row = 0; row<list.size(); row++){

model.addRow(new Object[]{list.get(row).start,list.get(row).end,list.get(row).step});

};

}

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

DefaultTableModel model = (DefaultTableModel)jTable2.getModel();

jFileChooser1.setDialogTitle("Выберите файл");

if(jFileChooser1.showOpenDialog(null) == jFileChooser1.APPROVE\_OPTION){

if(model.getRowCount() > 0){

JOptionPane.showMessageDialog(null, "Таблица перезаписана", "message",0);

list.clear();

model.setRowCount(0);

}

String file = jFileChooser1.getSelectedFile().toString();

long lines = 0;

try {

lines = Files.lines(jFileChooser1.getSelectedFile().toPath()).count();

System.out.println(lines);

} catch (IOException e) {

e.printStackTrace();

}

try{

BufferedReader in = new BufferedReader(new FileReader(file));

String[] s = new String[3];

try {

for(int i = 0; i < lines/3; i++){

for(int j = 0; j < 3; j++){

s[j] = in.readLine();

}

model.insertRow(model.getRowCount(), s);

}

}finally {

in.close();

}

}catch (IOException ex){

throw new RuntimeException(ex);

}

}

}

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

try {

int port = 8080;

TestServer serverUDP = new TestServer(port);

//TestClient clientUDP = new TestClient("localhost", port);

//TestClient clientUDP2 = new TestClient("localhost", port);

//new Thread(clientUDP).start();

//new Thread(clientUDP2).start();

new Thread(serverUDP).start();

try {

Thread.sleep(1000);

} catch (InterruptedException e) {

e.printStackTrace();

}

} catch (IOException ex) {

Logger.getLogger(frame1.class.getName()).log(Level.SEVERE, null, ex);

}

}

private void jTextField2ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

}

/\*\*

\* @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(frame1.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(frame1.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(frame1.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(frame1.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 frame1().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JButton addButton;

private javax.swing.JButton addToTableFromList;

private javax.swing.JButton deleteALLButton;

private javax.swing.JButton deleteButton;

private javax.swing.JButton jButton3;

private javax.swing.JFileChooser jFileChooser1;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JScrollPane jScrollPane2;

public static javax.swing.JTable jTable2;

private javax.swing.JTextField jTextField1;

private javax.swing.JTextField jTextField2;

private javax.swing.JTextField jTextField3;

private javax.swing.JButton openBinFile;

private javax.swing.JButton openFile;

private javax.swing.JButton saveBinFile;

private javax.swing.JButton saveFile;

private javax.swing.JButton startServer;

// End of variables declaration

}