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 №1

Выполнили студенты группы 20ВВП2:

Духнов О.О.

Тихонов А.А.

Приняли:

Юрова О. В.

Карамышева Н. С.

Пенза 2023

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

**Лабораторное задание:**

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.

**Листинг:**

**Файл Java6.java**

import java.io.BufferedInputStream;

import java.io.BufferedOutputStream;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.FileReader;

import java.io.IOException;

import java.io.ObjectInputStream;

import java.io.ObjectOutputStream;

import java.io.Serializable;

import java.net.DatagramPacket;

import java.net.DatagramSocket;

import java.net.InetAddress;

import java.net.SocketException;

import java.net.UnknownHostException;

import java.util.Vector;

import javax.swing.table.DefaultTableModel;

import static javax.swing.JOptionPane.showMessageDialog;

import java.util.ArrayList;

import java.util.Map;

import java.util.Collections;

import java.util.Scanner;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JFileChooser;

import javax.swing.filechooser.FileNameExtensionFilter;

import static sun.security.krb5.Confounder.bytes;

/\*

\* 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 student

\*/

public class NewJFrame extends javax.swing.JFrame

{

/\*\*

\* Creates new form NewJFrame

\*/

//CollectionList OurCollection = new CollectionList();

ArrayList<RecIntegral> OurArray = new ArrayList();

DatagramSocket socet;

InetAddress address;

public NewJFrame() throws SocketException, UnknownHostException {

socet = new DatagramSocket();

address = InetAddress.getByName("localhost");

initComponents();

}

@SuppressWarnings("unchecked")

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

private void initComponents() {

jScrollPane1 = new javax.swing.JScrollPane();

jTable1 = new javax.swing.JTable();

jScrollPane2 = new javax.swing.JScrollPane();

jTable2 = new javax.swing.JTable();

javax.swing.JPanel jPanel1 = new javax.swing.JPanel();

AddButton = new javax.swing.JButton();

DeleteButton = new javax.swing.JButton();

CalculateButton = new javax.swing.JButton();

ReadButton = new javax.swing.JButton();

ClearButton = new javax.swing.JButton();

javax.swing.JPanel jPanel2 = new javax.swing.JPanel();

UpperThreshold = new javax.swing.JTextField();

javax.swing.JLabel jLabel3 = new javax.swing.JLabel();

javax.swing.JLabel jLabel1 = new javax.swing.JLabel();

LowerThreshold = new javax.swing.JTextField();

javax.swing.JLabel jLabel2 = new javax.swing.JLabel();

Step = new javax.swing.JTextField();

javax.swing.JPanel jPanel3 = new javax.swing.JPanel();

jScrollPane3 = new javax.swing.JScrollPane();

MainTable = new javax.swing.JTable();

jMenuBar1 = new javax.swing.JMenuBar();

jMenu1 = new javax.swing.JMenu();

SaveBin = new javax.swing.JMenuItem();

SaveTxt = new javax.swing.JMenuItem();

DownloadBin = new javax.swing.JMenuItem();

DownloadTxt = new javax.swing.JMenuItem();

jMenu2 = new javax.swing.JMenu();

jTable1.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(jTable1);

jTable2.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"

}

));

jScrollPane2.setViewportView(jTable2);

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

setTitle("(1\\x) kan du inte lägga en automatisk tentamen och terminsuppsats utan laboratorium?");

setPreferredSize(new java.awt.Dimension(620, 432));

AddButton.setText("Lägg till");

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

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

AddButtonMouseClicked(evt);

}

});

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

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

AddButtonActionPerformed(evt);

}

});

DeleteButton.setText("Radera");

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

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

DeleteButtonActionPerformed(evt);

}

});

CalculateButton.setText("Beräkna");

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

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

CalculateButtonActionPerformed(evt);

}

});

ReadButton.setText("Föra in");

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

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

ReadButtonActionPerformed(evt);

}

});

ClearButton.setText("Klar");

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

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

ClearButtonActionPerformed(evt);

}

});

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);

jPanel1.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()

.addGap(118, 118, 118)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)

.addGroup(jPanel1Layout.createSequentialGroup()

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

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

.addComponent(CalculateButton, javax.swing.GroupLayout.PREFERRED\_SIZE, 103, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(jPanel1Layout.createSequentialGroup()

.addComponent(ReadButton, javax.swing.GroupLayout.PREFERRED\_SIZE, 72, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(AddButton, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.PREFERRED\_SIZE, 103, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(DeleteButton, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.PREFERRED\_SIZE, 103, javax.swing.GroupLayout.PREFERRED\_SIZE))))

.addGap(20, 20, 20))

);

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addContainerGap()

.addComponent(AddButton, javax.swing.GroupLayout.PREFERRED\_SIZE, 34, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(13, 13, 13)

.addComponent(DeleteButton, javax.swing.GroupLayout.PREFERRED\_SIZE, 34, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addComponent(CalculateButton, javax.swing.GroupLayout.PREFERRED\_SIZE, 34, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(41, 41, 41)

.addComponent(ReadButton)

.addGap(27, 27, 27)

.addComponent(ClearButton)))

.addContainerGap(49, Short.MAX\_VALUE))

);

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

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

UpperThresholdActionPerformed(evt);

}

});

jLabel3.setText("Steg");

jLabel1.setText("Topp");

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

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

LowerThresholdActionPerformed(evt);

}

});

jLabel2.setText("Botten");

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

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

StepActionPerformed(evt);

}

});

javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);

jPanel2.setLayout(jPanel2Layout);

jPanel2Layout.setHorizontalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel2Layout.createSequentialGroup()

.addGap(32, 32, 32)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel2)

.addComponent(jLabel3)

.addComponent(jLabel1))

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel2Layout.createSequentialGroup()

.addGap(18, 18, 18)

.addComponent(LowerThreshold, javax.swing.GroupLayout.DEFAULT\_SIZE, 134, Short.MAX\_VALUE))

.addGroup(jPanel2Layout.createSequentialGroup()

.addGap(16, 16, 16)

.addComponent(UpperThreshold))

.addGroup(jPanel2Layout.createSequentialGroup()

.addGap(18, 18, 18)

.addComponent(Step)))

.addGap(48, 48, 48))

);

jPanel2Layout.setVerticalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel2Layout.createSequentialGroup()

.addGap(12, 12, 12)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(UpperThreshold, javax.swing.GroupLayout.PREFERRED\_SIZE, 34, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel1))

.addGap(18, 18, 18)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel2)

.addComponent(LowerThreshold, javax.swing.GroupLayout.PREFERRED\_SIZE, 34, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 18, 18)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(Step, javax.swing.GroupLayout.PREFERRED\_SIZE, 34, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel3))

.addContainerGap(43, Short.MAX\_VALUE))

);

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

new Object [][] {

},

new String [] {

"Верхний порог", "Нижний порог", "Шаг", "Результат"

}

) {

Class[] types = new Class [] {

java.lang.Integer.class, java.lang.Integer.class, java.lang.Float.class, java.lang.Double.class

};

boolean[] canEdit = new boolean [] {

true, true, true, false

};

public Class getColumnClass(int columnIndex) {

return types [columnIndex];

}

public boolean isCellEditable(int rowIndex, int columnIndex) {

return canEdit [columnIndex];

}

});

MainTable.getTableHeader().setReorderingAllowed(false);

jScrollPane3.setViewportView(MainTable);

if (MainTable.getColumnModel().getColumnCount() > 0) {

MainTable.getColumnModel().getColumn(0).setResizable(false);

MainTable.getColumnModel().getColumn(1).setResizable(false);

MainTable.getColumnModel().getColumn(2).setResizable(false);

MainTable.getColumnModel().getColumn(3).setResizable(false);

}

javax.swing.GroupLayout jPanel3Layout = new javax.swing.GroupLayout(jPanel3);

jPanel3.setLayout(jPanel3Layout);

jPanel3Layout.setHorizontalGroup(

jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jScrollPane3)

);

jPanel3Layout.setVerticalGroup(

jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jScrollPane3, javax.swing.GroupLayout.DEFAULT\_SIZE, 212, Short.MAX\_VALUE)

);

jMenu1.setText("File");

SaveBin.setText("Сохранить bin");

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

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

SaveBinActionPerformed(evt);

}

});

jMenu1.add(SaveBin);

SaveTxt.setText("Сохранить txt");

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

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

SaveTxtActionPerformed(evt);

}

});

jMenu1.add(SaveTxt);

DownloadBin.setText("Загрузить bin");

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

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

DownloadBinActionPerformed(evt);

}

});

jMenu1.add(DownloadBin);

DownloadTxt.setText("Загрузить txt");

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

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

DownloadTxtActionPerformed(evt);

}

});

jMenu1.add(DownloadTxt);

jMenuBar1.add(jMenu1);

jMenu2.setText("Edit");

jMenuBar1.add(jMenu2);

setJMenuBar(jMenuBar1);

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

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

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

.addGroup(layout.createSequentialGroup()

.addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addComponent(jPanel3, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

);

layout.setVerticalGroup(

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

.addGroup(layout.createSequentialGroup()

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

.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

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

.addComponent(jPanel3, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

pack();

}// </editor-fold>

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

// TODO add your handling code here:

}

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

// TODO add your handling code here:

}

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

// TODO add your handling code here:

}

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

DefaultTableModel module = (DefaultTableModel)MainTable.getModel();

Vector data = module.getDataVector();

JThread MyThread = new JThread("MyThread", data.size());

MyThread.start();

// try {

// Thread.sleep(1000);

// } catch (InterruptedException ex) {

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

// }

for(int i = 0; i < data.size(); i++)

{

byte[] buf;

Vector CurrentData = (Vector)data.get(i);

String message = String.valueOf((int)CurrentData.get(0)) + " " + String.valueOf((int)CurrentData.get(1)) + " " + String.valueOf((float)CurrentData.get(2))+ " " + String.valueOf(i);

buf = message.getBytes();

DatagramPacket packet = new DatagramPacket(buf, buf.length, address, 17);

try {

socet.send(packet);

} catch (IOException ex) {

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

}

}

}

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

DefaultTableModel module = (DefaultTableModel)MainTable.getModel();

int SelectedRow = MainTable.getSelectedRow();

if(MainTable.getRowCount() != 0){

if(SelectedRow == -1)

module.removeRow(MainTable.getRowCount()-1);

else

module.removeRow(MainTable.getSelectedRow());

}

}

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

DefaultTableModel module = (DefaultTableModel)MainTable.getModel();

String a = UpperThreshold.getText();

String b = LowerThreshold.getText();

String c = Step.getText();

try

{

float step = Float.parseFloat(c);

int toplim = Integer.parseInt(a),

downLim = Integer.parseInt(b);

if ((toplim < 0.000001 || toplim >= 100000) || (downLim < 0.000001 || downLim >= 100000))

{

throw new MyException("Введён неверное значение предела");

}

else if (step <= 0)

{

throw new MyException("Неверно задан шаг");

}

else if (step >= (Math.abs(downLim - toplim)))

{

throw new MyException("Шаг не должен быть больше разницы порогов интегрирования");

}

else if (downLim <= toplim)

{

throw new MyException("Такого быть не может");

}

RecIntegral Node = new RecIntegral();

Node.addNode(Integer.parseInt(a), Integer.parseInt(b), Float.parseFloat(c));

module.addRow(new Object[]{Integer.parseInt(a), Integer.parseInt(b), Float.parseFloat(c), null});

OurArray.add(Node);

}

catch(MyException code)

{

showMessageDialog(null, code.msg);

return;

}

catch(Exception code)

{

showMessageDialog(null, "Är du en komplett nörd?");

return;

}

}

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

}

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

DefaultTableModel module = (DefaultTableModel)MainTable.getModel();

while(MainTable.getRowCount() != 0)

module.removeRow(MainTable.getRowCount()-1);

}

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

DefaultTableModel module = (DefaultTableModel)MainTable.getModel();

for(int i = 0; i < OurArray.size(); i++){

RecIntegral Node = OurArray.get(i);

module.addRow(new Object[]{Node.Top, Node.Lower, Node.Step, null});

}

}

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

OpenDialogBox odb = new OpenDialogBox();

String FileName = odb.DialogSave("bin") + ".bin";

ArrayList<String> arr = new ArrayList<String>();

try

{

FileOutputStream fos = new FileOutputStream(FileName);

BufferedOutputStream bis = new BufferedOutputStream(fos);

ObjectOutputStream oos = new ObjectOutputStream(bis);

for (int i = 0; i < OurArray.size(); i++)

{

Object Step = null, Lower = null, Top = null;

RecIntegral Node = OurArray.get(i);

Top = Node.Top;

Step = Node.Step;

Lower = Node.Lower;

arr.add(Top.toString() + ' ' + Lower.toString() + ' ' + Step.toString());

}

oos.writeObject(arr);

oos.close();

}

catch (FileNotFoundException ex)

{

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

}

catch (IOException ex)

{

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

}

}

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

OpenDialogBox odb = new OpenDialogBox();

String FileName = odb.DialogOpen("bin");

ArrayList <String> arr = new ArrayList<String>();

DefaultTableModel module = (DefaultTableModel)MainTable.getModel();

try

{

FileInputStream fis = new FileInputStream(FileName);

BufferedInputStream bis = new BufferedInputStream(fis);

ObjectInputStream ois = new ObjectInputStream(bis);

arr = (ArrayList<String>)ois.readObject();

for (int i = 0; i < arr.size(); i++)

{

String str = arr.get(i),

strTop = "",

strLower = "",

strStep = "";

int size = str.length();

int j = 0;

while (str.charAt(j) != ' ')

{

strTop += str.charAt(j);

j++;

}

j++;

while (str.charAt(j) != ' ')

{

strLower += str.charAt(j);

j++;

}

j++;

while (j != size)

{

strStep += str.charAt(j);

j++;

}

RecIntegral Node = new RecIntegral();

module.addRow(new Object[]{Integer.parseInt(strTop), Integer.parseInt(strLower), Float.parseFloat(strStep), null});

Node.addNode(Integer.parseInt(strTop), Integer.parseInt(strLower), Float.parseFloat(strStep));

OurArray.add(Node);

}

}

catch (FileNotFoundException ex)

{

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

}

catch (IOException ex)

{

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

}

catch (ClassNotFoundException ex)

{

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

}

}

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

OpenDialogBox odb = new OpenDialogBox();

String FileName = odb.DialogSave("txt") + ".txt";

ArrayList<String> arr = new ArrayList<String>();

try

{

FileOutputStream fos = new FileOutputStream(FileName);

BufferedOutputStream bis = new BufferedOutputStream(fos);

ObjectOutputStream oos = new ObjectOutputStream(bis);

for (int i = 0; i < OurArray.size(); i++)

{

Object Step = null, Lower = null, Top = null;

RecIntegral Node = OurArray.get(i);

Top = Node.Top;

Step = Node.Step;

Lower = Node.Lower;

arr.add(Top.toString() + ' ' + Lower.toString() + ' ' + Step.toString());

}

oos.writeObject(arr);

oos.close();

}

catch (FileNotFoundException ex)

{

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

}

catch (IOException ex)

{

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

}

}

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

OpenDialogBox odb = new OpenDialogBox();

String FileName = odb.DialogOpen("txt");

ArrayList <String> arr = new ArrayList<String>();

DefaultTableModel module = (DefaultTableModel)MainTable.getModel();

try

{

FileInputStream fis = new FileInputStream(FileName);

BufferedInputStream bis = new BufferedInputStream(fis);

ObjectInputStream ois = new ObjectInputStream(bis);

arr = (ArrayList<String>)ois.readObject();

for (int i = 0; i < arr.size(); i++)

{

String str = arr.get(i),

strTop = "",

strLower = "",

strStep = "";

int size = str.length();

int j = 0;

while (str.charAt(j) != ' ')

{

strTop += str.charAt(j);

j++;

}

j++;

while (str.charAt(j) != ' ')

{

strLower += str.charAt(j);

j++;

}

j++;

while (j != size)

{

strStep += str.charAt(j);

j++;

}

RecIntegral Node = new RecIntegral();

module.addRow(new Object[]{Integer.parseInt(strTop), Integer.parseInt(strLower), Float.parseFloat(strStep), null});

Node.addNode(Integer.parseInt(strTop), Integer.parseInt(strLower), Float.parseFloat(strStep));

OurArray.add(Node);

}

}

catch (FileNotFoundException ex)

{

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

}

catch (IOException ex)

{

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

}

catch (ClassNotFoundException ex)

{

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

}

}

/\*\*

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

} catch (InstantiationException ex) {

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

} catch (IllegalAccessException ex) {

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

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(NewJFrame.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() {

try {

new NewJFrame().setVisible(true);

} catch (SocketException ex) {

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

} catch (UnknownHostException ex) {

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

}

}

});

}

class FunctionIntegral

{

public double f(double x)

{

double F=1/x;

return F;

}

}

class FileReadFunctions{

public void textfile(String filename){

try{

FileReader FileToread = new FileReader(filename);

Scanner stringscaner = new Scanner(FileToread);

while(stringscaner.hasNextLine()){

UpperThreshold.setText(stringscaner.nextLine());

LowerThreshold.setText(stringscaner.nextLine());

Step.setText(stringscaner.nextLine());

AddButtonActionPerformed(null);

}

}

catch(Exception code)

{

}

// UpperThreshold.setText();

return;

}

}

class OpenDialogBox

{

String FileName;

public String DialogSave(String NameFilter)

{

JFileChooser chooser = new JFileChooser();

FileNameExtensionFilter filter = new FileNameExtensionFilter(

"txt & bin", NameFilter);

chooser.setFileFilter(filter);

int returnVal = chooser.showSaveDialog(null);

if(returnVal == JFileChooser.APPROVE\_OPTION)

{

FileName = chooser.getSelectedFile().getAbsolutePath();

}

return FileName;

}

public String DialogOpen(String NameFilter)

{

JFileChooser chooser = new JFileChooser();

FileNameExtensionFilter filter = new FileNameExtensionFilter(

"txt & bin", NameFilter);

chooser.setFileFilter(filter);

int returnVal = chooser.showOpenDialog(null);

if(returnVal == JFileChooser.APPROVE\_OPTION)

{

FileName = chooser.getSelectedFile().getAbsolutePath();

}

return FileName;

}

}

class RecIntegral implements Serializable

{

public int Lower, Top;

public float Step;

public void addNode(int top, int lower, float c)

{

Step = c;

Lower = lower;

Top = top;

}

}

class CollectionList

{

public ArrayList collectionarray;

public void addNewNode(Object element)

{

collectionarray.add(element);

}

}

static class MyException extends Exception

{

String msg;

MyException(String code)

{

msg = code;

}

}

class JThread extends Thread

{

int size;

JThread(String name, int \_size)

{

super(name);

size = \_size;

}

public void run()

{

DefaultTableModel module = (DefaultTableModel)MainTable.getModel();

DatagramSocket socket = null;

try {

socket = new DatagramSocket(26);

} catch (SocketException ex) {

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

}

for(int i = 0; i < size; i++)

{

byte[] buffer = new byte[256];

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

try {

socket.receive(request);

String Message = new String(request.getData(), 0, request.getLength());

String Resoult = "",

Num = "";

int j = 0;

while (Message.charAt(j) != ' ')

{

Resoult += Message.charAt(j);

j++;

}

j++;

while (j != Message.length())

{

Num += Message.charAt(j);

j++;

}

module.setValueAt(Float.parseFloat(Resoult), Integer.parseInt(Num), 3);

} catch (IOException ex) {

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

}

}

socket.close();

}

}

// Variables declaration - do not modify

private javax.swing.JButton AddButton;

private javax.swing.JButton CalculateButton;

private javax.swing.JButton ClearButton;

private javax.swing.JButton DeleteButton;

private javax.swing.JMenuItem DownloadBin;

private javax.swing.JMenuItem DownloadTxt;

private javax.swing.JTextField LowerThreshold;

private javax.swing.JTable MainTable;

private javax.swing.JButton ReadButton;

private javax.swing.JMenuItem SaveBin;

private javax.swing.JMenuItem SaveTxt;

private javax.swing.JTextField Step;

private javax.swing.JTextField UpperThreshold;

private javax.swing.JMenu jMenu1;

private javax.swing.JMenu jMenu2;

private javax.swing.JMenuBar jMenuBar1;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JScrollPane jScrollPane2;

private javax.swing.JScrollPane jScrollPane3;

private javax.swing.JTable jTable1;

private javax.swing.JTable jTable2;

// End of variables declaration

}

}

**Файл Client.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.

\*/

package client;

import java.io.IOException;

import java.net.DatagramPacket;

import java.net.DatagramSocket;

import java.net.InetAddress;

import java.net.SocketException;

import java.util.Vector;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.table.DefaultTableModel;

/\*\*

\*

\* @author eltix

\*/

class FunctionIntegral

{

public double f(double x)

{

double F=1/x;

return F;

}

}

class JThread extends Thread

{

int a;

int b;

float c;

int num;

DatagramSocket socket;

InetAddress address;

JThread(String name, int \_a, int \_b, float \_c, int \_n , DatagramSocket \_socket, InetAddress \_address)

{

super(name);

a = \_a;

b = \_b;

c = \_c;

num = \_n;

socket = \_socket;

address = \_address;

}

public void run()

{

FunctionIntegral funk = new FunctionIntegral();

int j;

double result=0;

int n = (int)((b-a)/c);

for(j = 1; j <= n; j++)

{

result += funk.f(a + j \*c)\* c;

}

if(n\*c < b - a)

{

float newstep = (b - a) - n\*c;

result += funk.f(b + newstep);

}

String message = String.valueOf(result) + " " + String.valueOf(num);

byte[] buf = message.getBytes();

DatagramPacket packet = new DatagramPacket(buf, buf.length, address, 26);

try {

socket.send(packet);

System.out.print("dadadadad ");

} catch (IOException ex) {

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

}

}

}

public class Client {

/\*\*

\* @param args the command line arguments

\*/

public static void main(String[] args) throws SocketException, IOException {

DatagramSocket socket = new DatagramSocket(17);

DatagramSocket socketSend = new DatagramSocket();

InetAddress address = InetAddress.getByName("localhost");

while(true){

byte[] buffer = new byte[256];

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

socket.receive(request);

if(request.getLength() != 0){

String Message = new String(request.getData(), 0, request.getLength());

String strTop = "",

strLower = "",

strStep = "",

strNum = "";

int size = Message.length();

int j = 0;

while (Message.charAt(j) != ' ')

{

strTop += Message.charAt(j);

j++;

}

j++;

while (Message.charAt(j) != ' ')

{

strLower += Message.charAt(j);

j++;

}

j++;

while (Message.charAt(j) != ' ')

{

strStep += Message.charAt(j);

j++;

}

j++;

while (j != size)

{

strNum += Message.charAt(j);

j++;

}

JThread MyThread = new JThread("MyThread", Integer.parseInt(strTop),Integer.parseInt(strLower),Float.parseFloat(strStep), Integer.parseInt(strNum), socketSend, address);

MyThread.start();

}

}

}

}

**Результат выполнения программы:**

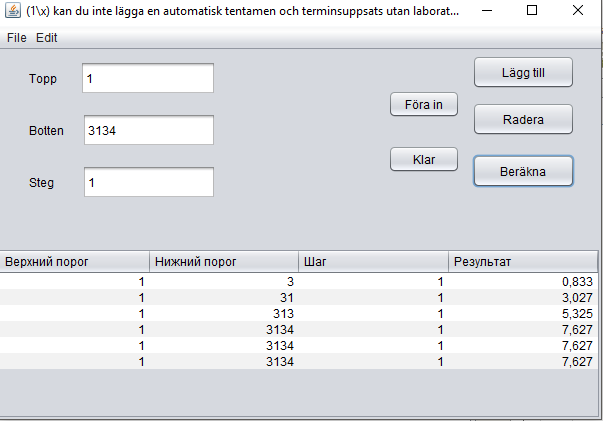


Рисунок 1 — Результат выполнения программы

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