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 №3

Выполнили студенты группы 20ВВП2:

Пантюшов Е.И.

Шмелёв Д.В.

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 LW4.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.io.BufferedInputStream;

import java.io.BufferedOutputStream;

import java.io.File;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.IOException;

import java.io.ObjectInputStream;

import java.io.ObjectOutputStream;

import java.io.Serializable;

import static java.lang.Math.abs;

import static java.lang.Math.cos;

import java.util.ArrayList;

import javax.swing.table.DefaultTableModel;

import java.util.Vector;

import java.util.logging.Level;

import java.util.logging.Logger;

//package com.csharpcoderr.messageDialog;

import javax.swing.\*;

import static javax.swing.JOptionPane.showMessageDialog;

import javax.swing.filechooser.FileNameExtensionFilter;

/\*\*

\*

\* @author PENZA

\*/

class FunctionIntegral {

// {Функция, площадь которой нужно вычислить}

public double f(double x) {

//double b = Math.toRadians(x)

double F = cos(x);

return F;

}

}

class RecIntegral implements Serializable {

public int Top, Button;

public float Step;

public void addUnit(int Top, int Button, float Step) {

this.Top = Top;

this.Button = Button;

this.Step = Step;

}

}

class FileTools {

String FileName;

public String Open(String fName) {

JFileChooser fileChooser = new JFileChooser();

fileChooser.setCurrentDirectory(new File("./src"));

FileNameExtensionFilter txt = new FileNameExtensionFilter("Text File(.txt)", "txt");

FileNameExtensionFilter bin = new FileNameExtensionFilter("Bin File(.bin)", "bin");

fileChooser.addChoosableFileFilter(txt);

fileChooser.addChoosableFileFilter(bin);

fileChooser.setFileFilter(txt);

int response = fileChooser.showDialog(null, null);

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

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

}

return FileName;

}

}

public class LW4 extends javax.swing.JFrame {

ArrayList<RecIntegral> Data = new ArrayList();

/\*\*

\* Creates new form NewJFrame

\*/

public LW4() {

initComponents();

}

static class MyException extends Exception {

String message;

MyException(String par) {

message = par;

}

}

/\*\*

\* 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() {

jScrollPane1 = new javax.swing.JScrollPane();

jTable1 = new javax.swing.JTable();

jTextField1 = new javax.swing.JTextField();

jTextField2 = new javax.swing.JTextField();

jTextField3 = new javax.swing.JTextField();

jLabel1 = new javax.swing.JLabel();

jLabel2 = new javax.swing.JLabel();

jLabel3 = new javax.swing.JLabel();

jButton1 = new javax.swing.JButton();

jButton2 = new javax.swing.JButton();

jButton3 = new javax.swing.JButton();

jButton4 = new javax.swing.JButton();

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);

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

new Object [][] {

},

new String [] {

"Top", "Bottom", "Step", "Result"

}

) {

boolean[] canEdit = new boolean [] {

true, true, true, false

};

public boolean isCellEditable(int rowIndex, int columnIndex) {

return canEdit [columnIndex];

}

});

jScrollPane1.setViewportView(jTable1);

if (jTable1.getColumnModel().getColumnCount() > 0) {

jTable1.getColumnModel().getColumn(0).setResizable(false);

jTable1.getColumnModel().getColumn(1).setResizable(false);

jTable1.getColumnModel().getColumn(2).setResizable(false);

jTable1.getColumnModel().getColumn(3).setResizable(false);

}

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

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

jTextField1ActionPerformed(evt);

}

});

jLabel1.setText("Up");

jLabel2.setText("Down");

jLabel3.setText("Step");

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("Fill");

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

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

jButton4ActionPerformed(evt);

}

});

jButton5.setText("Clear");

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

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

jButton5ActionPerformed(evt);

}

});

jButton6.setText("Save(.txt)");

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

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

jButton6ActionPerformed(evt);

}

});

jButton7.setText("Open(.txt)");

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

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

jButton7ActionPerformed(evt);

}

});

jButton8.setText("Save(.bin)");

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

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

jButton8ActionPerformed(evt);

}

});

jButton9.setText("Open(.bin)");

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(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

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

.addComponent(jButton5, javax.swing.GroupLayout.PREFERRED\_SIZE, 65, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

.addComponent(jButton4, javax.swing.GroupLayout.PREFERRED\_SIZE, 71, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(123, 123, 123))

.addGroup(layout.createSequentialGroup()

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

.addGroup(layout.createSequentialGroup()

.addContainerGap()

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

.addComponent(jLabel1)

.addComponent(jLabel2)

.addComponent(jLabel3))

.addGap(18, 18, 18)

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

.addComponent(jTextField3, javax.swing.GroupLayout.PREFERRED\_SIZE, 50, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED\_SIZE, 50, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED\_SIZE, 50, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(12, 12, 12)

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

.addComponent(jButton2, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.PREFERRED\_SIZE, 77, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGroup(layout.createSequentialGroup()

.addGap(6, 6, 6)

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

.addGroup(layout.createSequentialGroup()

.addComponent(jButton6)

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

.addComponent(jButton7, javax.swing.GroupLayout.PREFERRED\_SIZE, 100, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(layout.createSequentialGroup()

.addComponent(jButton8)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 8, Short.MAX\_VALUE)

.addComponent(jButton9, javax.swing.GroupLayout.PREFERRED\_SIZE, 100, javax.swing.GroupLayout.PREFERRED\_SIZE)))

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

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

.addComponent(jButton1, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.PREFERRED\_SIZE, 77, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jButton3, javax.swing.GroupLayout.Alignment.TRAILING)))))

.addGroup(javax.swing.GroupLayout.Alignment.LEADING, layout.createSequentialGroup()

.addGap(25, 25, 25)

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 345, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addContainerGap(8, Short.MAX\_VALUE))

);

layout.setVerticalGroup(

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

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

.addGap(6, 6, 6)

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

.addComponent(jButton4)

.addComponent(jButton5))

.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(jLabel1)

.addComponent(jButton2))

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

.addGroup(layout.createSequentialGroup()

.addGap(26, 26, 26)

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

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

.addComponent(jLabel2)

.addComponent(jButton1))

.addGap(18, 18, 18)

.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)

.addComponent(jButton3)))

.addGroup(layout.createSequentialGroup()

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

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

.addComponent(jButton7)

.addComponent(jButton6))

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

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

.addComponent(jButton8)

.addComponent(jButton9))))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 17, Short.MAX\_VALUE)

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 160, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap())

);

pack();

}// </editor-fold>

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

// TODO add your handling code here:

}

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

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

int SelectedRow = jTable1.getSelectedRow();

if (jTable1.getRowCount() != 0) {

if (SelectedRow == -1) {

model.removeRow(jTable1.getRowCount() - 1);

} else {

model.removeRow(jTable1.getSelectedRow());

}

} // TODO add your handling code here:

}

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

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

String TextField1 = jTextField1.getText();

String TextField2 = jTextField2.getText();

String TextField3 = jTextField3.getText();

try {

float step = Float.parseFloat(TextField3);

int up = Integer.parseInt(TextField1),

down = Integer.parseInt(TextField2);

if (((down < 0.000001 || down >= 100000) || (up < 0.000001 || up >= 100000))) {

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

} else if (step <= 0) {

throw new MyException("Шаг задан неверно!");

} else if (abs(up - down) <= step) {

throw new MyException("Слишком большой шаг!");

}

RecIntegral Unit = new RecIntegral();

model.addRow(new Object[]{Integer.parseInt(TextField1), Integer.parseInt(TextField2), Float.parseFloat(TextField3), null});

// TODO add your handling code here:

Unit.addUnit(Integer.parseInt(TextField1), Integer.parseInt(TextField2), Float.parseFloat(TextField3));

Data.add(Unit);

} catch (MyException code) {

showMessageDialog(null, code.message);

return;

} catch (Exception code) {

showMessageDialog(null, "Неверно ввели данные!");

return;

}

}

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

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

Vector data = model.getDataVector();

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

MyThread MyThread1 = new MyThread(i);

MyThread1.start();

}

}

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

// TODO add your handling code here:

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

while (jTable1.getRowCount() != 0)

module.removeRow(jTable1.getRowCount() - 1);

}

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

// TODO add your handling code here:

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

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

RecIntegral Unit = Data.get(i);

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

}

}

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

// TODO add your handling code here:

FileTools ft = new FileTools();

String FileName = ft.Open("save") + ".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 < Data.size(); i++) {

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

RecIntegral Node = Data.get(i);

Top = Node.Top;

Step = Node.Step;

Lower = Node.Button;

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

}

oos.writeObject(arr);

oos.close();

} catch (FileNotFoundException ex) {

Logger.getLogger(LW4.class.getName()).log(Level.SEVERE, null, ex);

} catch (IOException ex) {

Logger.getLogger(LW4.class.getName()).log(Level.SEVERE, null, ex);

}

}

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

// TODO add your handling code here:

FileTools ft = new FileTools();

String FileName = ft.Open("save");

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

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

while (jTable1.getRowCount() != 0) {

module.removeRow(jTable1.getRowCount() - 1);

}

while (Data.size() != 0) {

Data.remove(Data.size() - 1);

}

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

Data.add(Node);

}

} catch (FileNotFoundException ex) {

Logger.getLogger(LW4.class.getName()).log(Level.SEVERE, null, ex);

} catch (IOException ex) {

Logger.getLogger(LW4.class.getName()).log(Level.SEVERE, null, ex);

} catch (ClassNotFoundException ex) {

Logger.getLogger(LW4.class.getName()).log(Level.SEVERE, null, ex);

}

}

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

// TODO add your handling code here:

FileTools ft = new FileTools();

String FileName = ft.Open("save");

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 < Data.size(); i++) {

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

RecIntegral Node = Data.get(i);

Top = Node.Top;

Step = Node.Step;

Lower = Node.Button;

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

}

oos.writeObject(arr);

oos.close();

} catch (FileNotFoundException ex) {

Logger.getLogger(LW4.class.getName()).log(Level.SEVERE, null, ex);

} catch (IOException ex) {

Logger.getLogger(LW4.class.getName()).log(Level.SEVERE, null, ex);

}

}

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

// TODO add your handling code here:

FileTools ft = new FileTools();

String FileName = ft.Open("save");

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

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

while (jTable1.getRowCount() != 0) {

module.removeRow(jTable1.getRowCount() - 1);

}

while (Data.size() != 0) {

Data.remove(Data.size() - 1);

}

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

Data.add(Node);

}

} catch (FileNotFoundException ex) {

Logger.getLogger(LW4.class.getName()).log(Level.SEVERE, null, ex);

} catch (IOException ex) {

Logger.getLogger(LW4.class.getName()).log(Level.SEVERE, null, ex);

} catch (ClassNotFoundException ex) {

Logger.getLogger(LW4.class.getName()).log(Level.SEVERE, null, ex);

}

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

//<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(LW4.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(LW4.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(LW4.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(LW4.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

//</editor-fold>

/\* Create and display the form \*/

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

public void run() {

new LW4().setVisible(true);

}

});

}

class MyThread extends Thread {

private int INDEX;

MyThread(int index) {

INDEX = index;

}

public void run() {

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

FunctionIntegral funk = new FunctionIntegral();

Vector data = model.getDataVector();

for (int i = INDEX; i < data.size(); i = i + 3) {

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

double n, x1, x2;

int j, a, b;

double result = 0;

if ((int) CurrentData.get(1) < (int) CurrentData.get(0)) {

a = (int) CurrentData.get(1);

b = (int) CurrentData.get(0);

} else {

a = (int) CurrentData.get(0);

b = (int) CurrentData.get(1);

}

n = (int) ((b - a) / (float) CurrentData.get(2));

for (j = 0; j < n - 1; j++) {

x1 = a + j \* (float) CurrentData.get(2);

x2 = a + (float) CurrentData.get(2) \* (j + 1);

result += 0.5 \* (x2 - x1) \* (funk.f(x1) + funk.f(x2));

}

if ((n - 1) \* (float) CurrentData.get(2) < b) {

float newstep = (float) (b - (n - 1) \* (float) CurrentData.get(2));

x1 = a + (n - 1) \* (float) CurrentData.get(2);

result += 0.5 \* (b - x1) \* (funk.f(x1) + funk.f(b));

}

if ((int) CurrentData.get(1) < (int) CurrentData.get(0)) {

result = result \* (-1);

}

model.setValueAt(result, i, 3);

}

}

}

// 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.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

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;

// End of variables declaration

}

**Файл Server.java**

/\*

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

\*/

package com.mycompany.server;

import java.io.IOException;

import static java.lang.Math.cos;

import java.net.DatagramPacket;

import java.net.DatagramSocket;

import java.net.InetAddress;

import java.net.SocketException;

import java.net.UnknownHostException;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.table.DefaultTableModel;

/\*\*

\*

\* @author Dan

\*/

class FunctionIntegral {

public double f(double x) {

double F = cos(x);

return F;

}

}

class MyThread extends Thread {

DatagramSocket dSocket;

InetAddress iAddress;

private int strTop;

private int strBottom;

private float strStep;

private int num;

MyThread(String name, int strTOP, int strBOTTOM, float strSTEP, int N, DatagramSocket dSOCKET, InetAddress iADDRESS) {

super(name);

strTop = strTOP;

strBottom = strBOTTOM;

strStep = strSTEP;

num = N;

dSocket = dSOCKET;

iAddress = iADDRESS;

}

public void run() {

FunctionIntegral funk = new FunctionIntegral();

double n, x1, x2;

int j, a, b;

double result = 0;

if ((int) strBottom < (int) strTop) {

a = (int) strBottom;

b = (int) strTop;

} else {

a = (int) strTop;

b = (int) strBottom;

}

n = (int) ((b - a) / (float) strStep);

for (j = 0; j < n - 1; j++) {

x1 = a + j \* (float) strStep;

x2 = a + (float) strStep \* (j + 1);

result += 0.5 \* (x2 - x1) \* (funk.f(x1) + funk.f(x2));

}

if ((n - 1) \* (float) strStep < b) {

float newstep = (float) (b - (n - 1) \* (float) strStep);

x1 = a + (n - 1) \* (float) strStep;

result += 0.5 \* (b - x1) \* (funk.f(x1) + funk.f(b));

}

if ((int) strBottom < (int) strTop) {

result = result \* (-1);

}

// пиздец

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

byte[] buff = message.getBytes();

DatagramPacket dpacket = new DatagramPacket(buff, buff.length, iAddress, 4444);

try {

dSocket.send(dpacket);

} catch (IOException ex) {

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

}

}

}

public class Server {

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

DatagramSocket socketGet = new DatagramSocket(3333);

DatagramSocket socketPost = new DatagramSocket();

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

while (true) {

byte[] buffer = new byte[256];

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

socketGet.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++;

}

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

myThread.start();

}

}

}

}

**Результат выполнения программы:**

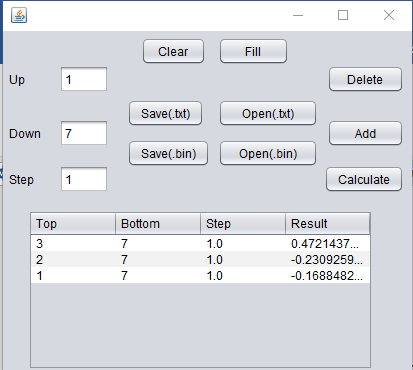


Рисунок 1 — Результат выполнения программы

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