Министерство 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»

Выполнили студенты группы 19ВВ3:

Яцков В.В

Татаршев А.А

Приняли:

Юрова О. В.

**2022**

### Название

### Сетевое взаимодействие в Java.

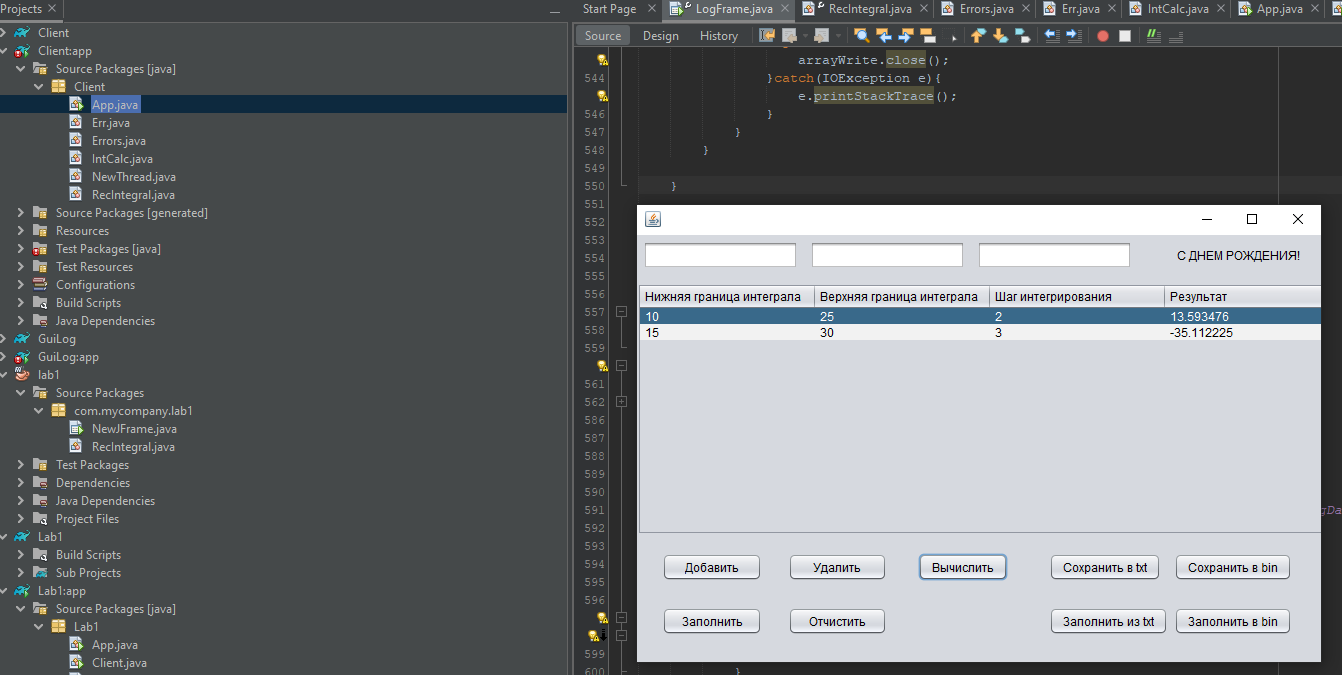
### Цель работы

### 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.

### Результаты работы:



### Рисунок 1 – Результат работы программы.

**Листинг**

**Server:**

**Файл NewJFrame.java**

/\*

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

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

\*/

package Lab1;

import static Lab1.Client.SERVER\_PORT;

import java.io.BufferedInputStream;

import java.io.BufferedOutputStream;

import java.io.BufferedReader;

import java.io.File;

import java.io.FileInputStream;

import java.io.FileOutputStream;

import java.io.FileReader;

import java.io.FileWriter;

import java.io.IOException;

import java.io.ObjectInputStream;

import java.io.ObjectOutputStream;

import java.util.LinkedList;

import javax.swing.JOptionPane;

import javax.swing.table.DefaultTableModel;

import java.math.MathContext;

import java.net.DatagramPacket;

import java.net.DatagramSocket;

import java.net.InetAddress;

import java.net.SocketException;

import java.util.ArrayList;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JFileChooser;

/\*\*

\*

@author Owl

\*/

public class LogFrame extends javax.swing.JFrame {

/\*\*

Creates new form LogFrame

\*/

public static int senderPort;

public static InetAddress senderAddress;

public static byte[] receivingDataBuffer = new byte[1024];

public final static int SERVER\_PORT = 30001;

public ArrayList array = new ArrayList();

public static DatagramSocket serverSocket;

public static double resultClient = 0.0;

public LogFrame() {

initComponents();

}

public LinkedList<RecIntegral> collection = new LinkedList<RecIntegral>();

/\*\*

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() {

jPanel1 = new javax.swing.JPanel();

jButton1 = new javax.swing.JButton();

jButton2 = new javax.swing.JButton();

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();

jButton4 = new javax.swing.JButton();

jButton5 = new javax.swing.JButton();

jLabel1 = new javax.swing.JLabel();

jButton6 = new javax.swing.JButton();

jButton7 = new javax.swing.JButton();

jButton3 = new javax.swing.JButton();

jButton8 = new javax.swing.JButton();

jButton9 = new javax.swing.JButton();

jPanel1.setBorder(new javax.swing.border.MatteBorder(null));

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

jPanel1.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup(

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

.addGap(0, 100, Short.MAX\_VALUE)

);

jPanel1Layout.setVerticalGroup(

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

.addGap(0, 100, Short.MAX\_VALUE)

);

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

setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT\_CURSOR));

setPreferredSize(new java.awt.Dimension(700, 380));

setSize(new java.awt.Dimension(700, 500));

jButton1.setText("Добавить");

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

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

jButton1ActionPerformed(evt);

}

});

jButton2.setText("Удалить");

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

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

jButton2ActionPerformed(evt);

}

});

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

new Object [][] {

},

new String [] {

"Нижняя граница интеграла", "Верхняя граница интеграла", "Шаг интегрирования", "Результат"

}

) {

boolean[] canEdit = new boolean [] {

false, false, false, false

};

public boolean isCellEditable(int rowIndex, int columnIndex) {

return canEdit [columnIndex];

}

});

jTable1.getTableHeader().setReorderingAllowed(false);

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);

}

});

jTextField2.setToolTipText("");

jButton4.setText("Заполнить");

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

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

jButton4ActionPerformed(evt);

}

});

jButton5.setText("Отчистить");

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

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

jButton5ActionPerformed(evt);

}

});

jLabel1.setText(" С ДНЕМ РОЖДЕНИЯ!");

jButton6.setText("Вычислить");

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

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

jButton6ActionPerformed(evt);

}

});

jButton7.setText("Сохранить в txt");

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

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

jButton7ActionPerformed(evt);

}

});

jButton3.setText("Заполнить из txt");

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

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

jButton3ActionPerformed(evt);

}

});

jButton8.setText("Сохранить в bin");

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

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

jButton8ActionPerformed(evt);

}

});

jButton9.setText("Заполнить в 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)

.addComponent(jScrollPane1)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED\_SIZE, 155, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(12, 12, 12)

.addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED\_SIZE, 155, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

.addComponent(jTextField3, javax.swing.GroupLayout.PREFERRED\_SIZE, 155, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

.addComponent(jLabel1, javax.swing.GroupLayout.DEFAULT\_SIZE, 204, Short.MAX\_VALUE)

.addContainerGap())

.addGroup(layout.createSequentialGroup()

.addGap(25, 25, 25)

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

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

.addComponent(jButton1, javax.swing.GroupLayout.DEFAULT\_SIZE, 100, Short.MAX\_VALUE))

.addGap(26, 26, 26)

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

.addComponent(jButton2, javax.swing.GroupLayout.PREFERRED\_SIZE, 99, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jButton5, javax.swing.GroupLayout.PREFERRED\_SIZE, 99, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(31, 31, 31)

.addComponent(jButton6)

.addGap(41, 41, 41)

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

.addComponent(jButton7)

.addComponent(jButton3))

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

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

.addComponent(jButton9)

.addComponent(jButton8))

.addGap(51, 51, 51))

);

layout.setVerticalGroup(

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

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

.addContainerGap()

.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, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

.addComponent(jLabel1))

.addGap(14, 14, 14)

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 252, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

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

.addComponent(jButton1)

.addComponent(jButton2)

.addComponent(jButton6)

.addComponent(jButton7)

.addComponent(jButton8))

.addGap(26, 26, 26)

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

.addComponent(jButton4)

.addComponent(jButton5)

.addComponent(jButton3)

.addComponent(jButton9))

.addContainerGap(18, Short.MAX\_VALUE))

);

pack();

}// </editor-fold>

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

// TODO add your handling code here:

}

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

// TODO add your handling code here:

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

if(!(jTextField1.getText().isEmpty())&&!(jTextField2.getText().isEmpty())&&!(jTextField3.getText().isEmpty()))

{

try{

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

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

}catch(Errors err){

JOptionPane.showMessageDialog(null, err.showErrorMessage());

}catch(Err err){

JOptionPane.showMessageDialog(null, err.showErrorMessage());

}

jTextField1.setText("");

jTextField2.setText("");

jTextField3.setText("");

}

else{

JOptionPane.showMessageDialog(this, "Введите значения");

}

}

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

// TODO add your handling code here:

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

if(jTable1.getSelectedRowCount() == 1){

collection.remove(jTable1.getSelectedRow());

model.removeRow(jTable1.getSelectedRow());

}else{

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

JOptionPane.showMessageDialog(this, "Таблица пуста");

}else{

JOptionPane.showMessageDialog(this, "Выберите строку, которую хотите удалить");

}

}

}

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

// TODO add your handling code here:

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

collection.forEach((g) -> {RecIntegral l = (RecIntegral)g; model.addRow(new Object[]{ l.step, l.lowerBound, l.upperBound});});

}

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

// TODO add your handling code here:

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

model.setRowCount(0);

}

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

IntCalc(Float.valueOf(jTable1.getValueAt(jTable1.getSelectedRow(), 0).toString()),Float.valueOf(jTable1.getValueAt(jTable1.getSelectedRow(), 1).toString())\*(float)0.25,Float.valueOf(jTable1.getValueAt(jTable1.getSelectedRow(), 2).toString()));

// IntCalc cu = new IntCalc(Float.valueOf(jTable1.getValueAt(jTable1.getSelectedRow(), 1).toString())\*(float)0.25,Float.valueOf(jTable1.getValueAt(jTable1.getSelectedRow(), 1).toString())\*(float)0.5,Float.valueOf(jTable1.getValueAt(jTable1.getSelectedRow(), 2).toString()));

// IntCalc la = new IntCalc(Float.valueOf(jTable1.getValueAt(jTable1.getSelectedRow(), 1).toString())\*(float)0.5,Float.valueOf(jTable1.getValueAt(jTable1.getSelectedRow(), 1).toString())\*(float)0.75,Float.valueOf(jTable1.getValueAt(jTable1.getSelectedRow(), 2).toString()));

// IntCalc te = new IntCalc(Float.valueOf(jTable1.getValueAt(jTable1.getSelectedRow(), 1).toString())\*(float)0.75,Float.valueOf(jTable1.getValueAt(jTable1.getSelectedRow(), 1).toString()),Float.valueOf(jTable1.getValueAt(jTable1.getSelectedRow(), 2).toString()));

double downValue;

double upValue;

double step;

DefaultTableModel table1 = (DefaultTableModel)jTable1.getModel();

try{

int count = jTable1.getSelectedRowCount();

int[] aRows = jTable1.getSelectedRows();

int[] aRows1 = jTable1.getSelectedRows();

double[] res = new double[count];

receivingDataBuffer = (String.valueOf(count) + " ").getBytes();

DatagramPacket msgCount = new DatagramPacket(receivingDataBuffer, receivingDataBuffer.length, senderAddress, senderPort );

serverSocket.send(msgCount);

String[] data = new String[aRows.length];

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

downValue = Double.valueOf(table1.getValueAt(aRows[i], 0).toString());

upValue = Double.valueOf(table1.getValueAt(aRows[i], 1).toString());

step = Double.valueOf(table1.getValueAt(aRows[i], 2).toString());

data[i] = String.valueOf(downValue) + " " + String.valueOf(upValue) + " " + String.valueOf(step);

receivingDataBuffer = data[i].getBytes();

DatagramPacket Value = new DatagramPacket(receivingDataBuffer, receivingDataBuffer.length, senderAddress, senderPort );

serverSocket.send(Value);

}

while(true){

byte[] resClient = new byte[1024];

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

DatagramPacket inputPacketAnsw = new DatagramPacket(resClient, resClient.length);

serverSocket.receive(inputPacketAnsw);

resultClient = Double.parseDouble(new String(inputPacketAnsw.getData()));

System.out.println("Result from client - " + resultClient);

res[i] = resultClient;

}

break;

}

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

table1.setValueAt(res[i], aRows1[i], 3);

}

}catch(SocketException e){

e.printStackTrace();;

}catch(IOException a){

a.printStackTrace();

}

}

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

JFileChooser openInputFile = new JFileChooser();

int dlgWnd = openInputFile.showDialog(null, "Сохранить в тектсовый файл");

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

File fileOpen = openInputFile.getSelectedFile();

try(FileWriter writeFile = new FileWriter(fileOpen, false)){

collection.forEach((s)->{

RecIntegral rc = (RecIntegral)s;

try{

writeFile.write(String.valueOf(rc.lowerBound) + " " + String.valueOf(rc.upperBound) + " " + String.valueOf(rc.step) + "\n");

}catch(IOException e){

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

}

});

}catch(IOException e){

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

}

}

}

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

collection.clear();

double downValue;

double upValue;

double step;

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

table.setRowCount(0);

JFileChooser openInputFile = new JFileChooser();

int dlgWnd = openInputFile.showDialog(null, "Открыть тектсовый файл");

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

File fileOpen = openInputFile.getSelectedFile();

try(FileReader readFile = new FileReader(fileOpen)){

String buf; BufferedReader rBuf = new BufferedReader(readFile);

while ((buf = rBuf.readLine()) != null){

String[] v = buf.split(" ");

downValue = Double.parseDouble(v[0]);

upValue = Double.parseDouble(v[1]);

step = Double.parseDouble(v[2]);

try{

RecIntegral obj = new RecIntegral(String.valueOf(downValue), String.valueOf(upValue), String.valueOf(step));

collection.add(obj);

}catch(Errors err){

JOptionPane.showMessageDialog(null, err.showErrorMessage());

} catch (Err ex) {

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

}

}

}catch(IOException e){

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

}

}

collection.forEach((s)->{

RecIntegral rc = (RecIntegral)s;

table.addRow(new Object[]{rc.lowerBound, rc.upperBound, rc.step});

});

}

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

collection.clear();

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

table.setRowCount(0);

JFileChooser openInputFile = new JFileChooser();

int dlgWnd = openInputFile.showDialog(null, "Открыть бинарный файл");

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

File fileOpen = openInputFile.getSelectedFile();

ObjectInputStream arrayRead = null;

try{

arrayRead = new ObjectInputStream(new BufferedInputStream(new FileInputStream(fileOpen)));

collection = (LinkedList)arrayRead.readObject();

}catch(IOException e){

e.printStackTrace();

}catch(ClassNotFoundException classErr){

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

}finally{

try{

arrayRead.close();

}catch(IOException e){

e.printStackTrace();

}

}

}

collection.forEach((s)->{

RecIntegral rc = (RecIntegral)s;

table.addRow(new Object[]{rc.lowerBound, rc.upperBound, rc.step});

});

}

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

JFileChooser openInputFile = new JFileChooser();

int dlgWnd = openInputFile.showDialog(null, "Сохранить в бинарный файл");

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

File fileOpen = openInputFile.getSelectedFile();

ObjectOutputStream arrayWrite = null;

try{

arrayWrite = new ObjectOutputStream(new BufferedOutputStream(new FileOutputStream(fileOpen)));

arrayWrite.writeObject(collection);

}catch(IOException e){

e.printStackTrace();

}finally{

try{

arrayWrite.close();

}catch(IOException e){

e.printStackTrace();

}

}

}

}

/\*\*

@param args the command line arguments

\*/

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

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

} catch (InstantiationException ex) {

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

} catch (IllegalAccessException ex) {

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

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

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

}

//</editor-fold>

/\* Create and display the form \*/

serverSocket = new DatagramSocket(SERVER\_PORT);

JOptionPane.showMessageDialog(null, "Waiting for a client to connect...");

DatagramPacket inputPacket = new DatagramPacket(receivingDataBuffer, receivingDataBuffer.length);

serverSocket.receive(inputPacket);

String receivedData = new String(inputPacket.getData());

JOptionPane.showMessageDialog(null, "Status: " + receivedData);

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

public void run() {

new LogFrame().setVisible(true);

}

});

System.out.println("Message from client - " + receivedData);

senderAddress = inputPacket.getAddress();

senderPort = inputPacket.getPort();

}

// 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.JPanel jPanel1;

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

**}**

**Client:**

/\*\*

\*

\* @author Owl

\*/

import java.io.IOException;

import java.net.DatagramPacket;

import java.net.DatagramSocket;

import java.net.InetAddress;

import java.net.SocketException;

import java.util.concurrent.\*;

import java.util.logging.Level;

import java.util.logging.Logger;

public class Client{

public final static int SERVER\_PORT = 30001;

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

try{

DatagramSocket clientSocket = new DatagramSocket();

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

byte[] sendingDataBuffer;

String sentence = "connected";

sendingDataBuffer = sentence.getBytes();

DatagramPacket sendingPacket = new DatagramPacket(sendingDataBuffer,sendingDataBuffer.length,IPAddress, SERVER\_PORT);

clientSocket.send(sendingPacket);

while(true) {

byte[] receivingDataBuffer = new byte[1024];

DatagramPacket receivingPacket = new DatagramPacket(receivingDataBuffer, receivingDataBuffer.length);

clientSocket.receive(receivingPacket);

while(true) {

String[] a = new String(receivingPacket.getData()).split(" ");

int count = Integer.parseInt(a[0], 10);

System.out.println("count from server: " + count);

RecIntegral[] rc = new RecIntegral[count];

try {

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

clientSocket.receive(receivingPacket);

String[] sVal = (new String(receivingPacket.getData())).split(" ");

rc[i] = new RecIntegral(sVal[0], sVal[1], sVal[2]);

}

double[] result = new double[count];

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

IntCalc cal = new IntCalc(Float.valueOf(rc[i].lowerBound),1,2);

// IntCalc cu = new IntCalc(Float.valueOf(jTable1.getValueAt(jTable1.getSelectedRow(), 1).toString())\*(float)0.25,Float.valueOf(jTable1.getValueAt(jTable1.getSelectedRow(), 1).toString())\*(float)0.5,Float.valueOf(jTable1.getValueAt(jTable1.getSelectedRow(), 2).toString()));

// IntCalc la = new IntCalc(Float.valueOf(jTable1.getValueAt(jTable1.getSelectedRow(), 1).toString())\*(float)0.5,Float.valueOf(jTable1.getValueAt(jTable1.getSelectedRow(), 1).toString())\*(float)0.75,Float.valueOf(jTable1.getValueAt(jTable1.getSelectedRow(), 2).toString()));

// IntCalc te = new IntCalc(Float.valueOf(jTable1.getValueAt(jTable1.getSelectedRow(), 1).toString())\*(float)0.75,Float.valueOf(jTable1.getValueAt(jTable1.getSelectedRow(), 1).toString()),Float.valueOf(jTable1.getValueAt(jTable1.getSelectedRow(), 2).toString()));

// cal.start();

// cu.start();

// la.start();

// te.start();

Thread t = new Thread()

{

public void run()

{

try

{

cal.join();

// cu.join();

// la.join();

// te.join();

}

catch (InterruptedException ex)

{

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

}

// jTable1.setValueAt(String.valueOf(cal.subanswer + cu.subanswer + la.subanswer + te.subanswer), jTable1.getSelectedRow(), 3);

// jTextField1.setText(String.valueOf(cal.subanswer + cu.subanswer + la.subanswer + te.subanswer));

// jButton4.setEnabled(true);

}

};

t.start();

// System.out.println("Result: " + result[i] + " current thread - " + tr[i].getName());

String resStr = String.valueOf(result[i]);

sendingDataBuffer = resStr.getBytes();

DatagramPacket sendingAnsw = new DatagramPacket(sendingDataBuffer, sendingDataBuffer.length, IPAddress, SERVER\_PORT);

clientSocket.send(sendingAnsw);

}

}catch(Errors e) {

e.showErrorMessage();

}catch(Err e){

e.getMessage();

}

break;

}

clientSocket.close();

}

}

catch(SocketException e) {

e.printStackTrace();

}

}

}

**Файл Client.java**

import java.io.IOException;  
import java.net.DatagramPacket;  
import java.net.DatagramSocket;  
import java.net.InetAddress;  
import java.net.SocketException;  
import java.util.concurrent.\*;  
  
public class client{  
  
 public final static int SERVER\_PORT = 30001;  
 public static void main(String[] args) throws IOException{  
 NewThread[] tr = new NewThread[5];  
  
 try{  
 Нет необходимости в привязке к определенному порту \*/  
 DatagramSocket clientSocket = new DatagramSocket();  
   
 InetAddress IPAddress = InetAddress.getByName("localhost");  
   
 byte[] sendingDataBuffer;  
  
 String sentence = "connected";  
 sendingDataBuffer = sentence.getBytes();  
   
 DatagramPacket sendingPacket = new DatagramPacket(sendingDataBuffer,sendingDataBuffer.length,IPAddress, SERVER\_PORT);  
   
 clientSocket.send(sendingPacket);  
 while(true) {  
 byte[] receivingDataBuffer = new byte[1024];  
 DatagramPacket receivingPacket = new DatagramPacket(receivingDataBuffer, receivingDataBuffer.length);  
 clientSocket.receive(receivingPacket);  
 while(true) {  
 String[] a = new String(receivingPacket.getData()).split(" ");  
 int count = Integer.parseInt(a[0], 10);  
 System.out.println("count from server: " + count);  
 RecIntegral[] rc = new RecIntegral[count];  
 try {  
 for (int i = 0; i < count; i++) {  
 clientSocket.receive(receivingPacket);  
 String[] sVal = (new String(receivingPacket.getData())).split(" ");  
 rc[i] = new RecIntegral(Double.parseDouble(sVal[0]), Double.parseDouble(sVal[1]), Double.parseDouble(sVal[2]));  
 }  
  
 double[] result = new double[count];  
 for (int i = 0; i < count; i++) {  
 tr[i] = new NewThread(rc[i]);  
 tr[i].start();  
 tr[i].join();  
 result[i] = tr[i].retRes();  
  
 System.out.println("Result: " + result[i] + " current thread - " + tr[i].getName());  
 String resStr = String.valueOf(result[i]);  
 sendingDataBuffer = resStr.getBytes();  
 DatagramPacket sendingAnsw = new DatagramPacket(sendingDataBuffer, sendingDataBuffer.length, IPAddress, SERVER\_PORT);  
 clientSocket.send(sendingAnsw);  
 }  
  
 }catch(Errors e) {  
 e.showErrorMessage();  
  
 }catch(InterruptedException e){  
 e.getMessage();  
 }  
 break;  
 }  
  
 clientSocket.close();  
 }  
 }  
 catch(SocketException e) {  
 e.printStackTrace();  
 }  
 }  
}

**Файл Errors.java\**

package Client;

/\*\*

\*

\* @author Owl

\*/

public class Errors extends Exception{

public String showErrorMessage(){return "Введите значение в диапазоне от 0,000001 до 1000000";};

}

**Файл Err.java**

package Client;

/\*\*

\*

\* @author Owl

\*/

public class Err extends Exception{

public String showErrorMessage(){return "Разность границ больше шага интегрирования";};

}

**Файл RecIntegral.java**

package Client;

import java.io.Serializable;

/\*\*

\*

\* @author Owl

\*/

class RecIntegral implements Serializable{

String step;

String lowerBound;

String upperBound;

public RecIntegral(String lowerBound, String upperBound, String step) throws Errors, Err{

if(Float.parseFloat(step) < 0.000001 || Float.parseFloat(step) > 1000000){

throw new Errors();

}

this.step = step;

if(Float.parseFloat(lowerBound) < 0.000001 || Float.parseFloat(lowerBound) > 1000000){

throw new Errors();

}

this.lowerBound = lowerBound;

if(Float.parseFloat(upperBound) < 0.000001 || Float.parseFloat(upperBound) > 1000000){

throw new Errors();

}

this.upperBound = upperBound;

Math.abs(Float.parseFloat(lowerBound));

if( Math.abs(Float.parseFloat(upperBound)) - Math.abs(Float.parseFloat(lowerBound)) < Float.parseFloat(step)){

throw new Err();

}

}

public RecIntegral(){

this.step = " ";

this.lowerBound = " ";

this.upperBound = " ";

}

public String getStep() {

return step;

}

public void setStep(String step) {

this.step = step;

}

public String getLowerBound() {

return lowerBound;

}

public void setLowerBound(String lowerBound) {

this.lowerBound = lowerBound;

}

public String getUpperBound() {

return upperBound;

}

public void setUpperBound(String upperBound) {

this.upperBound = upperBound;

}

public float IntegralFunc() {

float res = 0;

for(float i = Float.parseFloat(lowerBound); i < Float.parseFloat(upperBound); i += Float.parseFloat(step))

{

res += ((Math.tan(i)+Math.tan(i + Float.parseFloat(step)))/2)\*Float.parseFloat(step);

}

return res;

}

**Файл NewThread.java**

/\*

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

\* Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template

\*/

package Client;

/\*\*

\*

\* @author Owl

\*/

public class NewThread extends Thread{

RecIntegral trec;

float res;

NewThread(RecIntegral rc){

try{

trec = new RecIntegral(rc.lowerBound, rc.upperBound, rc.step);

}catch(Errors e){

e.showErrorMessage();

}

catch(Err e){

}

}

@Override

public void run(){

res = trec.IntegralFunc();

}

public float retRes(){

return res;

}

}

### Вывод: Научились создавать клиент-серверные приложения c использованием стандартных классов Java.