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:

Табрисов С. А.

Литвинов А. Ю.

Приняли:

Юрова О. В.

**2022**

### Название

### Сетевое взаимодействие в Java.

### Цель работы

### Научиться создавать клиент-серверные приложения c использованием стандартных классов Java.

### Лабораторное задание

Таблица 1

|  |  |
| --- | --- |
| Вариант | Функция |
| 5 |  |

### 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:

Изображение выглядит как текст

Автоматически созданное описание

### Рисунок 1 – Результат работы программы.

**Листинг**

**Server:**

**Файл NewJFrame.java**

package com.mycompany.lab1;

import javax.swing.JOptionPane;

import javax.swing.table.DefaultTableModel;

import java.math.MathContext;

import java.util.ArrayList;

import java.io.FileReader;

import java.io.FileWriter;

import javax.swing.JFileChooser;

import java.io.File;

import java.io.IOException;

import java.io.Serializable;

import java.io.ObjectInputStream;

import java.io.ObjectOutputStream;

import java.io.BufferedReader;

import java.io.BufferedInputStream;

import java.io.BufferedOutputStream;

import java.io.FileOutputStream;

import java.io.FileInputStream;

import java.io.IOException;

import java.net.DatagramPacket;

import java.net.DatagramSocket;

import java.net.InetAddress;

import java.net.SocketException;

public class NewJFrame extends javax.swing.JFrame {

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 NewJFrame() {

initComponents();

}

@SuppressWarnings("unchecked")

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

private void initComponents() {

jButton1 = new javax.swing.JButton();

jButton2 = new javax.swing.JButton();

jButton3 = new javax.swing.JButton();

jTextField1 = new javax.swing.JTextField();

jTextField2 = new javax.swing.JTextField();

jTextField3 = new javax.swing.JTextField();

jLabel4 = new javax.swing.JLabel();

jLabel5 = new javax.swing.JLabel();

jLabel6 = new javax.swing.JLabel();

jScrollPane1 = new javax.swing.JScrollPane();

jTable1 = new javax.swing.JTable();

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);

setTitle("Интеграл ");

setBackground(new java.awt.Color(51, 204, 0));

setName("Интеграл "); // NOI18N

setResizable(false);

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);

}

});

jButton3.setText("Вычислить");

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

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

jButton3ActionPerformed(evt);

}

});

jTextField1.setCursor(new java.awt.Cursor(java.awt.Cursor.TEXT\_CURSOR));

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);

}

});

jTextField3.addActionListener(new java.awt.event.ActionListener() {

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

jTextField3ActionPerformed(evt);

}

});

jLabel4.setFont(new java.awt.Font("Serif", 0, 14)); // NOI18N

jLabel4.setText("Нижняя граница интегрирования");

jLabel5.setFont(new java.awt.Font("Serif", 0, 14)); // NOI18N

jLabel5.setText("Верхняя граница интегрирования");

jLabel6.setFont(new java.awt.Font("Serif", 0, 14)); // NOI18N

jLabel6.setText("Шаг интегрирования ");

jTable1.setFont(new java.awt.Font("Serif", 0, 12)); // NOI18N

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 [] {

"Нижняя граница ", "Верхняя граница", "Шаг", "Результат"

}

) {

boolean[] canEdit = new boolean [] {

true, true, true, false

};

public boolean isCellEditable(int rowIndex, int columnIndex) {

return canEdit [columnIndex];

}

});

jTable1.setColumnSelectionAllowed(true);

jTable1.setDoubleBuffered(true);

jTable1.setDropMode(javax.swing.DropMode.INSERT\_ROWS);

jTable1.setGridColor(new java.awt.Color(228, 222, 222));

jTable1.setSelectionForeground(new java.awt.Color(204, 204, 204));

jTable1.setShowGrid(true);

jScrollPane1.setViewportView(jTable1);

jTable1.getColumnModel().getSelectionModel().setSelectionMode(javax.swing.ListSelectionModel.SINGLE\_INTERVAL\_SELECTION);

jButton4.setText("Очистить");

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

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

jButton4ActionPerformed(evt);

}

});

jButton5.setText("Заполнить");

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

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

jButton5ActionPerformed(evt);

}

});

jButton6.setText("Сохранить .txt");

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

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

jButton6ActionPerformed(evt);

}

});

jButton7.setText("Загрузить .txt");

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

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

jButton7ActionPerformed(evt);

}

});

jButton8.setText("Сохранить .bin");

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

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

jButton8ActionPerformed(evt);

}

});

jButton9.setText("Загрузить .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(layout.createSequentialGroup()

.addContainerGap()

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

.addGroup(layout.createSequentialGroup()

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

.addGroup(layout.createSequentialGroup()

.addComponent(jLabel4, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

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

.addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED\_SIZE, 70, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(layout.createSequentialGroup()

.addComponent(jLabel5, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

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

.addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED\_SIZE, 70, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(layout.createSequentialGroup()

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

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

.addComponent(jLabel6, javax.swing.GroupLayout.PREFERRED\_SIZE, 218, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGroup(layout.createSequentialGroup()

.addComponent(jButton1, javax.swing.GroupLayout.PREFERRED\_SIZE, 93, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(26, 26, 26)

.addComponent(jButton2, javax.swing.GroupLayout.PREFERRED\_SIZE, 93, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addComponent(jButton3)))))

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

.addGroup(layout.createSequentialGroup()

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

.addComponent(jTextField3, javax.swing.GroupLayout.PREFERRED\_SIZE, 70, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 323, javax.swing.GroupLayout.PREFERRED\_SIZE))

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

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

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

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

.addComponent(jButton8, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.DEFAULT\_SIZE, 135, Short.MAX\_VALUE)

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

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

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

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

);

layout.setVerticalGroup(

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

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

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

.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(jLabel4)

.addComponent(jButton6))

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

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

.addComponent(jButton7)

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

.addComponent(jLabel5))

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

.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(jLabel6)

.addComponent(jButton8))

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

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

.addGroup(layout.createSequentialGroup()

.addComponent(jButton9)

.addGap(13, 13, 13)

.addComponent(jButton5)

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

.addComponent(jButton4))

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 98, javax.swing.GroupLayout.PREFERRED\_SIZE))

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

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

.addComponent(jButton1)

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

.addComponent(jButton3))

.addGap(0, 13, Short.MAX\_VALUE))

);

pack();

}// </editor-fold>

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

// TODO add your handling code here:

double downValue;

double upValue;

double step;

try{

downValue = Double.parseDouble(jTextField1.getText());

upValue = Double.parseDouble(jTextField2.getText());

step = Double.parseDouble(jTextField3.getText());

}catch(NumberFormatException err){

JOptionPane.showMessageDialog(null, "Введите значения!");

return;

}

try{

RecIntegral recInt = new RecIntegral(downValue, upValue, step);

array.add(recInt);

}catch(Errors err){

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

}

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

table1.setRowCount(0);

array.forEach((s)->{RecIntegral rc = (RecIntegral)s; table1.addRow(new Object[]{rc.downValue, rc.upValue, rc.step});});

jTextField1.setText("");

jTextField2.setText("");

jTextField3.setText("");

}

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

// TODO add your handling code here:

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

int selectRow = jTable1.getSelectedRow();

if (selectRow == -1){

JOptionPane.showMessageDialog(null, "Выберете строку!");

return;

}

array.remove(selectRow);

table1.removeRow(selectRow);

}

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

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

// TODO add your handling code here:

}

private void jTextField3ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

}

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

// TODO add your handling code here:

}

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

// TODO add your handling code here:

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

table1.setRowCount(0);

}

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

// TODO add your handling code here:

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

table1.setRowCount(0);

array.forEach((s)->{RecIntegral rc = (RecIntegral)s;

table1.addRow(new Object[]{rc.downValue, rc.upValue, rc.step});});

}

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

// TODO add your handling code here:

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)){

array.forEach((s)->{

RecIntegral rc = (RecIntegral)s;

try{

writeFile.write(String.valueOf(rc.downValue) + " " + String.valueOf(rc.upValue) + " " + String.valueOf(rc.step) + "\n");

}catch(IOException e){

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

}

});

}catch(IOException e){

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

}

}

}

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

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(downValue, upValue, step);

array.add(obj);

}catch(Errors err){

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

}

}

}catch(IOException e){

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

}

}

array.forEach((s)->{

RecIntegral rc = (RecIntegral)s;

table.addRow(new Object[]{rc.downValue, rc.upValue, 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(array);

}catch(IOException e){

e.printStackTrace();

}finally{

try{

arrayWrite.close();

}catch(IOException e){

e.printStackTrace();

}

}

}

}

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

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)));

array = (ArrayList)arrayRead.readObject();

}catch(IOException e){

e.printStackTrace();

}catch(ClassNotFoundException classErr){

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

}finally{

try{

arrayRead.close();

}catch(IOException e){

e.printStackTrace();

}

}

}

array.forEach((s)->{

RecIntegral rc = (RecIntegral)s;

table.addRow(new Object[]{(double)rc.downValue, (double)rc.upValue, (double)rc.step});

});

}

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

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);

}

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 NewJFrame().setVisible(true);

}

});

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

senderAddress = inputPacket.getAddress();

senderPort = inputPacket.getPort();

}

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 jLabel4;

private javax.swing.JLabel jLabel5;

private javax.swing.JLabel jLabel6;

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;

}

**Client:**

**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**

public class Errors extends Exception{

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

}

**Файл StepErr.java**

public class StepErr extends Exception {

public String showErrorStep(){return "Шаг интегрирования превышает интервал";};

}

**Файл RecIntegral.java**

import java.io.Serializable;

public class RecIntegral implements Serializable{

double downValue;

double upValue;

double step;

public RecIntegral(double downValue, double upValue, double step) throws Errors {

if(downValue < 0.000001 || downValue > 1000000){

throw new Errors();

}

this.downValue = downValue;

if(upValue < 0.000001 || upValue > 1000000){

throw new Errors();

}

this.upValue = upValue;

if(step < 0.000001 || step > 1000000){

throw new Errors();

}

this.step = step;

}

private double MathFuncExp(double x){

return Math.exp(-x);

}

public double IntegralFunc() throws StepErr {

int i;

double n, res = 0;

if( Math.abs(upValue - downValue) < step ){

throw new StepErr();

}

n=(upValue-downValue)/step;

res = step \* (MathFuncExp(upValue) + MathFuncExp(downValue))/2.0;

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

res = res + step \* MathFuncExp(downValue + step \* i);

}

return res;

}

}

**Файл NewThread.java**

public class NewThread extends Thread{  
 RecIntegral trec;  
 double res;  
   
 NewThread(RecIntegral rc){  
 try{  
 trec = new RecIntegral(rc.downValue, rc.upValue, rc.step);  
 }catch(Errors e){  
 e.showErrorMessage();  
 }  
 }  
 @Override  
 public void run(){  
 try{  
 res = trec.IntegralFunc();  
 }catch(StepErr e){  
 e.showErrorStep();  
 }  
 }  
 public double retRes(){  
 return res;  
 }  
}

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