Министерство науки и высшего образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №6

по дисциплине: «Разработка КП приложений»

на тему «**Сетевое взаимодействие в Java**»

Выполнили:

студенты группы 19ВВ1

Артемов К.А.

Григорьева Д.Д.

Мереняшева Е.А.

Приняли:

Юрова О.В.

Карамышева Н.С.

Пенза 2022

**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.

# Вариант 4



**Ход работы**

Алгоритм работы программы.

Класс Fabrica запускает сервер, Server прослушивает порт на предмет подключения или отключения пользователей, так же осуществляет вычисления. Сообщения от клиента на сервер приходят через InputStreamReader и отправляются через OutputStreamWriter.

**Листинг**

**Класс Counter**

public class Counter {

private static int counter =4;

// Counter(){

// counter = 4;

// }

public static synchronized int get() {

return counter;

}

public static synchronized void inc() {

counter++;

}

public static synchronized void dec() {

counter--;

}

}

**Класс JThread**

package my.numberaddition;

import java.io.IOException;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JOptionPane;

import javax.swing.table.DefaultTableModel;

import static my.numberaddition.ContactEditorUI.reader;

import static my.numberaddition.ContactEditorUI.writer;

public class JThread extends Thread {

private double a;

private double b;

private double h;

public double res;

public int rowt;

public DefaultTableModel modelt;

public void setA(double a) throws MyException {

this.a = a;

}

public void setB(double b) throws MyException {

this.b = b;

}

public void setH(double h) throws MyException {

this.h = h;

}

public JThread(String name,Object a1, Object b1, Object h1, int row, DefaultTableModel model) throws MyException {

super(name);

res =0;

rowt = row;

modelt=model;

try {

this.setA(Double.parseDouble(a1.toString()));

this.setB(Double.parseDouble(b1.toString()));

this.setH(Double.parseDouble(h1.toString()));

if(b<a){

throw new MyException("Верхняя граница меньше чем нижняя");

}

if(h>(b-a)){

throw new MyException("Шаг перевашает");

}

} catch (NumberFormatException e) {

throw new MyException("Ошибка встречен символ или ,", e);

}catch (MyException ex) {

JOptionPane.showMessageDialog(null, ex.getMessage(),"err",JOptionPane.ERROR\_MESSAGE);

}

}

public void run(){

Counter.dec();

System.out.printf("%s started... \n", Thread.currentThread().getName());

try {

writer.write("#calc|"+a+"|"+b+"|"+h+"\n");

writer.flush();

String response = reader.readLine();

modelt.setValueAt(Double.parseDouble(response),

rowt,

3);

} catch (IOException ex) {

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

}

System.out.printf("%s fiished... \n", Thread.currentThread().getName());

Counter.inc();

}

}}

**Класс RecIntegral**

package my.numberaddition;

import java.io.Serializable;

import javax.swing.JOptionPane;

public class RecIntegral implements Serializable {

private double down;

private double up;

private double step;

private double res;

public double getDown() {

return down;

}

public void setDown(double down) throws MyException {

if(!prov(down)){

throw new MyException("Нижняя граница выходит из диапазона");

}

this.down = down;

}

public double getUp() {

return up;

}

public void setUp(double up) throws MyException {

if(!prov(up)){

throw new MyException("Верхняя граница выходит из диапазона");

}

if(this.getDown()>=up){

throw new MyException("верхняя граница меньше чем нижняя");

}

this.up = up;

}

public double getStep() {

return step;

}

public void setStep(double step) throws MyException {

if(!prov(step)){

throw new MyException("Шаг выходит из диапазона");

}

if(step>(this.getUp()-this.getDown())){

throw new MyException("Шаг перевашает");

}

this.step = step;

}

public double getRes() {

return res;

}

public void setRes(double res) throws MyException {

this.res = res;

}

public boolean prov(double a){

return a>0.000001 && a<1000000;

}

public RecIntegral(Object down,Object up,Object step,Object res) throws MyException{

try {

this.setDown(Double.parseDouble(down.toString()));

this.setUp(Double.parseDouble(up.toString()));

this.setStep(Double.parseDouble(step.toString()));

this.setRes(Double.parseDouble(res.toString()));

} catch (NumberFormatException e) {

throw new MyException("ошибка встречен символ или ,", e);

}catch (MyException ex) {

JOptionPane.showMessageDialog(null, ex.getMessage(),"err",JOptionPane.ERROR\_MESSAGE);

}

}

public RecIntegral(Object down,Object up,Object step) throws MyException{

try {

this.setDown(Double.parseDouble(down.toString()));

this.setUp(Double.parseDouble(up.toString()));

this.setStep(Double.parseDouble(step.toString()));

} catch (NumberFormatException e) {

throw new MyException("ошибка встречен символ или ,", e);

}catch (MyException ex) {

JOptionPane.showMessageDialog(null, ex.getMessage(),"err",JOptionPane.ERROR\_MESSAGE);

}

}

}

**Класс MyException**

package my.numberaddition;

public class MyException extends Exception{

public MyException() {

}

public MyException(String message) {

super(message);

}

public MyException(String message, Throwable cause) {

super(message, cause);

}

public MyException(Throwable cause) {

super(cause);

}

public MyException(String message, Throwable cause, boolean enableSuppression, boolean writableStackTrace) {

super(message, cause, enableSuppression, writableStackTrace);

}

}

**ContactEditorUI**

package my.numberaddition;

import java.awt.event.WindowAdapter;

import java.awt.event.WindowEvent;

import java.awt.event.WindowListener;

import java.net.\*;

import java.io.\*;

import java.util.Iterator;

import java.util.LinkedList;

import java.util.List;

import java.util.Scanner;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JFileChooser;

import javax.swing.JOptionPane;

import javax.swing.UIManager;

import javax.swing.table.DefaultTableModel;

public class ContactEditorUI extends javax.swing.JFrame implements WindowListener {

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

public static Socket clientSocket;

public static BufferedReader reader ;

public static OutputStreamWriter writer;

public ContactEditorUI() throws IOException{

clientSocket = new Socket("127.0.0.1", 8008);

if(clientSocket==null){

JOptionPane.showMessageDialog(null, "Не удалось подключиться к серверу","err",JOptionPane.ERROR\_MESSAGE);

}

reader = new BufferedReader (new InputStreamReader(clientSocket.getInputStream()));

writer = new OutputStreamWriter (clientSocket.getOutputStream());

addWindowListener(new WindowAdapter(){

@Override

public void windowClosing(WindowEvent e) {

System.out.println("2");

try {

writer.write("quit\n");

writer.flush();

reader.close();

writer.close();

clientSocket.close();

} catch (IOException ex) {

//System.out.println("Сервер стартовал локально (localhost) на порту: ");

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

}

//throw new UnsupportedOperationException("Not supported yet."); // Generated from nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody

}

});

initComponents();

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

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

private void initComponents() {

jButton1 = new javax.swing.JButton();

jtf\_down = new javax.swing.JTextField();

jtf\_up = new javax.swing.JTextField();

jtf\_step = new javax.swing.JTextField();

jScrollPane1 = new javax.swing.JScrollPane();

table\_integral = new javax.swing.JTable();

btn\_add = new javax.swing.JButton();

btn\_remove = new javax.swing.JButton();

btn\_calc = new javax.swing.JButton();

jLabel1 = new javax.swing.JLabel();

jLabel2 = new javax.swing.JLabel();

jLabel3 = new javax.swing.JLabel();

jLabel4 = new javax.swing.JLabel();

btn\_add\_table = new javax.swing.JButton();

btn\_clear = new javax.swing.JButton();

btn\_save = new javax.swing.JButton();

btn\_load\_t = new javax.swing.JButton();

btn\_save\_t = new javax.swing.JButton();

btn\_save\_b = new javax.swing.JButton();

btn\_load\_b = new javax.swing.JButton();

jButton1.setText("jButton1");

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

setTitle("1");

jtf\_down.setToolTipText("");

jtf\_down.setMinimumSize(new java.awt.Dimension(60, 20));

jtf\_down.addActionListener(new java.awt.event.ActionListener() {

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

jtf\_downActionPerformed(evt);

}

});

jtf\_up.setMinimumSize(new java.awt.Dimension(60, 20));

jtf\_up.setName(""); // NOI18N

jtf\_up.addActionListener(new java.awt.event.ActionListener() {

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

jtf\_upActionPerformed(evt);

}

});

jtf\_step.setToolTipText("");

jtf\_step.setMinimumSize(new java.awt.Dimension(60, 20));

table\_integral.setModel(new javax.swing.table.DefaultTableModel(

new Object [][] {

},

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];

}

});

jScrollPane1.setViewportView(table\_integral);

btn\_add.setText("Добавить");

btn\_add.setMaximumSize(new java.awt.Dimension(90, 23));

btn\_add.setMinimumSize(new java.awt.Dimension(90, 23));

btn\_add.setPreferredSize(new java.awt.Dimension(90, 23));

btn\_add.addActionListener(new java.awt.event.ActionListener() {

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

btn\_addActionPerformed(evt);

}

});

btn\_remove.setText("удалить");

btn\_remove.setMinimumSize(new java.awt.Dimension(90, 23));

btn\_remove.setPreferredSize(new java.awt.Dimension(90, 23));

btn\_remove.addActionListener(new java.awt.event.ActionListener() {

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

btn\_removeActionPerformed(evt);

}

});

btn\_calc.setText("Выполнить");

btn\_calc.setToolTipText("");

btn\_calc.setMinimumSize(new java.awt.Dimension(90, 23));

btn\_calc.setPreferredSize(new java.awt.Dimension(90, 23));

btn\_calc.addActionListener(new java.awt.event.ActionListener() {

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

btn\_calcActionPerformed(evt);

}

});

jLabel1.setFont(new java.awt.Font("Tahoma", 2, 12)); // NOI18N

jLabel1.setText("Шаг");

jLabel2.setFont(new java.awt.Font("Tahoma", 2, 12)); // NOI18N

jLabel2.setText("Верхняя граница");

jLabel3.setFont(new java.awt.Font("Tahoma", 2, 12)); // NOI18N

jLabel3.setText("Нижняя граница");

jLabel4.setFont(new java.awt.Font("Tahoma", 1, 24)); // NOI18N

jLabel4.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);

jLabel4.setText("Свойства интегрирования");

btn\_add\_table.setText("Заполнить");

btn\_add\_table.addActionListener(new java.awt.event.ActionListener() {

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

btn\_add\_tableActionPerformed(evt);

}

});

btn\_clear.setText("Очистить");

btn\_clear.addActionListener(new java.awt.event.ActionListener() {

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

btn\_clearActionPerformed(evt);

}

});

btn\_save.setText("Сохранить");

btn\_save.addActionListener(new java.awt.event.ActionListener() {

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

btn\_saveActionPerformed(evt);

}

});

btn\_load\_t.setText("Загрузить Т");

btn\_load\_t.addActionListener(new java.awt.event.ActionListener() {

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

btn\_load\_tActionPerformed(evt);

}

});

btn\_save\_t.setText("Сохранить как Т");

btn\_save\_t.addActionListener(new java.awt.event.ActionListener() {

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

btn\_save\_tActionPerformed(evt);

}

});

btn\_save\_b.setText("Сохранить как b");

btn\_save\_b.addActionListener(new java.awt.event.ActionListener() {

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

btn\_save\_bActionPerformed(evt);

}

});

btn\_load\_b.setText("Загрузить b");

btn\_load\_b.addActionListener(new java.awt.event.ActionListener() {

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

btn\_load\_bActionPerformed(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()

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

.addGroup(layout.createSequentialGroup()

.addGap(43, 43, 43)

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

.addComponent(jLabel2, javax.swing.GroupLayout.DEFAULT\_SIZE, 105, Short.MAX\_VALUE)

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

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

.addGap(55, 55, 55)

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

.addComponent(jtf\_down, javax.swing.GroupLayout.PREFERRED\_SIZE, 150, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jtf\_up, javax.swing.GroupLayout.PREFERRED\_SIZE, 150, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jtf\_step, javax.swing.GroupLayout.PREFERRED\_SIZE, 150, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addGroup(layout.createSequentialGroup()

.addGap(21, 21, 21)

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

.addGroup(layout.createSequentialGroup()

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 500, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

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

.addComponent(btn\_add, javax.swing.GroupLayout.PREFERRED\_SIZE, 105, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(btn\_remove, javax.swing.GroupLayout.PREFERRED\_SIZE, 105, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(btn\_calc, javax.swing.GroupLayout.PREFERRED\_SIZE, 105, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addComponent(jLabel4)

.addGroup(layout.createSequentialGroup()

.addGap(10, 10, 10)

.addComponent(btn\_save, javax.swing.GroupLayout.PREFERRED\_SIZE, 105, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addComponent(btn\_clear, javax.swing.GroupLayout.PREFERRED\_SIZE, 105, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addComponent(btn\_add\_table, javax.swing.GroupLayout.PREFERRED\_SIZE, 105, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(34, 34, 34)

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

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

.addComponent(btn\_save\_t, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

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

.addComponent(btn\_load\_t, javax.swing.GroupLayout.PREFERRED\_SIZE, 105, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(layout.createSequentialGroup()

.addComponent(btn\_save\_b)

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

.addComponent(btn\_load\_b, javax.swing.GroupLayout.PREFERRED\_SIZE, 105, javax.swing.GroupLayout.PREFERRED\_SIZE)))))))

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

);

layout.setVerticalGroup(

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

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addComponent(jLabel4)

.addGap(14, 14, 14)

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

.addComponent(jLabel3)

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

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

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

.addComponent(jLabel2)

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

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

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

.addComponent(jLabel1)

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

.addGap(18, 18, 18)

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

.addGroup(layout.createSequentialGroup()

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

.addGap(18, 18, 18)

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

.addGap(18, 18, 18)

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

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 125, javax.swing.GroupLayout.PREFERRED\_SIZE))

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

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

.addComponent(btn\_save)

.addComponent(btn\_clear)

.addComponent(btn\_add\_table)

.addComponent(btn\_load\_t)

.addComponent(btn\_save\_t))

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

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

.addComponent(btn\_save\_b)

.addComponent(btn\_load\_b))

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

);

pack();

}// </editor-fold>

private void jtf\_upActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

}

private void jtf\_downActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

}

private void btn\_addActionPerformed(java.awt.event.ActionEvent evt) {

DefaultTableModel model = (DefaultTableModel)table\_integral.getModel();

model.addRow(new Object[]{jtf\_down.getText(),jtf\_up.getText(), jtf\_step.getText()});

}

private void btn\_removeActionPerformed(java.awt.event.ActionEvent evt) {

DefaultTableModel model = (DefaultTableModel)table\_integral.getModel();

model.removeRow(table\_integral.getSelectedRow());

}

private void btn\_calcActionPerformed(java.awt.event.ActionEvent evt) {

DefaultTableModel model = (DefaultTableModel)table\_integral.getModel();

int row = table\_integral.getSelectedRow();

if (row>=0){

try {

if (Counter.get()>0){

JThread t = new JThread("JThread"+(5-Counter.get()),model.getValueAt(row, 0),model.getValueAt(row, 1),model.getValueAt(row, 2), row, model);

t.start();

}else{

JOptionPane.showMessageDialog(null,"Попробуйте позже, все потоки заняты","err",JOptionPane.ERROR\_MESSAGE );

}

} catch (MyException e) {

if(e.getCause()!= null){

JOptionPane.showMessageDialog(null, e.getMessage()+" ("+e.getCause()+")","err",JOptionPane.ERROR\_MESSAGE );

}else{

JOptionPane.showMessageDialog(null, e.getMessage(),"err",JOptionPane.ERROR\_MESSAGE );

}

}

}

}

private void btn\_add\_tableActionPerformed(java.awt.event.ActionEvent evt) {

DefaultTableModel model = (DefaultTableModel)table\_integral.getModel();

Iterator<RecIntegral> iterator = list.iterator();

while(iterator.hasNext()){

RecIntegral rec = iterator.next();

model.addRow(new Object[]{rec.getDown(),rec.getUp(),rec.getStep(),rec.getRes()});

iterator.remove();

}

}

private void btn\_clearActionPerformed(java.awt.event.ActionEvent evt) {

DefaultTableModel model = (DefaultTableModel)table\_integral.getModel();

model.setRowCount(0);

}

private void btn\_saveActionPerformed(java.awt.event.ActionEvent evt) {

DefaultTableModel model = (DefaultTableModel)table\_integral.getModel();

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

if(model.getValueAt(i, 3)== null){

try {

list.add(new RecIntegral(model.getValueAt(i, 0),model.getValueAt(i, 1),model.getValueAt(i, 2)));

} catch (MyException e) {

JOptionPane.showMessageDialog(null, e.getMessage()+" ("+e.getCause()+")","err",JOptionPane.ERROR\_MESSAGE);

}

//list.add(new RecIntegral(Double.parseDouble((model.getValueAt(i, 0)).toString()),Double.parseDouble((model.getValueAt(i, 1)).toString()),Double.parseDouble((model.getValueAt(i, 2)).toString())));

}

else{

try {

list.add(new RecIntegral(model.getValueAt(i, 0),model.getValueAt(i, 1),model.getValueAt(i, 2),model.getValueAt(i, 3)));

} catch (MyException e) {

JOptionPane.showMessageDialog(null, e.getMessage()+" ("+e.getCause()+")","err",JOptionPane.ERROR\_MESSAGE);

}

// list.add(new RecIntegral(Double.parseDouble((model.getValueAt(i, 0)).toString()),Double.parseDouble((model.getValueAt(i, 1)).toString()),Double.parseDouble((model.getValueAt(i, 2)).toString()),Double.parseDouble((model.getValueAt(i, 3)).toString())));

}

}

}

private void btn\_load\_tActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

btn\_clearActionPerformed(evt);

String line;

JFileChooser fc = new JFileChooser();

fc.setDialogTitle("Выбор директории");

if (fc.showOpenDialog(null) == JFileChooser.APPROVE\_OPTION) {

try{

BufferedReader reader = new BufferedReader (new FileReader(fc.getSelectedFile()));

while ((line = reader.readLine()) != null) {

Scanner scanner = new Scanner(line);

Object down = scanner.next();

if (Double.parseDouble(down.toString()) == -1.1) {

break;

}

Object up = scanner.next();

Object step = scanner.next();

Object res = scanner.next();

list.add(new RecIntegral(down,up,step,res));

}

btn\_add\_tableActionPerformed(evt);

}

catch(Exception e){

System.out.println("Что-то пошло не так...");

}

}

}

private void btn\_save\_tActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

btn\_saveActionPerformed(evt);

JFileChooser fc = new JFileChooser();

if (fc.showSaveDialog(null) == JFileChooser.APPROVE\_OPTION) {

try{

BufferedWriter writer = new BufferedWriter (new FileWriter(fc.getSelectedFile()));

Iterator<RecIntegral> iterator = list.iterator();

while(iterator.hasNext()){

RecIntegral rec = iterator.next();

writer.write(

Double.toString(rec.getDown()) + " "

+ Double.toString(rec.getUp()) + " "

+ Double.toString(rec.getStep()) + " "

+ Double.toString(rec.getRes()) + "\n");

iterator.remove();

}

writer.write(Double.toString(-1.1));

writer.close();

}

catch(Exception e){

System.out.println("Что-то пошло не так...");

}

}

}

private void btn\_save\_bActionPerformed(java.awt.event.ActionEvent evt) {

btn\_saveActionPerformed(evt);

JFileChooser fc = new JFileChooser();

if (fc.showSaveDialog(null) == JFileChooser.APPROVE\_OPTION) {

try {

FileOutputStream fos = new FileOutputStream(fc.getSelectedFile());

BufferedOutputStream bos = new BufferedOutputStream(fos);

ObjectOutputStream os = new ObjectOutputStream(bos);

os.writeObject(list);

os.flush();

os.close();

btn\_clearActionPerformed(evt);

btn\_add\_tableActionPerformed(evt);

}

catch (Exception e) {

System.out.println("Что-то пошло не так...");

}

}

}

private void btn\_load\_bActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

JFileChooser fc = new JFileChooser();

fc.setDialogTitle("Выбор директории");

if (fc.showOpenDialog(null) == JFileChooser.APPROVE\_OPTION) {

try{

FileInputStream fis = new FileInputStream(fc.getSelectedFile());

BufferedInputStream bis = new BufferedInputStream(fis);

ObjectInputStream is = new ObjectInputStream(bis);

LinkedList<RecIntegral> myList = (LinkedList<RecIntegral>) is.readObject();

list = myList;

btn\_clearActionPerformed(evt);

btn\_add\_tableActionPerformed(evt);

is.close();

}

catch(Exception e){

System.out.println("Что-то пошло не так...");

}

}

}

public double f(double x) {

double F=Math.tan(x);

return F;

}

public double integral(Object a1, Object b1, Object h1) throws MyException{

try{

double a =Double.parseDouble(a1.toString());

double b =Double.parseDouble(b1.toString());

double h = Double.parseDouble(h1.toString());

if(b<a){

throw new MyException("верхняя граница меньше чем нижняя");

}

if(h>(b-a)){

throw new MyException("Шаг перевашает");

}

double res=0;

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

res += f(a)+f(b);

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

res += 2\*f(a + h\*i );

}

return h\*res/2;

}catch(NumberFormatException e){

throw new MyException("ошибка встречен символ или ,", e);

}

}

в

UIManager.put(

"FileChooser.saveButtonText", "Сохранить");

UIManager.put(

"FileChooser.cancelButtonText", "Отмена");

UIManager.put(

"FileChooser.fileNameLabelText", "Наименование файла");

UIManager.put(

"FileChooser.filesOfTypeLabelText", "Типы файлов");

UIManager.put(

"FileChooser.lookInLabelText", "Директория");

UIManager.put(

"FileChooser.saveInLabelText", "Сохранить в директории");

UIManager.put(

"FileChooser.folderNameLabelText", "Путь директории");

UIManager.put(

"FileChooser.openButtonText", "Открыть");

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

} catch (InstantiationException ex) {

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

} catch (IllegalAccessException ex) {

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

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

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

writer.write(clientSocket.getLocalAddress()+"\n");

writer.flush();

JOptionPane.showMessageDialog(null,reader.readLine(),"yved",JOptionPane.INFORMATION\_MESSAGE );

} catch (IOException ex) {

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

}

}

});

}

// Variables declaration - do not modify

private javax.swing.JButton btn\_add;

private javax.swing.JButton btn\_add\_table;

private javax.swing.JButton btn\_calc;

private javax.swing.JButton btn\_clear;

private javax.swing.JButton btn\_load\_b;

private javax.swing.JButton btn\_load\_t;

private javax.swing.JButton btn\_remove;

private javax.swing.JButton btn\_save;

private javax.swing.JButton btn\_save\_b;

private javax.swing.JButton btn\_save\_t;

private javax.swing.JButton jButton1;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JTextField jtf\_down;

private javax.swing.JTextField jtf\_step;

private javax.swing.JTextField jtf\_up;

private javax.swing.JTable table\_integral;

// End of variables declaration

@Override

public void windowOpened(WindowEvent e) {

System.out.println("1");

throw new UnsupportedOperationException("Not supported yet."); // Generated from nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody

}

@Override

public void windowClosing(WindowEvent e) {

System.out.println("2");

try {

reader.close();

writer.write("quit \n");

writer.flush();

writer.close();

clientSocket.close();

} catch (IOException ex) {

//System.out.println("Сервер стартовал локально (localhost) на порту: ");

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

}

throw new UnsupportedOperationException("Not supported yet."); // Generated from nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody

}

@Override

public void windowClosed(WindowEvent e) {

System.out.println("3");

//throw new UnsupportedOperationException("Not supported yet."); // Generated from nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody

}

@Override

public void windowIconified(WindowEvent e) {

System.out.println("4");

throw new UnsupportedOperationException("Not supported yet."); // Generated from nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody

}

@Override

public void windowDeiconified(WindowEvent e) {

System.out.println("5");

throw new UnsupportedOperationException("Not supported yet."); // Generated from nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody

}

@Override

public void windowActivated(WindowEvent e) {

System.out.println("6");

throw new UnsupportedOperationException("Not supported yet."); // Generated from nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody

}

@Override

public void windowDeactivated(WindowEvent e) {

System.out.println("7");

throw new UnsupportedOperationException("Not supported yet."); // Generated from nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody

}

}

**class Fabrica**

import java.net.\*;

import java.util.concurrent.ExecutorService;

import java.util.concurrent.Executors;

import java.io.\*;

public class Fabrica {

static ExecutorService executeIt = Executors.newFixedThreadPool(5);

public static void main(String[] args) {

var port = 8008;

try (

ServerSocket serverSocket = new ServerSocket(port);

BufferedReader br = new BufferedReader(new InputStreamReader(System.in))) {

System.out.println("Сервер стартовал локально (localhost) на порту: "+port);

// стартуем цикл при условии что серверный сокет не закрыт

while (!serverSocket.isClosed()) {

// проверяем поступившие комманды из консоли сервера если такие

// были

if (br.ready()) {

System.out.println("Main Server found any messages in channel, let's look at them.");

// если команда - quit то инициализируем закрытие сервера и

// выход из цикла раздачии нитей монопоточных серверов

String serverCommand = br.readLine();

if (serverCommand.equalsIgnoreCase("quit")) {

System.out.println("Main Server initiate exiting...");

serverSocket.close();

break;

}

}

// если комманд от сервера нет то становимся в ожидание

// подключения к сокету общения под именем - "clientDialog" на

// серверной стороне

Socket client = serverSocket.accept();

// после получения запроса на подключение сервер создаёт сокет

// для общения с клиентом и отправляет его в отдельную нить

// в Runnable(при необходимости можно создать Callable)

executeIt.execute(new Server(client));

System.out.print("Connection accepted.");

}

// закрытие пула нитей после завершения работы всех нитей

executeIt.shutdown();

} catch (IOException e) {

e.printStackTrace();

}

}

}

**class Server**

import java.net.\*;

import java.io.\*;

public class Server implements Runnable{

private static Socket clientDialog;

public Server (Socket client) {

Server.clientDialog = client;

}

public static double f(double x) {

double F=Math.tan(x);

return F;

}

public static String calc(double a, double b, double h) {

double res=0;

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

res += f(a)+f(b);

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

res += 2\*f(a + h\*i );

}

res = h\*res/2;

System.out.println(res);

return String.valueOf(res);

}

@Override

public void run() {

try {

// инициируем каналы общения в сокете, для сервера

OutputStreamWriter writer = new OutputStreamWriter (clientDialog.getOutputStream());

BufferedReader reader =new BufferedReader (new InputStreamReader(clientDialog.getInputStream()));

System.out.println("Client "+ reader.readLine()+" conected");

writer.write("Успешное подключение \n");

writer.flush();

// начинаем диалог с подключенным клиентом в цикле, пока сокет не

// закрыт клиентом

while (!clientDialog.isClosed()) {

System.out.println("Server reading from channel");

// серверная нить ждёт в канале чтения (inputstream) получения

// данных клиента после получения данных считывает их

String response = reader.readLine();

System.out.println("запрос: "+response);

if (response.equalsIgnoreCase("quit")) {

// если кодовое слово получено то инициализируется закрытие

// серверной нити

System.out.println("Client initialize connections suicide ...");

writer.write("Server reply - " + response + "\n");

Thread.sleep(3000);

break;

}

String[] commands = response.split("#");

for (String comand : commands) {

if(comand.isEmpty()){

continue;

}

if(comand.contains("calc")){

System.out.println("Calc");

String[] params = comand.split("\\|");

for (String param : params) {

System.out.println(param);

}

if(params[1]!= null && params[2]!= null && params[3]!= null ){

System.out.println(params[1]);

System.out.println(params[2]);

System.out.println(params[3]);

try{

String req = calc(Double.parseDouble(params[1]),Double.parseDouble(params[2]),Double.parseDouble(params[3]));

System.out.println("return");

writer.write(req+"\n");

writer.flush();

}catch(Exception e){}

}

}

}

// освобождаем буфер сетевых сообщений

writer.flush();

// возвращаемся в началло для считывания нового сообщения

}

// если условие выхода - верно выключаем соединения

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

System.out.println("Closing connections & channels.");

// закрываем сначала каналы сокета !

//reader.close();

//writer.close();

// потом закрываем сокет общения с клиентом в нити моносервера

clientDialog.close();

System.out.println("Closing connections & channels - DONE.");

} catch (IOException e) {

e.printStackTrace();

} catch (InterruptedException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

**Результаты выполнения программы**

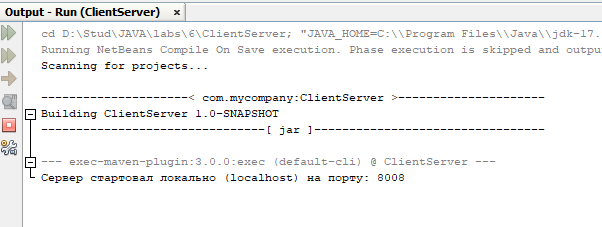


Рисунок 1 - Сервер запущен

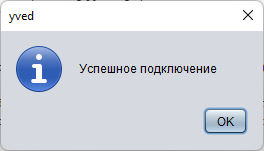


Рисунок 2 - Уведомление о подключении на стороне клиента

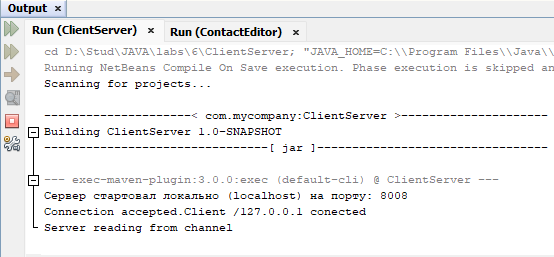


Рисунок 3 - Уведомление о подключении клиента на стороне сервера

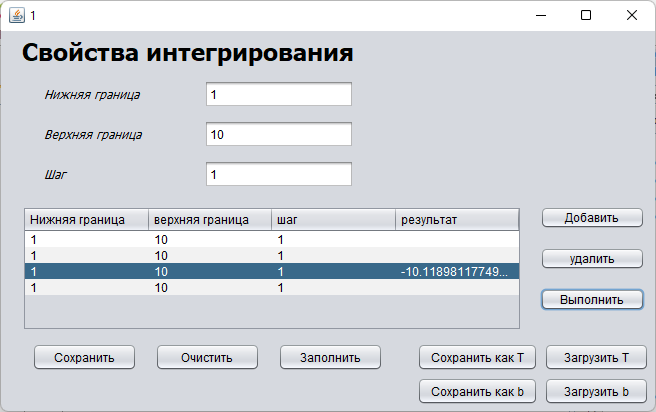


Рисунок 4 - Вычисление и результат на стороне клиента

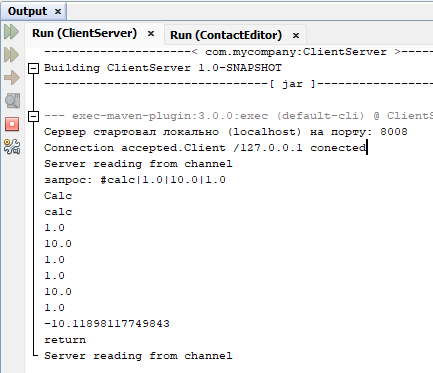


Рисунок 5 - Вычисление и результат на стороне сервера

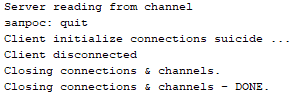


Рисунок 6 - Уведомление на сервере о том что пользователь отключился

**Вывод:** выполняя лабораторное задание, научились создавать многопоточные приложения c использованием стандартных средств языка Java.