СПбНИУ ИТМО

Отчет

По лабораторной работе №5

«Программирование интернет приложений»

Выполнил: Ратников В.И

Группа 3105

Преподаватель: Гаврилов А.В.

**Исходный код программы**

**Код Сервера( Server.java)**

package server;

import java.io.IOException;

public class Server

{

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

{

if(args.length==0)

{

System.err.println("\nUsage : Server [port]");

System.exit(-1);

}

Listener ls = new Listener( Integer.parseInt(args[0]) );

System.err.println("\nServer launched on port " + args[0] );

Thread main\_listen\_thread = new Thread(ls);

main\_listen\_thread.start();

}

}

**Listener.java**

package server;

import java.io.IOException;

import java.net.InetSocketAddress;

import java.net.ServerSocket;

import java.net.Socket;

import java.text.DateFormat;

import java.util.Calendar;

import java.util.logging.Level;

import java.util.logging.Logger;

import var333.Contour;

import var333.Mark;

public class Listener/\* implements Runnable\*/ extends Thread

{

ServerSocket sk;

public Listener(int port) throws IOException

{

//super(this.run());

sk = new ServerSocket();

sk.setReuseAddress(true);

sk.bind(new InetSocketAddress(port));

}

@Override

public synchronized void start()

{

//super.start(); //To change body of generated methods, choose Tools | Templates.

run();

}

public void run()

{

Socket client = null;

int CLID = 0;

//contains client message

String tmp = "";

while(true)

{

try

{

client = sk.accept();

CLID = client.getPort();

System.out.println( time() + " accepted client id = " + CLID);

//reading from client

byte[] buff = new byte[128];

String data = "";

//while not EOF continue reading data

int r = client.getInputStream().read(buff);

data = new String(buff);

tmp += data;

data = "";

//end reading

double X,Y,R;

int posXst,posYst,posRst,posXend,posYend,posRend;

posXst = tmp.indexOf("X::");

posXend = tmp.indexOf("::Y");

posYst = tmp.indexOf("Y::");

posYend = tmp.indexOf("::R");

posRst = tmp.indexOf("R::");

posRend = tmp.indexOf("::END");

X = Double.parseDouble(tmp.substring(posXst+3,posXend));

Y = Double.parseDouble(tmp.substring(posYst+3,posYend));

R = Double.parseDouble(tmp.substring(posRst+3,posRend));

System.out.println(CLID + " message : " + tmp);

tmp = "";

System.out.println(CLID + " creating calculate context");

Contour c = new Contour((float)R);

boolean hit = c.is\_hit(new Mark((float)X, (float)Y));

System.out.println(CLID + " context created. calculating response data : HIT = " + Boolean.toString(hit) );

System.out.println(CLID + " sending results to client");

String response="";

if(hit)

{

response="true";

}

else

{

response="false";

}

client.getOutputStream().write(response.getBytes());

//closing connection

client.close();

System.out.println( time() + " closed client id = " + CLID);

}

catch(IOException ioe)

{

System.err.println(ioe + ioe.toString());

tmp = "";

}

//reason : incorrect client request

catch(StringIndexOutOfBoundsException oub)

{

System.err.println(CLID + " incorrect client. closing");

try

{

client.close();

tmp = "";

} catch (IOException ex)

{

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

}

}

//reason : unable to parse double

catch(NumberFormatException nfe)

{

System.err.println(nfe + nfe.toString());

try

{

client.close();

tmp = "";

} catch (IOException ex)

{

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

}

}

}

}

//throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.

String time()

{

return DateFormat.getInstance().format(Calendar.getInstance().getTime());

}

}

**Код клиента(Сlient.java)**

package client;

import java.awt.BorderLayout;

import java.awt.Canvas;

import java.awt.Color;

import java.awt.Font;

import java.awt.GridLayout;

import java.awt.Panel;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.MouseEvent;

import java.awt.event.MouseListener;

import java.net.InetAddress;

import java.net.UnknownHostException;

import java.util.Locale;

import java.util.ResourceBundle;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JComboBox;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JRadioButton;

import javax.swing.JSlider;

import javax.swing.JTextField;

import javax.swing.SwingUtilities;

import javax.swing.WindowConstants;

import javax.swing.event.ChangeEvent;

import javax.swing.event.ChangeListener;

import var333.Contour;

import var333.Mark;

public class Client implements Runnable

{

//глобальные переменные: радиус,контур,точка из прошлой лабы

public static int radius = 1,

curr\_x,

curr\_y;

public static Contour contour;

public static Mark mark;

//элементы подключения к серверу

public static JFrame server\_frame;

public static JTextField server\_addr\_text;

public static JTextField server\_port\_text;

ActionListener serv\_addr\_changed;

ActionListener serv\_port\_changed;

//элементы управления

public static JFrame form;

public static Panel p\_elements,p\_canvas;

public static Canvas paint\_area;

public static JComboBox x\_coord;

public static JRadioButton y\_1,y\_2,y\_3,y\_4;

public static JSlider radius\_slider;

public static JLabel label\_coords;

public static int server\_port;

public static InetAddress server\_addr;

//слушатели событий

ActionListener x\_changed;

ActionListener y\_changed;

ChangeListener radius\_changed;

MouseListener mouse\_click;

//локализация

ResourceBundle rb\_def;

//точка входа

public static void main(String[] args)

{

SwingUtilities.invokeLater(new client.Client());

}

public void paint\_all()

{

paint\_area.getGraphics().clearRect(0, 0, paint\_area.getWidth(), paint\_area.getHeight());

painter.Painter.draw\_var333\_figure(paint\_area, 25, radius);

painter.Painter.draw\_grid\_on\_panel(paint\_area,25);

}

//инициализация глобальных переменных

public void init\_globals()

{

mark = new Mark(curr\_x, curr\_y);

contour = new Contour(radius);

form.setTitle( rb\_def.getString("radius") + " = " + String.valueOf(radius));

}

//инициализация элементов управления

public void init\_elements()

{

//Locale.setDefault(Locale.);

Locale loc = Locale.getDefault();

//System.err.println(loc.getCountry());

//System.err.println(loc.getLanguage());

rb\_def = ResourceBundle.getBundle("Labels",loc);

server\_frame = new JFrame( rb\_def.getString("type\_addr\_port") );

server\_frame.getContentPane().setLayout(new GridLayout(1,2));

server\_addr\_text = new JTextField("127.0.0.1");

server\_port\_text = new JTextField("1111");

server\_frame.add(server\_addr\_text);

server\_frame.add(server\_port\_text);

server\_frame.pack();

server\_frame.setSize(600, 50);

server\_frame.setVisible(true);

server\_frame.setAlwaysOnTop(true);

//server\_frame.add()

form = new JFrame( rb\_def.getString("labwork\_id") );

form.getContentPane().setLayout(new BorderLayout());

//панель с элементами управления

p\_elements = new Panel(new GridLayout(7,1));

p\_elements.setForeground(Color.WHITE);

p\_elements.add(x\_coord = new JComboBox());

x\_coord.addItem("X=1");

x\_coord.addItem("X=2");

x\_coord.addItem("X=3");

x\_coord.addItem("X=4");

p\_elements.add(y\_1 = new JRadioButton("Y=1"));

p\_elements.add(y\_2 = new JRadioButton("Y=2"));

p\_elements.add(y\_3 = new JRadioButton("Y=3"));

p\_elements.add(y\_4 = new JRadioButton("Y=4"));

p\_elements.add(radius\_slider = new JSlider(1, 10, 1));

p\_elements.add(label\_coords = new JLabel( rb\_def.getString("coordinates") ));

label\_coords.setFont(new Font(Font.MONOSPACED, Font.BOLD, 20));

//панель с областью рисования

p\_canvas = new Panel(new GridLayout());

paint\_area = new Canvas();

paint\_area.setBackground(new Color(0,250,0));

paint\_area.setSize(400, 300);

p\_canvas.add(paint\_area);

form.add(p\_elements,BorderLayout.WEST);

form.add(p\_canvas,BorderLayout.CENTER);

form.pack();

form.setVisible(true);

}

//инициализация событий

public void init\_events()

{

//обработчик события смены радиуса

radius\_changed = new ChangeListener() {

@Override

public void stateChanged(ChangeEvent e)

{

//поменять радиус

radius = radius\_slider.getValue();

radius\_slider.setToolTipText( String.valueOf(radius\_slider.getValue()) );

//пересчитать контур

init\_globals();

paint\_all();

}

};

radius\_slider.addChangeListener(radius\_changed);

//обработчик события смены координаты X

x\_changed = new ActionListener()

{

@Override

public void actionPerformed(ActionEvent e)

{

curr\_x = x\_coord.getSelectedIndex()+1;

init\_globals();

label\_coords.setText(mark.toString());

try

{ //c\s

TCPConn tc = new TCPConn(server\_port, server\_addr);

boolean hit = tc.get\_hit\_result(curr\_x, curr\_y, radius);

painter.Painter.draw\_animate\_cursor(

paint\_area,

25,

(int)Mark.Translate\_reverse(mark, paint\_area, 25).getX(),

(int)Mark.Translate\_reverse(mark, paint\_area, 25).getY(),

radius,

hit

);

} catch (InterruptedException ex)

{

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

}

}

};

x\_coord.addActionListener(x\_changed);

//обработчик события смены координаты Y

y\_changed = new ActionListener()

{

@Override

public void actionPerformed(ActionEvent e)

{

//some indian code

if(y\_1.isSelected())

{

y\_2.setSelected(false);

y\_3.setSelected(false);

y\_4.setSelected(false);

curr\_y = 1;

}

if(y\_2.isSelected())

{

y\_1.setSelected(false);

y\_3.setSelected(false);

y\_4.setSelected(false);

curr\_y = 2;

}

if(y\_3.isSelected())

{

y\_1.setSelected(false);

y\_2.setSelected(false);

y\_4.setSelected(false);

curr\_y = 3;

}

if(y\_4.isSelected())

{

y\_1.setSelected(false);

y\_2.setSelected(false);

y\_3.setSelected(false);

curr\_y = 4;

}

init\_globals();

try

{

//TCPConn tc = new TCPConn(curr\_x, null)

TCPConn tc = new TCPConn(server\_port, server\_addr);

boolean hit = tc.get\_hit\_result(curr\_x, curr\_y, radius);

painter.Painter.draw\_animate\_cursor(

paint\_area,

25,

(int)Mark.Translate\_reverse(mark, paint\_area, 25).getX(),

(int)Mark.Translate\_reverse(mark, paint\_area, 25).getY(),

radius,

hit

);

} catch (InterruptedException ex)

{

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

}

label\_coords.setText(mark.toString());

}

};

y\_1.addActionListener(y\_changed);

y\_2.addActionListener(y\_changed);

y\_3.addActionListener(y\_changed);

y\_4.addActionListener(y\_changed);

mouse\_click = new MouseListener()

{

@Override

public void mouseClicked(MouseEvent e)

{

try

{

Mark m\_tmp = Mark.Translate(paint\_area, e.getX(), e.getY(), 25);

TCPConn tc = new TCPConn(server\_port, server\_addr);

boolean hit = tc.get\_hit\_result(m\_tmp.getX(), m\_tmp.getY(), radius);

painter.Painter.draw\_animate\_cursor(

paint\_area,

25,

e.getX(),

e.getY(),

//(int)Mark.Translate\_reverse(mark, paint\_area, 25).getX(),

//(int)Mark.Translate\_reverse(mark, paint\_area, 25).getY(),

radius,

hit

);

label\_coords.setText(Mark.Translate(paint\_area, e.getX(), e.getY(), 25).toString());

} catch (InterruptedException ex)

{

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

}

}

@Override

public void mousePressed(MouseEvent e) {

//label\_coords.setText("Pressed");

//throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.

}

@Override

public void mouseReleased(MouseEvent e) {

//label\_coords.setText("Released");

//throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.

}

@Override

public void mouseEntered(MouseEvent e) {

//label\_coords.setText("Entered at ");

// throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.

}

@Override

public void mouseExited(MouseEvent e) {

//label\_coords.setText("Exited");

// throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.

}

};

paint\_area.addMouseListener(mouse\_click);

//событие установки хоста сервера

serv\_addr\_changed = new ActionListener()

{

@Override

public void actionPerformed(ActionEvent e)

{

try

{

server\_addr = InetAddress.getByName(server\_addr\_text.getText());

//throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.

} catch (UnknownHostException ex)

{

server\_frame.setTitle(rb\_def.getString("unknown\_host") );

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

}

}

};

server\_addr\_text.addActionListener(serv\_addr\_changed);

serv\_port\_changed = new ActionListener() {

@Override

public void actionPerformed(ActionEvent e)

{

try

{

server\_port = Integer.parseInt(server\_port\_text.getText());

}

catch(NumberFormatException nfe)

{

server\_frame.setTitle( rb\_def.getString("incorrect\_id") );

}

//throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.

}

};

server\_port\_text.addActionListener(serv\_port\_changed);

form.setDefaultCloseOperation(WindowConstants.EXIT\_ON\_CLOSE);

}

@Override

public void run()

{

init\_elements();

init\_events();

init\_globals();

}

}

**TCPconn.java**

/\*

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

package client;

import java.io.IOException;

import java.lang.reflect.Field;

import java.net.InetAddress;

import java.net.InetSocketAddress;

import java.net.Socket;

import java.net.SocketAddress;

import java.text.FieldPosition;

import java.text.NumberFormat;

import java.text.ParsePosition;

public class TCPConn

{

int port;

InetAddress addr;

Socket clientSock;

public TCPConn(int \_port,InetAddress \_servAddr)

{

port = \_port;

addr = \_servAddr;

clientSock = new Socket();

}

boolean get\_hit\_result(double x,double y,double radius)

{

try

{

SocketAddress sa = new InetSocketAddress(addr, port);

clientSock.connect(sa, 5000);

StringBuffer sbuf = new StringBuffer();

sbuf.append("X::");

sbuf.append(x);

sbuf.append("::Y::");

sbuf.append(y);

sbuf.append("::R::");

sbuf.append(radius);

sbuf.append("::END");

String outs = sbuf.toString();

clientSock.getOutputStream().write(outs.getBytes());

byte[] tmp = new byte[32];

clientSock.getInputStream().read(tmp);

String interchange = new String(tmp);

// System.out.println(interchange);

clientSock.close();

if(interchange.contains("true"))

return true;

//return Boolean.parseBoolean(interchange);

//clientSock.getOutputStream().write();

}

catch(IOException ioe)

{

System.err.println(ioe + ioe.toString());

}

return false;

}

}