Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №6

По дисциплине: «СПП»

Тема: “Клиент-серверная архитектура приложений Java.”

Выполнил:

Студент 3 курса

Группы ПО-7

Комиссаров А.Е.

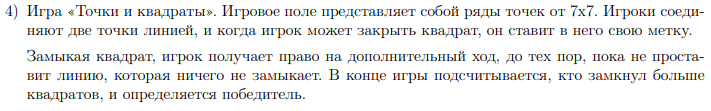
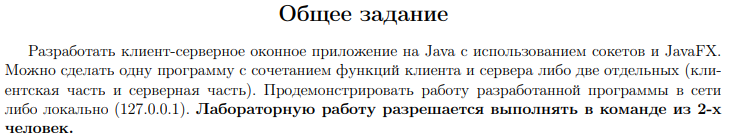
Дмитрук М.А.

Проверил:

Монтик Н.С.

2023

**Цель работы:** освоить приёмы разработки оконных клиент-серверных приложений на Java с использованием сокетов.



**Код программы:**

**Server.java**

package gameserver;

import java.io.DataInputStream;

import java.io.DataOutputStream;

import java.io.IOException;

import java.net.ServerSocket;

import java.net.Socket;

public class GameServer {

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

ServerSocket myServer= new ServerSocket(8080);

while(true){

// Player1

Socket player1 = myServer.accept();

DataInputStream in1 = new DataInputStream(player1.getInputStream());

DataOutputStream out1 = new DataOutputStream(player1.getOutputStream());

out1.writeUTF("1");

String name1 = in1.readUTF();

// Player2

Socket player2= myServer.accept();

DataInputStream in2 = new DataInputStream(player2.getInputStream());

DataOutputStream out2 = new DataOutputStream(player2.getOutputStream());

out2.writeUTF("2");

String name2 = in2.readUTF();

// для первого окна

out1.writeUTF(name1);

out1.writeUTF(name2);

// для второго окна

out2.writeUTF(name1);

out2.writeUTF(name2);

NewClient c = new NewClient(in1,out1,in2,out2);

Thread t = new Thread(c);

t.start();

}

}

}

**Player.java**

package projectrs;

public class Player {

int score;

String name;

boolean turn = false;

boolean isWinner=false;

boolean current=false;

boolean square =false;

Player() {

score=0;

}

Player(String s)

{

score = 0;

name = s;

}

}

**Multiplayer.java**

package projectrs;

import java.awt.BorderLayout;

import java.io.DataInputStream;

import java.io.DataOutputStream;

import java.io.IOException;

import java.net.Socket;

import java.util.LinkedList;

import java.util.List;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

public class Multiplayer {

//GameOver go;

Multiplayer(String s,Socket socket) throws IOException {

Player player1,player2;

JFrame board;

player1 = new Player();

player2 = new Player();

List l = new LinkedList();

DataOutputStream out = new DataOutputStream(socket.getOutputStream());

DataInputStream in = new DataInputStream(socket.getInputStream());

if("1".equals(s))

{

player1.name = JOptionPane.showInputDialog(null, "Введите имя первого игрока" , "Player 1");

out.writeUTF(player1.name);

player1.current=true;

player1.turn=true;

player2.turn = false;

}

if("2".equals(s))

{

player2.name = JOptionPane.showInputDialog(null, "Введите имя второго игрока" , "Player 2");

out.writeUTF(player2.name);

player2.current=true;

player2.turn=false;

player1.turn=true;

}

if(player1.name == null)

{

player1.name = "Player 1";

}

if(player2.name == null)

{

player2.name = "Player 2";

}

//go = new GameOver();

player1.name = in.readUTF();

player2.name = in.readUTF();

System.out.println(player1.name+" "+player2.name);

board = new Draw(player1, player2,l,in,out);

board.setVisible(true);

}

}

**Draw.java**

package projectrs;

import java.awt.Color;

import java.awt.Component;

import java.awt.Dimension;

import java.awt.Font;

import java.awt.Graphics;

import java.awt.Point;

import java.awt.Rectangle;

import java.awt.event.MouseAdapter;

import java.awt.event.MouseEvent;

import java.io.DataInputStream;

import java.io.DataOutputStream;

import java.io.IOException;

import java.util.List;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JFrame;

import static javax.swing.JFrame.EXIT\_ON\_CLOSE;

import javax.swing.JLabel;

import javax.swing.JPanel;

public class Draw extends JFrame {

JPanel jp;

GameOver go;

int mousx,mousy;

int ii;

//Offline:

public Draw(Player Player1, Player Player2,List l) {

super("Точки и Квадраты");

setSize(800, 400);

setDefaultCloseOperation(EXIT\_ON\_CLOSE);

go = new GameOver(this);

jp = new GPanel(Player1, Player2, l,go);

jp.setVisible(true);

jp.addMouseListener(new MouseAdapter() {

@Override

public void mouseClicked(MouseEvent e) {

GPanel Temp = (GPanel) jp;

Temp.isTurn = true;

mousx = e.getX();

mousy = e.getY();

int i = mousx;

int j = mousy;

if(needsLine(i,j))

{

if(((i+ii-20)%60)==0)

{

i = i + ii - 20;

int yyy = (j-20)/60;

yyy=yyy\*60 + 20;

if(yyy>=0 && yyy < 281)

{

l.add((i+27)+" "+ (yyy+7)+" " +(i+27)+" "+(yyy+60+7));

repaint();

}

}

if(((j+ii-20)%60)==0 && i+ii<=340)

{

j = j + ii -20;

int xxx = (i-20)/60;

xxx=xxx\*60 +20;

if(xxx>=0 && xxx < 281)

{

l.add((xxx+7)+" "+ (j+27)+" " +(xxx+60+7)+" "+(j+27));

repaint();

}

}

}

if(Temp.gameOver)

setVisible(false);

}

});

Component add = add(jp);

jp.setVisible(true);

}

//Online:

void detectMouse(Player Player1, Player Player2, List l, DataInputStream in, DataOutputStream out, JPanel jp) {

jp.addMouseListener(new MouseAdapter() {

@Override

public void mouseClicked(MouseEvent e) {

GpanelM Temp = (GpanelM) jp;

mousx = e.getX();

mousy = e.getY();

int i=mousx;

int j=mousy;

if(needsLine(i,j))

{

if(((i+ii-20)%60)==0)

{

i = i+ii-20;

int yyy = (j-20)/60;

yyy = yyy\*60+20;

if(yyy>=0 && yyy <281)

{

Line tmp = new Line((i+27),(yyy+7),(i+27),(yyy+60+7));

if(tmp.getTrueValue(mc)==false)

{

if((Player1.current && Player1.turn) || (Player2.current && Player2.turn)) {

l.add((i+27)+" "+(yyy+7)+" "+(i+27)+" "+(yyy+60+7));

Player1.square=false;

Player2.square=false;

try {

out.writeUTF((i+27)+" "+(yyy+7)+" "+(i+27)+" "+(yyy+60+7));

String s= in.readUTF();

if("p1".equals(s)) {

Player1.turn = false;

Player2.turn = true;

s=in.readUTF();

}

if("p2".equals(s)) {

Player2.turn=false;

Player1.turn = true;

s=in.readUTF();

}

if(Player1.turn && Player1.current && !"bla".equals(s))

{

Player1.turn = false;

Player2.turn = true;

}

else if(Player2.turn && Player2.current && !"bla".equals(s))

{

Player2.turn = false;

Player1.turn = true;

}

} catch (IOException ex) {

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

}

}

}

//repaint();

}

}

else if(((j+ii-20)%60)==0 && i+ii <=340)

{

j = j+ii-20;

int xxx = (i-20)/60;

xxx=xxx\*60+20;

if(xxx>=0 && xxx < 281)

{

Line tmp = new Line((xxx+7),(j+27),(xxx+60+7),(j+27));

if(tmp.getTrueValue(mc)==false)

{

if((Player1.current && Player1.turn )||(Player2.turn && Player2.current)) {

l.add((xxx+7)+" "+(j+27)+" "+(xxx+60+7)+" "+(j+27));

try {

out.writeUTF((xxx+7)+" "+(j+27)+" "+(xxx+60+7)+" "+(j+27));

String s= in.readUTF();

if(Player1.turn && !"bla".equals(s))

{

Player1.turn = false;

Player2.turn = true;

}

else if(Player2.turn && !"bla".equals(s))

{

Player2.turn = false;

Player1.turn = true;

}

} catch (IOException ex) {

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

}

}

//repaint();

}

}

}

}

}

public void ismouseReleased(MouseEvent e)

{

repaint();

}

});

}

boolean first = true;

myCollections mc = new myCollections(100);

public Draw(Player Player1, Player Player2, List l, DataInputStream in, DataOutputStream out) throws IOException {

super("Точки и Квадраты");

setSize(800, 400);

go = new GameOver(this);

String lines;

jp = new GpanelM(Player1, Player2, l, go, in, out, mc);

setDefaultCloseOperation(EXIT\_ON\_CLOSE);

Component add2;

add2 = add(jp);

jp.setVisible(true);

detectMouse(Player1,Player2,l,in,out,jp);

}

boolean needsLine(int x,int y)

{

int xx = x- 20 ;

int yy = y -20 ;

for(ii=-15;ii<=0;ii++)

if((((xx+ii)%60 == 0)|| ((yy+ii)%60==0)) && (!((xx%60 ==0) && (yy%60==0))) && (xx+ii)>=0 && (xx+ii) <=320 && (yy+ii)>=0 && (yy+ii) <=320)return true;

return false;

}

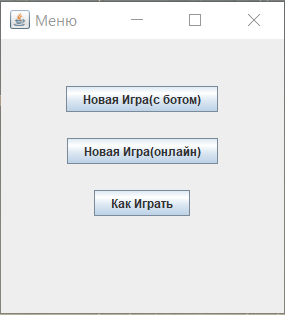
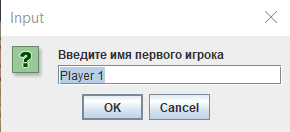
void close(){

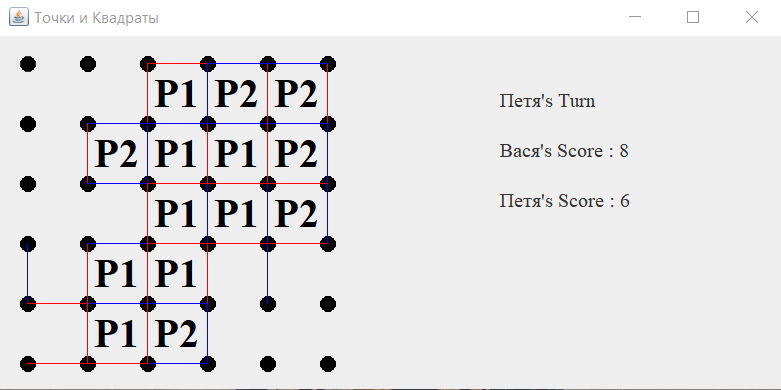
setVisible(false);

}

}

**Результат работы программы:**





**Вывод:** освоили приёмы разработки оконных клиент-серверных приложений на Java с использованием сокетов.