Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

**Лабораторная работа №12**

По дисциплине «СПП»

за 6-й семестр

Выполнил:

студент 3 курса

группы ПО-5

Харкевич Д.А.

Проверил:

Крощенко А.А.

Брест, 2022

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

**Задание:**

2) Простейший многопользовательский чат. Простой чат с возможностью подключения до 5 пользователей. Все пользователи подключаются к серверу, задача сервера – отображение сообщений конкретного пользователя (приват) или общего чата.

**Код прогаммы:**

ClientGUI.java

package Chat;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

public class ClientGUI extends JFrame implements ActionListener {

private static final long serialVersionUID = 1L;

private JLabel label; // For username/enter message

private JTextField textField;

private JTextField tfServer, tfPort;

private JButton login, logout, whoIsIn;

private JTextArea textArea;

private boolean connected;

private Client client;

private int defaultPort;

private String defaultHost;

ClientGUI(String host, int port) { // Receive a socket number

super("Chat Client");

defaultPort = port;

defaultHost = host;

JPanel nordPanel = new JPanel(new GridLayout(3,1));

JPanel serverAndPort = new JPanel(new GridLayout(1,5, 1, 3));

tfServer = new JTextField(host);

tfPort = new JTextField("" + port);

tfPort.setHorizontalAlignment(SwingConstants.RIGHT);

serverAndPort.add(new JLabel("Address: "));

serverAndPort.add(tfServer);

serverAndPort.add(new JLabel("Port: "));

serverAndPort.add(tfPort);

serverAndPort.add(new JLabel(""));

nordPanel.add(serverAndPort);

label = new JLabel("Enter your username ", SwingConstants.CENTER);

nordPanel.add(label);

textField = new JTextField("Guest");

textField.setBackground(Color.WHITE);

nordPanel.add(textField);

add(nordPanel, BorderLayout.NORTH);

textArea = new JTextArea("Welcome to the chat room\n", 70, 70);

JPanel centralPanel = new JPanel(new GridLayout(1,1));

centralPanel.add(new JScrollPane(textArea));

textArea.setEditable(false);

add(centralPanel, BorderLayout.CENTER);

login = new JButton("Login");

login.addActionListener(this);

logout = new JButton("Logout");

logout.addActionListener(this);

logout.setEnabled(false);

whoIsIn = new JButton("Who is in");

whoIsIn.addActionListener(this);

whoIsIn.setEnabled(false);

JPanel downPanel = new JPanel();

downPanel.add(login);

downPanel.add(logout);

downPanel.add(whoIsIn);

add(downPanel, BorderLayout.SOUTH);

setDefaultCloseOperation(EXIT\_ON\_CLOSE);

setSize(400, 400);

setVisible(true);

textField.requestFocus();

}

void append(String str) { // Appending the text in the TextArea

textArea.append(str);

textArea.setCaretPosition(textArea.getText().length() - 1);

}

void connectionFailed() {

login.setEnabled(true);

logout.setEnabled(false);

whoIsIn.setEnabled(false);

label.setText("Enter your username");

textField.setText("Guest");

tfPort.setText("" + defaultPort);tfServer.setText(defaultHost);

tfServer.setEditable(false);

tfPort.setEditable(false);

textField.removeActionListener(this);

connected = false;

}

public void actionPerformed(ActionEvent event) {

Object tempObject = event.getSource();

if (tempObject == logout) {

connectionFailed();

client.sendMessage(new Message(Message.LOGOUT, ""));

return;

}

if (tempObject == whoIsIn) {

client.sendMessage(new Message(Message.WHOISIN, ""));

return;

}

if (connected) { // Come from the JTextField

client.sendMessage(new Message(Message.MESSAGE, textField.getText()));

textField.setText("");

return;

}

if (tempObject == login) {

String username = textField.getText().trim(); // A connection request

if (username.length() == 0)

return;

String server = tfServer.getText().trim();

if (server.length() == 0)

return;

String portNumber = tfPort.getText().trim();

if (portNumber.length() == 0)

return;

int port = 0;

try {

port = Integer.parseInt(portNumber);

}

catch(Exception exception) {

return;

}

client = new Client(server, port, username, this);

if (!client.start())

return;

textField.setText("");label.setText("Enter your message below");

connected = true;

login.setEnabled(false);

logout.setEnabled(true);

whoIsIn.setEnabled(true);

tfServer.setEditable(false);

tfPort.setEditable(false);

textField.addActionListener(this); // Action listener for when the user enter a message

}

}

public static void main(String[] args) {

new ClientGUI("localhost", 1200);

} // Start the whole server

@SuppressWarnings("unchecked")

private void initComponents() {

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

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

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

.addGap(0, 400, Short.MAX\_VALUE)

);

layout.setVerticalGroup(

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

.addGap(0, 300, Short.MAX\_VALUE)

);

pack();

}

}

**ServerGUI.java**

package Chat;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

public class ServerGUI extends JFrame implements ActionListener, WindowListener {

private static final long serialVersionUID = 1L;

private JButton stopStart;

private JTextArea chat, event;

private JTextField tPortNumber;

private Server server;

ServerGUI(int port) {

super("NordChat (Server)");

server = null;

JPanel nord = new JPanel();

nord.add(new JLabel("Port: "));

tPortNumber = new JTextField(" " + port);

nord.add(tPortNumber);

stopStart = new JButton("Start");

stopStart.addActionListener(this);

nord.add(stopStart);

add(nord, BorderLayout.NORTH); // The event and chat room

JPanel center = new JPanel(new GridLayout(2, 1));

chat = new JTextArea(90, 90);

chat.setEditable(false);

appendRoom("Chating room\n");

center.add(new JScrollPane(chat));

event = new JTextArea(90, 90);

event.setEditable(false);

appendEvent("Events log.\n");

center.add(new JScrollPane(event));

add(center);

addWindowListener(this);

setSize(400, 500);

setVisible(true);

}

void appendRoom(String str) { // Append a message to the two JTextAreas

chat.append(str);

chat.setCaretPosition(chat.getText().length() - 1);

}

void appendEvent(String str) {

event.append(str);

event.setCaretPosition(chat.getText().length() - 1);

}

public void actionPerformed(ActionEvent event) {

if (server != null) {

server.stop();

server = null;

tPortNumber.setEditable(true);

stopStart.setText("Starting");

return;

}

int port = 0;

try {

port = Integer.parseInt(tPortNumber.getText().trim());

}

catch (Exception exception) {

appendEvent("Invalid port number");

return;

}

server = new Server(port, this);

new ServerRunning().start();

stopStart.setText("Stop");

tPortNumber.setEditable(false);

}

public static void main(String[] arg) { // Starting the server

new ServerGUI(1200);

}

public void windowClosing(WindowEvent event) { // Closing by Windows event

if (server != null) {

try {

server.stop();

}

catch (Exception eClose) {}

server = null;

}

dispose();

System.exit(0);

}

public void windowClosed(WindowEvent event) {}

public void windowOpened(WindowEvent event) {}

public void windowIconified(WindowEvent event) {}

public void windowDeiconified(WindowEvent event) {}

public void windowActivated(WindowEvent event) {}

public void windowDeactivated(WindowEvent event) {}

class ServerRunning extends Thread {

public void run() {

server.start(); // Should executing until if fails

stopStart.setText("Start");

tPortNumber.setEditable(true);

appendEvent("Server is stopped\n");

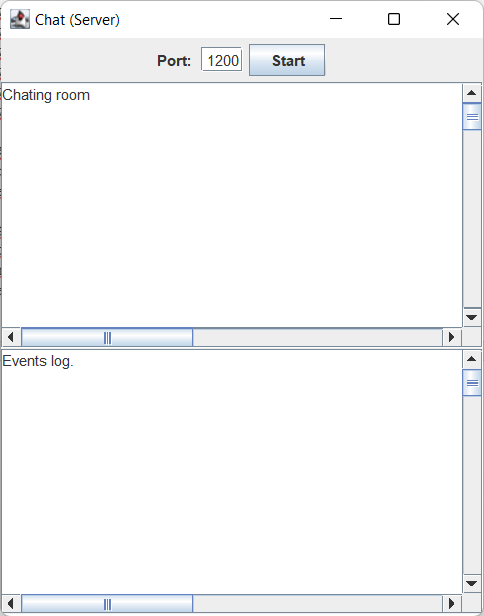
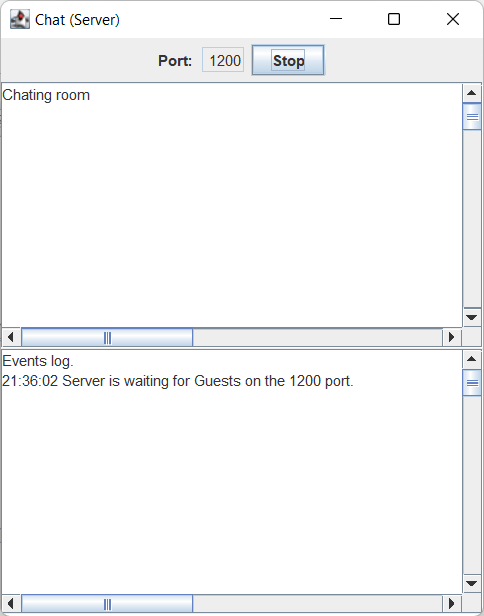
server = null;

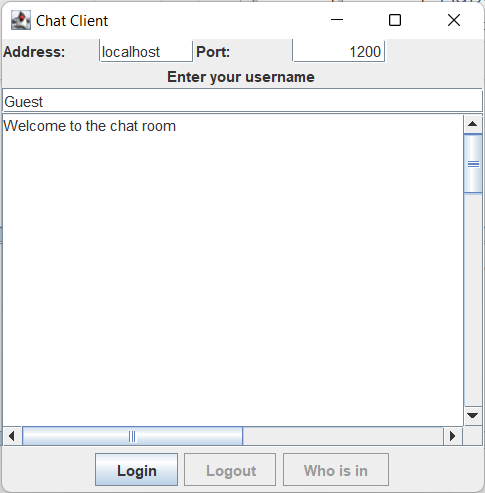
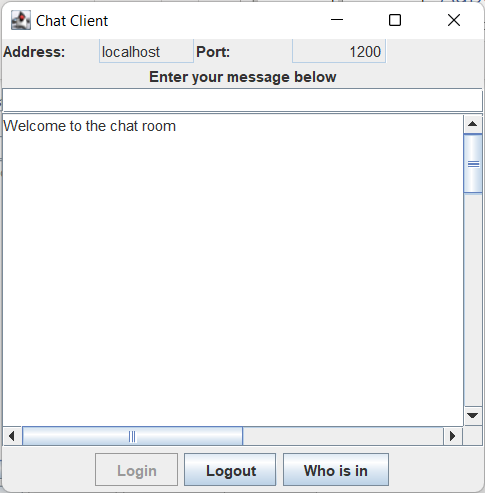
}

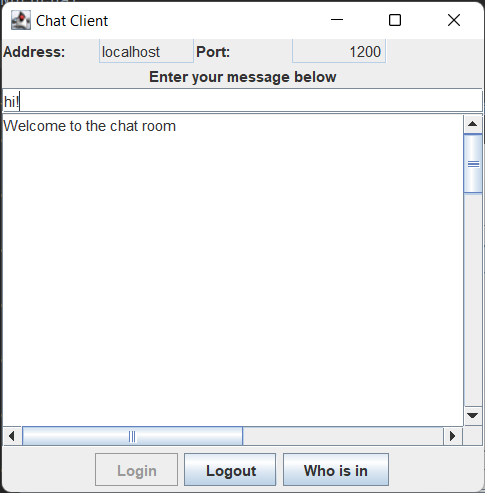
}

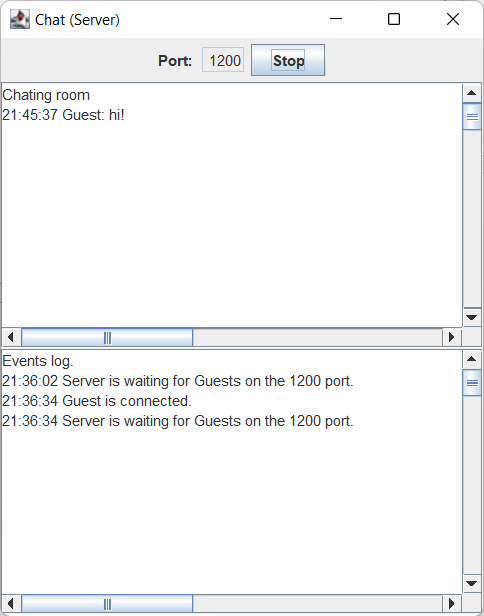
}

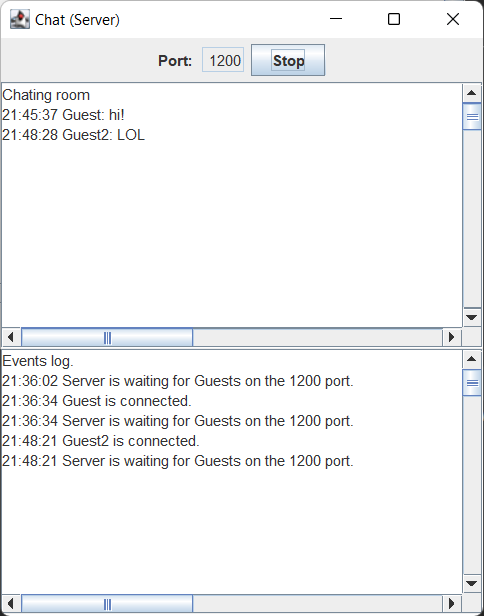
**Результаты:**

**** ****

**** ****

****

****

****

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