import java.io.IOException;

import java.io.ObjectInputStream;

import java.io.ObjectOutputStream;

import java.net.InetAddress;

import java.net.ServerSocket;

import java.net.Socket;

public class Server {

private static final int PORT = 12345;

private static Boolean firstClient = true;

private static String lastMessage;

public static void main(String[] args) {

try {

InetAddress address = InetAddress.getLocalHost();

System.out.println("IP address: " + address.getHostName());

System.out.println("creating server socket");

ServerSocket serverSocket = new ServerSocket(PORT);

System.out.println("waiting for connection");

Socket socket = serverSocket.accept();

System.out.println("accepted connection");

ObjectOutputStream outputStream = new ObjectOutputStream(socket.getOutputStream());

outputStream.flush();

ObjectInputStream inputStream = new ObjectInputStream(socket.getInputStream());

boolean running = true;

while (running) {

outputStream.writeUTF("Enter a message:");

outputStream.flush();

String message = inputStream.readUTF();

if (message.equalsIgnoreCase("shutdown")) {

running = false;

System.out.println("\n!shutdown detected!\n");

}

outputStream.writeUTF(message);

outputStream.flush();

if (firstClient){

outputStream.writeUTF("The last user entered: You're the first to connect!\n");

outputStream.flush();

firstClient = false;

}

else{

outputStream.writeUTF("The last user entered: " + lastMessage + "\n");

outputStream.flush();

}

lastMessage = message;

}

inputStream.close();

outputStream.close();

socket.close();

System.out.println("shutting down");

} catch (IOException ex) {

ex.printStackTrace();

}

}

}

import java.awt.BorderLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.EOFException;

import java.io.IOException;

import java.io.ObjectInputStream;

import java.io.ObjectOutputStream;

import java.net.InetAddress;

import java.net.Socket;

import javax.swing.JFrame;

import javax.swing.JScrollPane;

import javax.swing.JTextArea;

import javax.swing.JTextField;

public class Client extends JFrame {

private static final int PORT = 12345;

private JTextArea \_textArea;

private JTextField \_textField;

private ObjectOutputStream \_outputStream;

private ObjectInputStream \_inputStream;

public Client() {

super("Chat Client version 0.212");

setLayout(new BorderLayout());

\_textArea = new JTextArea(12, 40);

\_textArea.setEditable(false);

add(new JScrollPane(\_textArea), BorderLayout.CENTER);

\_textField = new JTextField(40);

\_textField.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent ev) {

sendMessage(\_textField.getText());

\_textField.setText("");

}

});

add(\_textField, BorderLayout.SOUTH);

setDefaultCloseOperation(DISPOSE\_ON\_CLOSE);

pack();

setVisible(true);

try {

InetAddress address = InetAddress.getLocalHost();

Socket socket = new Socket(address, PORT);

System.out.println("connected!");

\_outputStream = new ObjectOutputStream(socket.getOutputStream());

\_outputStream.flush();

\_inputStream = new ObjectInputStream(socket.getInputStream());

boolean running = true;

while (running) {

try {

String message = \_inputStream.readUTF();

\_textArea.append(message + "\n");

}

catch (EOFException e){

running = false;

System.out.println("\n!shutdown detected!\n");

System.out.println("server is dead");

}

// TODO: Scroll to bottom of scroll pane

}

\_inputStream.close();

\_outputStream.close();

socket.close();

System.out.println("shutting down");

} catch (IOException ex) {

ex.printStackTrace();

}

}

private void sendMessage(String message) {

if (\_outputStream == null) {

return;

}

try {

\_outputStream.writeUTF(message);

\_outputStream.flush();

} catch (IOException ex) {

System.err.println("Unable to send message" + ex);

}

}

public static void main(String[] args) {

new Client();

}

}