**《Java语言高级特性》实验 七**

**网络编程**

班级 学号 姓名

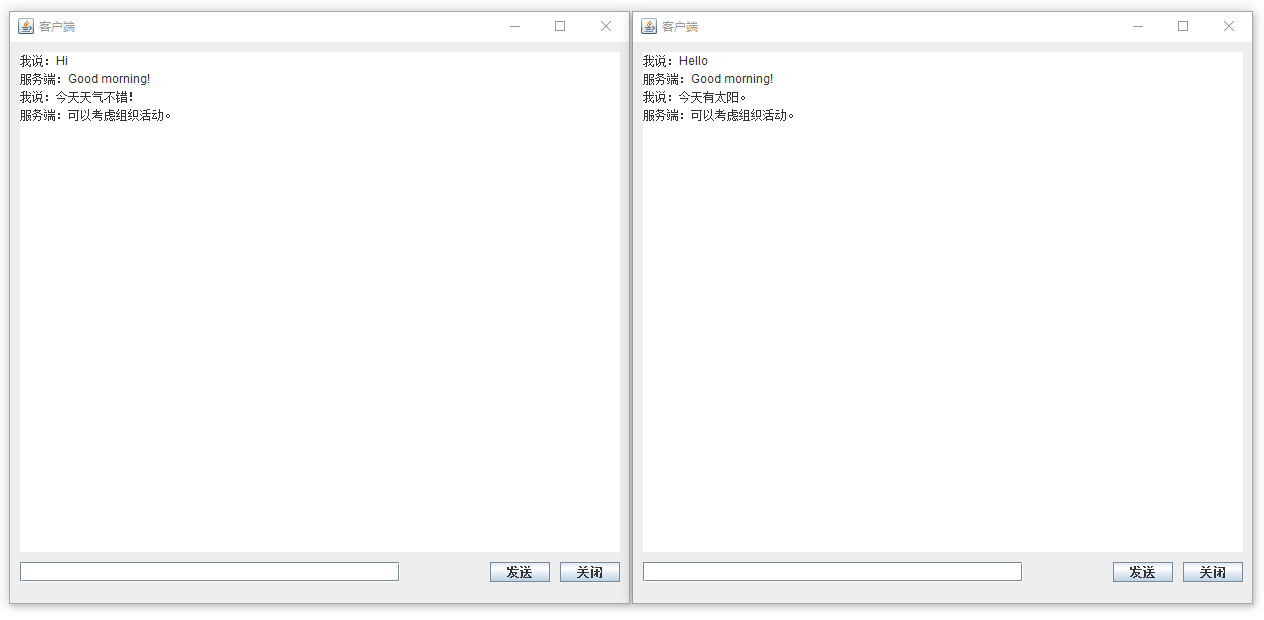
**[实验任务1**]**基于TCP的Socket网络编程**

**[任务介绍**]

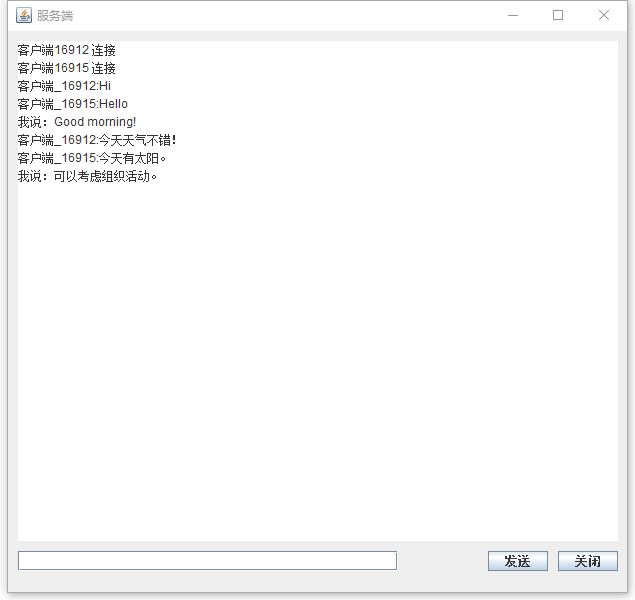
1. **任务描述**

利用TCP协议进行Socket编程，实现如下图所示的聊天器。

1. **运行结果**



**图1客户端运行效果**



**图2服务器端运行效果**

**[任务目标]**

* 学会分析“基于TCP的Socket网络编程”任务结合Socket技术实现思路；
* 根据思路独立完成“基于TCP的Socket网络编程”任务结合网络编程技术的源代码编写、编译和运行。

**[实现思路]**

1. **先创建基于TCP的Socket服务**
2. **命令行输入ip地址以及端口号**
3. **创建一个线程监听Socket链接**
4. **客户端连接成功显示聊天窗口**

**[实现代码及运行结果]**

**代码:**

1. **创建Socket服务**

**package TCP;**

**import javax.swing.\*;**

**import java.awt.event.ActionEvent;**

**import java.io.DataOutputStream;**

**import java.io.IOException;**

**import java.net.ServerSocket;**

**import java.net.Socket;**

**import java.util.Map;**

**import java.util.concurrent.ConcurrentHashMap;**

**import java.util.concurrent.ConcurrentMap;**

**public class TcpSvrChatWin extends ChatWin{**

**public static void main(String[] args) {**

**TcpSvrChatWin win = new TcpSvrChatWin();**

**win.setTitle("服务器");**

**win.setVisible(true);**

**win.setBounds(100,100,700,700);**

**}**

**ServerSocket server=null;**

**ServerSocket server2=null;**

**//ServerThread thread;**

**//Socket you=null;**

**Map<Integer,Socket> clients = new ConcurrentHashMap();**

**public TcpSvrChatWin(){**

**try{**

**server=new ServerSocket(1111);**

**server2=new ServerSocket(2111);**

**}**

**catch(IOException e1) {**

**//ServerSocket对象不能重复创建，除非更换端口号**

**System.out.println("正在监听");**

**}**

**try{**

**System.out.println("等待客户呼叫");**

**Socket you=server.accept();**

**clients.put(you.getPort(),you);**

**System.out.println("客户的地址:"+you.getInetAddress());**

**if(you!=null) {**

**new ServerThread(you,this).start(); //为每个客户启动一个专门的线程**

**}**

**}**

**catch (IOException e) {**

**System.out.println("正在等待客户");**

**}**

**try{**

**System.out.println("等待客户呼叫");**

**Socket you=server2.accept();**

**clients.put(you.getPort(),you);**

**System.out.println("客户的地址:"+you.getInetAddress());**

**if(you!=null) {**

**new ServerThread(you,this).start(); //为每个客户启动一个专门的线程**

**}**

**}**

**catch (IOException e) {**

**System.out.println("正在等待客户");**

**}**

**}**

**@Override**

**public void actionPerformed(ActionEvent arg0){**

**String s =arg0.getActionCommand();**

**Object obj = arg0.getSource();**

**if (obj instanceof JButton){**

**JButton btn = (JButton) obj;**

**if ("发送".equals(s)){**

**String txt = txtSnd.getText();**

**for(Map.Entry<Integer,Socket> e:clients.entrySet())**

**{**

**try {**

**DataOutputStream w = new DataOutputStream(e.getValue().getOutputStream());**

**w.writeUTF(txt);**

**w.flush();**

**} catch (IOException ex) {**

**ex.printStackTrace();**

**}**

**}**

**txtChat.setText(txtChat.getText() + "我说：" + txt+ "\n");**

**txtSnd.setText("");**

**}**

**else if(btn.equals(btnClose)){**

**}**

**}**

**}**

**}**

1. **客户端**

**package TCP;**

**import javax.swing.\*;**

**import java.awt.event.ActionEvent;**

**import java.io.DataInputStream;**

**import java.io.DataOutputStream;**

**import java.io.IOException;**

**import java.net.InetAddress;**

**import java.net.InetSocketAddress;**

**import java.net.Socket;**

**import java.util.InputMismatchException;**

**import java.util.Map;**

**import java.util.Scanner;**

**public class TcpClientCharWin extends ChatWin{**

**DataOutputStream out=null;**

**public static void main(String[] args) {**

**TcpClientCharWin win = new TcpClientCharWin();**

**win.setTitle("客户端1");**

**win.setVisible(true);**

**win.setBounds(100,100,700,700);**

**}**

**@Override**

**public void actionPerformed(ActionEvent arg0){**

**String s =arg0.getActionCommand();**

**Object obj = arg0.getSource();**

**if (obj instanceof JButton){**

**JButton btn = (JButton) obj;**

**if ("发送".equals(s)){**

**String txt = txtSnd.getText();**

**//System.out.println(txt);**

**txtChat.setText(txtChat.getText()+"我说："+txt+"\n");**

**try {**

**out.writeUTF(txt);**

**out.flush();**

**} catch (IOException e) {**

**e.printStackTrace();**

**}**

**txtSnd.setText("");**

**}**

**else if(btn.equals(btnClose)){**

**}**

**}**

**}**

**public TcpClientCharWin(){**

**Scanner scanner = new Scanner(System.in);**

**Socket mysocket=null;**

**DataInputStream in=null;**

**Thread readData ;**

**Read read=null;**

**try{**

**mysocket=new Socket();**

**System.out.print("输入服务器的IP:");**

**String IP = scanner.nextLine();**

**System.out.print("输入端口号:");**

**int port = scanner.nextInt();**

**if(mysocket.isConnected()){}**

**else{**

**InetAddress address=InetAddress.getByName(IP);**

**InetSocketAddress socketAddress=new InetSocketAddress(address,port);**

**mysocket.connect(socketAddress);**

**in =new DataInputStream(mysocket.getInputStream());**

**out = new DataOutputStream(mysocket.getOutputStream());**

**read = new Read(in,this);**

**readData = new Thread(read);**

**read.setDataInputStream(in);**

**readData.start();**

**}**

**}**

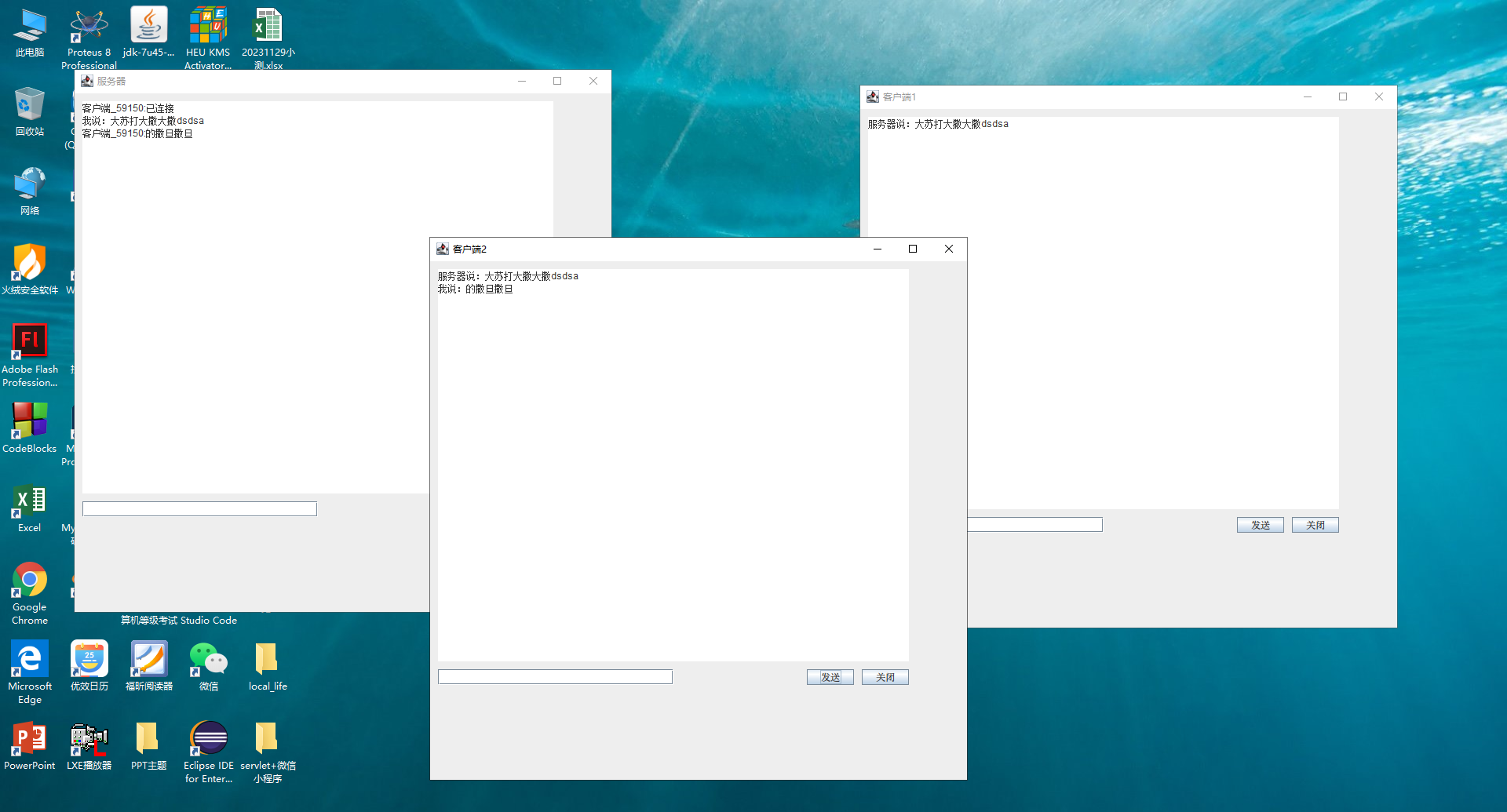
**catch(Exception e) {**

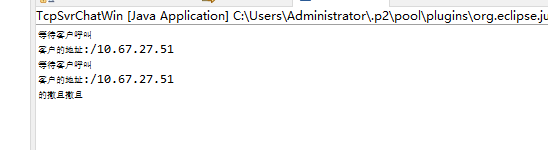
**System.out.println("服务器已断开"+e);**

**}**

**}**

**}**

****

****

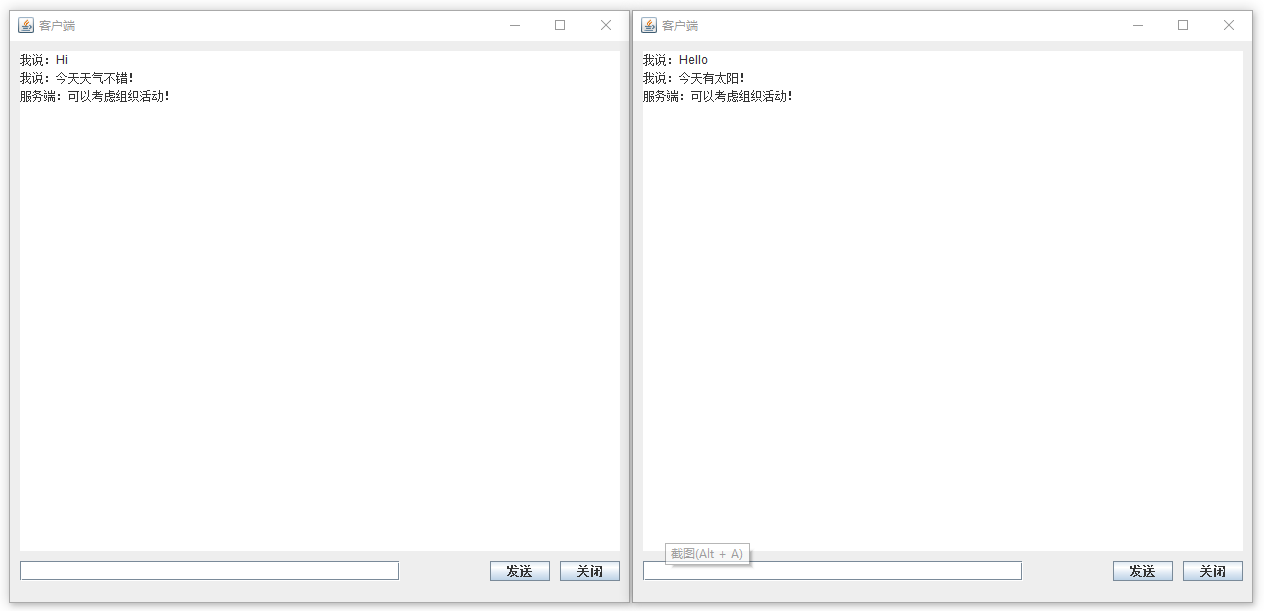
**[实验任务2**]**基于UDP的Socket网络编程**

**[任务介绍**]

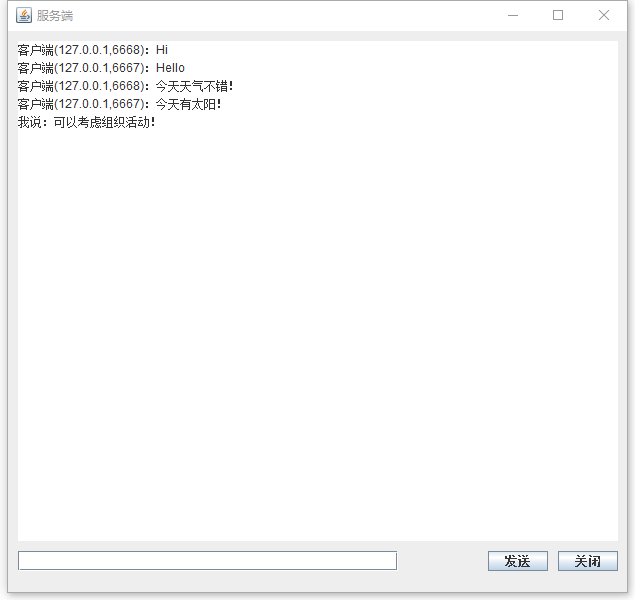
1. **任务描述**

利用UDP协议进行Socket编程，实现如下图所示的聊天器。

1. **运行结果**



**图3客户端运行效果**



**图4服务器端运行效果**

**[任务目标]**

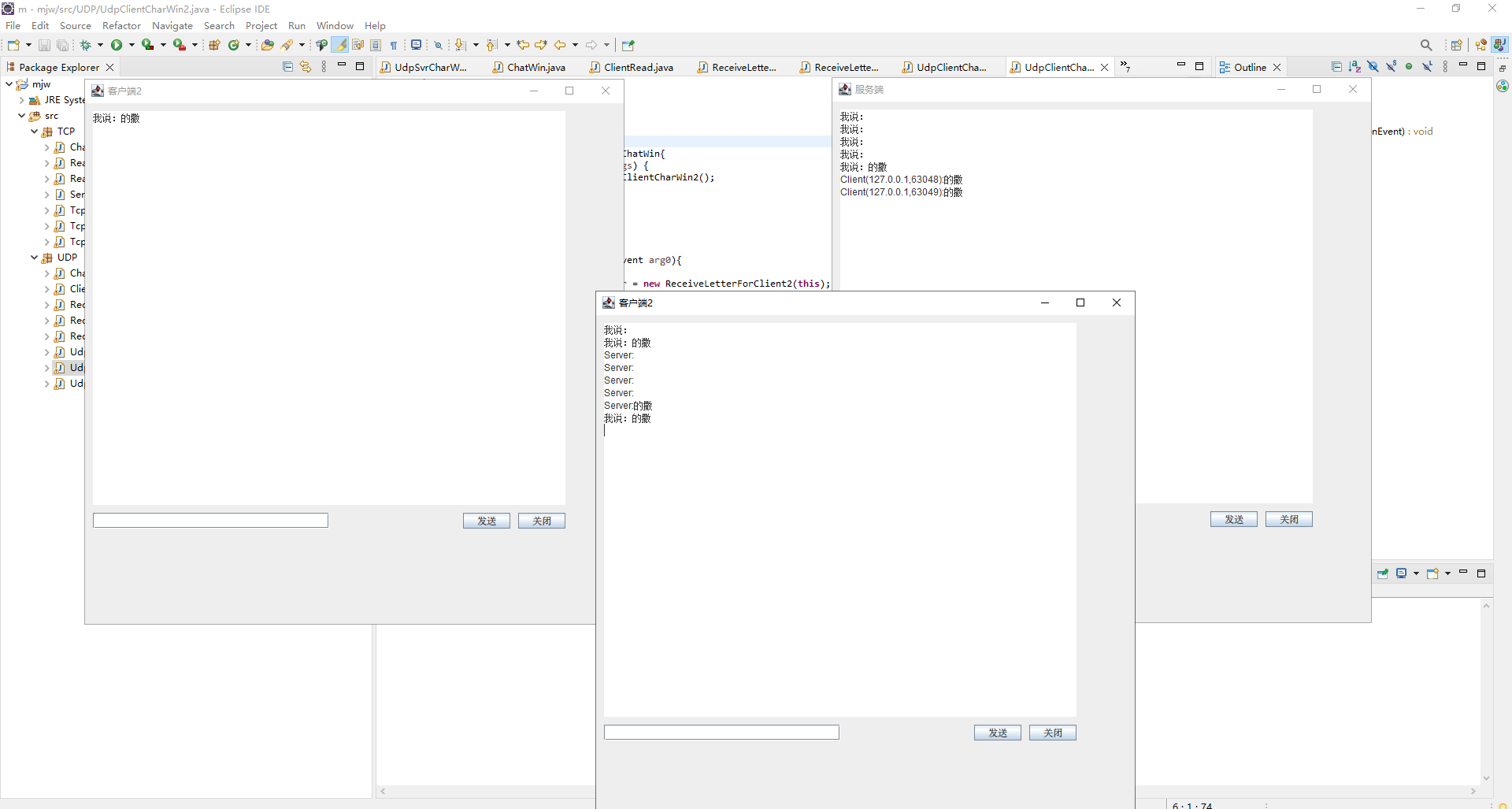
* 学会分析“基于UDP的Socket网络编程”任务结合Socket技术实现思路；
* 根据思路独立完成“基于UDP的Socket网络编程”任务结合网络编程技术的源代码编写、编译和运行。

**[实现思路]**

**1、**

**[实现代码及运行结果]**

**[总结或感悟]（**对运行结果所作的分析以及本次调试程序所取得的经验。如果程序未能通过，分析其原因。**）**

****