Lecture 03
Custom Networking
Part 2

Sockets

Chapter 13- Object Streams and RMI

(The java.net package)

Objectives

- What is a socket?
- How to develop applications using TCP protocol such as chatter?

Contents

- Java Sockets
- The java.net.Socket and java.net.ServerSocket classes
- Sockets: How do they work?
- Socket: How do we code?
- Demonstration

1- Java Sockets

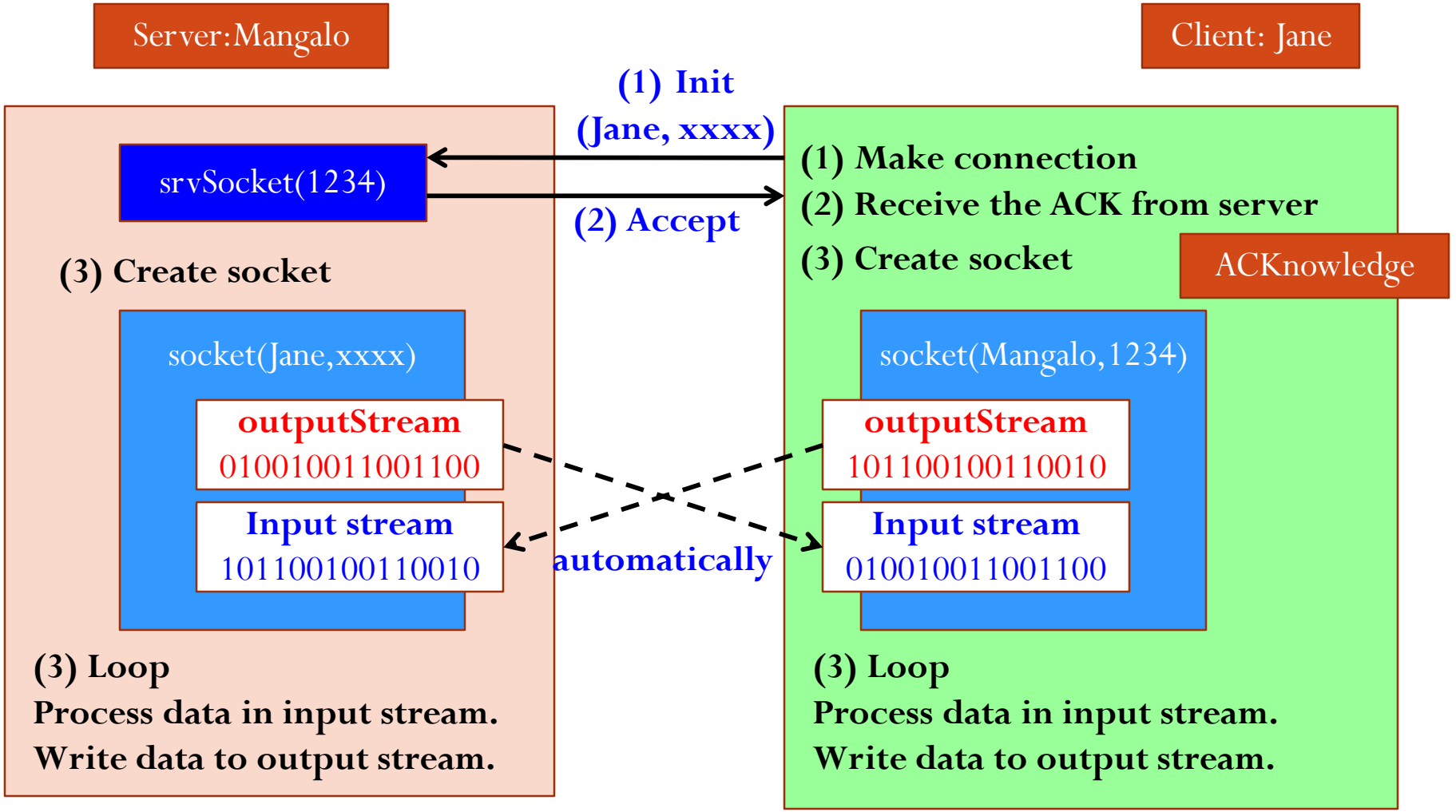
- A *socket* is one endpoint of a two-way communication link between two programs running on the network. A socket is bound to a port number so that the TCP layer can identify the application that data is destined to be sent.
- Main members:

  Socket = IP + Port + IO Streams + Methods

2- Socket and ServerSocket Classes

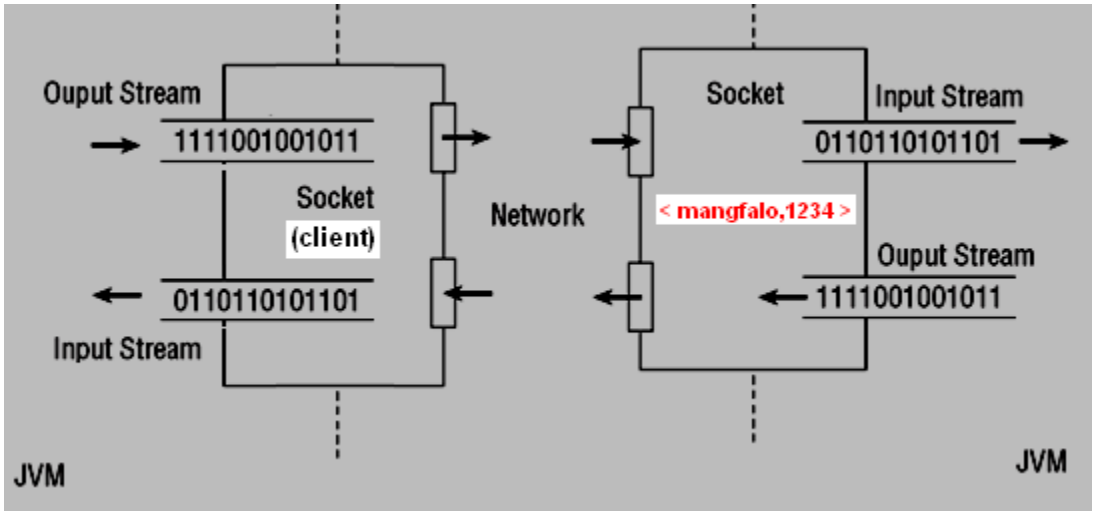
- The java.net.Socket class implements one side of a two-way connection between your Java program and another program on the network
- The java.net.ServerSocket implements server sockets. A server socket waits for requests to come in over the network. It performs some operation based on that request, and then possibly returns a result to the requester.

3- Sockets: How do they works?



4- Sockets: How do we code?

Server Program
Name: mangfalo
or IP



Client
Program

ServerSocket ss= new ServerSocket(1234);
Socket clientSoctket=ss.accept();

Socket srvSocket= new Socket("mangfalo",1234);

```
// Receive data
bf= new BufferedReader( new InputStreamReader(socket.getInputStream()));
aString= bf.readLine();
// Send data
os= new DataOutputStream (socket.getOutputStream());
os.writeBytes( aString); os.write(13); os.write(10); os.flush();
```

5- Java TCP Socket Demo.

Problem

- A manager wants to communicate with his/her staffs. Each communication is distinct from others.
- Write Java programs which allow this manager and his/her staffs carrying out their communications.

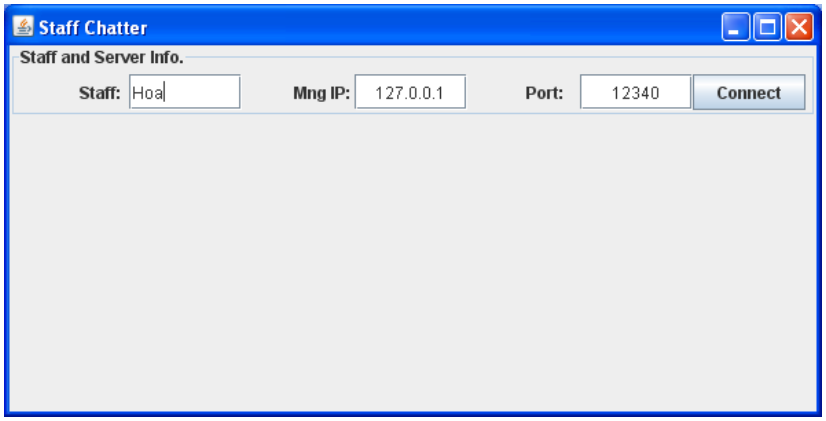
➔Each client program is used by a staff.

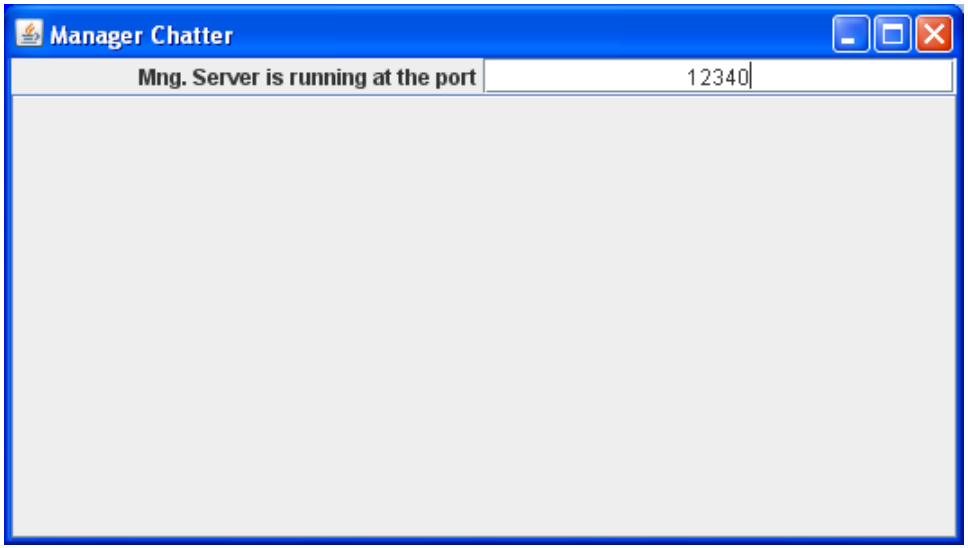
➔The program used by the manager has some threads, each thread supports a communication between the manager and a staff.

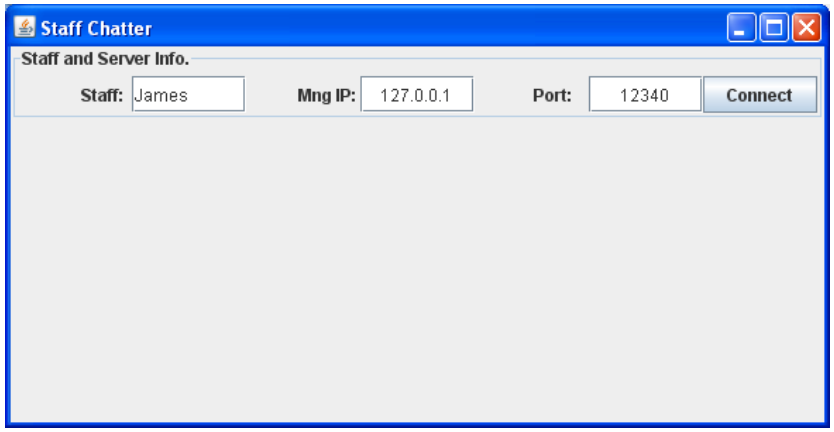
➔Manager site is specified by IP and the port 12340.

Java TCP Socket Demo.: GUIs

2 staffs and manager – when no connection is established

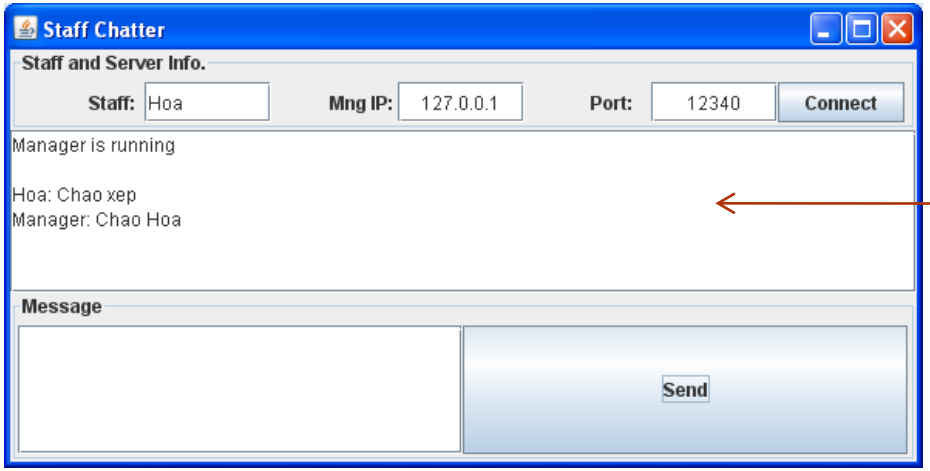


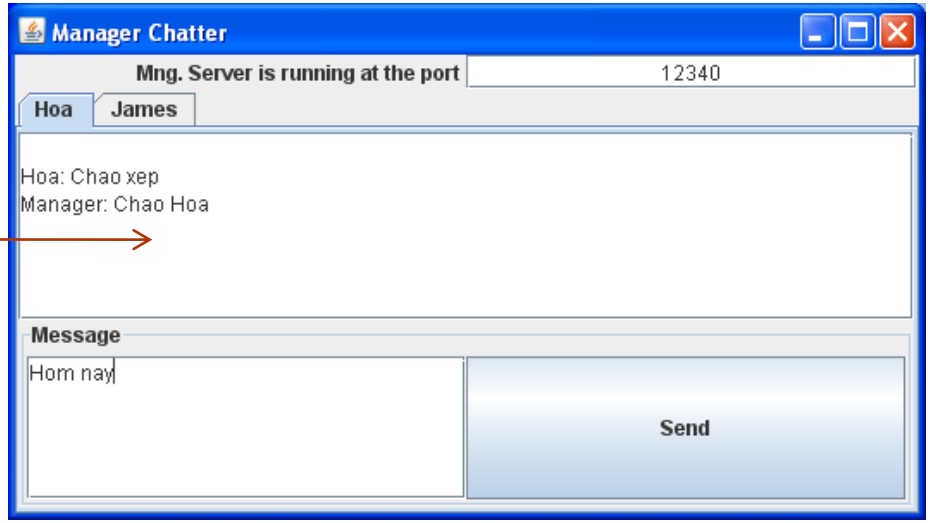


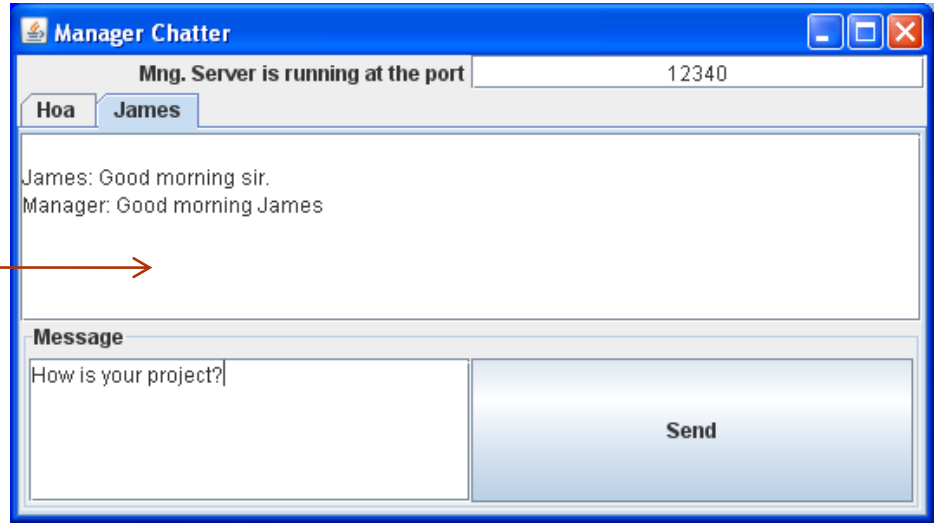


Java TCP Socket Demo.: GUIs

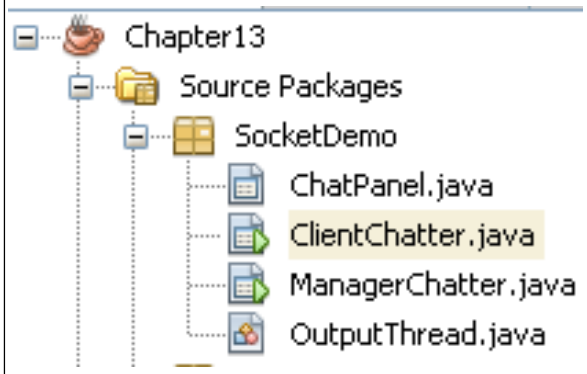
2 staffs and manager – when 2 connections is established

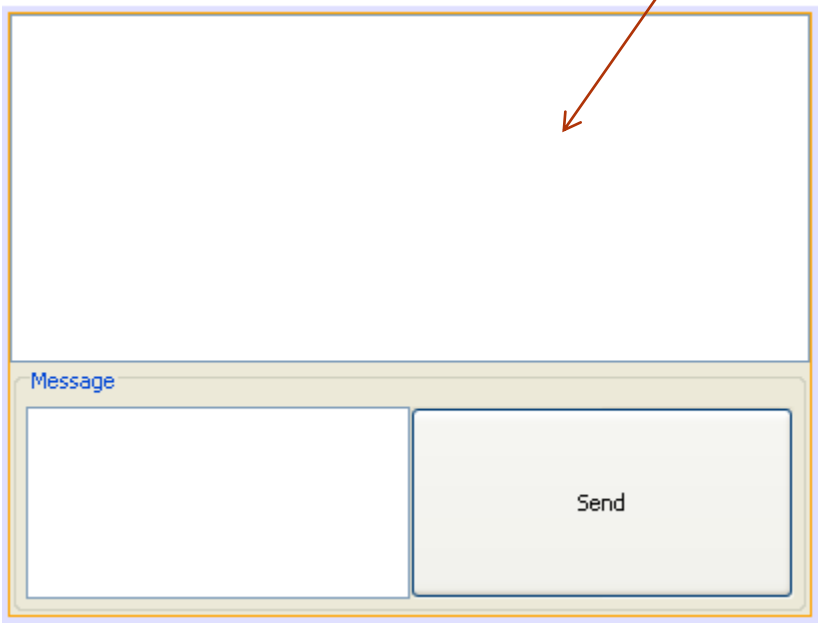






Java TCP Socket Demo.: Project architecture





ChatPanel: Panel for chatting, it is used in client and server.

ClientChatter: GUI client program for staffs

ManagerChatter: GUI Server program for manager

OutputThread: a thread helps presenting received data (1 time/second)

TCP Socket Demo...

OutputThread.java

```
1  /* Thread presents received messages automatically */
2  package SocketDemo;
3  import javax.swing.JTextArea;
4  import java.io.BufferedReader;
5  import java.io.InputStreamReader;
6  import java.net.Socket;
7  import javax.swing.JOptionPane;
8  public class OutputThread extends Thread {
9      Socket socket;      // socket is joining to the communication
10     JTextArea txt;  // text-area contains communicated message
11     BufferedReader bf; // in put buffer of the socket
12     String sender;         // sender, a site of the communication
13     String receiver;       // receiver, other site of the communication
14
15     public OutputThread ( Socket s, JTextArea txt, String sender, String receiver){
16         super();
17         this.socket =s; this.txt=txt; this.sender=sender; this.receiver=receiver;
18         try{
19          bf= new BufferedReader( new InputStreamReader(socket.getInputStream()));
20         }
21         catch (Exception e){
22             JOptionPane.showMessageDialog(null, "Network Error!");
23             System.exit(0);
24         }
25     }
```

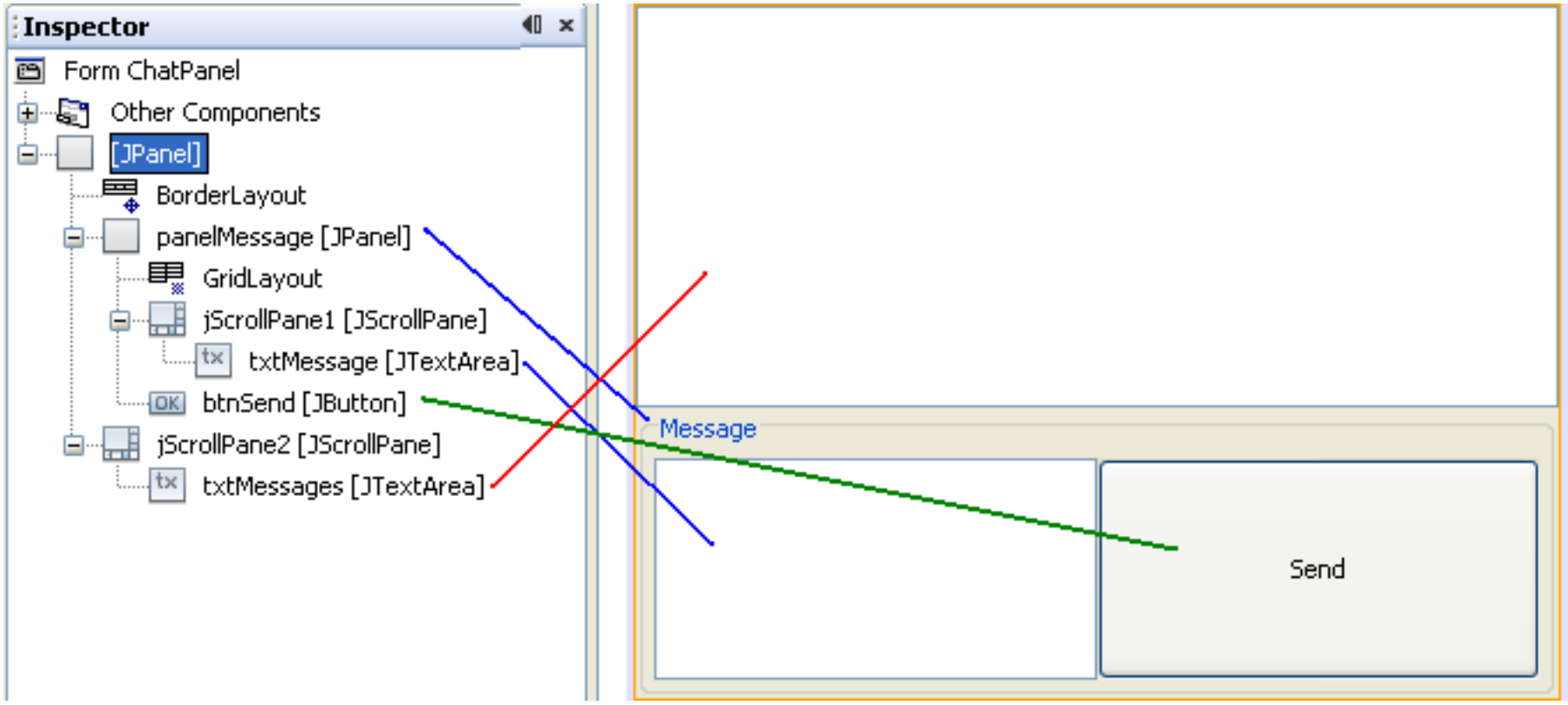
TCP Socket Demo...

OutputThread.java

```
26        // get data from the input stream periodically (1 time/ sec
27        // The time when data comes can nt be known in advance
          public void run()
29         { while (true)
30           try {
31               if (socket!=null) {
32                 String msg=""; // get data from the input stream
33                 if ((msg=bf.readLine())!=null && msg.length()>0)
34                     txt.append("\n" + receiver + ": " + msg);
35               }
                 sleep(1000);
37          }
38            catch (Exception e){}
39          }
40    }
```

TCP Socket Demo...

ChatPanel.java



TCP Socket Demo….

ChatPanel.java

```
public class ChatPanel extends javax.swing.JPanel {

    Socket socket = null;
    BufferedReader bf = null;
    DataOutputStream os = null;
    OutputThread t = null;
    String sender;
    String receiver;

    public ChatPanel(Socket s, String sender, String receiver) {
        initComponents();
        setSize(600, 400);
        txtMessages.setEditable(false);
        socket = s;
        this.sender = sender;
        this.receiver = receiver;
        try {
            os = new DataOutputStream(socket.getOutputStream());
            t = new OutputThread(s, txtMessages, sender, receiver);
            t.start();
        } catch (Exception e) {
        }
    }
```

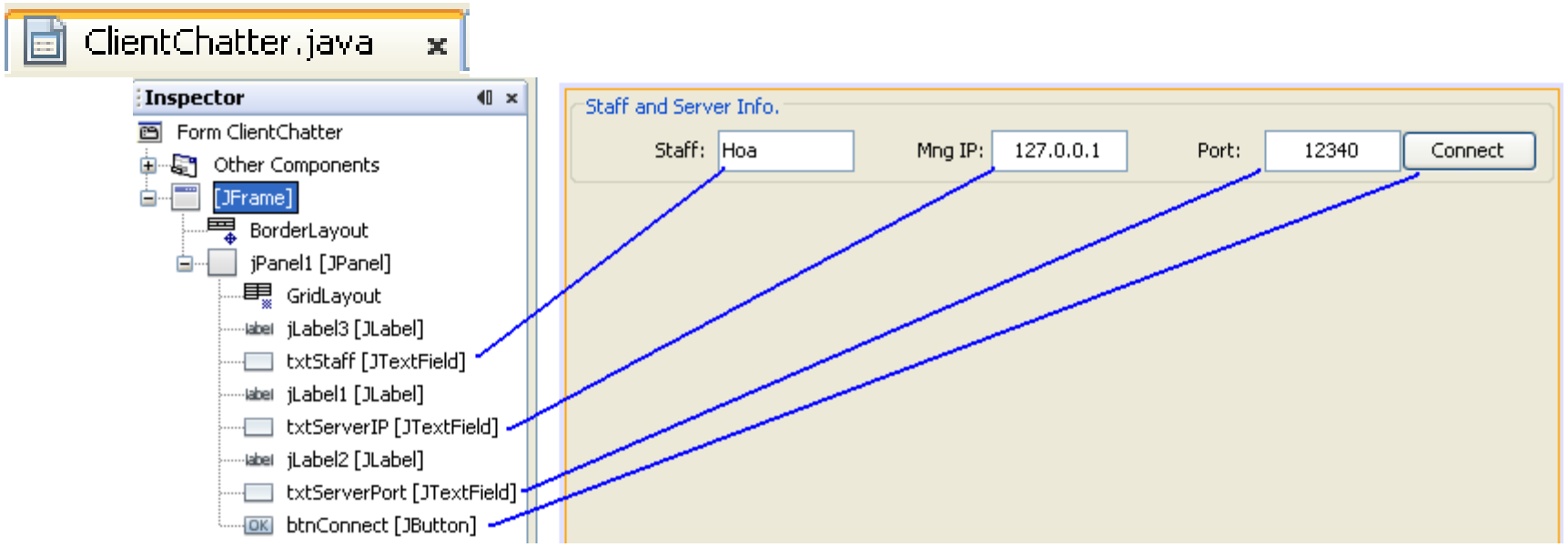
TCP Socket Demo….

ChatPanel.java

```
30      public JTextArea getTxtMessages(){
31          return this.txtMessages;
32      }

78      private void btnSendActionPerformed(java.awt.event.ActionEvent evt) {
79          // TODO add your handling code here:
80          if (txtMessage.getText().trim().length()==0) return;
81          try {
82            os.writeBytes(txtMessage.getText());
83            os.write(13); os.write(10);
84            os.flush();
85            this.txtMessages.append("\n" + sender + ": " + txtMessage.getText());
86            txtMessage.setText("");
87          }
88          catch(Exception e){
89          }
90      }
```

TCP Socket Demo….




```
public class ClientChatter extends javax.swing.JFrame {

    /**
     * Creates new form ClientChatter
     */
    Socket mngSocket = null;
    String mngIP = "";
    int mngPort = 0;
    String staffName = "";
    BufferedReader bf = null;
    DataOutputStream os = null;
```

TCP Socket Demo….

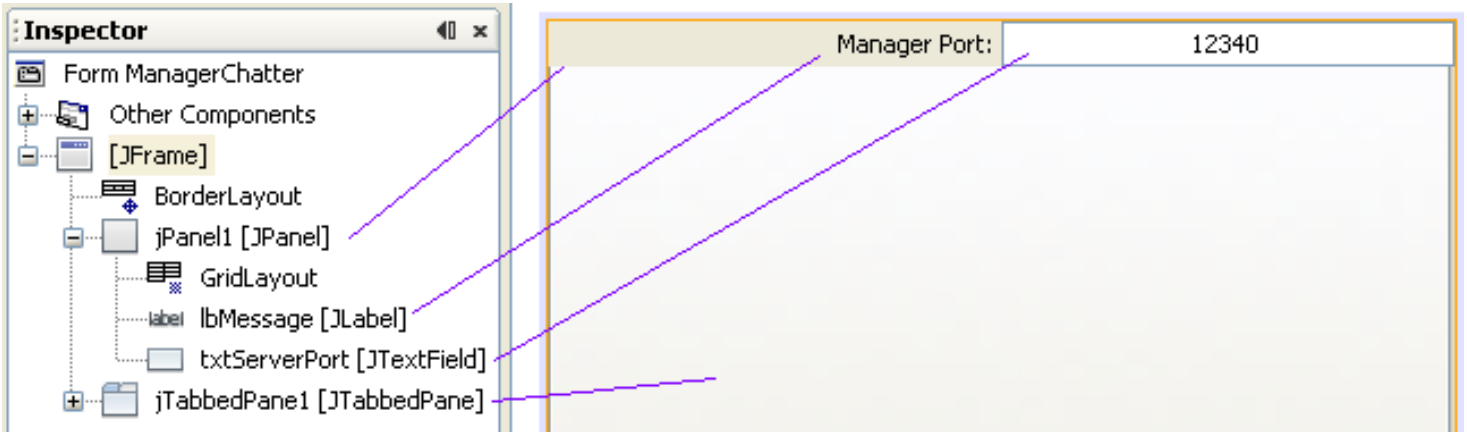
ClientChatter.java

```
private void btnConnectActionPerformed(java.awt.event.ActionEvent evt) {
    mngIP = this.txtServerIP.getText();
    mngPort = Integer.parseInt(this.txtServerPort.getText());
    staffName = this.txtStaff.getText();
    try {
        mngSocket = new Socket(mngIP, mngPort);
        if (mngSocket != null) {
            ChatPanel p = new ChatPanel(mngSocket, staffName, "Manager");
            this.getContentPane().add(p);
            jPanel1.setVisible(false);
            p.getTxtMessages().append("Manager is running\n");
            p.updateUI();

            os = new DataOutputStream(mngSocket.getOutputStream());
            os.writeBytes("Staff: " + staffName);
            os.write(10);
            os.flush();
        }
    } catch (Exception e) {
        JOptionPane.showMessageDialog(this, "Manager is not running");
        System.exit(0);
    }
}
```

TCP Socket Demo….

ManagerChatter.java




```
1   package SocketDemo;
2   /* @author SuTV */
3   import java.net.Socket;
4   import java.net.ServerSocket;
5   import java.io.*;
6   public class ManagerChatter extends javax.swing.JFrame implements Runnable {
7       ServerSocket srvSocket=null;
8       BufferedReader br=null;
9       Thread t; // thread for exploring connections from staffs
```

TCP Socket Demo….

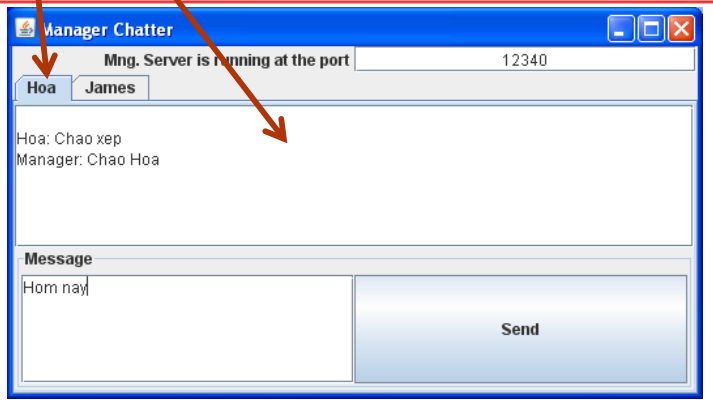
ManagerChatter.java

```
10      /** Creates new form ManagerGUI */
11      public ManagerChatter() {
12          initComponents();
13          this.setSize(600,300);
14          int serverPort=Integer.parseInt(txtServerPort.getText());
15          try {
16             srvSocket= new ServerSocket(serverPort);
17             this.lbMessage.setText("Mng. Server is running at the port ");
18          }
19          catch(Exception e) {
20          }
21          t= new Thread (this);
            t.start();
23      }
```

TCP Socket Demo….

ManagerChatter.java

```
26      public void run(){
27          while (true){
28            try {  // Wait for a client
29                Socket aStaffSocket = srvSocket.accept();
30                if (aStaffSocket!=null) { // If there is a connection
31                    // Get staffname
32                    // When a staff inits a connection, he/she sends his/her name first
33                    br= new BufferedReader (new InputStreamReader(
34                                            aStaffSocket.getInputStream()));
35                    String S= br.readLine();
36                    int pos = S.indexOf(":"); // Fortmat:   Staff:Hoa
37                    String staffName = S.substring(pos+1); // Get name
38                    // Crate a tab for this connection
39                    ChatPanel p= new ChatPanel(aStaffSocket, "Manager",staffName);
40                    jTabbedPane1.add(staffName,p);
41                    p.updateUI();
42                }
43                Thread.sleep(1000);
44            }
45            catch(Exception e) {
46            }
          }
      }
```



Summary

- IP and Port
- TCP, UDP Protocols
- Sockets and Ports
- Client Sockets/ Server Sockets in Java
- How to use TCP sockets

Thank You