

学号：2013458

姓名：刘嗣旸

专业：计算机科学与技术

一、源代码截图

(一) Server.java

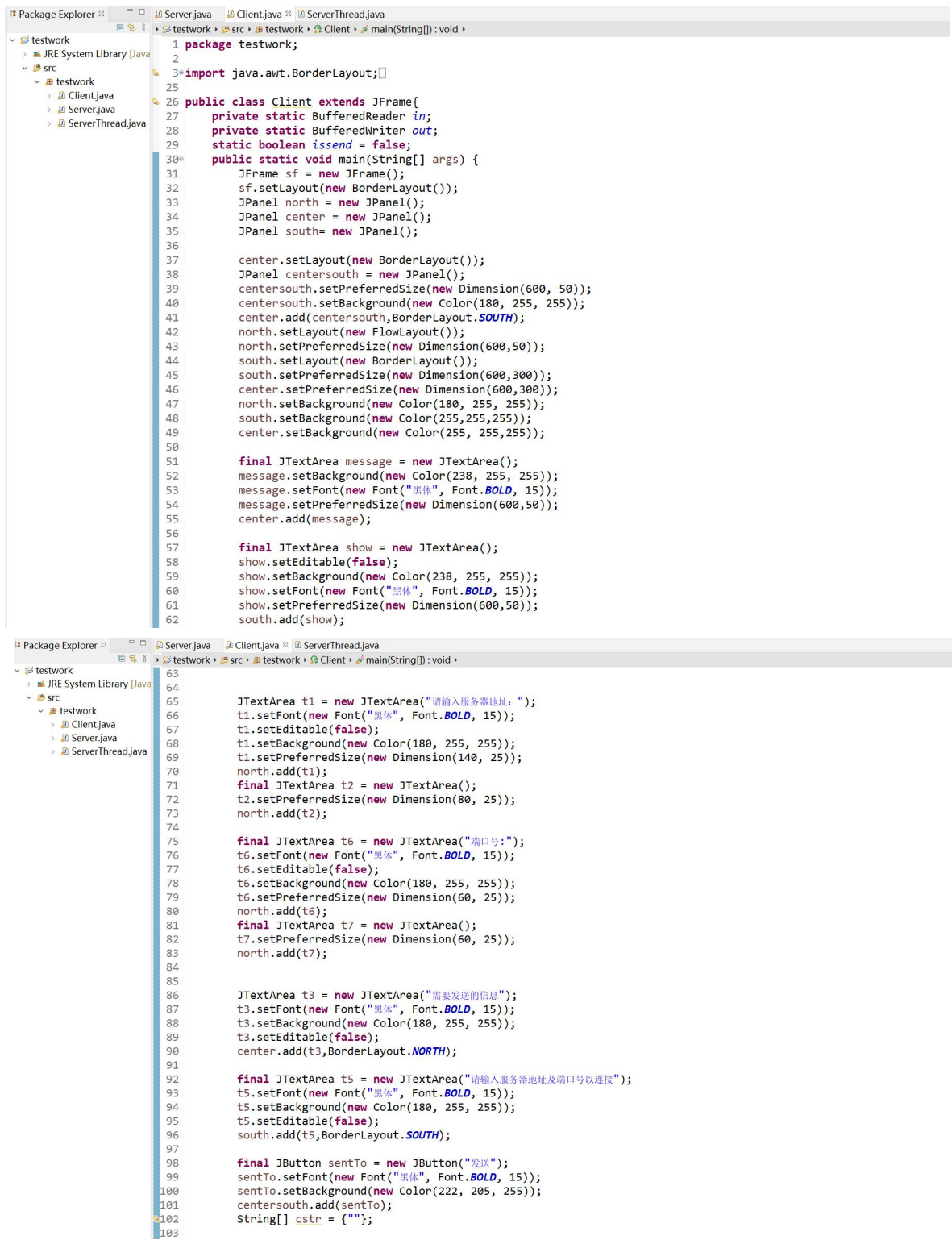
```
Package Explorer
testwork
  JRE System Library [Java]
  src
    testwork
      Client.java
      Server.java
      ServerThread.java

Server.java
1 package testwork;
2
3 import java.awt.BorderLayout;
4
24
25 public class Server extends JFrame{
26     static boolean close = false;
27     static int iclient = 0;
28     static int index = 0;
29     static BufferedReader[] sin = new BufferedReader[1000];
30     static BufferedWriter[] sout = new BufferedWriter[1000];
31     static ServerSocket ss;
32
33     public static void main(String[] args) {
34         JFrame sf = new JFrame();
35         sf.setLayout(new BorderLayout());
36         JPanel north = new JPanel();
37         JPanel center = new JPanel();
38         JPanel south = new JPanel();
39
40         center.setLayout(new BorderLayout());
41         final JPanel centersouth = new JPanel();
42         centersouth.setPreferredSize(new Dimension(600, 50));
43         centersouth.setBackground(new Color(180, 255, 255));
44         center.add(centersouth, BorderLayout.SOUTH);
45         north.setLayout(new FlowLayout());
46         north.setPreferredSize(new Dimension(600, 50));
47         south.setLayout(new BorderLayout());
48         south.setPreferredSize(new Dimension(600, 300));
49         center.setPreferredSize(new Dimension(600, 300));
50         north.setBackground(new Color(180, 255, 255));
51         south.setBackground(new Color(255, 255, 255));
52         center.setBackground(new Color(255, 255, 255));
53
54         final JTextArea message = new JTextArea();
55         message.setBackground(new Color(238, 255, 255));
56         message.setFont(new Font("黑体", Font.BOLD, 15));
57         message.setPreferredSize(new Dimension(600, 50));
58         center.add(message);
59
60         final JTextArea show = new JTextArea();
61         show.setEditable(false);
62
63         show.setBackground(new Color(238, 255, 255));
64         show.setFont(new Font("黑体", Font.BOLD, 15));
65         show.setPreferredSize(new Dimension(600, 50));
66         south.add(show);
67
68         JTextArea t1 = new JTextArea("请输入服务端口号:");
69         t1.setFont(new Font("黑体", Font.BOLD, 15));
70         t1.setEditable(false);
71         t1.setBackground(new Color(180, 255, 255));
72         t1.setPreferredSize(new Dimension(140, 25));
73         final JTextArea t2 = new JTextArea();
74         t2.setPreferredSize(new Dimension(80, 25));
75         north.add(t1);
76         north.add(t2);
77
78         final JButton sentTo = new JButton("发送至");
79         sentTo.setFont(new Font("黑体", Font.BOLD, 15));
80         sentTo.setBackground(new Color(222, 205, 255));
81         centersouth.add(sentTo);
82
83         final JTextArea t5 = new JTextArea("输入端口号启动服务");
84         t5.setBackground(new Color(180, 255, 255));
85         t5.setFont(new Font("黑体", Font.BOLD, 15));
86         t5.setEditable(false);
87         south.add(t5, BorderLayout.SOUTH);
88
89         final JComboBox<String> boxClients = new JComboBox<>();
90         boxClients.setPreferredSize(new Dimension(100, 28));
91         boxClients.setBackground(new Color(222, 205, 255));
92         centersouth.add(boxClients);
93
94         final JButton bstart = new JButton("start");
95         bstart.setBackground(new Color(222, 205, 255));
96         bstart.addActionListener(new ActionListener() {
97             @Override
98             public void actionPerformed(ActionEvent e) {
99                 String b = e.getActionCommand();
100                 if(b.equals("start")) {
101                     close = false;
102                     bstart.setText("stop");
103                     new Thread() {
```

```
Package Explorer :: testwork
JRE System Library [Java]
src
testwork
Client.java
Server.java
ServerThread.java

Server.java
103 public void run() {
104     boxClients.addActionListener(new ActionListener() {
105         @Override
106         public void actionPerformed(ActionEvent e) {
107             JComboBox<String> b = (JComboBox<String>) e.getSource();
108             Integer a = new Integer(b.getSelectedIndex()+1);
109             index = a;
110         }
111     });
112     sentTo.addActionListener(new ActionListener() {
113         @Override
114         public void actionPerformed(ActionEvent e) {
115             String m = message.getText();
116             try {
117                 sout[index].write(m+'\n');
118                 sout[index].flush();
119                 show.append("向客户端"+index+"发送: "+m+"\n");
120                 show.append("成功\n");
121             } catch (IOException e1) {
122                 e1.printStackTrace();
123             }
124         }
125     });
126     try {
127         String text = t2.getText();
128         Integer port = new Integer(text);
129         ss = new ServerSocket(port);
130         show.append("服务器已启动, 端口号: "+text+"\n");
131         t5.setText("启动服务, 端口号为"+text);
132         while(!close) {
133             Socket s = ss.accept();
134             new ServerThread(s) {
135                 public void run() {
136                     try {
137                         int cidex = 0;
138                         iclient++;
139                         cidex = iclient;
140                         show.append("接受客户端"+cidex+"的请求"+"n");
141                         String cstr = new String("Client"+cidex);
142                         boxClients.addItem(cstr);
143                     }
144                 }
145             }
146             sin[cidex] = new BufferedReader(new InputStreamReader(s.getInputStream()));
147             sout[cidex] = new BufferedWriter(new OutputStreamWriter(s.getOutputStream()));
148             while(!close) {
149                 String line;
150                 line = sin[cidex].readLine();
151                 show.append("接收到客户端: "+cidex+" "+line+"\n");
152             }
153             catch (IOException e) {
154                 e.printStackTrace();
155             }
156         }
157     }
158     }
159     }
160     }
161     }
162     }
163     }
164     }
165     }
166     }
167     }
168     }
169     }
170     }
171     }
172     }
173     }
174     }
175     }
176     }
177     }
178     }
179     }
180     }
181     }
182     }
183     }
184     }
185     }
186     }
187     }
188     }
189     }
190     }
191     }
192     }
193     }
194     }
195     }
196     }
197     }
198     }
199     }
200     }
201     }
202     }
203     }
204 }
```

(二) Client.java



```
1 package testwork;
2
3 import java.awt.BorderLayout;
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26 public class Client extends JFrame{
27     private static BufferedReader in;
28     private static BufferedWriter out;
29     static boolean issend = false;
30     public static void main(String[] args) {
31         JFrame sf = new JFrame();
32         sf.setLayout(new BorderLayout());
33         JPanel north = new JPanel();
34         JPanel center = new JPanel();
35         JPanel south= new JPanel();
36
37         center.setLayout(new BorderLayout());
38         JPanel centersouth = new JPanel();
39         centersouth.setPreferredSize(new Dimension(600, 50));
40         centersouth.setBackground(new Color(180, 255, 255));
41         center.add(centersouth, BorderLayout.SOUTH);
42         north.setLayout(new FlowLayout());
43         north.setPreferredSize(new Dimension(600,50));
44         south.setLayout(new BorderLayout());
45         south.setPreferredSize(new Dimension(600,300));
46         center.setPreferredSize(new Dimension(600,300));
47         north.setBackground(new Color(180, 255, 255));
48         south.setBackground(new Color(255,255,255));
49         center.setBackground(new Color(255, 255,255));
50
51         final JTextArea message = new JTextArea();
52         message.setBackground(new Color(238, 255, 255));
53         message.setFont(new Font("黑体", Font.BOLD, 15));
54         message.setPreferredSize(new Dimension(600,50));
55         center.add(message);
56
57         final JTextArea show = new JTextArea();
58         show.setEditable(false);
59         show.setBackground(new Color(238, 255, 255));
60         show.setFont(new Font("黑体", Font.BOLD, 15));
61         show.setPreferredSize(new Dimension(600,50));
62         south.add(show);
63
64
65         JTextArea t1 = new JTextArea("请输入服务器地址: ");
66         t1.setFont(new Font("黑体", Font.BOLD, 15));
67         t1.setEditable(false);
68         t1.setBackground(new Color(180, 255, 255));
69         t1.setPreferredSize(new Dimension(140, 25));
70         north.add(t1);
71         final JTextArea t2 = new JTextArea();
72         t2.setPreferredSize(new Dimension(80, 25));
73         north.add(t2);
74
75         final JTextArea t6 = new JTextArea("端口号:");
76         t6.setFont(new Font("黑体", Font.BOLD, 15));
77         t6.setEditable(false);
78         t6.setBackground(new Color(180, 255, 255));
79         t6.setPreferredSize(new Dimension(60, 25));
80         north.add(t6);
81         final JTextArea t7 = new JTextArea();
82         t7.setPreferredSize(new Dimension(60, 25));
83         north.add(t7);
84
85
86         JTextArea t3 = new JTextArea("需要发送的信息");
87         t3.setFont(new Font("黑体", Font.BOLD, 15));
88         t3.setBackground(new Color(180, 255, 255));
89         t3.setEditable(false);
90         center.add(t3, BorderLayout.NORTH);
91
92         final JTextArea t5 = new JTextArea("请输入服务器地址及端口号以连接");
93         t5.setFont(new Font("黑体", Font.BOLD, 15));
94         t5.setBackground(new Color(180, 255, 255));
95         t5.setEditable(false);
96         south.add(t5, BorderLayout.SOUTH);
97
98         final JButton sentTo = new JButton("发送");
99         sentTo.setFont(new Font("黑体", Font.BOLD, 15));
100        sentTo.setBackground(new Color(222, 205, 255));
101        centersouth.add(sentTo);
102        String[] cstr = {" "};
103
```

```
Package Explorer
testwork
  JRE System Library [Java]
  src
    testwork
      Client.java
      Server.java
      ServerThread.java

ServerThread.java
104 JButton link = new JButton("连接");
105 link.setFont(new Font("黑体", Font.BOLD, 15));
106 link.setPreferredSize(new Dimension(70, 25));
107 link.setBackground(new Color(222, 205, 255));
108 north.add(link);
109 link.addActionListener(new ActionListener() {
110     @Override
111     public void actionPerformed(ActionEvent e) {
112         new Thread() {
113             public void run() {
114                 sentTo.addActionListener(new ActionListener() {
115                     @Override
116                     public void actionPerformed(ActionEvent e) {
117                         String m = message.getText();
118                         try {
119                             out.write(m+'\n');
120                             out.flush();
121                             show.append("向服务器发送: "+m+"\n");
122                             show.append("成功!\n");
123                         } catch (IOException e1) {
124                             e1.printStackTrace();
125                         }
126                     }
127                 });
128             try {
129                 String textport = t7.getText();
130                 String textip = t2.getText();
131                 Integer port = new Integer(textport);
132                 Socket c = new Socket(textip, port);
133                 t5.setText("连接服务器"+textip+": "+port+"成功: ");
134                 show.append("连接服务器"+textip+": "+port+"成功! \n");
135                 in = new BufferedReader(new InputStreamReader(c.getInputStream()));
136                 out = new BufferedWriter(new OutputStreamWriter(c.getOutputStream()));
137                 while(true) {
138                     String line = in.readLine();
139                     show.append("接收到服务器消息: "+line+"\n");
140                 }
141             } catch (IOException e1) {
142                 e1.printStackTrace();
143             }
144         }
145     }.start();
146 }
147 });
148
149 JTextArea t4 = new JTextArea("日志");
150 t4.setFont(new Font("黑体", Font.BOLD, 15));
151 t4.setBackground(new Color(180, 255, 255));
152 t4.setEditable(false);
153 south.add(t4, BorderLayout.NORTH);
154
155 sf.add(north, BorderLayout.NORTH);
156 sf.add(south, BorderLayout.SOUTH);
157 sf.add(center, BorderLayout.CENTER);
158
159 sf.setBounds(300, 100, 750, 800);
160 sf.setDefaultCloseOperation(EXIT_ON_CLOSE);
161 sf.setVisible(true);
162 sf.setTitle("Client");
163 }
164 }
165 }
```

(三) ServerThread.java

```
Package Explorer
testwork
  JRE System Library [Java]
  src
    testwork
      Client.java
      Server.java
      ServerThread.java

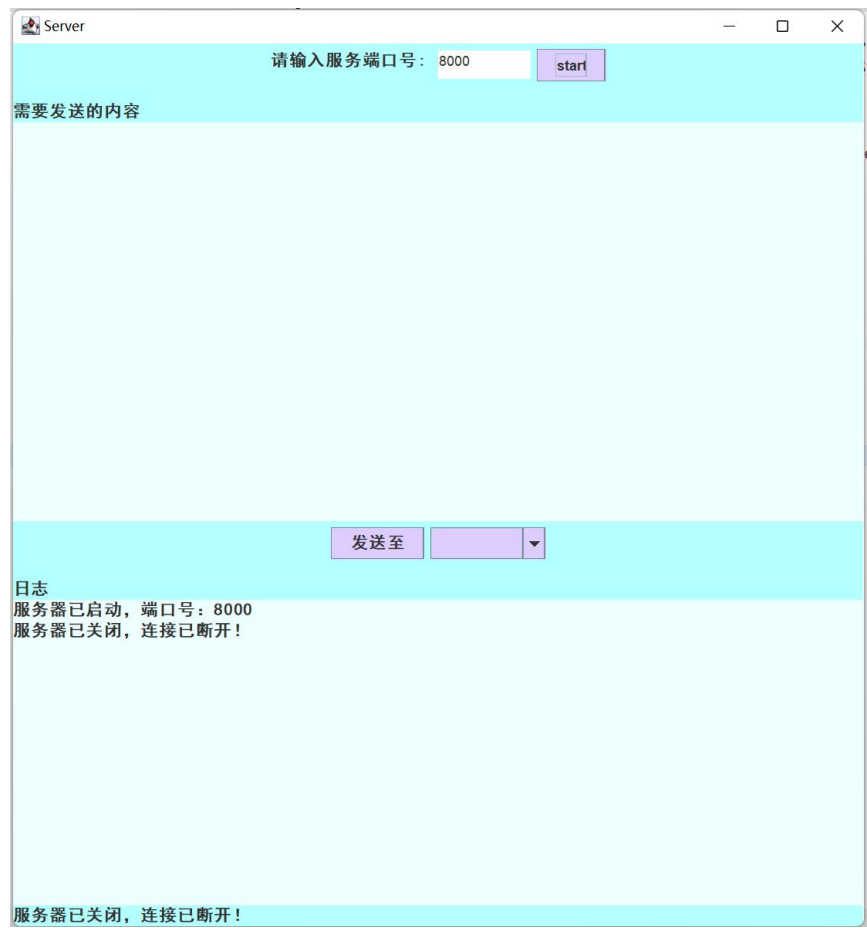
ServerThread.java
1 package testwork;
2
3 import java.io.BufferedReader;
4
5
6
7
8
9
10
11
12
13 public class ServerThread extends Thread{
14     Socket s;
15     public ServerThread(Socket s) {
16         this.s=s;
17     }
18 }
19 }
```


二、运行截图

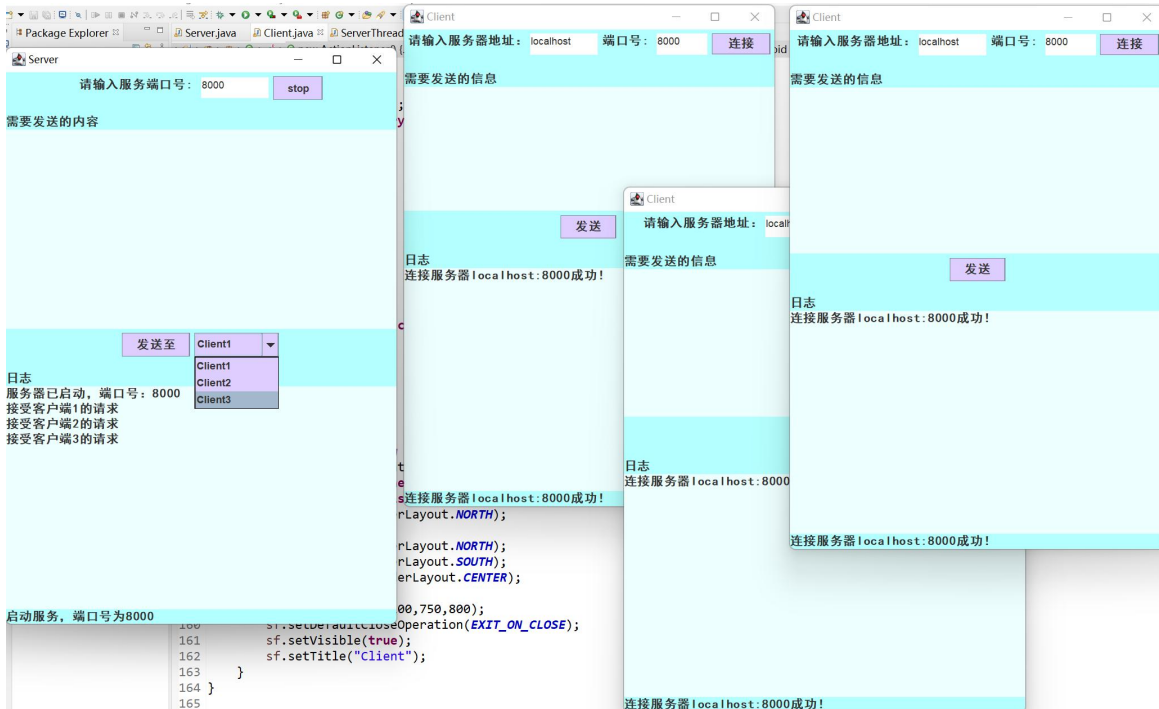
(一) 初始界面（输入 8000 为例，启动服务器）



(二) 关闭服务器



(三) 下面以启动三个客户端为例，输入端口号和 IP 连接，左下角提示成功连接信息。



以上连接了三个客户端，中间的 ComboBox 就会相应增加三个客户选项。

三个客户端分别向服务器发送消息“你好！我是客户 XXX”，服务器和客户端的聊天记录栏分别提示收到的消息内容和发送的消息内容。

点击 ComboBox 可切换不同客户，让服务器与不同的客户分别发送消息，例子向三个客户分别发送了消息“客户 XXX 你好”，日志栏展示了聊天记录。

下图展示了上述聊天交互过程。

