네트워크프로그래밍-7주
Java 소켓 프로그래밍

정인환교수

7주 : Java 소켓 프로그래밍

- Class 기반, Stream 기반
- Encoding 문제
- JavaEchoClient/Server
- Java jar 파일 이용하기
- Java 채팅
- JavaChatClient/Server GUI Version
- WindowBuilder 설치 및 사용법
- JavaChatClient, JavaChatServer
- 7주 과제 설명
  - 과제1 : JavaEchoClient/Server, JavaChatClient/Server, Windows chat_client/server4 실습
  - 과제2 : JavaChatServer 에 Windows chat_server4 기능 모두 구현하기

Class 기반, Stream 기반

- Server : ServerSocket 으로 시작
  - ServerSocket serverSocket = new ServerSocket(30000); // accept 용
  - Socket clientSocket = serverSocket.accept();
- Client : Socket 으로 connect
  - Socket socket = new Socket("127.0.0.1", 30000); // connect 포함
- Stream 기반
  - OutputStream out = socket.getOutputStream();
  - DataOutputStream dos = new DataOutputStream(out);
  - InputStream in = socket.getInputStream();
  - DataInputStream dis = new DataInputStream(in);
- Send/Recv 대신 Read/Write
  - Srting msg; dos.writeUTF(msg); String msg = dis.readUTF();
- Encoding 문제
  - writeUTF()/readUTF()
    - Java 사이에 통신 OK
    - Java / C (ASCII code) 통신 X

Encoding 문제 해결

- String buf;
- buf = sc.nextLine(); // keyboard 입력
- // dos.writeUTF(buf);
- byte[] bb;
- bb = buf.getBytes("euc-kr");// 한글 완성형
- dos.write(bb,0, bb.length); // send()와 동일

- // msg = dis.readUTF();
- byte[] b = new byte[128];
- int ret;
- ret = dis.read(b, 0, 128); // recv()와 동일
- String msg = new String(b, "euc-kr");

JavaEchoClient.java

```java
// JavaEchoClient.java

import java.util.Scanner;

public class JavaEchoClient {
    public static void main(String[] args) {
        try {
            String serverIP = "127.0.0.1"; // 127.0.0.1 & localhost 본인
            String serverPort = "30000";
            if (args.length == 2) {
                serverIP = args[0];
                serverPort = args[1];
            }

            System.out.println("Connecting " + serverIP + " " + serverPort);
            Socket socket = new Socket(serverIP, Integer.parseInt(serverPort));
            System.out.println("Connected.");

            OutputStream out = socket.getOutputStream();
            DataOutputStream dos = new DataOutputStream(out);
            InputStream in = socket.getInputStream();
            DataInputStream dis = new DataInputStream(in);

            Scanner sc = new Scanner(System.in);
            String buf;
            while (true) {
                // keyboard에서 읽고
                System.out.print("Input string : ");
                buf = sc.nextLine();
                //dos.writeUTF(buf);

                byte[] bb;
                bb = buf.getBytes("euc-kr");// 한글 완성형
                dos.write(bb,0, bb.length);

                if (buf.contains("exit"))
                    break;

                // server에서 수신하고
                String msg;
                //msg = dis.readUTF();

                byte[] b = new byte[128];
                int ret;
                ret = dis.read(b, 0, 128);
                msg = new String(b, "euc-kr");
                //msg = msg.trim(); // 앞뒤 blank 제거
                // 화면에 출력
                System.out.println(msg);
            }
            dos.close();
            dis.close();
            socket.close();
```

JavaEchoServer.java

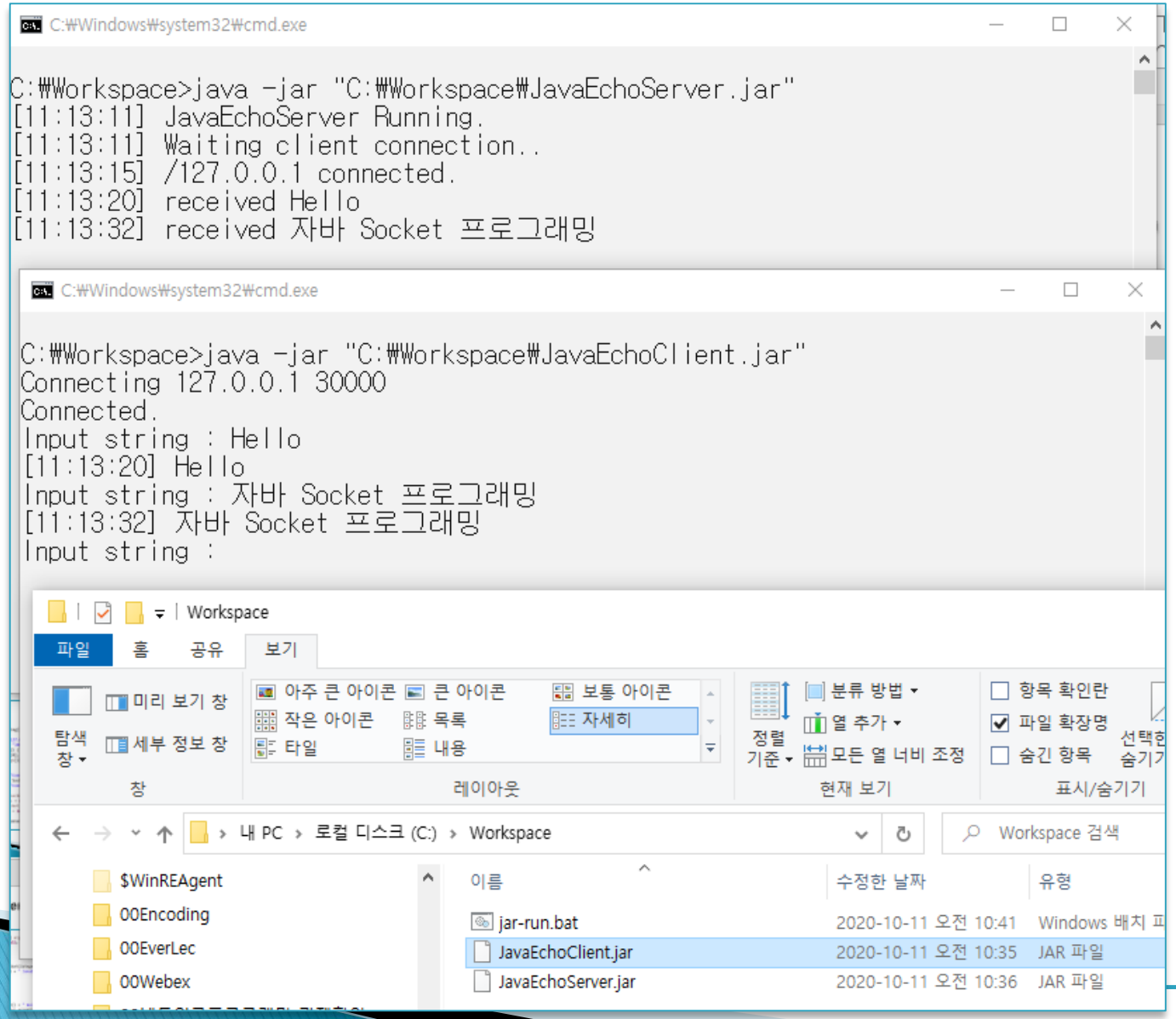
```
13 public class JavaEchoServer {
14   public static void main(String[] args) {
15     ServerSocket serverSocket = null;
16     String serverPort = "30000";
17     if (args.length==1)
18         serverPort = args[0];
19     try {
20       serverSocket = new ServerSocket(Integer.parseInt(serverPort));
21       System.out.println(getTime() + " JavaEchoServer Running.");
22     } catch (IOException e) {
23       e.printStackTrace();
24     } // try - catch
25
26     while (true) {
27       try {
28         System.out.println(getTime() + " Waiting client connection..");
29         Socket socket = serverSocket.accept();
30         System.out.println(getTime() + " " + socket.getInetAddress()
31                            + " connected.");
32
33         OutputStream out = socket.getOutputStream();
34         DataOutputStream dos = new DataOutputStream(out);
35         InputStream in = socket.getInputStream();
36         DataInputStream dis = new DataInputStream(in);
```

```
37         while (true) {
38             String msg;
39             // client 로부터 읽고
40             //msg  = dis.readUTF();
41             //System.out.println(getTime() + " received " +  msg);
42
43             byte[] b = new byte[128];
44             int ret;
45             ret = dis.read(b, 0, 128);
46             msg = new String(b, "euc-kr");
47             msg = msg.trim(); // 앞뒤 blank 제거
48             System.out.println(getTime() + " received " +  ret + " " + msg);
49
50             if (msg.contains("exit"))
51                 break;
52             // client 에게 시간 붙여서 전송
53             msg = getTime() + " " + msg;
54             //dos.writeUTF(msg);
55
56             // 128 byte로 고정시키기 위함
57             msg = String.format("%-128s", msg);
58             byte[] bb;
59             bb = msg.getBytes("euc-kr"); // 한글 완성형 코드 사용
60             //System.out.println("sending = " + msg.length());
61             dos.write(bb,0, msg.length());
62
63         }
64         dis.close();
65         dos.close();
66         socket.close();
```

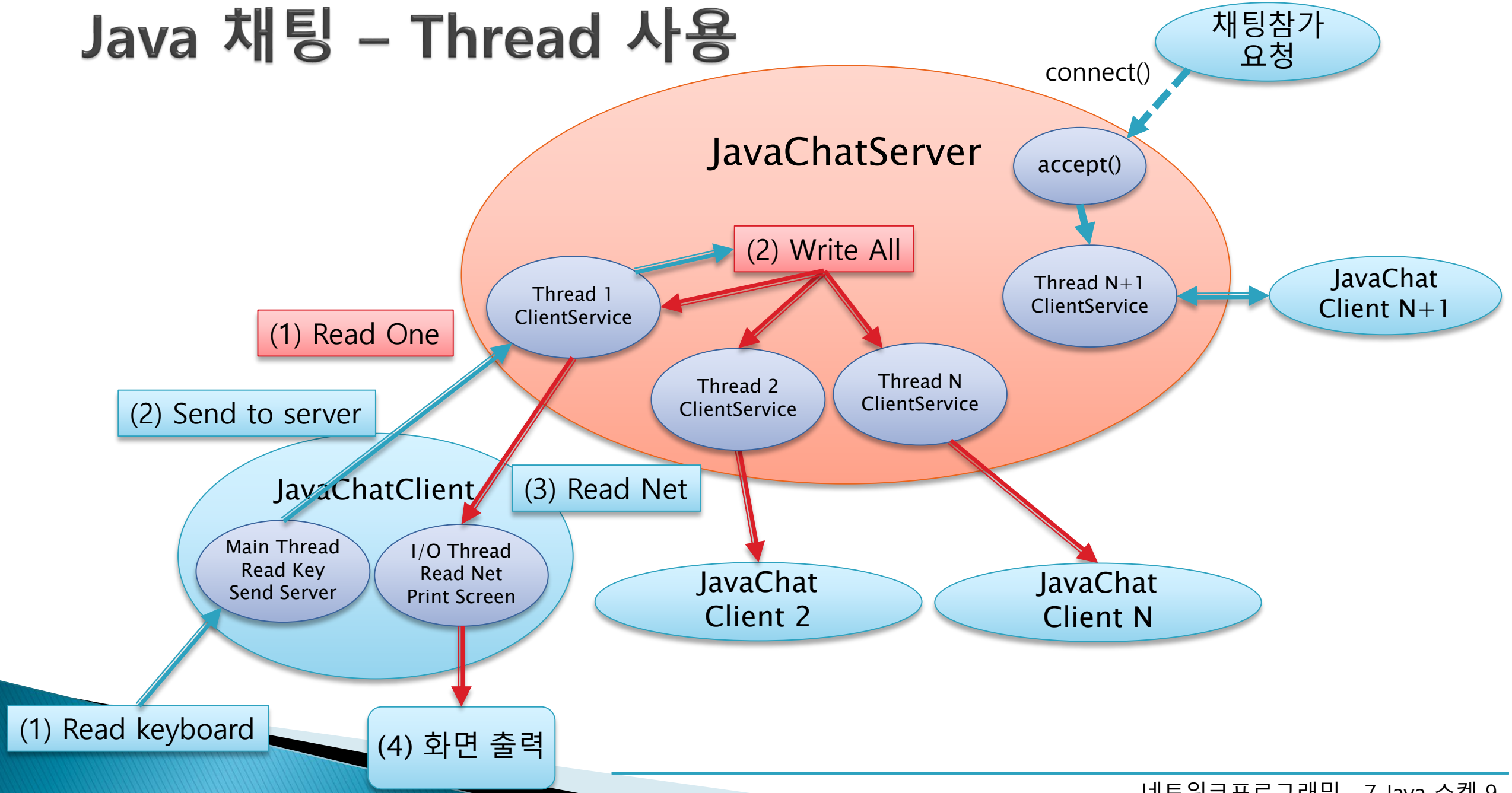
Java 실행파일 jar 이용하기

- Eclipse 에서 Run 하는 경우 불편 > cmd 창에서 실행으로 변경
- 실행 가능한 jar 파일 생성하기
  - File > export > Java > Runnable JAR file
  - c:\Workspace\JavaEchoServer.jar 저장
- jar 파일 실행하기
  - c:\Workspace> java –jar JavaEchoServer.jar
- 탐색기에서 jar 자동 실행하기
  - jar-run.bat 생성
    - java –jar %1
  - jar파일 우측 버튼 > 연결 프로그램 > 추가 앱 > PC에서 다른앱
  - jar-run.nat 선택
  - jar 파일 더블클릭 > cmd 창에서 실행

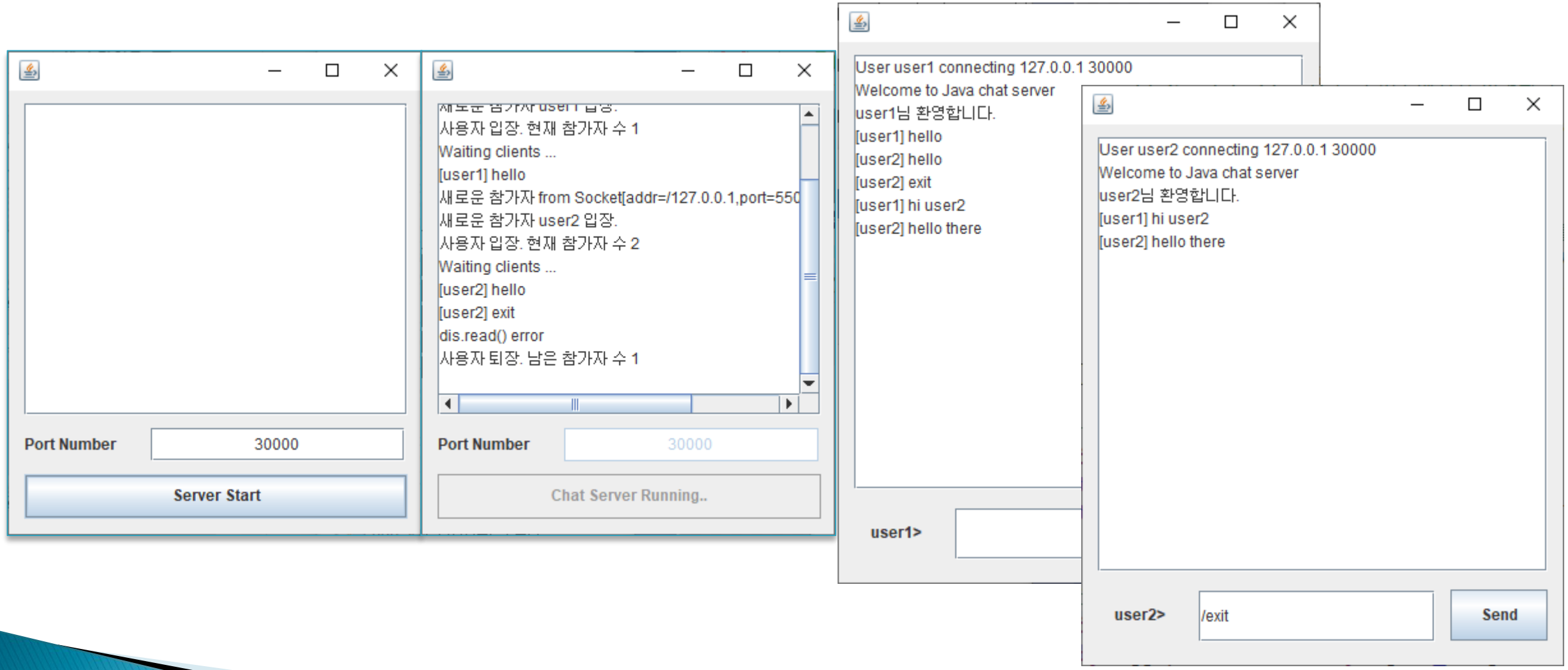
Java jar 파일 직접 실행하기



Java 채팅 – Thread 사용

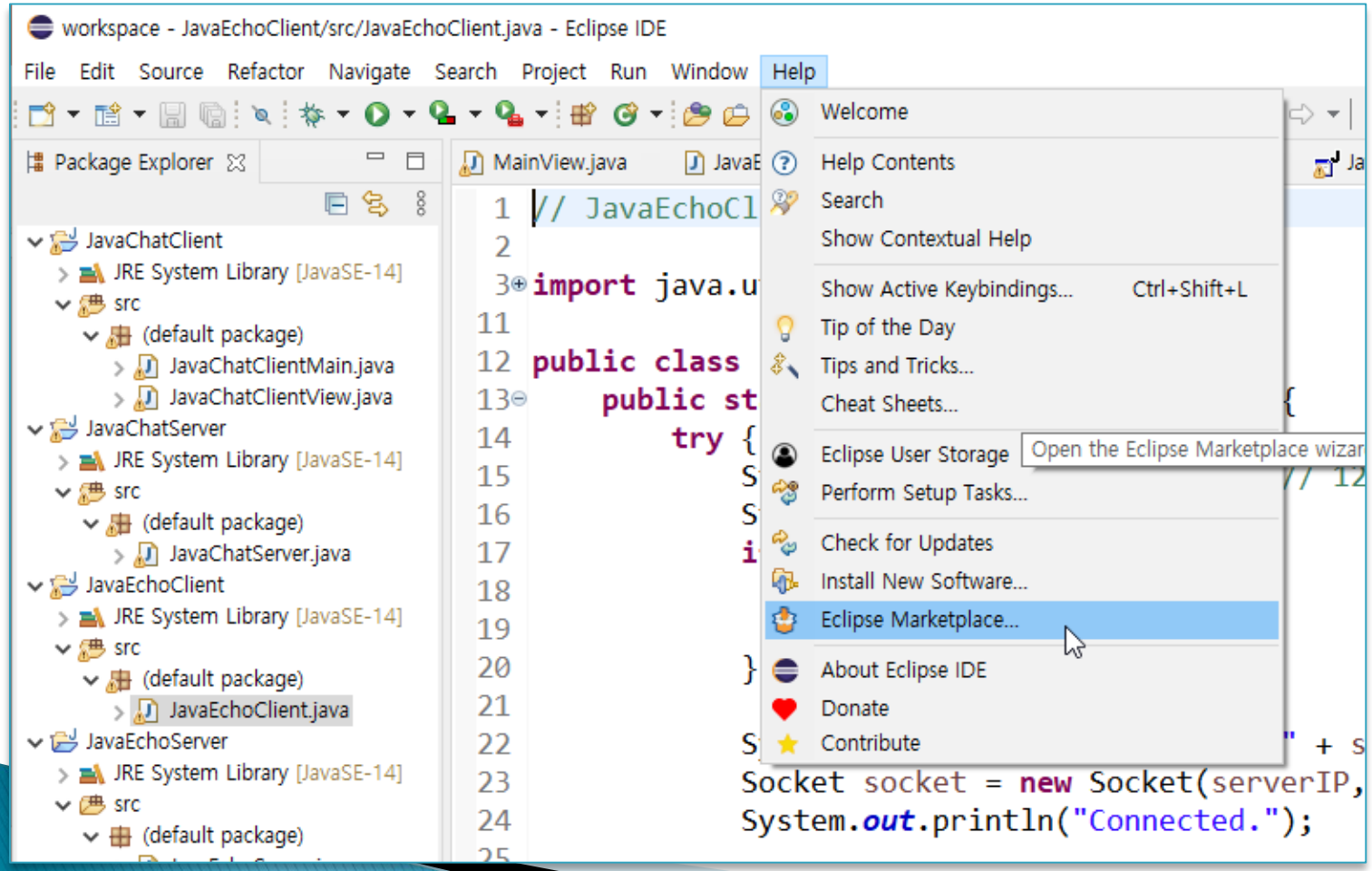


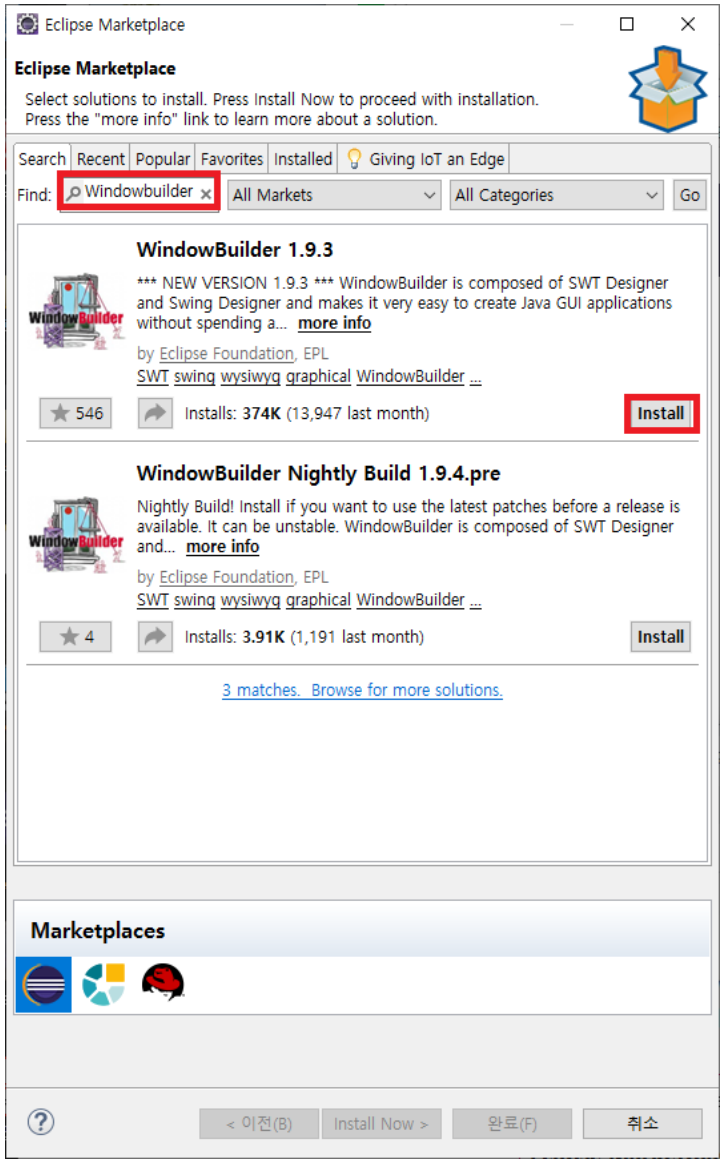
JavaChatServer, JavaChatClient GUI Version

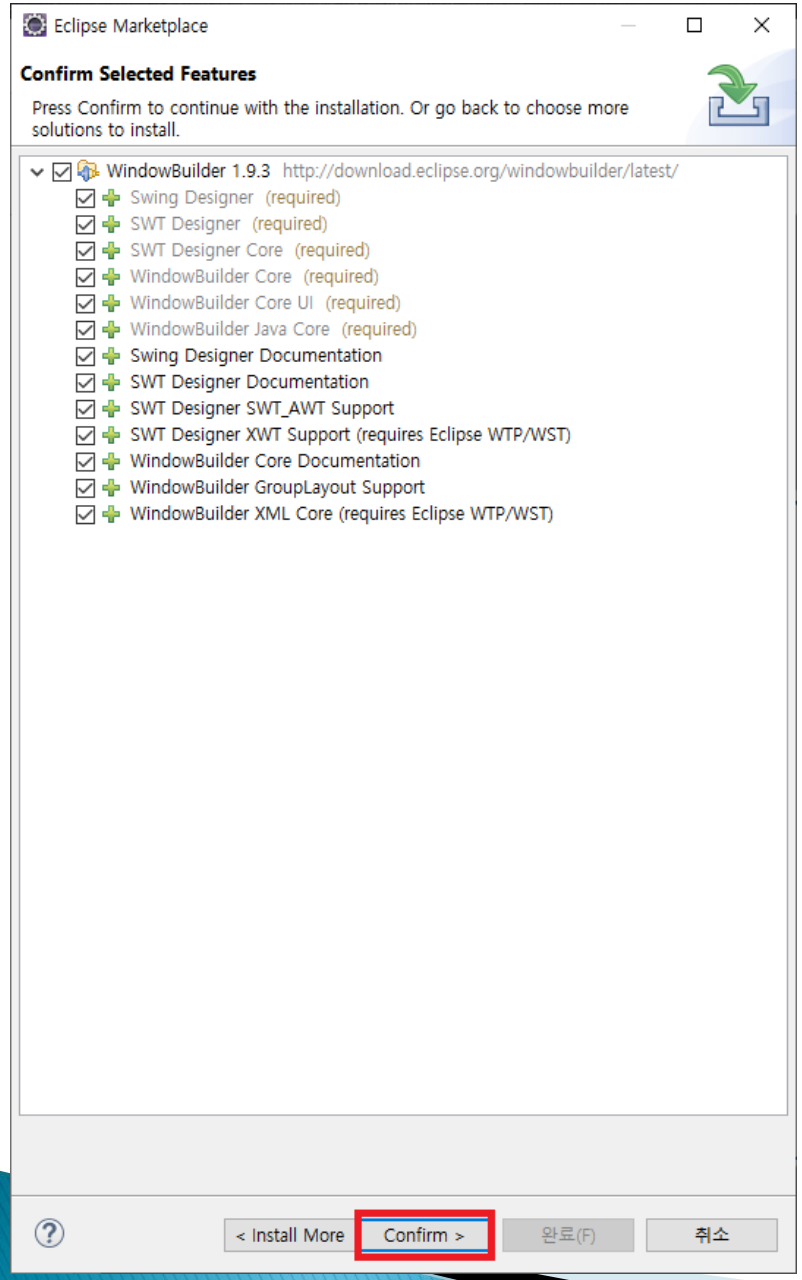


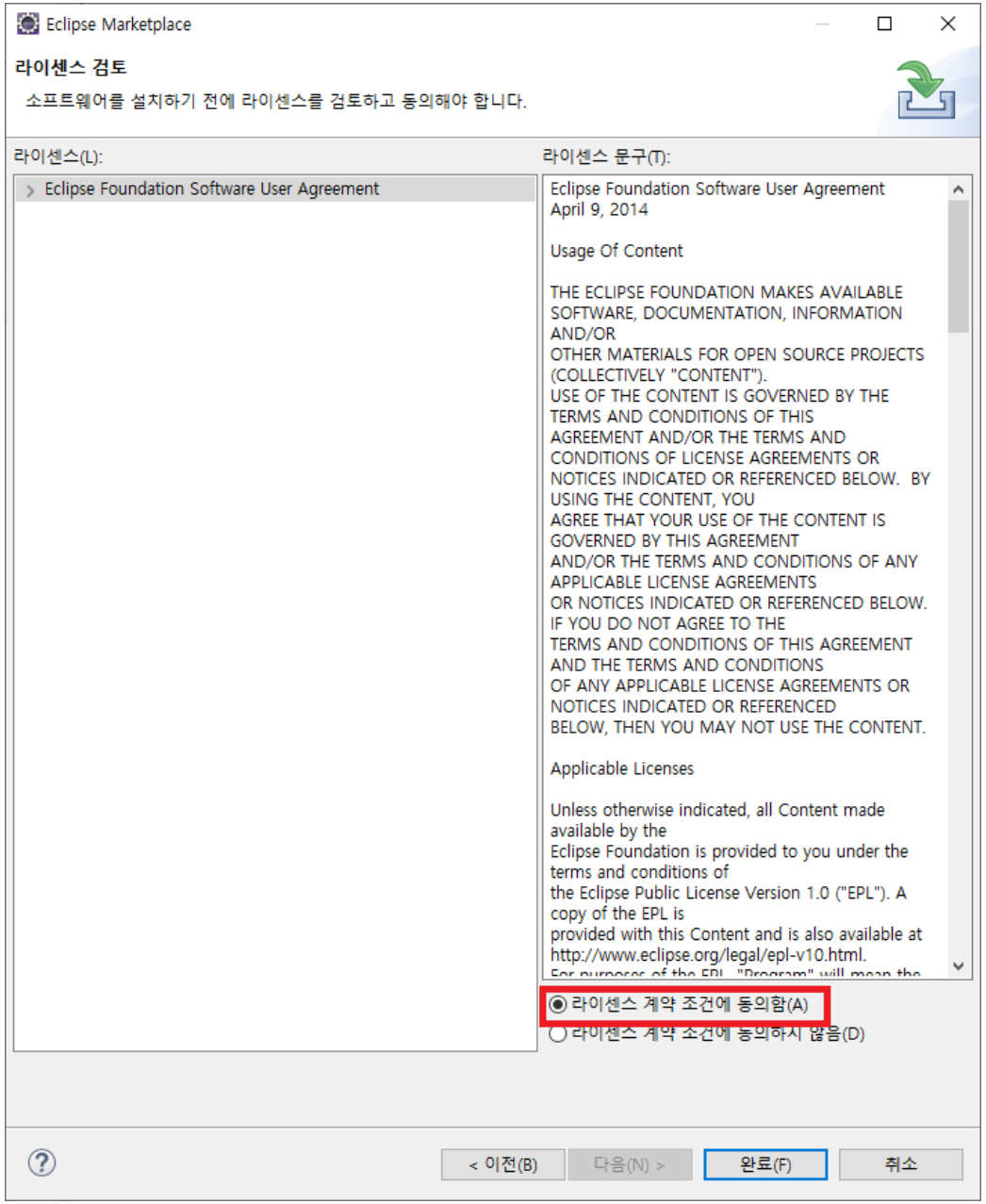
Java Swing GUI 화면 사용

- Eclipse WindowBuilder 설치

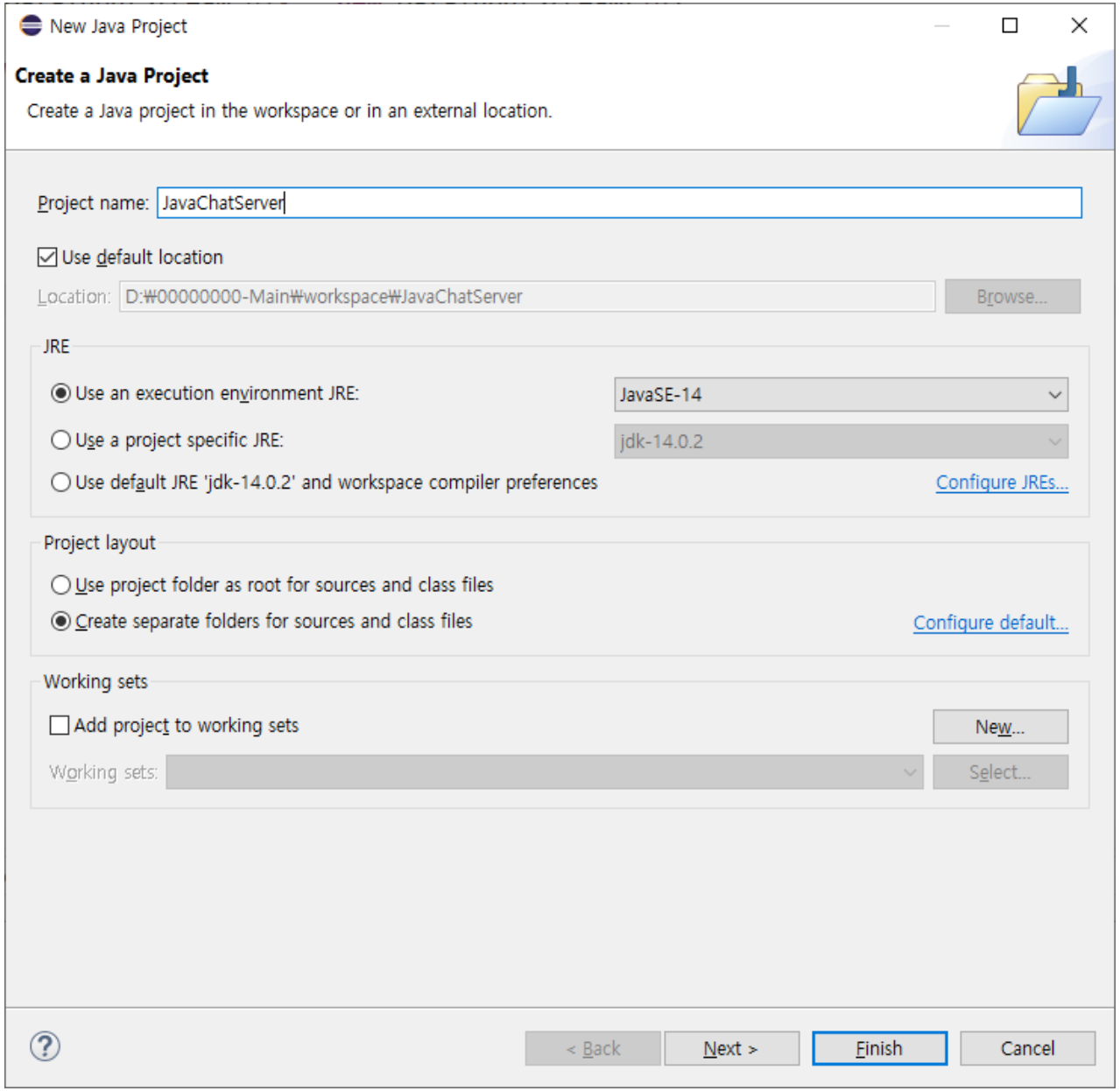




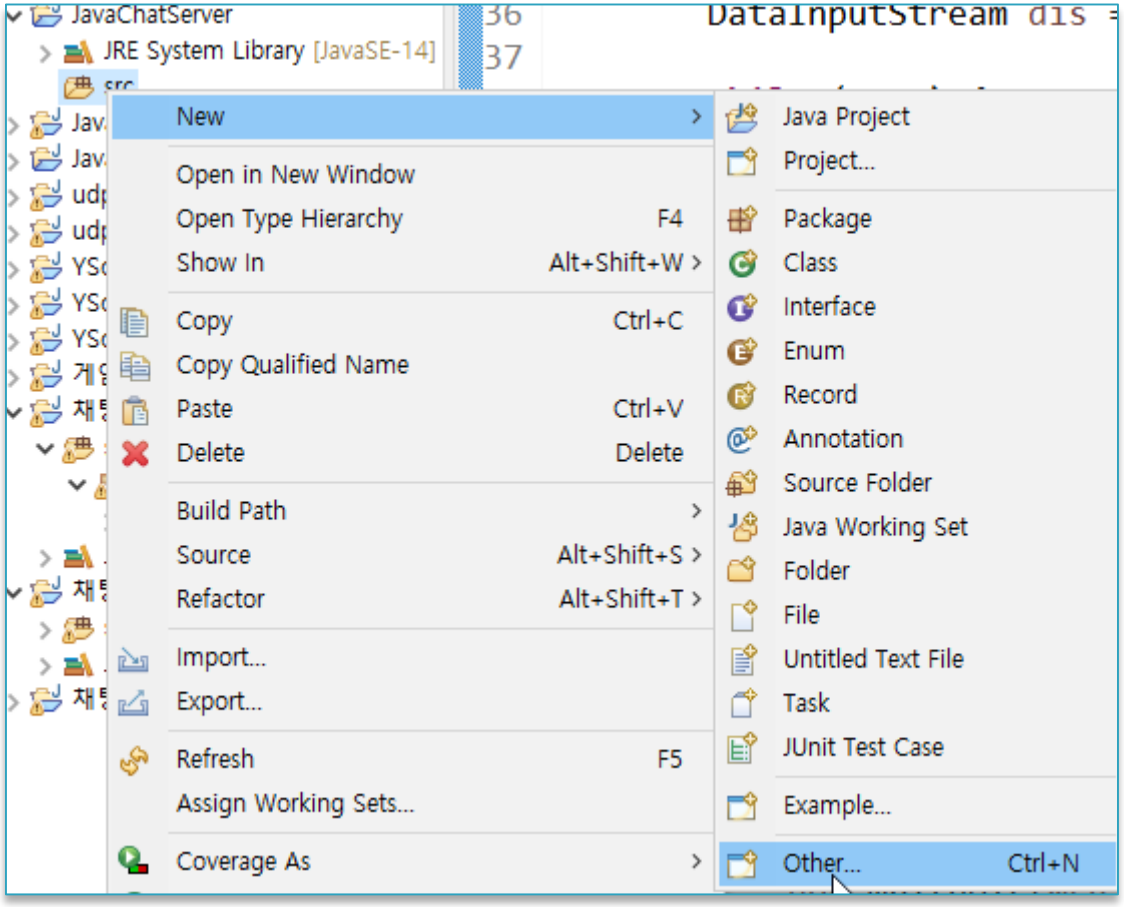

Eclipse Marketplace
Confirm Selected Features
Press Confirm to continue with the installation. Or go back to choose more solutions to install.
WindowBuilder 1.9.3 http://download.eclipse.org/windowbuilder/latest/
Swing Designer (required)
SWT Designer (required)
SWT Designer Core (required)
WindowBuilder Core (required)
WindowBuilder Core UI (required)
WindowBuilder Java Core (required)
Swing Designer Documentation
SWT Designer Documentation
SWT Designer SWT_AWT Support
SWT Designer XWT Support (requires Eclipse WTP/WST)
WindowBuilder Core Documentation
WindowBuilder GroupLayout Support
WindowBuilder XML Core (requires Eclipse WTP/WST)
< Install More
Confirm >
완료(F)
취소


Eclipse Marketplace
라이센스 검토
소프트웨어를 설치하기 전에 라이센스를 검토하고 동의해야 합니다.
라이센스(L):
Eclipse Foundation Software User Agreement
라이센스 문구(T):
Eclipse Foundation Software User Agreement
April 9, 2014
Usage Of Content
THE ECLIPSE FOUNDATION MAKES AVAILABLE SOFTWARE, DOCUMENTATION, INFORMATION AND/OR
OTHER MATERIALS FOR OPEN SOURCE PROJECTS (COLLECTIVELY "CONTENT").
USE OF THE CONTENT IS GOVERNED BY THE TERMS AND CONDITIONS OF THIS
AGREEMENT AND/OR THE TERMS AND CONDITIONS OF LICENSE AGREEMENTS OR NOTICES INDICATED OR REFERENCED BELOW. BY USING THE CONTENT, YOU
AGREE THAT YOUR USE OF THE CONTENT IS GOVERNED BY THIS AGREEMENT
AND/OR THE TERMS AND CONDITIONS OF ANY APPLICABLE LICENSE AGREEMENTS
OR NOTICES INDICATED OR REFERENCED BELOW. IF YOU DO NOT AGREE TO THE
TERMS AND CONDITIONS OF THIS AGREEMENT AND THE TERMS AND CONDITIONS
OF ANY APPLICABLE LICENSE AGREEMENTS OR NOTICES INDICATED OR REFERENCED
BELOW, THEN YOU MAY NOT USE THE CONTENT.
Applicable Licenses
Unless otherwise indicated, all Content made available by the
Eclipse Foundation is provided to you under the terms and conditions of
the Eclipse Public License Version 1.0 ("EPL"). A copy of the EPL is
provided with this Content and is also available at http://www.eclipse.org/legal/epl-v10.html.
라이센스 계약 조건에 동의함(A)
라이센스 계약 조건에 동의하지 않음(D)
< 이전(B)
다음(N) >
완료(F)
취소

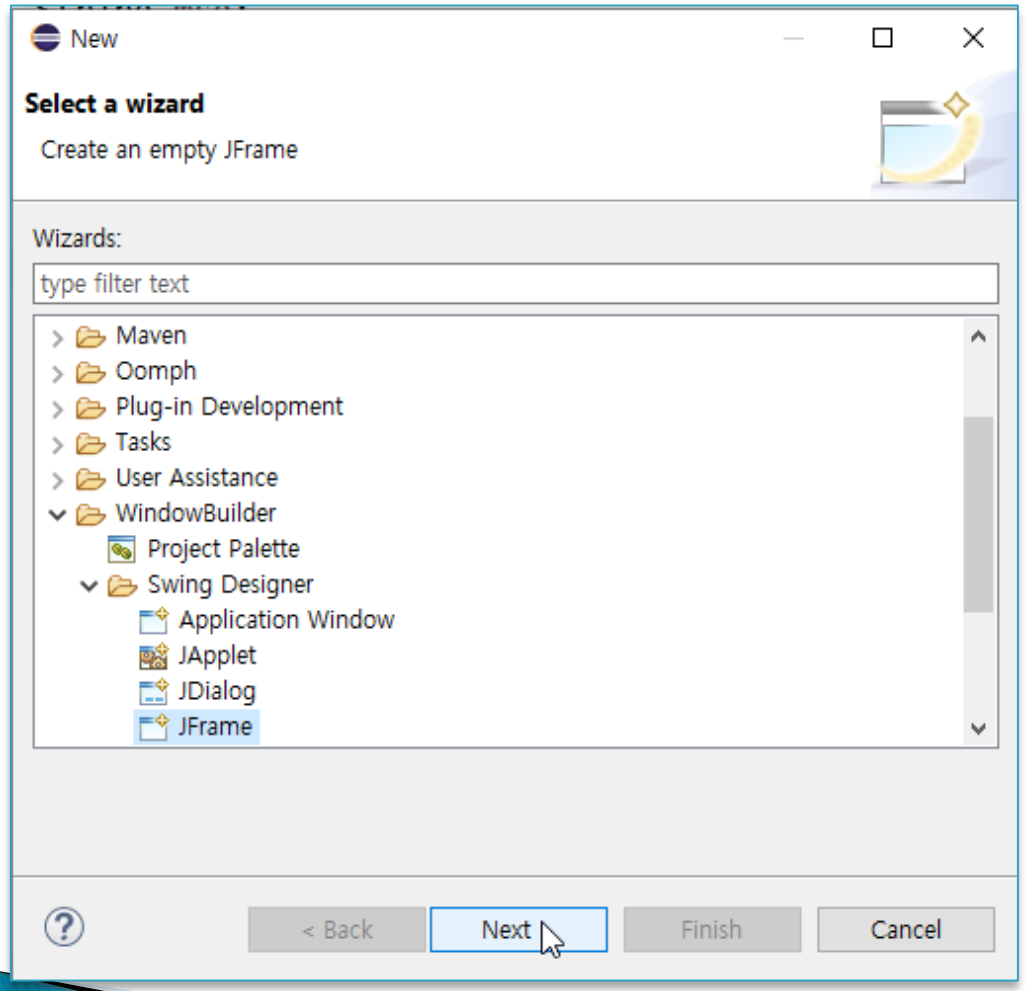
GUI 만들기

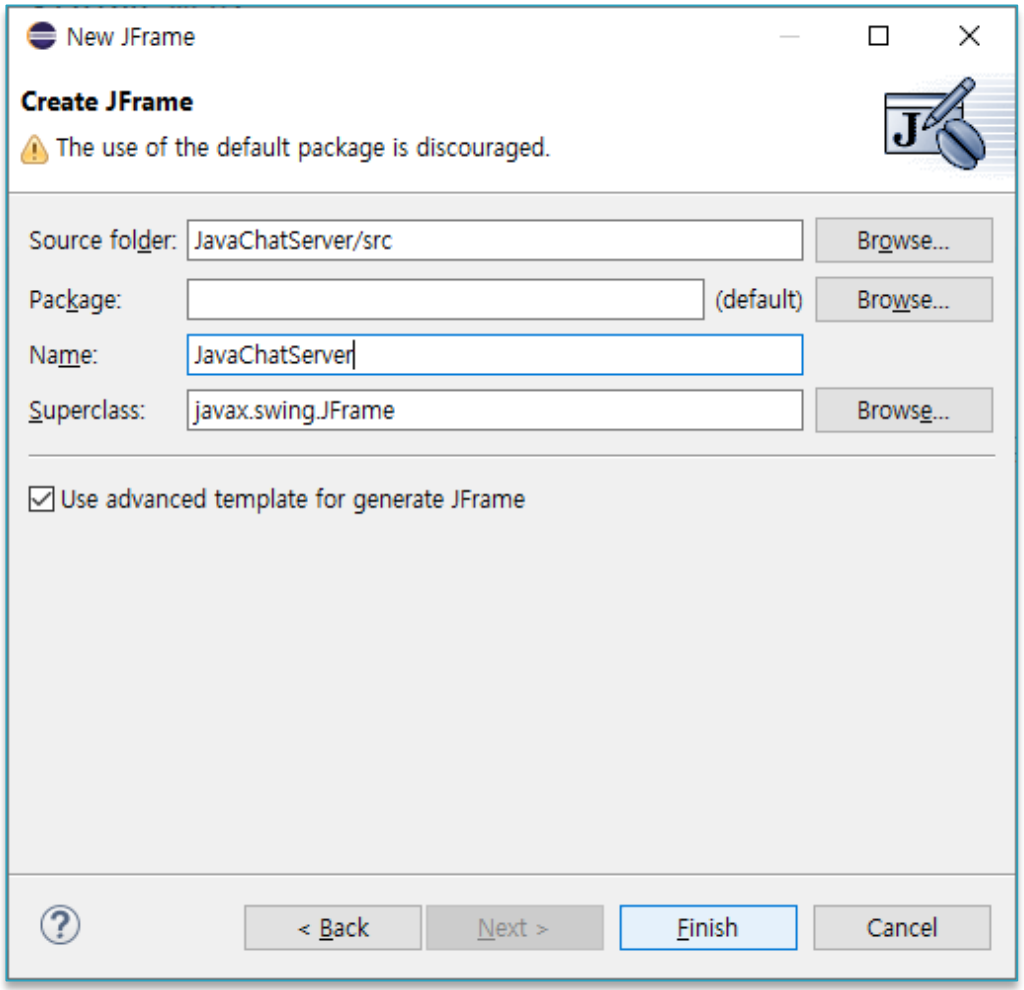

New Java Project
Create a Java Project
Create a Java project in the workspace or in an external location.
Project name: JavaChatServer
Use default location
Location: D:₩00000000-Main₩workspace₩JavaChatServer
Browse...
JRE
Use an execution environment JRE:
JavaSE-14
Use a project specific JRE:
jdk-14.0.2
Use default JRE 'jdk-14.0.2' and workspace compiler preferences
Configure JREs...
Project layout
Use project folder as root for sources and class files
Create separate folders for sources and class files
Configure default...
Working sets
Add project to working sets
New...
Working sets:
Select...
< Back
Next >
Finish
Cancel

Project 만들고 – src / 새로 만들기 - 기타

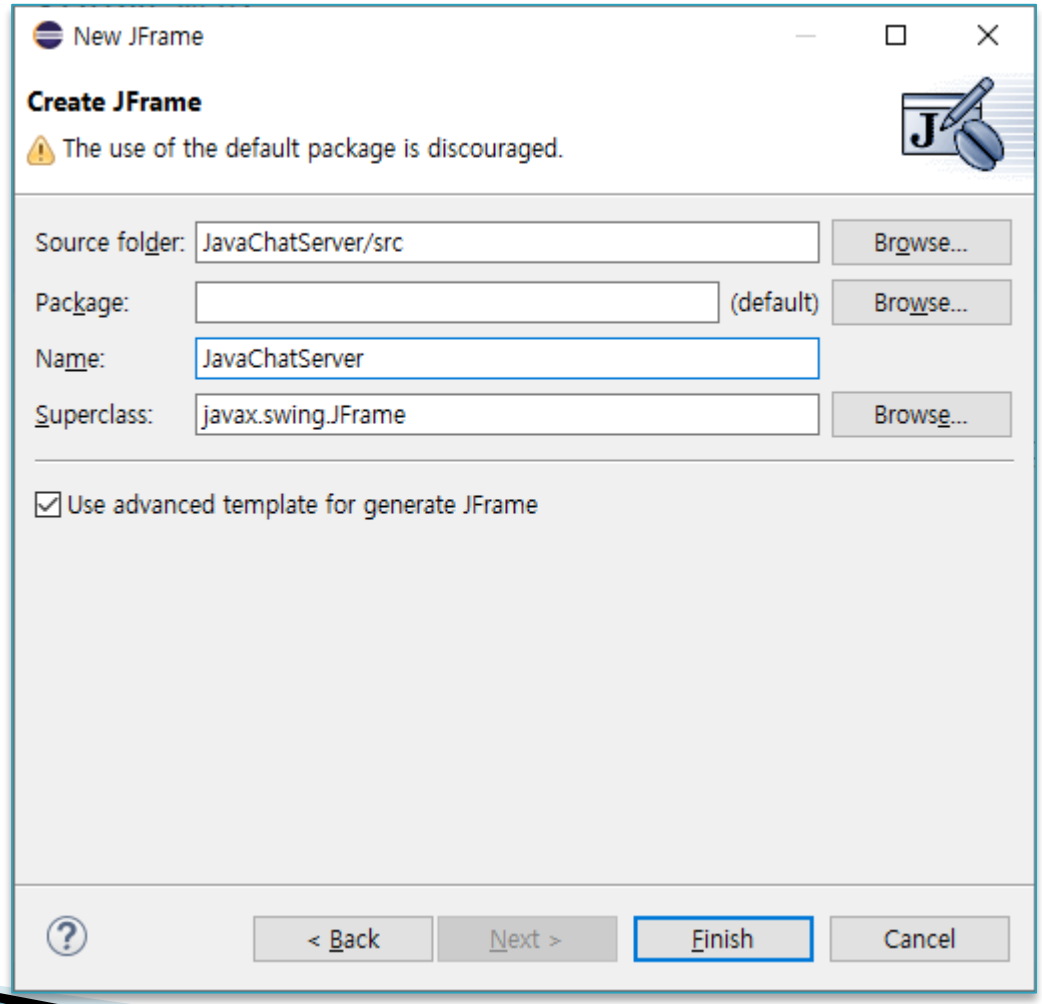


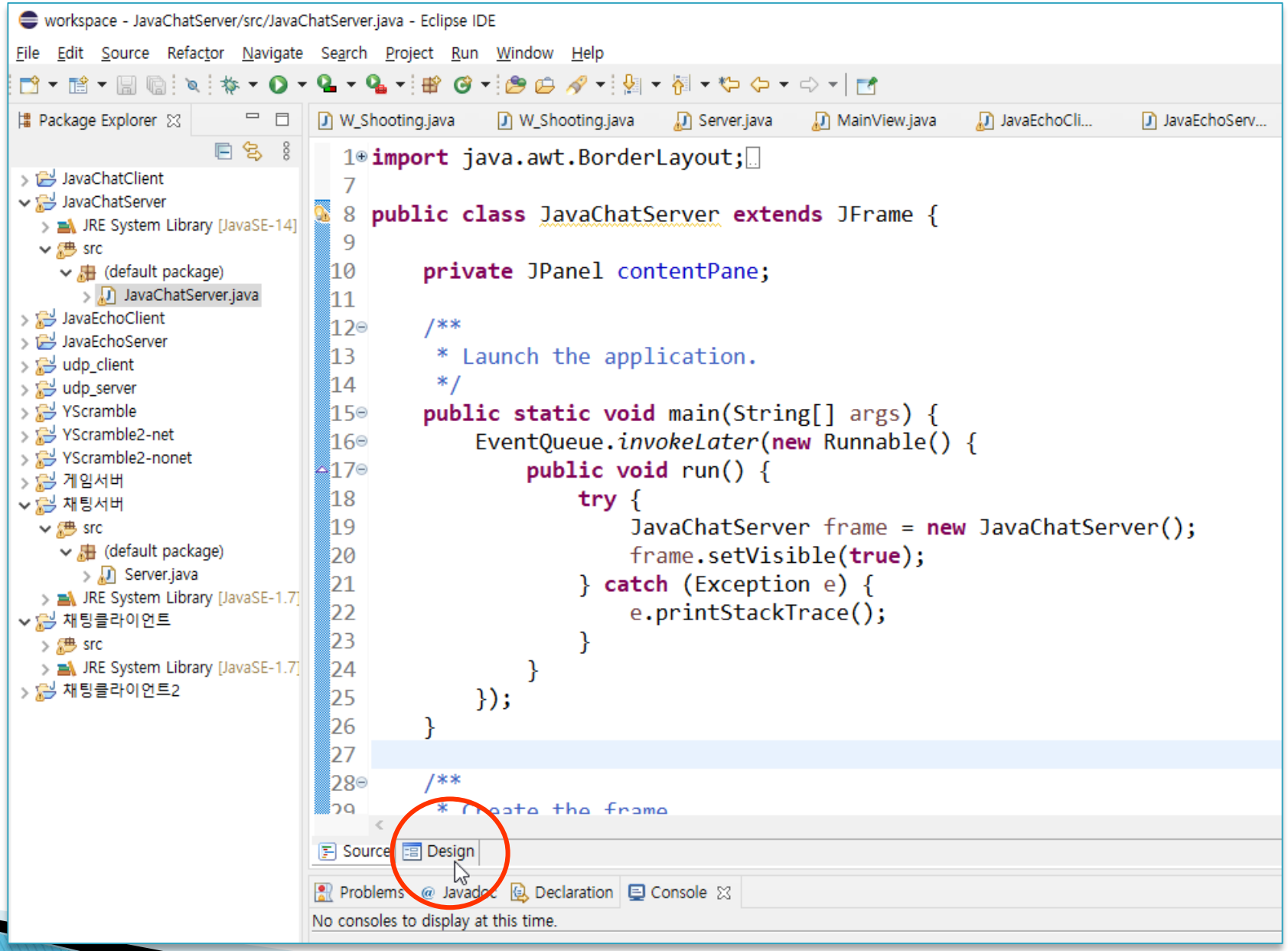
기타 > WindowBuilder > Swing Desinger > JFrame





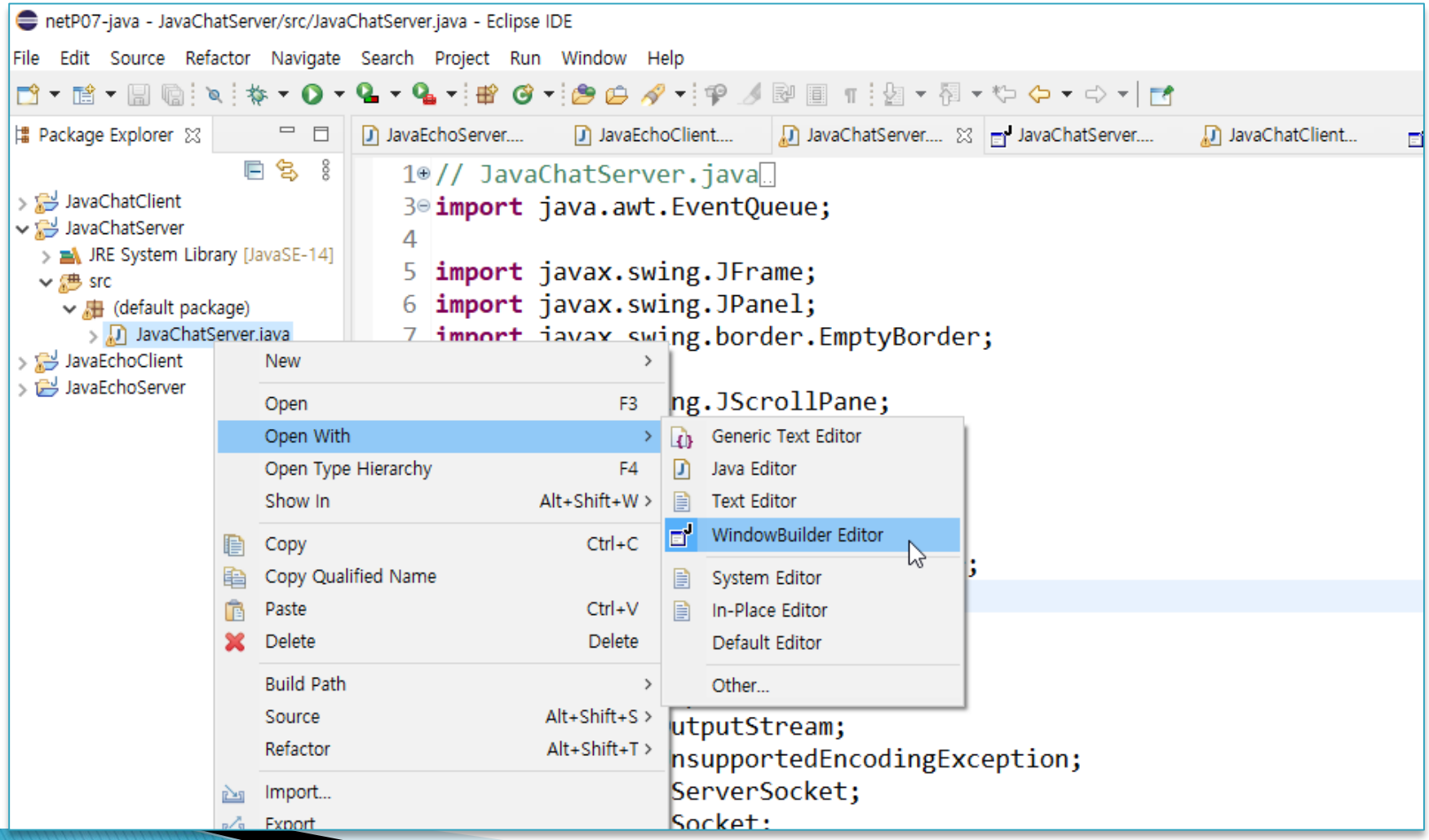
기타 > WindowBuilder > Swing Desinger > JFrame

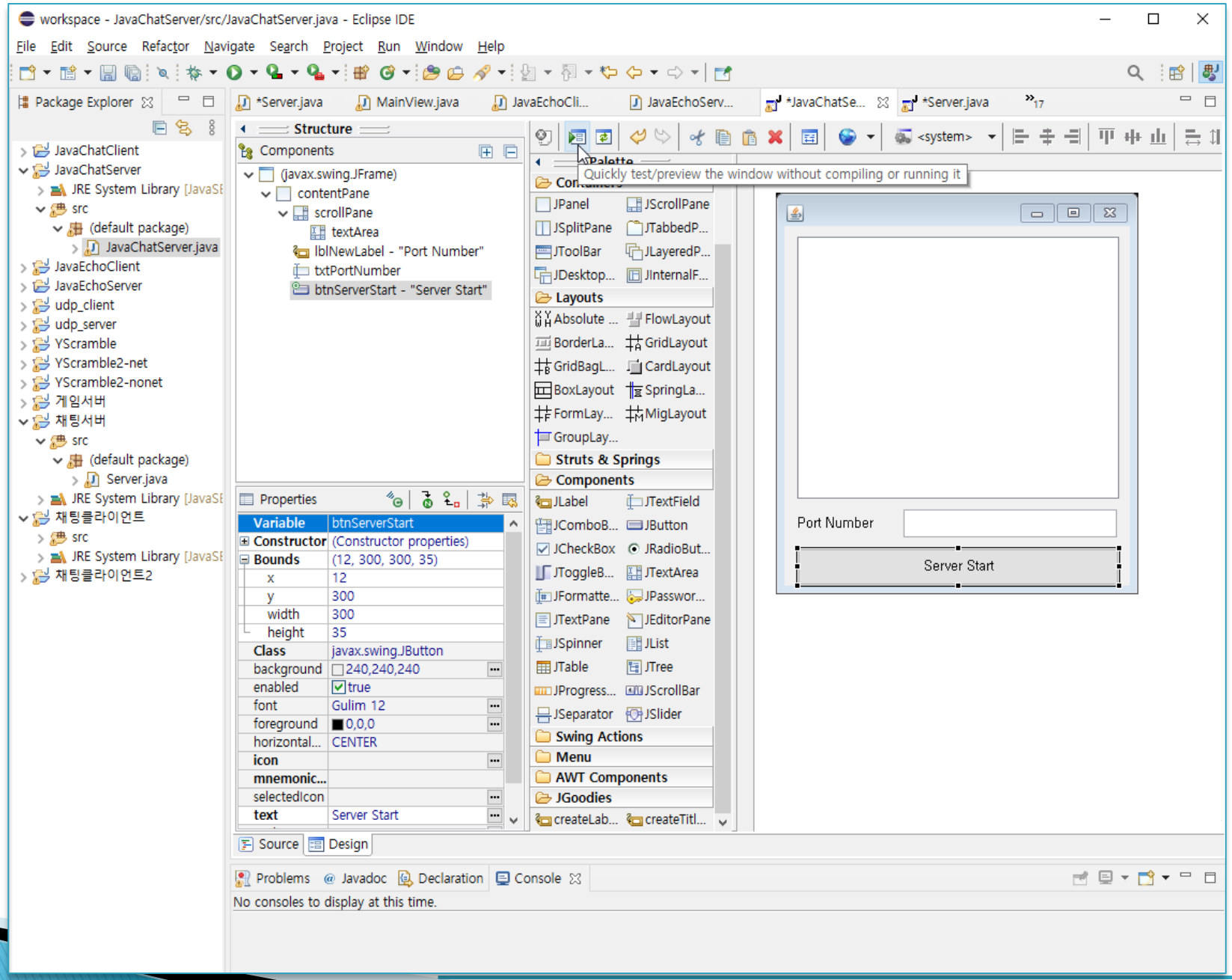



workspace - JavaChatServer/src/JavaChatServer.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
JavaChatClient
JavaChatServer
JRE System Library [JavaSE-14]
src
(default package)
JavaChatServer.java
JavaEchoClient
JavaEchoServer
udp_client
udp_server
YScramble
YScramble2-net
YScramble2-nonet
게임서버
채팅서버
src
(default package)
Server.java
JRE System Library [JavaSE-1.7]
채팅클라이언트
src
JRE System Library [JavaSE-1.7]
채팅클라이언트2
W_Shooting.java
W_Shooting.java
Server.java
MainView.java
JavaEchoCli...
JavaEchoServ...
1 import java.awt.BorderLayout;
7
8 public class JavaChatServer extends JFrame {
9
10 private JPanel contentPane;
11
12 /**
13 * Launch the application.
14 */
15 public static void main(String[] args) {
16 EventQueue.invokeLater(new Runnable() {
17 public void run() {
18 try {
19 JavaChatServer frame = new JavaChatServer();
20 frame.setVisible(true);
21 } catch (Exception e) {
22 e.printStackTrace();
23 }
24 }
25 });
26 }
27
28 /**
Source
Design
Problems
Javadoc
Declaration
Console
No consoles to display at this time.

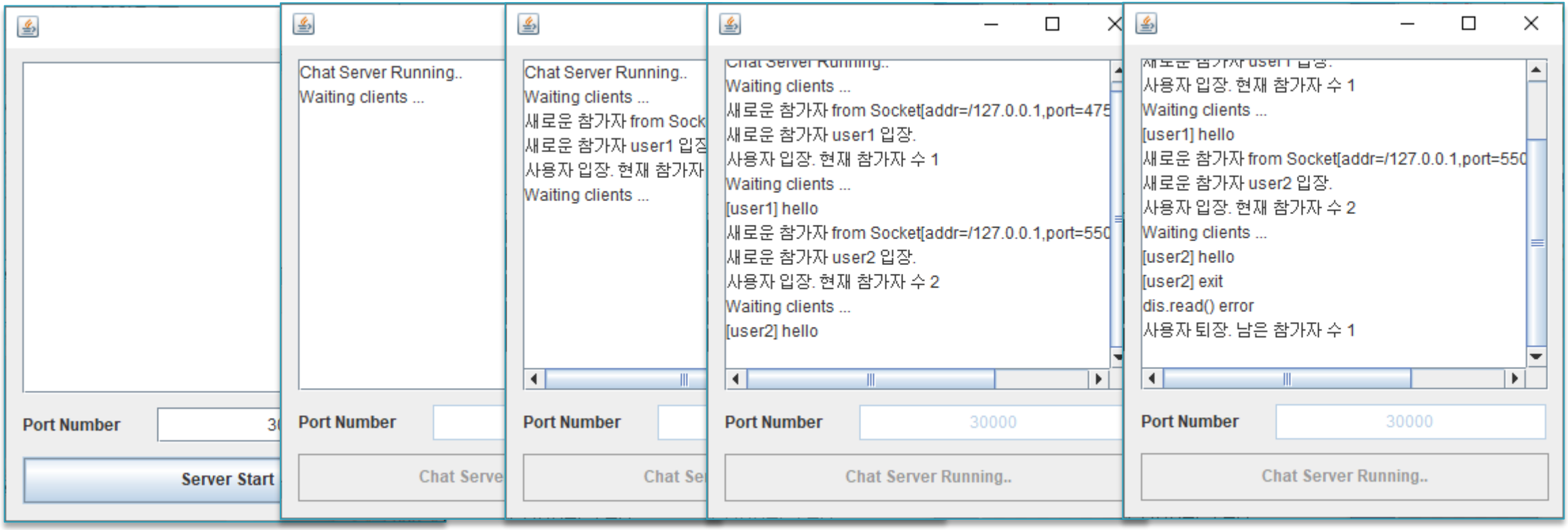
Design Button 이 안보이는 경우

- java파일 > 우측버튼 > Open With > WindowBuilder Editor

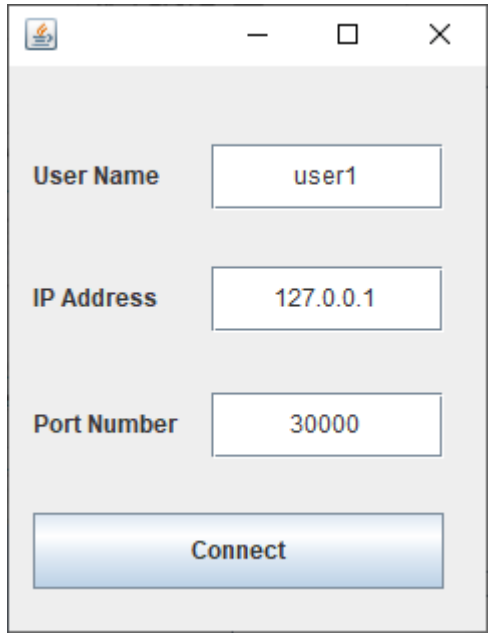


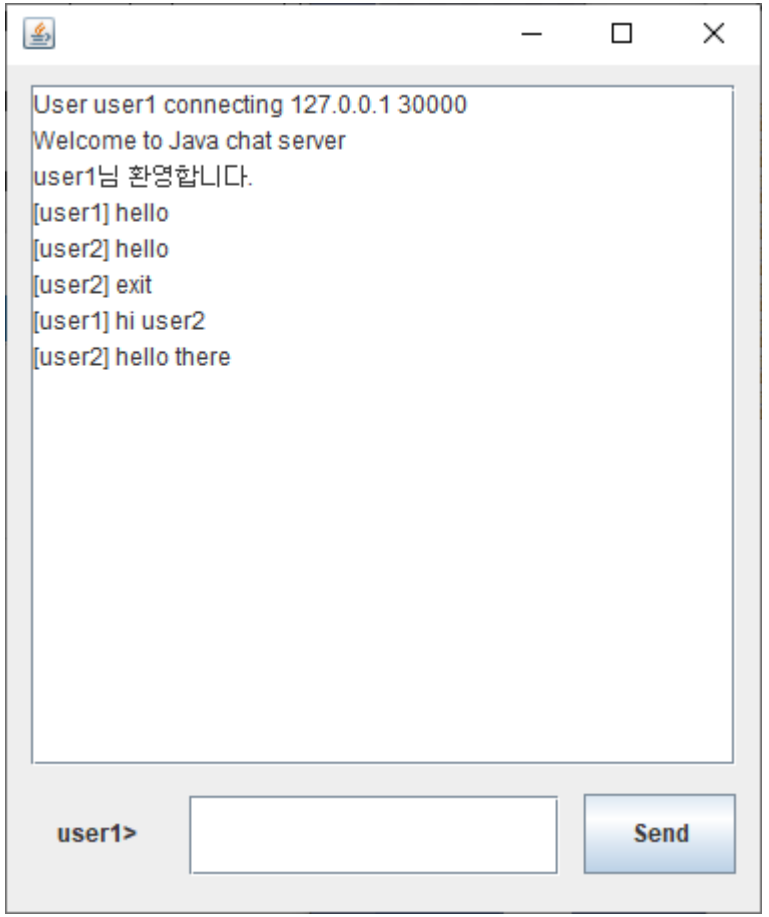

workspace - JavaChatServer/src/JavaChatServer.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
JavaChatClient
JavaChatServer
JRE System Library [JavaSE
src
(default package)
JavaChatServer.java
JavaEchoClient
JavaEchoServer
udp_client
udp_server
YScramble
YScramble2-net
YScramble2-nonet
게임서버
채팅서버
src
(default package)
Server.java
JRE System Library [JavaSE
채팅클라이언트
src
JRE System Library [JavaSE
채팅클라이언트2
*Server.java
MainView.java
JavaEchoCli...
JavaEchoServ...
*JavaChatSe...
*Server.java
Structure
Components
(javax.swing.JFrame)
contentPane
scrollPane
textArea
lblNewLabel - "Port Number"
txtPortNumber
btnServerStart - "Server Start"
Quickly test/preview the window without compiling or running it
Palette
JPanel
JScrollPane
JSplitPane
JTabbedP...
JToolBar
JLayeredP...
JDesktop...
JInternalF...
Layouts
Absolute ...
FlowLayout
BorderLa...
GridLayout
GridBagL...
CardLayout
BoxLayout
SpringLa...
FormLay...
MigLayout
GroupLay...
Struts & Springs
Components
JLabel
JTextField
JComboB...
JButton
JCheckBox
JRadioBut...
JToggleB...
JTextArea
JFormatte...
JPasswor...
JTextPane
JEditorPane
JSpinner
JList
JTable
JTree
JProgress...
JScrollBar
JSeparator
JSlider
Swing Actions
Menu
AWT Components
JGoodies
createLab...
createTitl...
Properties
Variable btnServerStart
Constructor (Constructor properties)
Bounds (12, 300, 300, 35)
x 12
y 300
width 300
height 35
Class javax.swing.JButton
background 240,240,240
enabled true
font Gulim 12
foreground 0,0,0
horizontal... CENTER
icon
mnemonic...
selectedIcon
text Server Start
Port Number
Server Start
Source
Design
Problems
Javadoc
Declaration
Console
No consoles to display at this time.

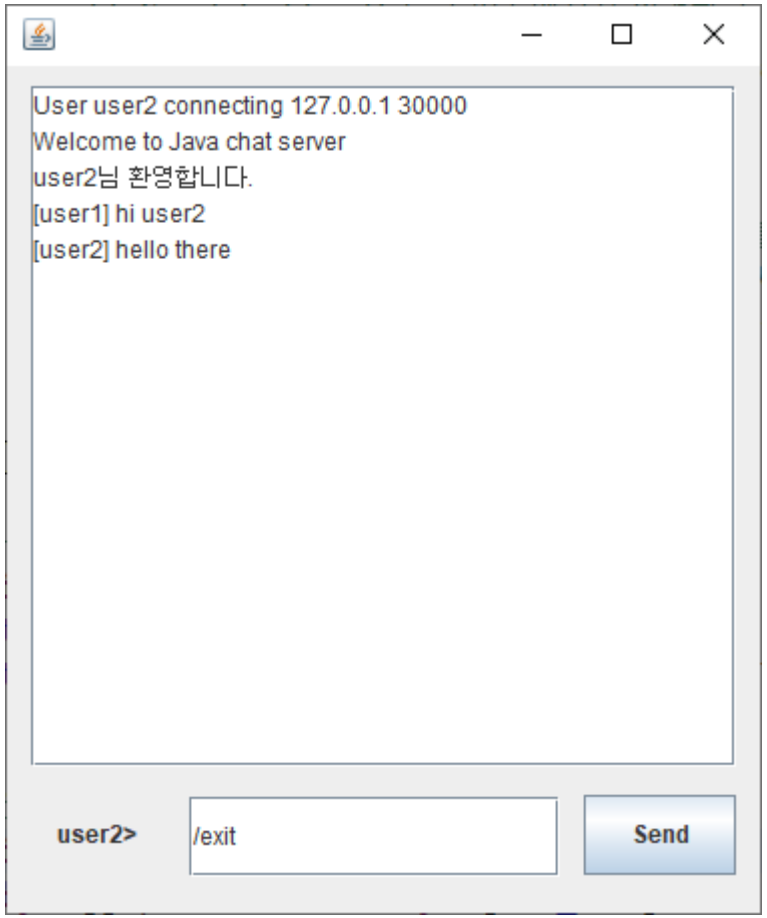
JavaChatServer.java 실행 화면



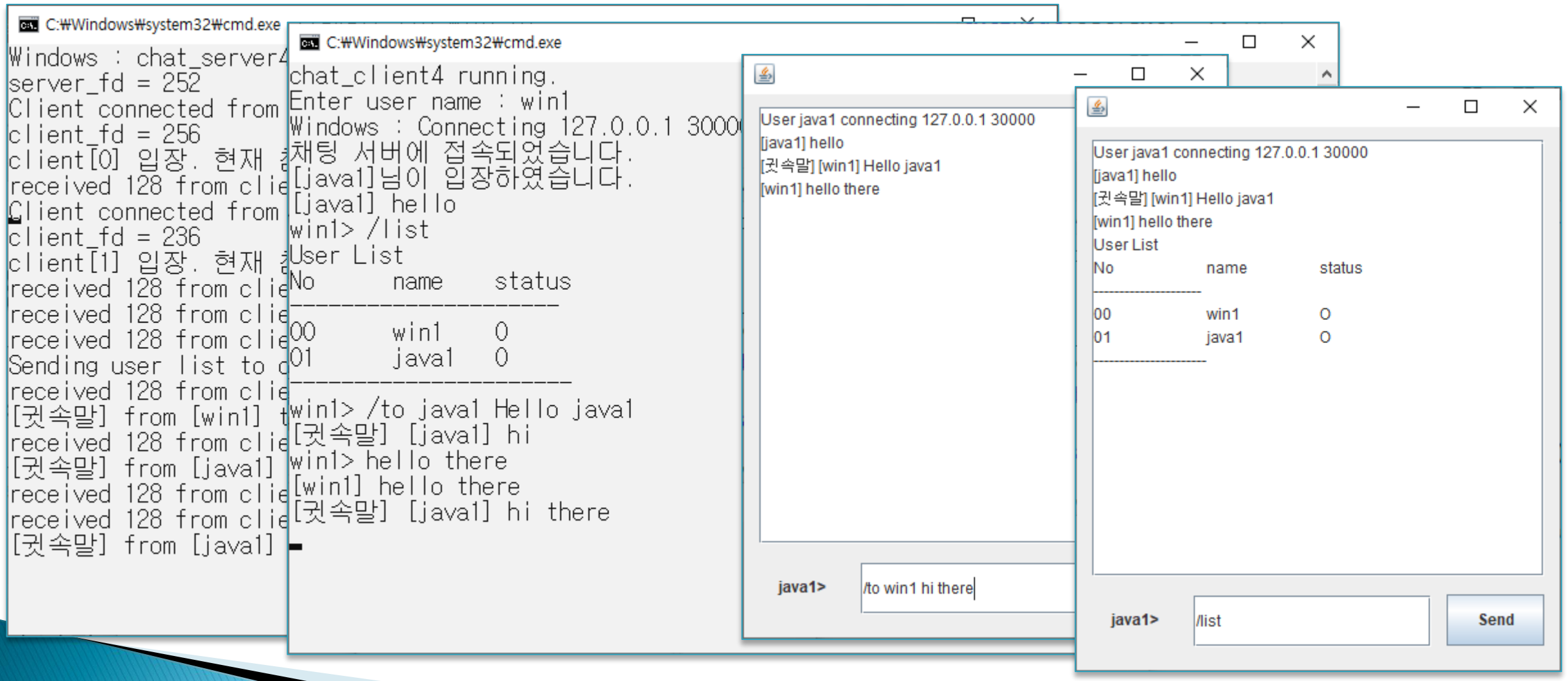
JavaChatClient.java 실행 화면



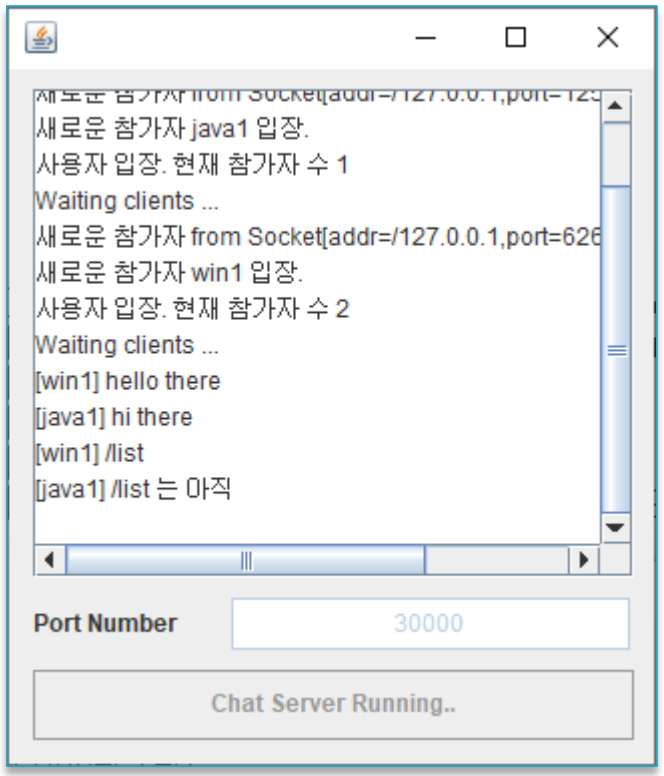


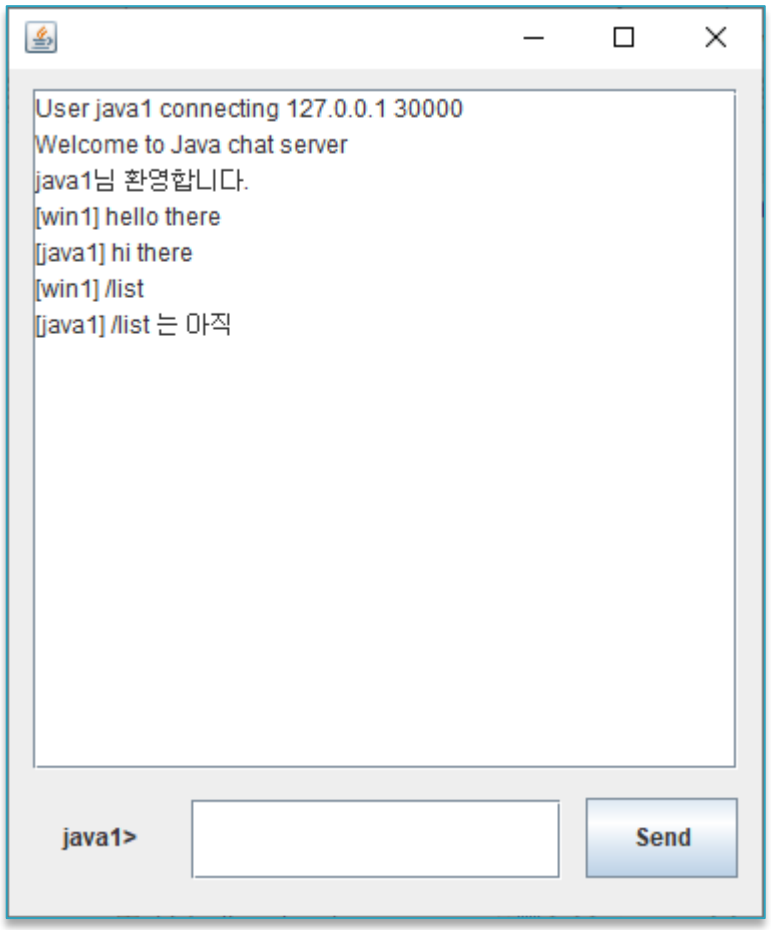


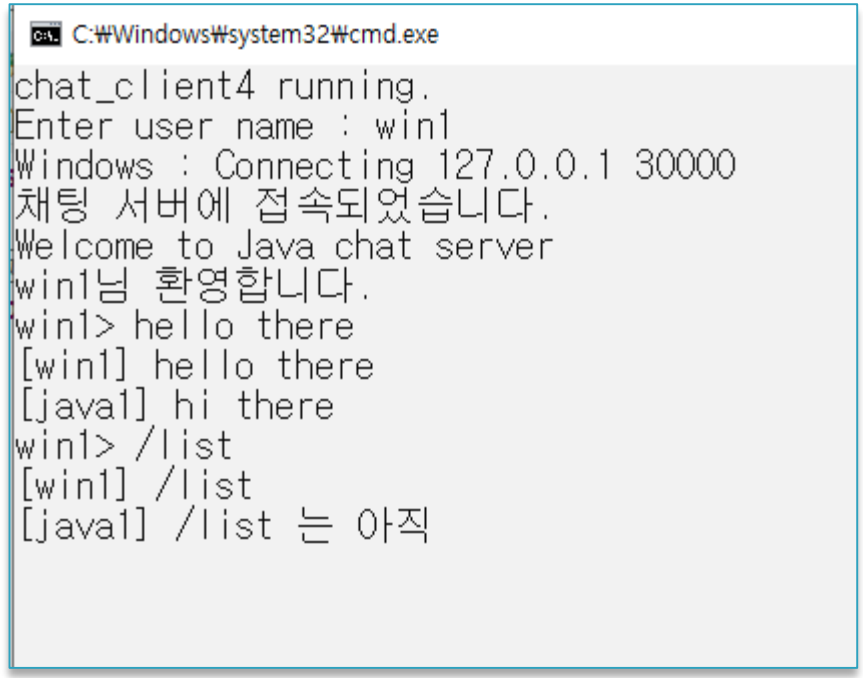
Windows Client/Server Java Client



Java Client/Server Windows Client







7주 과제1 : Java EchoClient/Server, ChatClient/Server 실습

- Java Echo Client/Server 실습
- Chat Client/Server 실습
  - Java Client/Server, Windows Client4
    - /login user1 (User Name 입력하고 접속하면 됨)
    - /exit
  - Windows Client/Server – 4, Java Client
    - /list, /to 귓속말, /sleep, /wakeup

7주 과제2 : Win chat_server4 기능 JavaChatServer에 구현하기

- Client는 추가 기능 없음
- Windows chat_server4
  - 입장/퇴장 알리기
  - /exit, /list, /to 귓속말, /sleep, /wakeup
- 입장/퇴장 알리기
  - 입장 : AcceptServer Accept하고, UserService() 생성되면 생성자에서 처리
    - UserName = msg[1].trim() → UserName을 이용 WriteAll() 하면 됨
  - 퇴장 : UserVec.removeElement(**this**); 하기 전에 퇴장 알림 처리.
- /exit, /list, /to
  - dis.read() 다음 처리, UserVec.removeElement(this); 로 제거
  - JavaChatServer의 class UserService {}
  - private → public String UserName 변수를 이용
    - /list 목록을 보여주고
    - /to 상대를 찾는다.
- /sleep, /wakeup
  - JavaChatServer의 class UserService {}
  - public String UserStatus 변수 추가