

컴퓨터 네트워크 학기말 과제 보고서

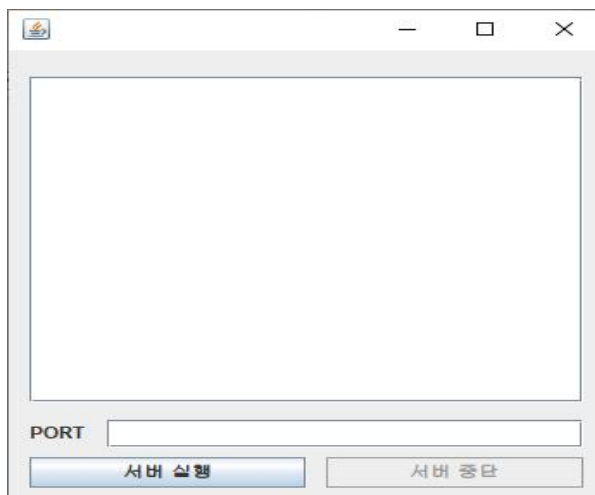
2018147549 이해지

I. 개발 환경

Eclipse IDE(버전 : 2020-09 (4.17.0))에서 JAVA로 개발하였다.

II. 실행 방법 및 구현한 기능

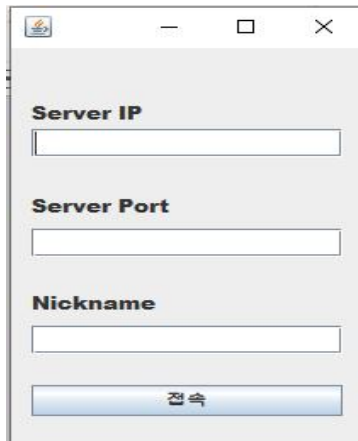
먼저 Server.java파일을 Run시킨다. 그러면 아래와 같은 화면이 뜬다.



<PORT> 입력칸에 실행할 포트번호를 입력하고 <서버 실행>버튼을 클릭한다. 예를 들어 포트 번호를 '111'이라 하자. 111을 입력하고 <서버 실행>버튼을 클릭하면 아래와 같은 화면이 뜬다.

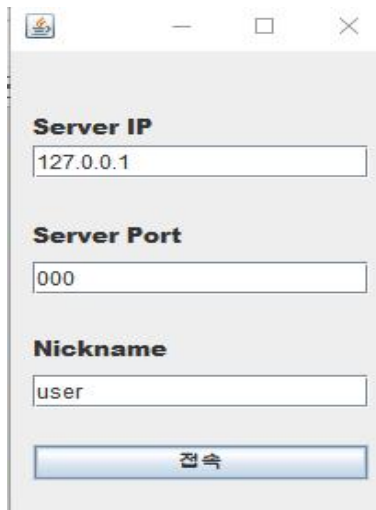


이제 Client.java파일을 Run시킨다. 그러면 아래와 같은 화면이 뜬다.

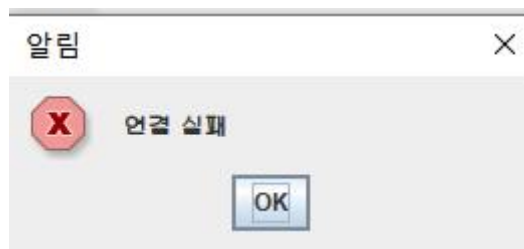


A Java Swing window titled 'Client' with a standard title bar (minimize, maximize, close). It contains three text input fields labeled 'Server IP', 'Server Port', and 'Nickname'. Below these fields is a blue button labeled '접속' (Connect).

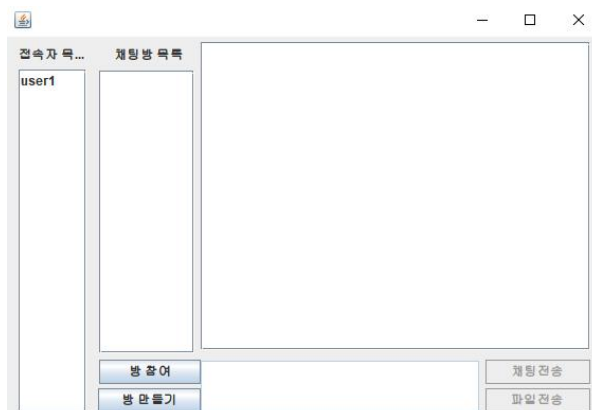
IP주소와 Port번호, Nickname을 입력하고 접속을 누른다. 이때 서버에서 실행되고 있지 않은 port번호를 입력하고 접속을 누르면 아래와 같이 뜬다.



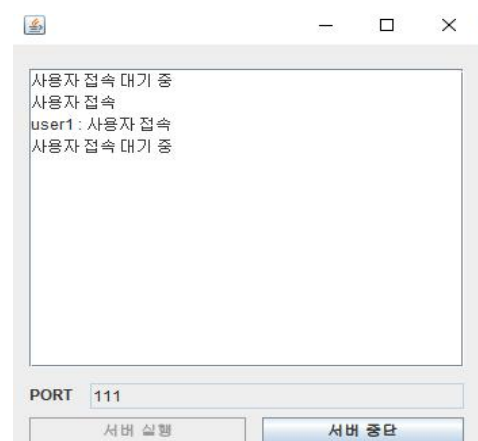
The same 'Client' window as before, but with the following values entered: 'Server IP' is '127.0.0.1', 'Server Port' is '000', and 'Nickname' is 'user'. The '접속' button is still visible.



아까 111포트번호를 서버에서 실행시켰으니 올바르게 포트번호를 111로 입력하고 접속하면 아래와 같은 창이 뜨고 서버 측에서는 이렇게 뜬다.

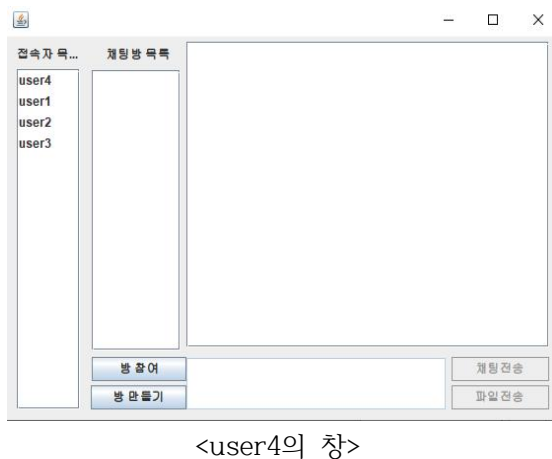
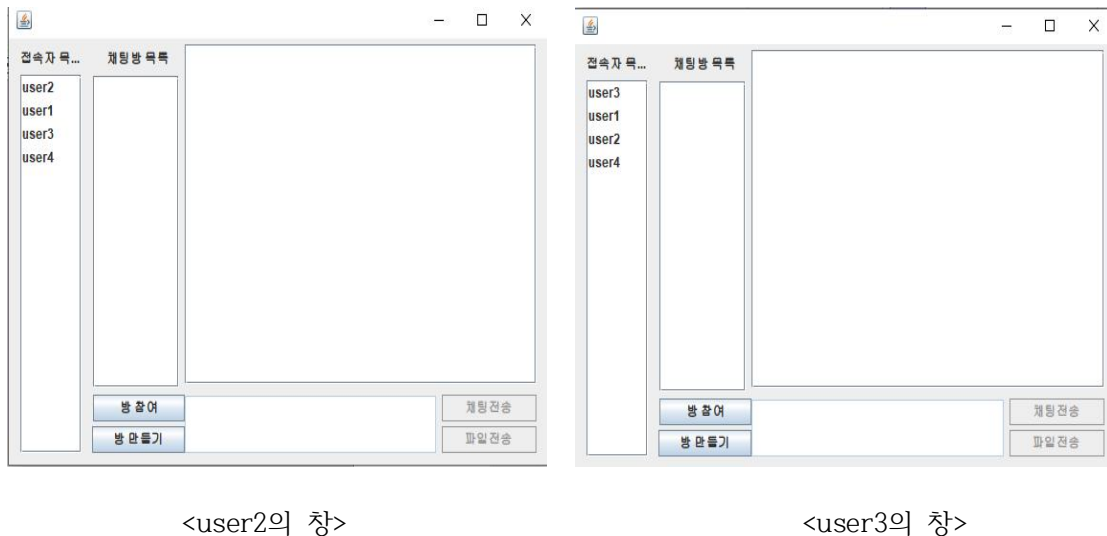
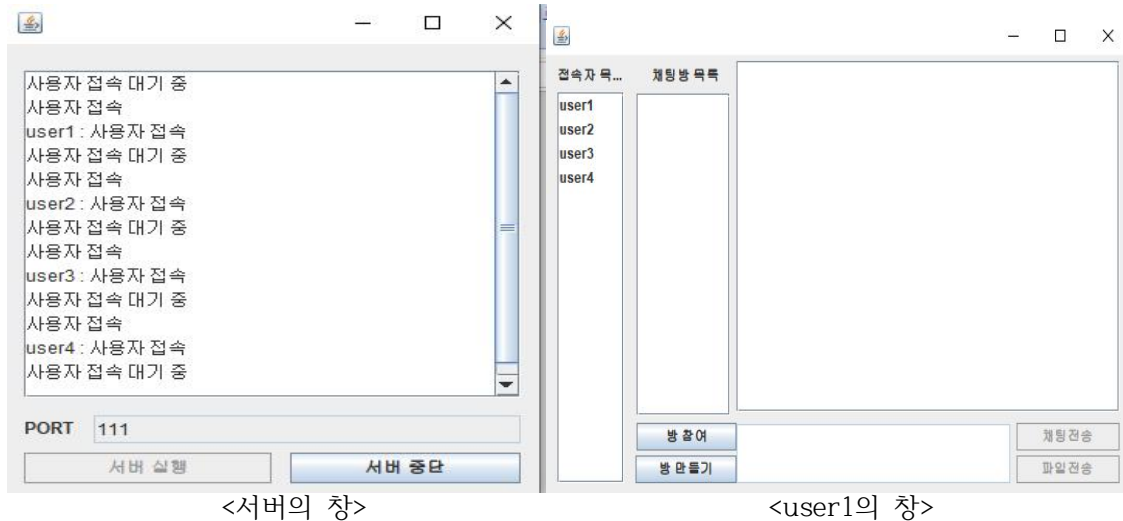


A Java Swing window titled 'Server' with a standard title bar. It has two vertical list boxes on the left: '접속자 목록...' (Connected users list...) containing 'user1' and '채팅방 목록...' (Chat room list...). To the right is a large text area. At the bottom are four buttons: '방 참여' (Join room), '방 안 들기' (Listen to room), '채팅 전송' (Send chat), and '파일 전송' (Send file).

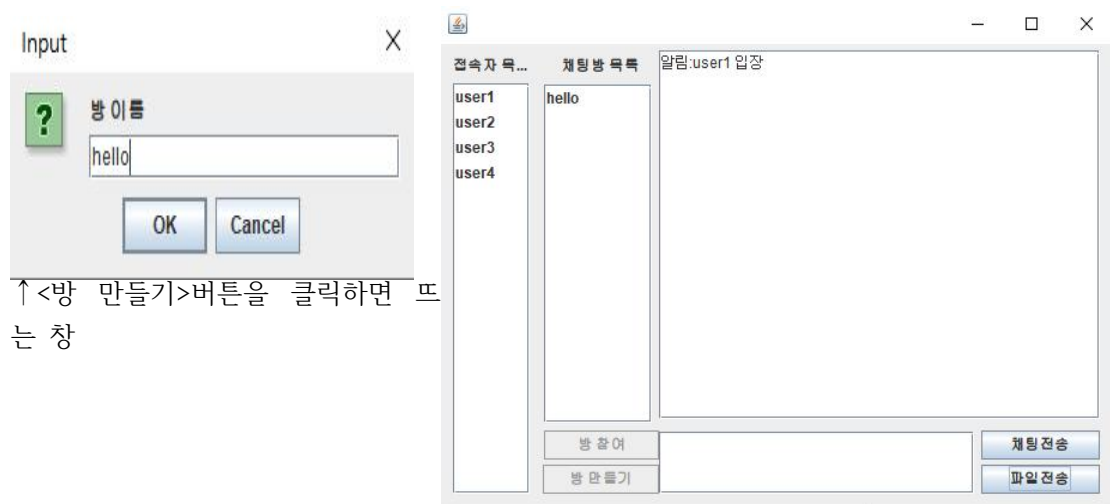


A Java Swing window titled 'Server Log' with a standard title bar. It contains a large text area with the following log messages: '사용자 접속 대기 중' (Waiting for user connection), '사용자 접속' (User connection), 'user1 : 사용자 접속' (user1 : User connection), and '사용자 접속 대기 중' (Waiting for user connection). Below the text area is a 'PORT' label followed by the value '111'. At the bottom are two buttons: '서버 실행' (Run server) and '서버 중단' (Stop server).

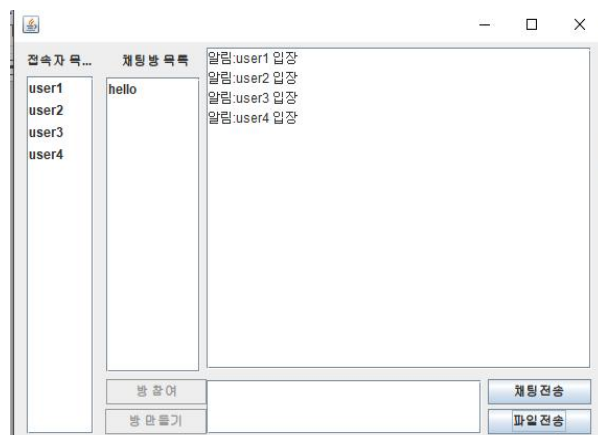
클라이언트는 여러 명일 수 있다. Client.java파일을 여러번 run하면 여러개의 클라이언트를 만들 수 있다. 위와 같은 방법으로 일단 클라이언트를 4명 만들면 창은 클라이언트 4명 각각 1개, 서버 1개로 총 5개가 된다. 창 5개는 아래와 같다.(접속자 목록에서 맨 위에 있는 이름이 자기자신이다.)



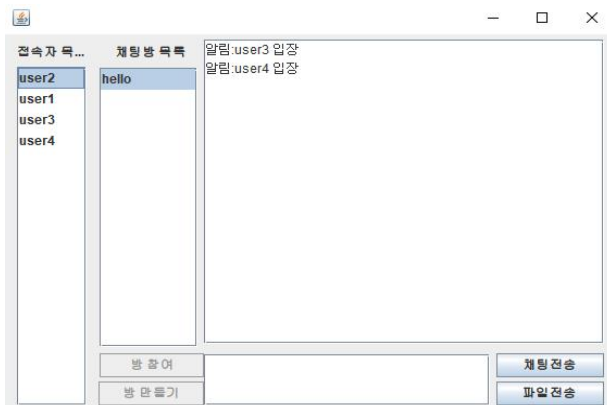
이제 채팅을 하려면 채팅방을 만들어야한다. <방 만들기>버튼을 클릭하면 방을 만들 수 있고 방의 이름을 입력하면 방이 생성된다. 일단 user1이 'hello'라는 방을 만들면 각각의 클라이언트들의 창에 모두 방이 뜬고 user1은 저절로 'hello'라는 방에 들어가진다. 나머지 클라이언트들은 참여하고 싶은 방을 클릭하고 <방 참여>버튼을 누르면 그 방에 들어가진다. user2, user3, user4를 차례대로 참여시킨다.



<방을 만든 후 user1의 창>

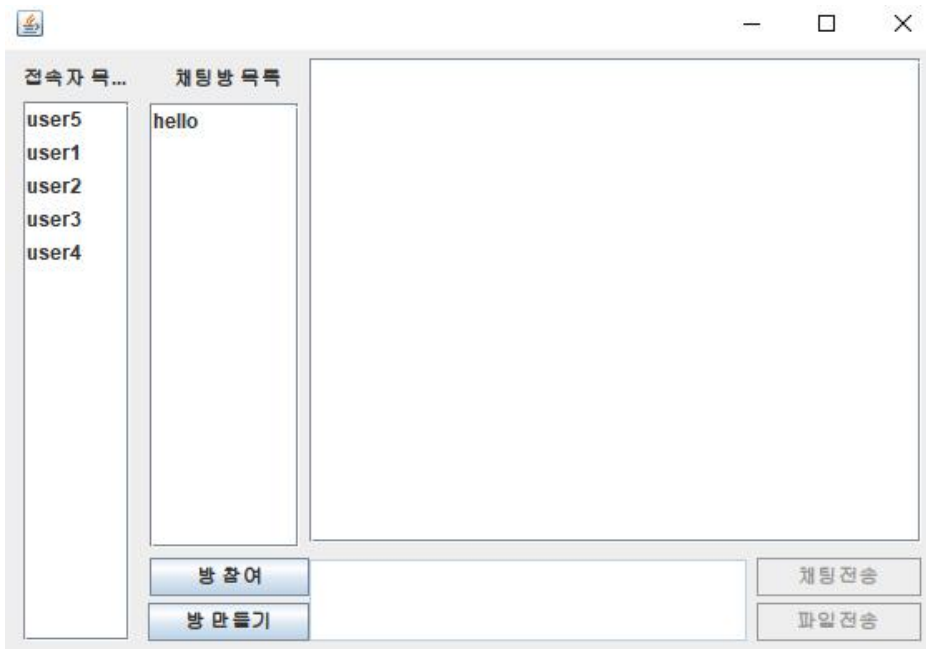


<user2, user3, user4를 'hello'방에 참여시킨 후 user1의 창>

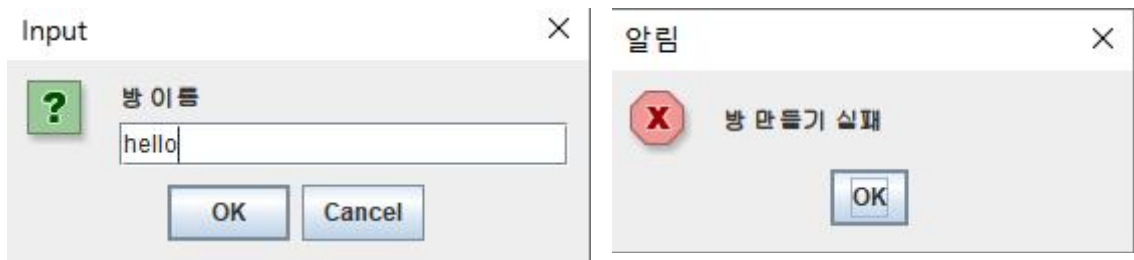


<user2 입장 후 user3, user4를 'hello'방에 참여시킨 후 user2의 창>

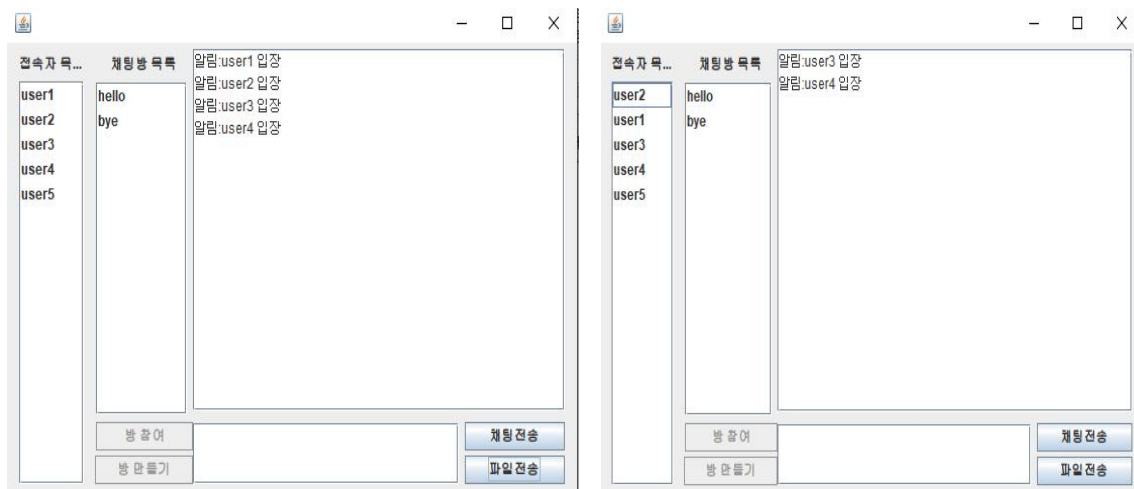
클라이언트가 방이 이미 만들어진 후 뒤늦게 접속해도 이미 존재하는 채팅방을 볼 수 있다.
user5를 만들어 확인하면 아래와 같다.



이미 존재하는 채팅방의 이름과 같은 방을 만들면 오류가 뜬다.



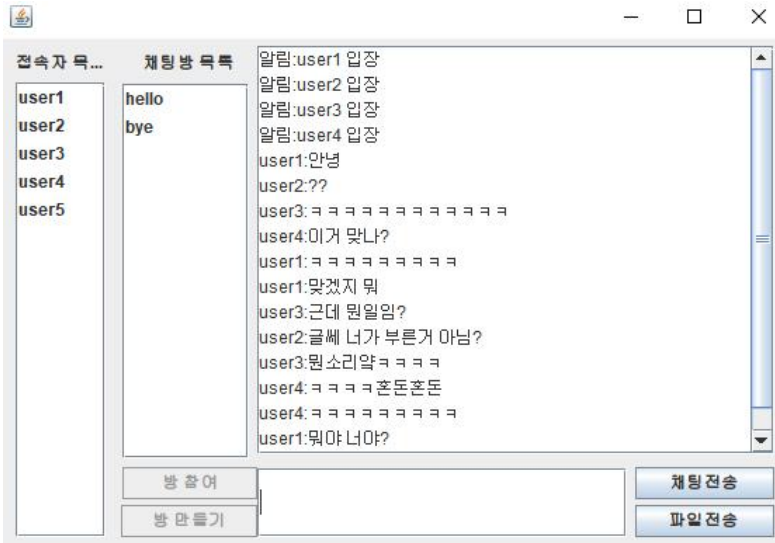
채팅방은 한 포트에 여러개일 수 있다. 예를 들어 'bye'채팅방을 만들 수 있고 같은 포트에 접속해 있는 모든 클라이언트는 새로 생긴 채팅방을 볼 수 있다.



<bye방을 만든 후 user1의 창>

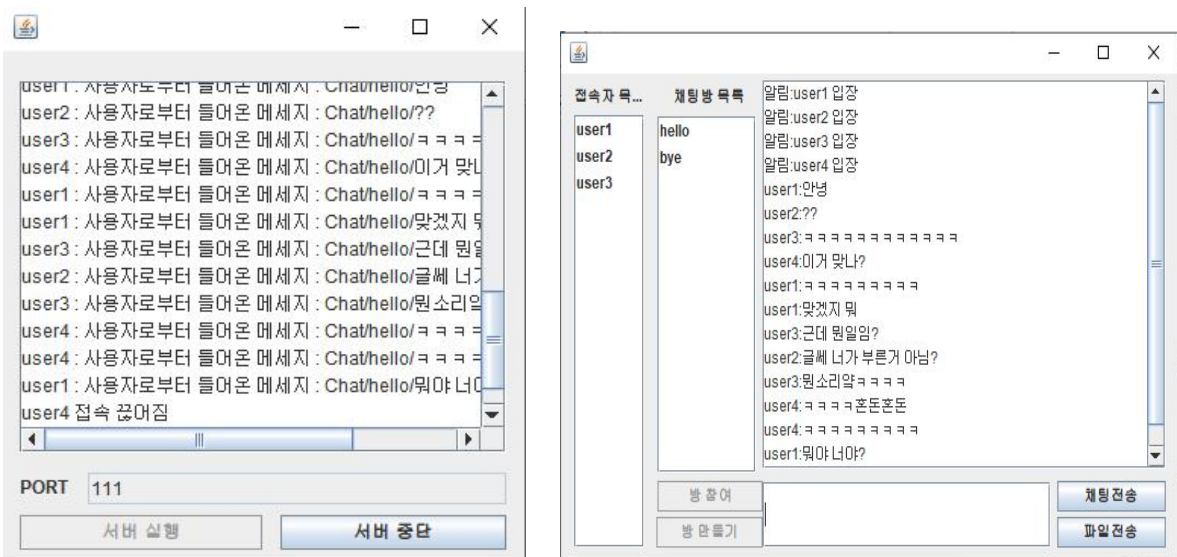
<bye방을 만든 후 user2의 창>

같은 방에 있는 사람들끼리 채팅을 할 수 있다.



채팅방 중간에 들어온 사람은 앞의 내용을 볼 수 없다.

이 때 갑자기 user4가 창의 X버튼을 눌러 강제로 접속을 중지하면 서버가 인지한다. 클라이언트는 접속자 목록에서 없어진 user4를 볼 수 있다.



<파일 전송>버튼을 클릭하면 전송할 파일을 선택할 수 있다. 한 클라이언트가 파일을 전송하면 이 파일은 Client.java파일이 있는 폴더에 저장된다.

server측의 <서버 중단>버튼을 클릭하면 모든 클라이언트는 채팅을 할 수 없게 된다. 그리고 서버와 연결이 끊어졌다는 알림창을 받게 된다.