네트워크 학기말 과제

<JAVA 소켓으로 채팅 프로그램 구현하기>

2015143505 강병헌

2015171008 장서연

소켓 프로그래밍을 통해 클라이언트 간 text를 주고받을 수 있는 프로그램을 JAVA의 Swing을 이용해서 만들었습니다.

클라이언트는 서버로 글을 전송하는 역할을 하며, 서버는 특정 다수의 클라이언트를 연결하고 서로 대화를 할 수 있도록 채팅 내용을 각각 클라이언트에 모두 전달합니다.

각각의 행동은 모두 소켓을 통해서 서버로 전송되게 되어 있습니다. 따라서 특정한 행동을 할 경우 특정 info값이 전달되어 서버는 그 것을 감지해 각기 다른 행동을 할 수 있게 됩니다.

1. 서버

1. Server.java

Main 메소드가 있는 java 파일입니다. 주로 서버의 기능을 실행할 수 있는 버튼들을 생성하고 버튼에 할당된 event listener를 통해 서버의 기능을 수행합니다.

sendMsg()

운영자가 메시지를 유저들에게 전송할 때 특정한 event 상황에 맞춰서 상황에 맞는 메시지를 운영자에게 보여줍니다. 주로 ServerRun.java의 메소드를 사용해서 만든 user의 소켓 리스트를 이용해서 특정상황을 검출합니다.

getPort()

운영자가 사용하고 있는 IP의 port 번호를 등록합니다. Start 라는 메소드를 실행시키는 버튼을 누르면 동작합니다. Port 번호가 비어 있거나 운영자의 이름이 없으면 알람 메시지를 보여줍니다.

Window 리스너

소켓에서 메시지를 받아와 메시지들을 출력해주는 메소드를 실행시킵니다.

1. ServerRun.java

서버를 구동 시키는 run 메소드가 있는 java 파일입니다. 소켓통신의 핵심이 되는 클래스라 할 수 있습니다.

run( ) {}

서버가 지정한 port번호를 받아온 후에 받은 포트 번호로 서버 소켓을 만듭니다.   
별다른 일이 없는 이상 flag는 true로 유지되며 서버가 열어놓은 소켓에 연결하려는 client를 감지하고 클라이언트 소켓을 생성 그 후 해당 스레드를 실행하는 일을 반복 수행합니다.   
클라이언트당 스레드는 1:1로 매칭되어 돌아갑니다.

1. ServerRecord.java

클라이언트들을 관리하는 클래스가 있는 파일입니다. 각각의 클라이언트 소켓이 보낸 메시지를 분석해서 어떤 종류의 요청인지 알아내고 이에 해당하는 내용을 콘솔로 출력해서 다른 클라이언트들에게 전송합니다.

Run()

BufferedReader를 통해서 클라이언트 소켓의 인풋 스트림을 사용하기 편하게 변경하고 데이터를 받아옵니다.   
클라이언트가 전송하는 String은 구분자가 “,,” 로 되어 있습니다.

해당 소켓의 메시지를 분별하기 위해서 info항목을 사용합니다.   
클라이언트의 행동에 따라서 소켓으로 오는 message의 info항목이 변하게 됩니다.   
1은 일반적인 메시지 전송 2는 입장 버튼 클릭 메시지, 3은 종료 메시지를 나타냅니다.

메시지 전송에는 ServerEnter라는 객체를 쓰며 이 객체를 통해 전달한 userList를 통해서 모든 client 소켓으로 메시지를 전송합니다.   
userList에는 이전부터 받아온 소켓들이 담겨 있습니다.

1. ServerEnter.java

서버에서 메시지를 클라이언트들에게 전파할 때 사용하는 클래스입니다. ServerEnter로 파라미터들이 들어오게 되면 userList에 담긴 모든 소켓에게 메시지를 보냅니다.

2. 클라이언트

1. Client.java

Main 메소드가 있는 java 파일입니다. 서버의 메인과 마찬가지로 여러 swing 버튼에 따른 정해진 동작을 하도록 설계되어 있습니다. 동작의 검출은 eventListner를 통해서 검출합니다. Event는 myEvent라는 곳에 모여 있습니다.

우선 ID와 입장하고 싶은 IP의 포트번호를 입력해서 입장버튼을 클릭하게 되면 클라이언트 파일에서 설정해둔 IP를 이용 InterSocketAddress에 서버의 IP와 입력한 port번호를 할당합니다.

할당과정에서 connect의 timeout 파라미터를 이용 해당하는 ip나 port를 5초동안 발견하지 못하면 exception을 발생시킵니다. 이후 클라이언트의 전송을 담당하는 클래스에 id와 등록 정보를 할당하고 서버로 알려줍니다. 이후 클라이언트도 수신 스레드를 열어줍니다.

나머지 버튼들은 sendMsg()에 지정된 조건들로 예상하지 못한 행동들을 검출해 사용자가 올바른 행동을 하도록 알람 메시지를 보여줍니다.

1. ClientReceive.java

클라이언트 자바파일에서 만들어준 client의 receive 스레드를 통해서 메시지를 읽어옵니다.

BufferedReader를 통해서 메시지를 관리하기 편하도록 만들어 줍니다. 그리고 메시지에 포함된 정보를 이용해서 각각에 해당하는 메시지를 콘솔창에 띄웁니다. 일반적인 대화 메시지는 1, 다른 클라이언트들의 입장이나 퇴장은 2,3 아이디가 이미 올라가 있는 건 4 입장 중인데 다시 입장하려는 경우는 5번을 사용합니다.

1. ClientSend.java

열려 있는 서버의 소켓으로 메시지를 보냅니다. 메시지는 어떤 actionListener(버튼)인지에 따라서 info값이 변경되어 전송됩니다. 소켓과 메시지 info값 클라이언트의 이름이 담겨서 전송됩니다.

3. 비정상연결 대처