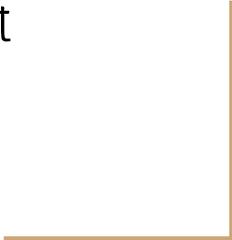




Chat server and client

Master and servant



Simple server

Listen for connections. Start a thread for each connection that reads input from a client and writes it to all other clients.

The server needs a list of connections so that it can loop through them when broadcasting message to all of them.

When a client disconnects, the server should remove the connection (and make sure it's closed).

Simple server example

```
public class Server{
    //Inner class Runnable
    class ClientThread implements Runnable{
        //constructor that takes a socket and a server...
        public ClientThread(.....)
        // run()-method that loops until client disconnect or quits
        // get the inputstream from the socket, and read lines
        // print lines to the server for broadcast
        // if client disconnects, ask server to remove connection
    }

    // Server needs to listen forever and create client threads
    // also, it needs a broadcast method and a remove connection method
    // from its internal list of connections
    // broadcast is to loop through connections and write message from a client
    // to all of them
}
```

Simple client

A GUI with a textarea and an input field

A thread that reads broadcasts from the server and appends them to the textarea

Textfield needs an actionlistener that sends the clients message to the server

When started, the client connects to the server

Simple client example

```
class Window implements Runnable{
    // run method reads from server in the background and appends to the textarea
    // constructor (takes a network object that holds the connection)
    // the textfield reads input from user and writes to server using Jchat's getSocket()
    // initComponents()
    // layout()
    // show()
}

class Jchat{ // responsible of the connection
    getSocket() // returns the socket
    connect() // connects to the server
    public Jchat() // Constructor // creates and starts the window/thread
}

// Main:
create a new Jchat (which creates the window/thread)
```