Nel momento in cui il Client vuole connettersi, istanzia una ClientSocket, che esegue il metodo connect (di Socket) passando una porta e un ip.
Nel frattempo il Server avrà creato una SocketServer che è un Thread che si mette in attesa di ricevere una connessione attraverso il metodo accept().

Una volta effettuata la connessione, il metodo accept() ritorna una Socket che viene usata per creare il ClientHandler (anch'esso un Thread) e viene chiamato il metodo addClient del Server.

Quando il Client vuole connettersi a una Lobby viene invocato il metodo Login che manda un messaggio con sendMessage. Il messaggio viene ricevuto dal ClientHandler che attraverso l'onMessageReceived dell'interfaccia MessageReceiver, dopo aver controllato l'univocità del nickname chiede al Server di far connettere il client a una lobby esistente o, nel caso non esistesse, di crearne una.

I messaggi sono rappresentati da una classe "Message" astratta (o Interfaccia); ogni sottotipo di messaggio utilizza una sottoclasse che implementa la classe astratta.

La logica della Lobby verrà espansa più avanti una volta che avremo più chiaro come sarà strutturato il package controller.

In allegato troverete un UML esplicativo e il formato dei messaggi.

N.B.: Nell'UML il package dei messaggi per ora è solo di Mockup si veda il file dei messaggi