Comunicazione tra Server e Client

<Agostino Contemi>, <Federico Consorte>, <Filippo Dodi>, <Jacopo Corsi>
Gruppo <AM04>

1 Introduzione

La gestione della comunicazione dal lato Server avviene attraverso la classe GeneralServerConnectionHandler, che gira su un thread a parte e si occupa di gestire tutte le connessioni. Il GeneralServerConnectionHandler contiene un ServerConnectionHandlerRMI e un ServerConnectionHandlerSocket, incaricati ciascuno di gestire tutte le connessioni della tipologia corrispondente. Per ogni connessione socket il ServerConnectionHandlerSocket associa un ClientHandler che comunicherà al client attraverso i propri ServerSender e ServerReceiver, i quali girano ognuno sul corrispettivo thread. Il ServerSender si occuperà dell'invio dei messaggi da parte del Controller, il quale passa prima attraverso il ServerConnectionHandler e poi il ClientHandler. Il ServerReceiver invece ha un riferimento al Controller e usa i metodi pubblici di quest' ultimo per comunicare i messaggi.

Per ogni connessione RMI, invece, il ServerConnectionHandlerRMI associa direttamente il ClientConnectionHandler corrispondente, di cui chiamerà le funzioni di esecuzione messaggi.

Dal lato Client viene utilizzato lo stesso procedimento ma ClientConnectionHandler crea direttamente il ClientSender e ClientReceiver.

2 Socket

Il ServerConnectionHandlerSocket resta in ascolto della connessione di nuovi client, effettuando il binding e poi istanziando ClientHandler, che si occuperà di tutte le successive comunicazioni in ingresso da tale client. Il ServerReceiver è in continuo ascolto sulla socket,
mentre il ServerSender inoltra le informazioni dal client al controller.
Viene utilizzato lo stesso procedimento per il client.

3 RMI

Il ServerConnectionHandlerRMI e il ClientConnectionHandlerRMI, una volta effettuato il binding e una volta connessi fra di loro, si possono chiamare i metodi a vicenda per l'esecuzione di messaggi. Inoltre, ogni 5 secondi chiamano uno la funzione ping dell'altro, per verificare la presenza continua dell'altro nella rete e gestirne la disconnessione nel caso non sia più connesso. Il ping, inoltre, viene effettuato prima dell'invio di ciascun messaggio.

4 Messaggi

Per lo scambio di pacchetti tra server e client vengono usate le classi Message, tutte in formato serializable, che implementano le interfacce ToClientMessage e ToServerMessage, a seconda che il messaggio sia diretto al client, diretto al server, o a entrambi: ToClientMessage contiene il metodo clientExecute, usato dal client per eseguire il messaggio; ToServerMessage contiene invece serverExecute, usato dal server per eseguire il messaggio. Nei casi in cui il messaggio implementi entrambe le interfacce, vi sono due costruttori: uno viene

utilizzato per creare un nuovo messaggio diretto al server e uno al client.