

Classi non raffigurate all’interno dello schema UML

* ChatMultiThread: Launcher della classe SchermataClient
* ChatMultiThreadServer: Launcher del server

I metodi **sottolineati** si occupano della costruzione del socket e dello scambio vero e proprio dei messaggi.

Parte Client :

Dopo che **ChatMultiThread** istanzia la classe **SchermataClient** essa inizializza la classe **Client** trascrivendo dunque il suo nome e le sue informazioni personali descritte dall’utente. Nella sua inizializzazione il client si occuperà di costruire la classe **Utente** e instanziare la classe **ThreadClient** che si occuperà della ricezione dei messaggi.

Parte Server :

Il **ChatMultiThreadServer** inizializza la classe **Server** che resterà quindi in attesa delle richieste di connessione da parte dei client. Una volta determinata la connessione e salvato il socket all’interno di un’hashmap avente come chiavi i nomi utente dei client, la classe server principale si occuperà quindi di instanziare una classe **Cliente** per ogni singolo client ai fini di restare in ascolto dei messaggi in arrivo.

Comunicazione :

Lo scambio di messaggi tra client è determinato dal punto di connessione principale ossia il server: il mittente invia il messaggio al server che lo analizzerà e lo rinvierà ai destinatari, quindi, destinatari e mittenti si alterneranno tra loro. Se un messaggio rappresenta un comando all’ora dovrà presentare il carattere “@” iniziale. I comandi disponibili sono: **@ext** per uscire **@ls** per mostrare tutti i partecipanti e **@<nomeutente> <messaggio>** per inviare un messaggio in privato all’utente indicato.