

Programmazione di Sistemi in Rete (A.A. 2017/18)

Esercitazione n. 6 del 27 ottobre 2017

Prof. Eugenio Zimeo

Esercizio 6.1

Realizzare un'applicazione secondo il modello Producer/Consumer in cui un produttore scrive degli interi su un oggetto di tipo `BoundedBuffer` (realizzato come da lezione 13) e un consumatore estrae i valori per visualizzarli sullo standard output. Si associi sia il produttore che il consumatore ad un gruppo di thread. Si sospenda l'esecuzione del main in attesa di ricevere un comando di interruzione da standard input (es. "Posso terminare l'applicazione ? si/no"). Quando l'utente decide di terminare l'applicazione, digita il comando atteso e il metodo main invia un segnale di interruzione al gruppo per interrompere i due thread. Si preveda un messaggio sullo standard output quando un thread è in fase di terminazione (es. Il thread Consumer sta per terminare).

Esercizio 6.2

Scrivere un'applicazione Java di "instant messaging" multi utente. L'applicazione deve prevedere un server in grado di servire diversi processi client. Ogni client può inviare messaggi (stringhe di caratteri) a tutti gli altri clienti connessi al server. La comunicazione dovrà essere mediata dal server: ogni volta che un client invia una stringa, questa viene ricevuta dal server e inoltrata a tutti i client che in quel momento sono connessi al server. Allo scopo, il server dovrà prevedere un'attività (`IMServer`) dedicata alla ricezione delle connessioni da parte dei client e un'attività (`IMServerHandler`), per ogni client che ha richiesto di partecipare alla conversazione multi-utente attraverso l'attivazione di una connessione, dedicata alla conversazione.

`IMServer` e `IMServerHandler` condivideranno un oggetto di tipo `Room` usato per gestire le connessioni correntemente attive. Per l'implementazione di `Room` si può utilizzare `ArrayList` o `LinkedList`.

Per la realizzazione del client si può procedere in modo simile a quanto fatto per l'esercizio 5.3. In alternativa, si può provare ad usare l'applicazione telnet.

Completare l'applicazione con la gestione dell'uscita dalla room da parte di ciascun client. Un client termina la propria conversazione e abbandona la room quando digita il carattere '.'