

Esercizio 1. Programmazione concorrente

Si consideri un programma multi-thread in cui i thread rappresentano i giocatori di una partita.

Il gioco consiste in una sequenza di mani.

In ciascuna mano i giocatori devono fare la loro mossa in un certo ordine, che può essere diverso in ciascuna mano. L'ordine di gioco è deciso all'inizio di ogni mano in base allo stato della partita.

Programmare il sistema, in modo che i thread giocatori facciano la propria mossa nell'ordine corretto. A questo scopo, occorre che un giocatore

1. aspetti il proprio turno
2. legga la situazione del gioco
3. decida quale mossa fare
4. faccia la propria mossa.

NB: è essenziale che il giocatore faccia la mossa conoscendo la situazione del gioco risultante dalle mosse dei giocatori che muovono prima di lui.

Si modifichi il codice dato, in modo che ciascun giocatore faccia la propria mossa nell'ordine corretto.

Come sempre, occorre evitare i problemi tipici della programmazione concorrente (corse, critiche, deadlock, starvation, ecc.).

Si ricorda che bisogna caricare i file .java, NON i .class, meglio se raggruppati in un unico file zip.

Esercizio 2. Programmazione distribuita / socket

Si consideri il problema di programmazione concorrente dato.

Si chiede di trasformare il programma in un sistema distribuito, in cui i client operano come giocatori e il server gestisce la partita.

Si realizzi il sistema mediante l'uso di socket.

Come sempre, occorre evitare i problemi tipici della programmazione concorrente (corse, critiche, deadlock, starvation, ecc.).

Si ricorda che bisogna caricare i file .java, NON i .class, meglio se raggruppati in un unico file zip.

Esercizio 3. Programmazione distribuita / RMI

Si consideri il problema di programmazione concorrente dato.

Si chiede di trasformare il programma in un sistema distribuito, in cui i client operano come giocatori e il server gestisce la partita.

Si realizzi il sistema mediante l'uso di RMI.

Come sempre, occorre evitare i problemi tipici della programmazione concorrente (corse, critiche, deadlock, starvation, ecc.).

Si ricorda che bisogna caricare i file .java, NON i .class, meglio se raggruppati in un unico file zip.