

Commento ai problemi e alle possibili soluzioni

Esercizio 1

Come scritto esplicitamente nel testo del problema, occorre assicurarsi che 1) la lettura del punteggio avvenga dopo che tutti i giocatori hanno giocato, e 2) che una nuova mano cominci solo quando tutti i giocatori sono pronti, avendo letto l'esito della mano precedente. L'unica difficoltà del problema dato consiste nel garantire questi due punti di sincronizzazione.

Bisogna quindi fare in modo che

- 1) le giocate avvengano solo dopo che tutti i giocatori sono pronti a giocare
- 2) le letture dei punteggi avvengano solo dopo che tutti hanno giocato.

Pertanto, il problema si presta a essere risolto mediante l'uso di **CyclicBarrier**. Essendoci due punti di sincronizzazione, occorrono due barriere. Avendo fatto questa osservazione, l'implementazione è piuttosto banale, come illustrato nel codice in `Es1_soluzione_semplice.zip`.

La soluzione prospettata fa sì che siano i giocatori a dover usare opportunamente le barriere date. Quindi sono i giocatori responsabili della sincronizzazione, mentre la classe **Partita** non ha bisogno di essere modificata, perché il metodo `giocata` viene chiamato solo quando tutti sono pronti a giocare, mentre il metodo `puntiMano` viene chiamato solo dopo che tutti hanno giocato.

Se non si vuole che i giocatori siano responsabili della sincronizzazione si può modificare opportunamente la classe **Partita**, in modo che i metodi `giocata` e `puntiMano` risultino sospensivi: se chiamati quando non ci sono le condizioni (ad es. un thread cerca di leggere il punteggio di una mano ancora in corso) il thread chiamante viene sospeso fino a quando le condizioni non sono verificate. Questa soluzione è illustrata nel codice del file `Es1_soluzione.zip`.

Esercizio 2.

In questo caso l'unica difficoltà consiste nel fare in modo che il server faccia le estrazioni periodicamente, e contemporaneamente serva i client. Questo è un problema perché ad esempio, un server che sia in attesa di leggere da un client non può effettuare l'estrazione.

Ci sono diversi possibili modi per fare in modo che le estrazioni avvengano periodicamente, indipendentemente da quello che il server sta facendo con i client. Nella soluzione proposta il server crea un thread dedicato ad effettuare le estrazioni. Naturalmente il server master, gli slave e il thread dedicato alle estrazioni condividono tutti l'istanza del gioco.

Esercizio 3.

La difficoltà illustrata per l'esercizio 2 si supera in modo banale con RMI (come illustrato nel codice di `Es3A_soluzione.zip`. Il main del server, una volta pubblicato il riferimento remoto nel registry mediante una `rebind` esegue il ciclo delle estrazioni.

La domanda B richiedeva solo di implementare il solito meccanismo della callback.