

Progetto Laboratorio di Programmazione Java

Modulo B

Esercizio 1:

Si consideri un piccolo Aeroporto, nella quale è presente una sola pista, utilizzata sia per i decolli che per gli atterraggi. Un addetto al servizio dell'aeroporto, si occupa di far partire e decollare gli aerei una volta che la pista è piena.

La pista può accettare contemporaneamente UN aereo pronto al decollo ed UN aereo pronto all'atterraggio. In questo caso la pista si considera piena e l'addetto la libera dando il via agli aerei.

Gli aerei sbloccati attendono un certo periodo random ed eseguono nuovamente la richiesta, stavolta inversa. Quindi se erano precedentemente partiti, adesso chiedono di atterrare.

Gli aerei in attesa di partire o di atterrare, controllano se il posto sulla pista è libero, nel caso, la occupano con la propria richiesta ed attendono che l'addetto dia il via libera per partire.

Modellare il funzionamento del piccolo aeroporto considerando i seguenti punti:

- la pista può essere contemporaneamente condivisa da un aereo in partenza ed uno in atterraggio;
- l'addetto al servizio attende che ci sia 1 aereo pronto per il decollo ed uno per l'atterraggio e in tal caso dà il via ai due aerei e si rimette in attesa.
- Un aereo che richiede di atterrare, controlla se c'è posto sulla pista, altrimenti si mette in attesa. Stesso comportamento per il decollo.
- prevedere il comportamento degli aerei e dell'addetto all'interno di un ciclo infinito.
- Atterraggio: la pista è allocata ad un unico aereo che deve

Realizzare un'applicazione concorrente in Java che, rappresentando aerei e l'addetto mediante thread concorrenti, implementi la politica di sincronizzazione per l'accesso alla pista.

Suggerimento:

- prevedere un numero random di aerei generato all'interno del Main, che non sia superiore a 10, per semplicità di controllo.
- commentare opportunamente il codice e le scelte effettuate.

Esercizio 2:

Scrivere un'applicazione Client/Server che simuli un gioco. Il gioco è un multithread, quindi il server può gestire più giochi contemporaneamente. Il client si connette al server e deve provare ad indovinare un numero compreso tra 1 e 20. Il server, una volta che il client si è connesso, avvia un thread per gestire quella specifica connessione. Seguire i seguenti passi per lo sviluppo del gioco:

- il client genera un numero random compreso tra 1 e 20;
- il client invia il numero al server e attende la risposta;
- il server avvia un thread per gestire ogni singola connessione di gioco.
- il thread gestisce il messaggio in arrivo dal client e controlla se il numero inviato corrisponde ad un numero generato random dal server (sempre compreso tra 1 e 20) una volta avviato il gioco.
- Se il numero corrisponde, invia la risposta OK e il client elaborerà il messaggio lato client visualizzando la scritta "Bravo hai indovinato, il numero era X"
- Se il numero non corrisponde, il server scrive A se il numero da indovinare è più grande, oppure B se il numero da indovinare è minore.
- il client continua ad inviare i numeri fino a quando non indovina, quindi se riceve A o B deve continuare a chiedere l'input all'utente.

Commentare opportunamente il codice per rendere comprensibili le scelte effettuate.