

Esercizi consigliati sulla sincronizzazione fra thread:

1) Un gruppo di 50 turisti si recano a visitare il Louvre per vedere la Gioconda. Per motivi di sicurezza possono entrare nel museo solo 5 persone alla volta.

Simulare la suddetta situazione in Java, utilizzando un Monitor (costruito con `synchronized` di Java). Ogni turista sarà rappresentato da un Thread; i turisti attenderanno un tempo casuale, quindi cercheranno di entrare nel museo, di vedere il quadro e alla fine usciranno dal museo.

Si deve utilizzare un Monitor per gestire i vincoli sulla mutua esclusione e sugli accessi limitati.

2) Un gruppo di 20 bevitori ha a disposizione una botte di vino con 3 rubinetti, contenente 100 litri di vino. Di conseguenza possono bere solo tre persone alla volta, sempre che la botte contenga ancora vino. Si suppone per semplicità che ogni bevuta faccia diminuire il vino nella botte di 1 litro.

Si simuli la suddetta situazione in Java, utilizzando un Monitor (costruito con `synchronized` di Java).

Suggerimento: Il thread bevitore, dopo aver atteso un tempo casuale, deciderà di bere e si metterà in attesa se il numero di rubinetti liberi è 0; quindi controllerà la variabile contenente il vino. Qualora non ci sia più vino il thread terminerà la sua esecuzione).

3) Un gruppo di 200 persone deve comprare i biglietti per il concerto di Sting. Al botteghino sono a disposizione solo 100 biglietti e vi sono solo 5 persone che li vendono. Perciò solo 5 persone alla volta potranno comprare i biglietti e solo se ve ne sono ancora a disposizione..

Si simuli la suddetta situazione in Java, utilizzando un Monitor (costruito con `synchronized` di Java). Ogni compratore sarà rappresentato da un thread..

(Suggerimento: Il thread compratore, dopo aver atteso un tempo casuale, cercherà di comprare il biglietto e si metterà in attesa se altre 5 persone sono già al botteghino; quindi controllerà la variabile contenente il numero di biglietti e la decrementerà).

4) Otto amici decidono di fare una gara di velocità. Scelgono quindi di andare a correre sulla pista di go-kart che si trova a pochi chilometri di distanza dalla loro città. Il gestore della pista, per questioni di sicurezza, ha stabilito che possono correre solo 4 kart per volta e che ciascun pilota può effettuare al più 15 giri. Inoltre prima di salire sul kart il pilota deve indossare una tuta protettiva ed un casco. L'accesso agli spogliatoi per indossare la tuta è consentito a 2 persone per volta.

Scrivere un programma che simuli la gara di velocità rispettando i vincoli imposti dal gestore della pista. Ciascun pilota, rappresentato da un thread, intraprende le seguenti azioni:

- entra negli spogliatoi per indossare la tuta ed il casco (se non c'è posto attende)
- sale sul kart e entra in pista (se in pista ci sono già 4 kart attende che uno dei piloti termini i giri a disposizione)
- effettua 15 giri di pista
- lascia la pista
- torna negli spogliatoi per indossare i propri indumenti

Utilizzare il costrutto `synchronized` per gestire l'accesso concorrente alle risorse condivise.