

## CLASSE JAVA THREAD-SAFE PER LA GESTIONE DI EVENTI

Consideriamo un sistema di gestione di eventi (es concerti, conferenze, ecc).

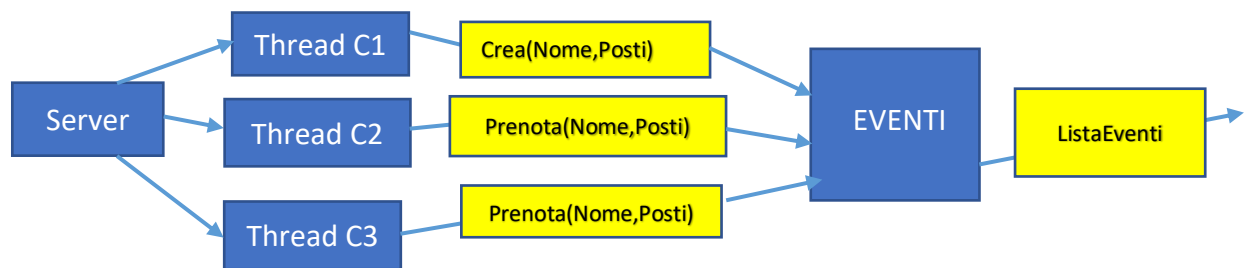
Si vuole definire una classe Java per garantire l'accesso thread-safe ad una classe Java con metodi per gestire eventi e posti disponibili.

In particolare utilizzando i monitor di sincronizzazione di Java, la classe EVENTI deve fornire

- un costruttore per inizializzare le proprie istanze
- un metodo "Crea(Nome,Posti)" per aggiungere un nuovo evento e i relativi posti disponibili solo se non esiste già un evento con lo stesso nome.
- un metodo "Aggiungi(Nome,Posti)" per aggiungere nuovi posti ad un determinato evento
- un metodo "Prenota(Nome,Posti)" per prenotare posti per un dato evento, il metodo deve essere bloccante se non ci sono abbastanza posti
- un metodo "ListaEventi" per visualizzare su console eventi e posti ancora disponibili
- un metodo "Chiudi(Nome)" che cancella l'evento e sblocca tutti i clienti in attesa di posti

Per clienti intende i thread che cercano di accedere a quel evento

Assumiamo che ogni richiesta relativa ad eventi sia eseguita da thread (es. un handler di gestione di richieste che arrivano ad un server di qualche tipo) e che diversi utenti possano richiedere di inserire eventi o prenotare posti anche simultaneamente.



Una volta definita la classe implementate un programma di test per simulare la creazione di alcuni eventi e un certo numero di richieste concorrenti effettuare sulla stessa istanza di EVENTI da thread di due tipi: ADMIN e UTENTE.

- Un thread ADMIN esegue la sequenza Crea, pausa, Aggiungi, pausa, Chiudi per diversi nomi di eventi.
- Un thread di tipo UTENTE invia richieste di prenotazione

Consegnate il progetto Java in un file zip nomegruppo\_foglio2.zip