

Soluzione per la Simulazione della Pizzeria con Thread in Java

Obiettivi del Progetto

L'obiettivo è simulare una pizzeria con le seguenti caratteristiche:

1. Un forno produce da 1 a 8 pizze alla volta.
2. Tre camerieri possono prendere ordini e servire le pizze ai tavoli.
3. La pizzeria ha 20 tavoli e una capienza massima di 100 persone.
4. La simulazione termina quando tutte le pizze sono state servite.
5. La sincronizzazione tra i thread avviene utilizzando esclusivamente `synchronized`, `wait()`, `notify()` e `notifyAll()`.

Scelte Progettuali

- **Forno:** Un thread che simula la produzione di pizze. Le pizze vengono aggiunte a una lista condivisa e i camerieri vengono notificati quando sono pronte.
- **Cameriere:** Thread che simulano i camerieri, i quali prelevano pizze dal forno e le servono ai tavoli.
- **Tavolo:** Classe che rappresenta i tavoli della pizzeria e il numero di persone sedute.
- **Ordine:** Rappresenta gli ordini dei clienti, utilizzata dai camerieri per prelevare pizze dal forno.

Sincronizzazione e Gestione dei Thread

La sincronizzazione tra i thread viene gestita utilizzando i seguenti meccanismi:

1. Il Forno produce le pizze e le aggiunge a una lista condivisa. I camerieri attendono la disponibilità delle pizze tramite l'uso di `wait()`. Una volta prodotte le pizze, il forno invoca `notifyAll()` per risvegliare i camerieri.
2. I Camerieri controllano se ci sono pizze disponibili e, se presenti, prelevano una pizza dalla lista.
3. Per evitare condizioni di gara, ogni accesso alle risorse condivise (pizze e tavoli) è protetto mediante blocchi `synchronized`.