Sincronização em Sistemas Multithreading - Organização de Atributos Independentes em Classes Sincronizadas

Objetivo do Projeto

Este documento apresenta os testes realizados em dois cenários distintos para controle de concorrência na classe Cinema, que gerencia a venda e devolução de ingressos para duas salas de cinema. O objetivo é comparar o impacto da sincronização global da classe com a sincronização específica por atributo usando Locks Os dois cenários analisados são: Sincronização Global da Classe - Utiliza a palavra-chave synchronized em todos os métodos. Sincronização Específica por Atributo - Utiliza ReentrantLock para cada sala de cinema separadamente.

🗒️ Cenários teste

Cenário 1: Sincronização Global da Classe

Descrição

Neste cenário, todos os métodos da classe Cinema são sincronizados globalmente, garantindo que apenas uma thread possa acessar qualquer método por vez. Isso previne condições de corrida, mas pode reduzir o desempenho devido ao bloqueio de threads concorrentes.

Comportamento Esperado

Apenas uma thread pode acessar qualquer método da classe por vez. Condições de corrida são evitadas. Pode ocorrer um gargalo de desempenho, pois threads que poderiam operar em cinemas diferentes são bloqueadas.

Resultados Obtidos

O sistema manteve consistência nos dados. O desempenho foi afetado devido ao bloqueio global, reduzindo a paralelização.

=======

Cenário 2: Sincronização Específica por Atributo (Usando Locks)

Descrição

Neste cenário, cada atributo (vacanciesCinema1 e vacanciesCinema2) tem seu próprio Lock, permitindo que threads operando em cinemas diferentes possam executar em paralelo, melhorando o desempenho sem comprometer a consistência dos dados.

Comportamento Esperado

Threads que operam em cinemas diferentes podem executar simultaneamente. Redução de bloqueios desnecessários, melhorando o desempenho. Manutenção da consistência dos dados.

Resultados Obtidos

O sistema manteve consistência nos dados. O desempenho melhorou significativamente devido à paralelização de threads.

🔧 Conclusão:

🔗

A escolha da melhor abordagem depende dos requisitos do sistema. Para aplicações de alto desempenho com concorrência elevada, a sincronização específica por atributo é a melhor opção.

💻 Desenvolvedor

🔗

Marcio Fonseca
👨‍💻

=======