
Coordenação de Threads com Phaser - Controle da Mudança de Fase em Tarefas Concorrentes



STATUS

DESENVOLVIDA

Objetivo do Projeto



Este projeto demonstra o uso da classe Phaser do Java para coordenar a execução de threads em fases distintas. A aplicação simula um exame realizado por estudantes, onde cada um deve completar três exercícios antes de finalizar o exame. O Phaser garante que todas as threads sincronizem entre as fases antes de prosseguir.

Estrutura do Código

O projeto é composto por três classes principais:

MyPhaser: Estende Phaser para personalizar o comportamento em cada fase do exame. Student: Implementa Runnable para simular a execução dos estudantes em cada fase. Main: Inicializa o Phaser, registra os estudantes e inicia as threads.

Fluxo de Execução

Os estudantes são registrados no Phaser e chegam para fazer o exame. O exame é dividido em três fases: primeiro exercício, segundo exercício e terceiro exercício. Em cada fase, os estudantes concluem o exercício e aguardam a sincronização com os demais. O Phaser finaliza sua execução após a conclusão do exame.



Cenários teste



Foram realizados três cenários de testes para validar o funcionamento do Phaser:

Cenário 1: Execução Normal

Descrição: Todos os 5 estudantes completam o exame em fases.

Saída Esperada:

Todos os estudantes chegam e sincronizam na fase 0. Cada fase é concluída após todos terminarem seus exercícios. O Phaser finaliza corretamente.

=====

Cenário 2: Abandono de Estudantes

Descrição: Um ou mais estudantes abandonam o exame antes de terminar.

Saída Esperada:

O número de estudantes registrados no Phaser diminui. As fases são concluídas apenas pelos estudantes restantes.

=====

Cenário 3: Registro Dinâmico

Descrição: Novos estudantes se registram durante a execução do exame.

Saída Esperada:

O número de estudantes registrados no Phaser aumenta dinamicamente. As fases são concluídas com o novo número de participantes.

Conclusão:



Este projeto demonstrou como a classe Phaser pode ser utilizada para coordenar a execução de threads em fases distintas. Os testes realizados confirmam o correto funcionamento da sincronização e abrem possibilidades para futuras melhorias na implementação.

Desenvolvedor



Marcio Fonseca



=====