RelogioArquivo (FileClock) - Dormindo e Retomando uma Thread

Objetivo do Projeto

Este documento descreve os testes realizados no código da classe RelogioArquivo, que implementa a interface Runnable para exibir a data e hora atuais em um loop controlado. Foram explorados três cenários distintos para avaliar o comportamento da thread.

🗒️ Cenários teste

Cenário 1: Interrupção da Thread após 5 Segundos

Objetivo: Avaliar o comportamento da thread ao ser interrompida antes de completar sua execução.

Procedimento:

A thread principal aguarda por 5 segundos antes de chamar interrupt() na thread do RelogioArquivo. A thread do RelogioArquivo verifica a interrupção dentro do bloco catch, imprime uma mensagem e termina a execução.

Resultado esperado:

A thread exibe a data e hora durante 5 segundos e, ao ser interrompida, imprime "O RelogioArquivo foi interrompido" antes de encerrar.

=======

Cenário 2: Aumentar o Tempo de Execução da Thread

Objetivo: Verificar o comportamento da thread quando ela tem tempo suficiente para completar todas as iterações.

Procedimento:

A thread principal aguarda por 15 segundos antes de chamar interrupt(). O loop do RelogioArquivo imprime a data e hora por 10 iterações (10 segundos).

Resultado esperado:

A thread do RelogioArquivo conclui suas 10 iterações normalmente sem ser interrompida.

=======

Cenário 3: Reduzir o Número de Iterações

Objetivo: Observar como a thread se comporta ao encerrar sua execução naturalmente, sem ser interrompida.

Procedimento:

O número de iterações do loop dentro do RelogioArquivo é reduzido para 5. A thread principal aguarda por 10 segundos antes de chamar interrupt().

Resultado esperado:

A thread finaliza naturalmente após 5 segundos, sem necessidade de interrupção.

🔧 Conclusão:

🔗

Os testes demonstraram que a thread do RelogioArquivo responde corretamente à interrupção e pode também concluir sua execução normalmente dependendo das condições estabelecidas. Essas observações ajudam a validar o funcionamento do código em diferentes situações, garantindo que a gestão de threads esteja funcionando conforme esperado.

💻 Desenvolvedor

🔗

Marcio Fonseca
👨‍💻

=======