

Repositório de Exemplos e Relatórios de Programação Concorrente

Este repositório contém diversos exemplos de código e relatórios criados durante o estudo de programação concorrente. Cada exemplo possui um relatório detalhado explicando sua implementação e funcionamento.

Diretório de Exemplos

- **barbeiro**
- **demo05_dormir** - Exemplo de implementação de threads em Java para simular um processo simples de concorrência.
- **demo06_join** - Exemplo de uso de `wait()` e `notify()` para sincronização entre threads.
- **demo07_daemon** - Demonstração de uso de um contador compartilhado entre múltiplas threads.
- **demo08_excecoesNaoControladas** - Simulação de um sistema de produtor-consumidor utilizando `synchronized`.
- **demo09_variaveisLocais** - Exemplo de um sistema com threads e bloqueios explícitos usando `ReentrantLock`.
- **demo10_grupoThreads** - Implementação de um sistema de filas utilizando threads e locks.
- **demo11_tratarExcecoes** - Demonstração de sincronização com `volatile` em Java.
- **demo12_fabricaThreads** - Exemplo de uso de `ExecutorService` para gerenciar pools de threads.
- **demo13_sincronizar** - Implementação de um banco de dados simples com acesso concorrente.
- **demo14_organizar** - Demonstração de uso de `ThreadLocal` para variáveis específicas de thread.
- **demo15_condicoes** - Exemplo de implementação de produtores e consumidores com `wait()` e `notify()`.
- **demo16_syncLock** - Exemplo de fila de impressão usando `ReentrantLock` para controlar o acesso a um recurso compartilhado.
- **demo17_readWriteLock** - Demonstração de uso de `ReadWriteLock` para gerenciar concorrência em leituras e escritas.
- **demo18_modifyLock** - Implementação de bloqueio de modificação em um recurso compartilhado.
- **demo19_multiCondicoes** - Exemplo de uso de múltiplas condições dentro de um `Lock`.
- **demo20_controleRecursos** - Controle de acesso a recursos limitados usando `Semaphore`.
- **demo21_semaforoMultiplasCopias** - Exemplo de controle de múltiplas cópias de um recurso compartilhado.
- **demo22_aguardandoMultiplosEventos** - Demonstração de espera por

múltiplos eventos usando `CountDownLatch`.

- **demo23_syncPontosComuns** - Exemplo de sincronização de pontos comuns em um fluxo de execução.
- **demo24_tarefasEmFases** - Implementação de execução de tarefas em fases usando `CyclicBarrier`.
- **demo25_mudancaDeFasesControle** - Controle de fases de execução com `Phaser`.
- **demo26_mudancaDados** - Troca de dados entre threads usando `Exchanger`.

Instruções de Uso

Para executar os exemplos, siga os passos abaixo:

```
javac Principal.java  
java Principal
```

Cada exemplo possui arquivos específicos e podem ter classes diferentes como ponto de entrada. Verifique o relatório de cada exemplo para mais detalhes sobre sua execução.

Notas Finais

Todos os relatórios foram organizados em pastas específicas. Os exemplos são descritos com o objetivo de simular situações de concorrência e sincronização em sistemas multi-threaded.