

Relatório do Código

1. Enumeração `InterruptType`

Função: Define os tipos de interrupções e suas prioridades.

- **Constantes:** `TIMER`, `IO`, e `SYSTEM_ERROR` com prioridades associadas (respectivamente 3, 2, e 1).
- **Método `getPrioridade`:** Retorna a prioridade do tipo de interrupção.

2. Classe `Interruption`

Função: Representa uma interrupção com tipo e prioridade.

- **Atributo:** `tipo` determina o tipo e a prioridade da interrupção.
 - **Métodos:**
 - `getTipo` e `getPrioridade`: Retornam informações da interrupção.
 - `compareTo`: Compara interrupções com base na prioridade.
 - `toString`: Formata a interrupção como string.
-

3. Classe `InterruptionManager`

Função: Gerencia a fila de interrupções.

- **Gerar interrupções:** Métodos `gerarInterrupcaoTimer`, `gerarInterrupcaoIO`, e `gerarInterrupcaoErroSistema`.
- **Processar interrupções:** Remove e exibe interrupções da fila por prioridade.
- **Verificar status:** Lista todas as interrupções na fila.
- **Formatar interrupção:** Exibe tipo e prioridade de forma amigável.

4. Classe `Main`

A classe `Main` organiza o funcionamento do sistema utilizando threads para simular um ambiente multitarefa.

Principais Componentes:

1. **Fila de prioridade:** Uma `PriorityQueue` é utilizada para armazenar e ordenar as interrupções. A ordenação é baseada no método `compareTo` da classe `Interruption`.
2. **Thread Geradora de Interrupções:**
 - Essa thread simula a geração contínua de interrupções em intervalos de tempo aleatórios (entre 1 e 5 segundos).
 - Em cada ciclo, gera de 1 a 3 interrupções de tipos aleatórios (TIMER, IO ou SYSTEM_ERROR) e as adiciona à fila de prioridade.
 - Permite uma dinâmica de fluxo contínuo de eventos no sistema.
3. **Thread Processadora de Interrupções:**
 - Essa thread processa interrupções da fila a cada 5 segundos.
 - Primeiro, exibe o estado atual da fila, listando as interrupções ordenadas por prioridade.
 - Em seguida, processa e remove cada interrupção da fila até que ela esteja vazia.
4. **Concorrência:** As duas threads (geradora e processadora) executam simultaneamente, criando uma simulação realista de um sistema multitarefa. Isso assegura que novas interrupções podem ser adicionadas à fila enquanto as existentes são processadas.