

# Exercício: O Problema da Montanha Russa (CyclicBarrier)

Suponha que existam  $N$  passageiros e um carro em uma montanha russa. O carro tem capacidade para  $C$  passageiros, com  $C < N$ , e  $N$  sendo múltiplo de  $C$ . Assuma que o carro só pode partir quando estiver completamente cheio. Assuma que quando o carro está cheio ele dá uma volta na montanha russa e estará disponível para receber novos passageiros e continuar dando voltas na montanha russa.

Implemente esse sistema usando o mecanismo de sincronismo CyclicBarrier (barreira cíclica) com a opção de incluir um método a ser chamado automaticamente quando as  $C$  threads dos passageiros tiverem chegado na barreira. Este método automático, que simula a volta na montanha russa, pode ser simplesmente um sleep de 5 segundos. O carro será a thread principal, e os passageiros são outras threads separadas. Inclua um ID para cada thread (pode ser usando o método setName()). Para cada passageiro (thread) imprima que ele chegou na fila da Montanha Russa (está esperando a barreira) e entrou no carro da montanha russa (ultrapassou a barreira). No "método automático" imprima que está iniciando a volta, depois aguarde 5 segundos, e depois imprima que a volta foi completada.

## OBSERVAÇÕES:

- 1) Implemente as  $N$  threads dos passageiros como uma Thread Pool;
- 2) Os valores de  $N$  e  $C$  podem ser constantes (enviados para os construtores);
- 3) Envie somente **um** códigos Java.