Exercício: CountDownLatch e Thread Pools

Implemente um sistema concorrente usando o sincronismo CountDownLatch. Lembrando que nessa estratégia de sincronismo você define um ferrolho (latch), com um certo valor inicial **N**, e que esse latch aguarda que **N** tarefas decrementem o latch até que ele chegue ao valor zero. A partir desse momento, a thread que estava bloqueada (usualmente é a thread principal), e aguardando pelo latch, é liberada para continuar.

Todas as threads receberão como parâmetro do construtor o mesmo latch definido anteriormente. Note que, quando as primeiras **N** threads executarem o método countDown(), para decrementar o contador, o ferrolho se abrirá e a thread principal, que executou o método await(), vai poder prosseguir.

Defina **2*N** threads usando Thread Pool.

No código das threads use o método getCount() da classe **CountDownLatch**, antes e depois da chamada do método **countDown()**, para checar e imprimir se a thread foi ou não responsável por abrir o ferrolho (**latch**).

Na thread principal imprima duas mensagens, uma antes de iniciar o Thread Pool e outra após o retorno do método **await()** do **latch**, o que implica que a thread principal, que estava bloqueada, foi liberada.

Envie somente o código Java que resolve o problema acima.