

Exercício - Threads

1. Faça um programa que leia um número “n” informado pelo usuário e diga quantos números primos há entre 0 e “n”. Esse seu programa deve rodar em duas threads, de forma que o esforço computacional seja dividido entre as threads.
2. Esse exercício demonstra como dois métodos sincronizados têm que se comunicar para manter um contador entre 0 e 3 quando uma thread está tentando repetidamente aumentá-lo e a outra thread tenta repetidamente diminuí-lo. Ele usa o método `Math.random()` que não emprega parâmetros e retorna um valor `double` aleatório entre 0 e 1. Execute todas as etapas a seguir:

- a. Crie uma classe `Counter` com uma variável de instância privada `count` e dois métodos. O primeiro método

```
synchronized void increment()
```

tenta incrementar `count` em 1 unidade. Se `count` já tiver atingido seu máximo, que é 3, ele esperará até `count` ser menor do que 3 antes de incrementá-lo. O outro método

```
synchronized void decrement()
```

tenta diminuir `count` em 1 unidade. Se `count` já tiver atingido seu mínimo, que é 0, ele esperará até `count` ser maior do que 0 antes de decrementá-lo. Sempre que um dos métodos tem que esperar, ele exibe uma declaração dizendo por que está esperando. Além disso, sempre que ocorre um incremento ou decremento, o contador exibe uma declaração informando o que ocorreu e mostra o novo valor de `count`.

- b. Crie uma classe de thread cujo método `run()` chame o método `increment()` de `Counter` 20 vezes. Entre cada chamada, ele entra em suspensão por um período de tempo aleatório entre 0 e 500 milissegundos.
- c. Crie uma classe de thread cujo método `run()` chame o método `decrement()` de `Counter` 20 vezes. Entre cada chamada, ele entra em suspensão por um período de tempo aleatório entre 0 e 500 milissegundos.
- d. Crie uma classe `CounterUser` com um método `main()` que gere um `Counter` e as duas threads e comece a executá-las.

Nota: em vez de criar duas classes de threads, você pode criar apenas uma classe de thread que incremente ou decmente o contador de acordo com um parâmetro passado para seu construtor.