

## Exercícios

**1)** Uma barreira é uma construção para a sincronização de processos que é útil para assegurar que todos os processos envolvidos em uma tarefa atingiram um ponto determinado em suas computações. A barreira é inicializada com um valor inteiro. Se um processo executa `barrier(int b)`, ele é bloqueado na barreira `bar` até que o número de processos igualmente bloqueados na barreira atinja `b`, quando a progressão de todos os processos em espera na barreira é liberada. Programe uma classe `Barrier` que ofereça este tipo de sincronização.

**2)** Seja o problema dos fumadores de cigarro, no qual há três threads fumantes e uma thread agente. Cada fumante continuamente prepara um cigarro e fuma-o. Preparar um cigarro exige três ingredientes: tabaco, papel e um fósforo. Um fumante tem o tabaco, outro tem o papel e o terceiro tem fósforos. Cada um tem um fornecedor infinito de tais ingredientes (ou seja, os ingredientes são infinitos). O agente coloca, de maneira aleatória, dois ingredientes na mesa. O fumante que tem o terceiro ingrediente pega os outros dois que estão na mesa, faz um cigarro e fuma-o. O agente espera até o fumante terminar e repete o ciclo. Dado o código da função do processo agente.

```
semaphore mutex = 0, habilita[3] = 0; // Semaforos
void agente(){
    int vai_fumar;
    while(1){
        vai_fumar = random()%3; // Numeros entre 0 e 2
        printf("%d vai fumar\n",vai_fumar);
        V(habilita[vai_fumar]);
        P(mutex); // Espera que fume
    }
}
```

desenvolva em java o código da função para as threads fumantes cujo protótipo é `void fumante(int id)`.

**3)** Considere um parque de estacionamento com capacidade máxima para `MAX` viaturas. Existem três tipos de utilizadores deste parque: docentes, funcionários e alunos. Enquanto há lugares livres no parque, a ordem de entrada é por ordem de chegada. A partir do momento em que o parque fica cheio, a entrada de viaturas fica condicionada à saída de outras, devendo ser dada prioridade às viaturas dos docentes, funcionários e alunos (por esta ordem). Utilizando semáforos, implemente em Java as funções entrar e sair do estacionamento.

**4)** Pombos correio foram treinados para transportar cartas da cidade A para a cidade B. Na cidade A existe uma Caixa Postal onde as pessoas depositam suas cartas para que sejam entregues na cidade B. Os pombos consultam a Caixa Postal e retiram sempre 5 cartas para transportar na sua viagem. Caso não existam ainda 5 cartas na caixa, o pombo espera até que esse número seja atingido. Na cidade B existe uma gaiola onde os pombos depositam suas cartas e ficam presos até que um processo tratador libere-os. O tratador verifica a gaiola a cada 20 clocks e libera todos os pombos somente se o número de pombos for maior ou igual a 2. Os pombos somente transportam cartas de A para B, na sua volta eles voam vazios. Construa um monitor para controlar a utilização da Caixa Postal pelas pessoas, pombos e o tratador.