

=====

Escrever um programa em JAVA para simular um sistema de reserva de salas usando múltiplas linhas de execução. O conjunto de salas disponíveis deve ter pelo menos 4 salas, que são representadas em um objeto compartilhado pelas linhas de execução. As linhas de execução representam os pontos de acesso ao sistema para fazer reserva e liberação de salas (pelo menos 6). As seguintes especificações devem ser seguidas:

- Utilizar uma classe AulasTeoricas para representar o objeto compartilhado (array de objetos ou arrayList), que tenha, no mínimo campos de dados que representam as salas e os métodos reservar (...) e liberar(...).
- Cada sala deve ser um objeto com os seguintes campos: identificador da sala (int ou String), o tamanho da sala (número de lugares - int), se a sala tem projetor (boolean), o nome do ponto de acesso que fez a reserva (por exemplo, secretaria dos professores, secretaria de pós, etc - String) e se a sala está livre ou ocupada (boolean).
- O método reservar(...) procura uma sala livre (sequencialmente) que atenda os requisitos de tamanho e existência de projetor (parâmetros do método) e, se encontra, faz a reserva. Se não, espera até que uma sala com as condições especificadas seja liberada. A operação de reservar a sala consiste em inicializar o nome do ponto de acesso que reservou a sala e alterar o estado da sala sendo utilizada de livre para ocupada.
- O método liberar(...) libera a sala que estava sendo utilizada. A operação de liberar sala consiste em alterar o estado da sala sendo liberada de ocupada para livre.
- Sincronizar os métodos reservar(...) e liberar(...) para garantir que essas operações não sejam feitas simultaneamente por mais de uma linha de execução.
- Usar os métodos wait() e notify(), dentro do método reservar(...), para fazer o sincronismo de cooperação, de tal forma que uma reserva seja feita somente quando houver uma sala adequada livre e que a conclusão da operação de reserva de sala seja comunicada ao sistema de reservas.
- Usar o método notify() dentro do método liberar(...) para fazer o sincronismo de cooperação, avisando que a sala está liberada.
- Definir uma classe ReservaThread (estendendo a classe Thread ou implementando a interface Runnable) para iniciar uma linha de execução para cada ponto de acesso para reserva.
- Cada ponto de acesso para reserva deve fazer de 2 a 4 pedidos de reserva e liberação salas. Esse número pode ser gerado aleatoriamente ou pode ser definido por atribuição, mas não pode ser o mesmo para todas as linhas de execução.
- A cada pedido de reserva, deve ser chamado o método reservar(...) passando as informações de nome do ponto de acesso, tamanho da sala requisitada e necessidade ou não de projetor. Quando a reserva for feita, esperar algum tempo e chamar o método liberar(...).
- O tempo que a sala fica reservada deve ser gerado aleatoriamente e passado para o método sleep.
- A cada operação de reserva, imprimir uma mensagem indicando qual sala foi reservada e o nome do ponto de acesso que fez a reserva. Ao liberar a sala, imprimir uma mensagem dizendo qual sala está sendo liberada e qual o nome do ponto de acesso que fez a reserva.
- O programa pode parar depois que todos os pontos de acesso tiverem realizado o número de reservas determinado para cada um.

Observações:

- Os trabalhos podem ser feitos em duplas, sendo as mesmas dos trabalhos anteriores;
- As classes e métodos definidos neste enunciado devem ser obrigatoriamente utilizados como foram apresentados.
- Entregar, no ambiente da disciplina no Moodle-DC, listagem do programa (arquivo .java), programa compilado (arquivo .class), relatório com documentação do programa (explicação do funcionamento de cada classe) e exemplos de execução.
- Data de entrega: 20/07/2017.