Ministério da Educação
Universidade Federal de Ouro Preto (Ufop)
Departamento de Computação e Sistemas (Decsi)

Professor: Rafael Alexandre
rfalexandre@ufop.edu.br
Disciplina: CSI032 || CSI102

Questão 1. A linguagem Java dispõe de um suporte nativo a vetores, que exige a definição de seu tamanho no momento da instanciação. Depois de instanciado, o tamanho do vetor não pode ser modificado. Escreva uma classe chamada Vetor cujos objetos simulem vetores de tamanho variável. A classe define os seguintes métodos:

a) **construtor:** recebe como parâmetro o tamanho inicial do vetor insert recebe como parâmetro uma string e a coloca na próxima posição disponível do vetor; note que o vetor cresce automaticamente, portanto, se a inserção ultrapassar o tamanho inicial estabelecido na criação, por exemplo, o vetor deve aumentar seu tamanho automaticamente
b) **get:** recebe como parâmetro uma posição do vetor e retorna a string que estiver naquela posição; se a posição não estiver ocupada ou ultrapassar o tamanho do vetor, este método retorna nulo
c) **size:** retorna o número de elementos inseridos no vetor (independente do tamanho do mesmo)

Questão 2. Escreva um modelo para representar uma lâmpada que está à venda em um supermercado. Considere as seguintes oprações para a lâmpada:

a) **estaAcesa:** retorna true ou false caso ela esteja acesa ou apagada, respectivamente;
b) **getLuminosidade:** retorna a luminosidade que pode estar entre 0 (apagada) e 100 (acesa)
c) **ajustarLuminosidade:** recebe o novo valor de sua luminosidade no intervalor 0 e 100. Caso seja fornecido um valor fora do intervalo especificado, a luminosidade atribuída deverá ser 0 e uma mensagem no console será exibida ao usuário indicando ”Luminosidade inválida!”.

Questão 3. Considere que você foi contratado para desenvolver um sistema para a gestão de estoque de um galpão, que possui um número fixo de praleleiras (3 dimensões) onde deveremos estocar materiais. Cada material poderá ser armazenado no galpão tal que o usuário deverá informar o índice da prateleira, o andar e o seguimento. Desenvolva uma classe para a gestão de estoque que possua as seguintes características:

a) **construtor:** recebe as informações do número de prateleiras, número de andares e quantidade de seguimentos de cada prateleira;
b) **inserirMaterial:** permite armazenar um material indicando o índice da prateleira, andar e seguimento;
c) **obterMaterialArmazenado:** retorna o material conforme especificação do número da prateleira, número do andar e identificador do seguimento. O espaço deverá ser liberado para que outro material possa ser armazenado no local. Caso não exista meterial estocado no local, o método deverá apresentar uma mensagem no console e retorna ”null”para o usuário.
d) **obterNumeroDeSlotesVagos:** retorna a quantidade de espaços disponíveis no galpão, que ainda não possuem materiais estocados.
e) **apresentaMapa:** dado o identificador de uma prateleira, apresentar o mapa de slots (andar e seguimento). Caso o slot esteja ocupado, o método deverá apresentar o identificador do Material armazenado. Caso o slot esteja vazio, indicar ”LIVRE”para a posição.