

**Questão 1.** Escreva um programa cujos objetos representam alunos matriculados em uma disciplina. Cada aluno deve guardar os seguintes dados: matrícula, nome, 2 notas de prova e 1 nota de trabalho. A relação associação entre a Disciplina e o Aluno se dará por agregação. Escreva os seguintes métodos para a classe Disciplina:

- a) **media:** dada a matrícula de um aluno, calcule a média final do aluno (cada prova tem peso 2,5 e o trabalho tem peso 2)
- b) **final:** dada a matrícula de um aluno, calcula quanto o aluno precisa para a prova final (retorna zero se ele não for para a final)

**Questão 2.** Escreva uma classe Data cuja instância (objeto) represente uma data. Esta classe deverá dispor dos seguintes métodos:

- a) **construtor:** define a data que determinado objeto (através de parâmetro), este método verifica se a data está correta, caso não esteja a data é configurada como 01/01/0001.
- b) **compara:** recebe como parâmetro um outro objeto da Classe data, compare com a data corrente e retorne: 0 se as datas forem iguais; 1 se a data corrente for maior que a do parâmetro; -1 se a data do parâmetro for maior que a corrente.
- c) **getDia:** retorna o dia da data
- d) **getMes:** retorna o mês da data
- e) **getMesExtenso:** retorna o mês da data corrente por extenso
- f) **getAno:** retorna o ano da data
- g) **isBissexto:** retorna verdadeiro se o ano da data corrente for bissexto e falso caso contrário (obs: considere que qualquer ano que seja uniformemente divisível por 4 é um ano bissexto)
- h) **clone:** o objeto clona a si próprio, para isto, ele cria um novo objeto da classe Data com os mesmos valores de atributos e retorna sua referência pelo método

Crie uma classe denominada AppData que faça os testes de cada um dos métodos propostos.

**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 prateleiras (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 material estocado no local, o método deverá apresentar uma mensagem no console e retorna "null" para o usuário.
- d) **obterNumeroDeSlotsVagos:** 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.