

## Curso Java

Prof. Júlio Machado

1) Implemente uma função recursiva  $fib(n)$  que calcula o  $n$ -ésimo valor da sequência de Fibonacci para qualquer valor inteiro  $n > 0$ . A sequência são os números 1,1,2,3,5,8,13,21,34,55,... A fórmula de cálculo é dada por:

$$\begin{cases} fib(1) = 1 \\ fib(2) = 1 \\ fib(n) = fib(n-1) + fib(n-2) \end{cases}$$

2) Dadas as classes *Circulo* e *Ponto*, reescreva o construtor *Circulo(Ponto p, int r)* para que:

- Gere uma exceção *NullPointerException* caso receba *null* no parâmetro do ponto;
- Gere uma exceção *IllegalArgumentException* caso receba um número não positivo no parâmetro do raio.

3) Reveja a implementação da classe *Lampada*. Analise a implementação dos métodos que permitem ligar ou desligar a lâmpada. Altere-os para que gerem uma exceção *IllegalStateException* caso sejam executados em momentos impróprios.

4) Dado um programa em Java que cria um arranjo contendo 10 números inteiros:

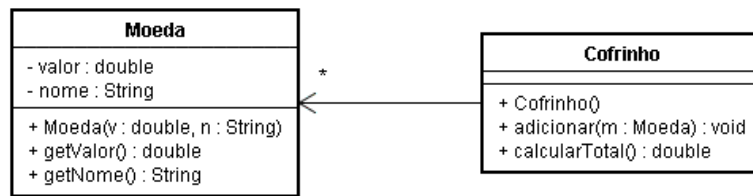
- Escreva um método que recebe um arranjo de inteiros e retorna o somatório dos seus valores.
- Escreva um método que recebe um arranjo de inteiros e retorna o maior elemento.

Na implementação dos dois métodos, caso o arranjo esteja vazio, qual seria uma boa solução para a questão?

5) Escreva uma classe *MatrizQuadrada* capaz de armazenar os dados de uma matriz cujo número de linhas e colunas é igual. A classe deve possuir as seguintes características:

- Construtor que informa a dimensão da matriz. Somente são aceitos números positivos.
- Métodos para obter e alterar o valor de qualquer posição da matriz.
- Método para calcular a soma dos elementos de uma determinada linha.
- Método para calcular a soma dos elementos de uma determinada coluna.
- Método para calcular a soma dos elementos da diagonal principal, ou seja, dos elementos  $M[i,j]$  onde  $i=j$ .
- Método para calcular a soma de duas matrizes. Discuta com os colegas diferentes formas de implementar a assinatura desse método.

6) Implementar completamente as classes descritas no diagrama UML:



7) Implemente um lustre que contém diversas lâmpadas e que pode ser ligado e desligado. Assuma que o número de lâmpadas do lustre deve ser definido através do construtor e que não pode ser alterado posteriormente. Inclua as seguintes funcionalidades no lustre:

- Um método que retorne verdadeiro caso o lustre tenha alguma lâmpada queimada, e falso caso contrário.
- Um método que retorne o número de lâmpadas queimadas no lustre.
- Um método que retorne e remova todas as lâmpadas queimadas no lustre. Utilize um arranjo para retornar a coleção de lâmpadas queimadas.

8) Alterar a classe Cofrinho do exercício anterior para incluir:

- Método que retorna o valor da menor moeda armazenada.
- Método que retorna uma instância da menor moeda armazenada.
- Método que retorna um mapa com a frequência (o número de moedas) de cada moeda existente (1 centavo, 5 centavos, 10 centavos, 25 centavos, 50 centavos, 1 real) no cofrinho.