

## Lista de exercícios – Programação Orientada a Objetos

Professor: Giomar Sequeiros

Semestre 2019-II

**Instruções Gerais:** Implementar em Java as seguintes questões

**Data de Entrega:** 10 de outubro de 2019 (no laboratório e por e-mail)

**1. (200 pontos)** Crie um programa que receba três números: a, b, e c. Determine se esses números podem formar um triângulo.

**Observação:** Considere que a condição de existência de um triângulo é definida por:

$$|b - c| < a < b + c$$

$$|a - c| < b < a + c$$

$$|a - b| < c < a + b$$

**2. (200 pontos)** Escreva um programa que calcule os quadrados e cubos dos números de 1 a n e imprima os valores resultantes em formato de tabela como mostrado a seguir:

Para n=5, temos:

Número	Quadrado	Cubo
1	1	1
2	4	8
3	9	27
4	16	64
5	25	125

**3. (200 pontos)** Crie um programa em Java que permita ordenar um array de n elementos.

**4. (200 pontos)** Crie uma classe denominada **Elevador** para armazenar as informações de um elevador dentro de um prédio. A classe deve armazenar o andar atual (térreo = 0), total de andares no prédio, excluindo o térreo, capacidade do elevador, e quantas pessoas estão presentes nele. A classe deve também disponibilizar os seguintes métodos:

- Inicializa: que deve receber como parâmetros a capacidade do elevador e o total de andares no prédio (os elevadores sempre começam no térreo e vazio);
- Entra: para acrescentar uma pessoa no elevador (só deve acrescentar se ainda houver espaço);
- Sai: para remover uma pessoa do elevador (só deve remover se houver alguém dentro dele);
- Sobe: para subir um andar (não deve subir se já estiver no último andar);
- Desce: para descer um andar (não deve descer se já estiver no térreo);
- Encapsular todos os atributos da classe (criar os métodos set e get).
- Crie uma classe para testar a funcionalidade da classe Elevador

**5. (200 pontos)** Escreva uma classe **Ponto2D** que represente um ponto no plano cartesiano. Além dos atributos x e y que representam as coordenadas do ponto (tipo double), a classe deve oferecer os seguintes membros:

- a) Construtores que permitam a inicialização do ponto:
  - i. Por default (sem parâmetros) na origem do espaço 2D;
  - ii. Num local indicado por dois parâmetros do tipo double (indicando o valor de abscissa e ordenada do ponto que está sendo criado);
- b) Métodos de acesso (getter/setter) para os atributos do ponto;
- c) Método que permita calcular a distância euclidiana de um ponto com outro, definido pela expressão abaixo;

Sejam os pontos  $P_1(x_1, y_1)$  e  $P_2(x_2, y_2)$ , a distância euclidiana está dado por:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

- d) Crie uma classe para testar a funcionalidade da classe Ponto2D