

Programação orientada a objetos

Exercício 1 *Escreva em Java uma classe `Ponto2D` que represente um ponto no plano cartesiano. Além dos atributos por você identificados, a classe deve oferecer os seguintes membros:*

- *atributos privados do ponto;*
- *método boolean que compara se um ponto é igual a outro;*
- *método que permita calcular a distância do ponto até outro ponto.*

Exercício 2 *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, capacidade do elevador (número máximo de pessoas) 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`).*

Exercício 3 *Um supermercado vende n diferentes tipos de produtos, onde n é dado como entrada. Cada produto tem um nome, um preço e uma quantidade em estoque. Sabendo disso:*

- *crie uma classe Produto;*
- *utilize atributos privados, construtor e métodos get/set;*
- *na função principal, declare e inicialize um vetor com n objetos do tipo Produto.*

Cada pedido de um cliente possui as quantidades para cada um dos n produtos e o valor a ser pago por ele. Na função principal:

- *declare e inicialize um vetor com as quantidades para cada produto;*
- *calcule o valor total do pedido;*
- *atualize o estoque do supermercado;*
- *é possível que um pedido não seja atendido completamente.*