

Lista de Exercícios

Programação Orientada a Objetos

Prof. Pedro Gabriel Calíope Dantas Pinheiro

24 de setembro de 2025

1. Escreva em Java uma classe **Contador**, que encapsule um valor usado para contagem de itens ou eventos. A classe deve oferecer métodos para:
 - a) Zerar o contador;
 - b) Incrementar o valor;
 - c) Retornar o valor atual do contador.
2. 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:
 - a) Construtores:
 - i) Por default (sem parâmetros) na origem do espaço 2D;
 - ii) Num local indicado por dois parâmetros do tipo **double**;
 - iii) Em um local indicado por outro ponto.
 - b) Métodos de acesso (**getter/setter**);
 - c) Método de comparação semântica (**equals**);
 - d) Método para calcular a distância do ponto que recebe a mensagem para outro ponto.
3. Escreva em Java uma classe que represente uma reta ($y = ax + b$). Forneça os seguintes membros:
 - a) Construtores:
 - i) Dois valores: coeficiente angular e coeficiente linear;
 - ii) Dois pontos.

- b) Métodos de acesso para os coeficientes;
 - c) Método que verifique se um ponto pertence à reta;
 - d) Método que, dada outra reta, retorne o ponto de interseção ou `null` se forem paralelas.
4. Escreva em Java uma classe que represente um círculo no plano cartesiano. Forneça:
- a) Construtor que receba o raio e o centro do círculo;
 - b) Construtor que receba o raio e posicione o círculo na origem;
 - c) Métodos de acesso ao raio;
 - d) Métodos `inflar` e `desinflar`;
 - e) Método que retorne a área do círculo.
5. Escreva uma classe que represente um país com os atributos: código ISO, nome, população e dimensão em km^2 . Além disso, mantenha uma lista de países vizinhos.
- a) Construtor que inicialize código ISO, nome e dimensão;
 - b) Getters e setters;
 - c) Método de igualdade semântica (mesmo código ISO);
 - d) Método que retorne a densidade populacional.
6. Escreva uma classe **Pessoa** para árvore genealógica.
- a) Construtores:
 - i) Inicialize nome, pai e mãe;
 - ii) Inicialize apenas o nome (pai e mãe nulos).
 - b) Método de igualdade semântica (mesmo nome e mesma mãe).
7. Escreva uma classe **Conjunto** de elementos **String**.
- a) Método para adicionar elemento (sem repetição);
 - b) Método para verificar pertinência de elemento;
 - c) Método `uniao`, que retorne novo conjunto com a união;
 - d) Método `inter`, que retorne novo conjunto com a interseção;
 - e) Método `menos`, que retorne novo conjunto com a diferença.
8. Crie uma classe **Matriz**.
- a) Construtor para inicializar dimensões;

- b) Métodos de acesso a elementos;
 - c) Método para adicionar matrizes;
 - d) Operações:
 - i) Comparação semântica;
 - ii) Transposta;
 - iii) Oposta;
 - iv) Matriz nula;
 - v) Verificar se é identidade;
 - vi) Verificar se é diagonal;
 - vii) Verificar se é singular;
 - viii) Verificar se é simétrica;
 - ix) Verificar se é anti-simétrica;
 - x) Subtração;
 - xi) Multiplicação;
 - xii) Cópia da matriz.
9. Crie uma classe **Veiculo** com os atributos básicos **marca** e **modelo**.
- a) Encapsule os atributos;
 - b) Crie getters e setters;
 - c) Crie uma classe **Carro** que herde de **Veiculo** e adicione número de portas;
 - d) Crie uma classe **Moto** que herde de **Veiculo** e adicione cilindradas;
 - e) Implemente **toString()** em cada veículo.
10. Crie uma classe abstrata **Funcionario** com atributos privados **nome** e **salario**.
- a) Getters e setters;
 - b) Método abstrato **calcularBonus()**;
 - c) Subclasses:
 - i) **Gerente**: bônus de 20%;
 - ii) **Desenvolvedor**: bônus de 10%.
 - d) Teste criando lista de funcionários e exibindo os bônus.
11. Crie uma hierarquia de classes para representar animais.
- a) Classe abstrata **Animal** com atributos **nome** e **idade**;
 - b) Getters e setters;

- c) Método abstrato `emitirSom()`;
 - d) Classes `Cachorro` e `Gato` implementando `emitirSom()`;
 - e) Demonstre polimorfismo em uma lista de animais.
12. Crie uma classe `ContaBancaria` com atributos privados `numero` e `saldo`.
- a) Getters e setters;
 - b) Métodos `depositar` e `sacar`;
 - c) Classe `ContaCorrente`, que cobra taxa fixa em cada saque;
 - d) Classe `ContaPoupanca`, que adiciona o método `renderJuros(double taxa)`.