

Exercícios de Programação Orientada a Objetos em Java

1. Parte 1 - Conceitos Básicos de Objetos

2. **Criação de Classe e Objeto:** Crie uma classe Carro com atributos marca, modelo e ano. Instancie objetos dessa classe e imprima seus valores.
 - a. **Métodos Simples:** Adicione um método `exibirDetalhes()` na classe Carro que imprime os detalhes do carro.
 - b. **Encapsulamento:** Torne os atributos marca, modelo e ano privados e crie métodos getters e setters.
 - c. **Construtores:** Adicione um construtor que receba os valores dos atributos da classe Carro no momento da instanciação.
 - d. **Contador de Objetos:** Adicione um atributo estático na classe Carro para contar quantos carros foram criados.
 - e. **Métodos Estáticos:** Crie um método estático na classe Carro que exibe quantos carros foram criados.
3. **Classe Pessoa:** Crie uma classe Pessoa com nome e idade. Adicione um método para verificar se a pessoa é maior de idade.
4. **Classe Produto:** Crie uma classe Produto com nome, preço e quantidade. Implemente um método para calcular o valor total em estoque.
5. **Associação de Objetos:** Crie uma classe Dono que possui um Carro. Instancie um objeto e relacione-o com um carro.
6. **Classe ContaBancaria:** Implemente uma classe com saldo e métodos para depositar e sacar dinheiro.

7. Parte 2 - Herança

8. **Criando uma Hierarquia de Classes:** Crie uma classe Animal com os atributos nome e idade. Depois, crie as classes Cachorro e Gato que herdam de Animal.
 - a. **Métodos Específicos:** Adicione métodos `latir()` em Cachorro e `miar()` em Gato.
 - b. **Sobrecarga de Construtores:** Adicione várias formas de inicializar um Animal, incluindo um construtor padrão e outro que recebe nome e idade.
 - c. **Superclasse e Subclasses:** Utilize `super()` dentro das subclasses para chamar o construtor da superclasse.
9. **Classe Funcionario e Gerente:** Crie uma classe Funcionario com nome e salário. Crie uma subclasse Gerente que tem um bônus adicional.
10. **Classe Veiculo e Moto:** Crie uma classe Veiculo e uma subclasse Moto. Adicione um atributo cilindrada e um método para exibi-lo.
11. **Classe Professor e Aluno:** Crie uma hierarquia onde Aluno e Professor herdam de Pessoa, cada um com atributos específicos.

12. **Classe InstrumentoMusical:** Crie uma superclasse InstrumentoMusical e subclasses como Violao e Piano, cada uma com um método tocar().
13. **Classe Empregado e Especializações:** Crie uma hierarquia onde Empregado tem subclasses Programador e Designer, cada uma com um método que imprime sua função.
14. **Parte 3 - Polimorfismo**
15. **Polimorfismo de Sobrescrita:** Modifique Animal para que tenha um método emitirSom(). Em Cachorro e Gato, sobrescreva esse método.
16. **Uso de @Override:** Garanta que as subclasses sobrescrevem corretamente os métodos usando @Override.
17. **Classe Abstrata:** Transforme Animal em uma classe abstrata e faça emitirSom() ser obrigatório nas subclasses.
18. **Polimorfismo em Métodos:** Crie um método que recebe Animal como parâmetro e chama emitirSom().
19. **Hierarquia de Veículos:** Crie uma hierarquia de classes Veiculo -> Carro -> CarroEletrico, e sobrescreva um método abastecer() para se adaptar a cada tipo de veículo.