

Lista de 100 Exercícios de Orientação a Objetos

Parte 1 – Fundamentos (1–20)

- 1 Defina o que é classe e objeto com exemplos em pseudocódigo.
- 2 Crie uma classe Carro com atributos básicos (marca, modelo, ano).
- 3 Instancie três objetos da classe Carro e exiba seus atributos.
- 4 Diferencie atributos de instância e de classe com exemplos.
- 5 Crie uma classe Pessoa com atributos nome e idade, e um método apresentar().
- 6 Explique a diferença entre atributo e método.
- 7 Escreva um construtor para a classe Pessoa.
- 8 Crie uma classe ContaBancaria com métodos depositar() e sacar().
- 9 Defina encapsulamento. Crie atributos privados em ContaBancaria.
- 10 Implemente métodos getters e setters em ContaBancaria.
- 11 Explique o conceito de abstração em OO.
- 12 Modele uma classe Livro com atributos e um método exibirInformacoes().
- 13 Faça uma classe Retângulo com métodos calcularArea() e calcularPerimetro().
- 14 Crie uma classe Círculo que calcule área e circunferência.
- 15 Modele a classe Aluno com atributos e um método para calcular média.
- 16 Explique a diferença entre classe e objeto usando Aluno como exemplo.
- 17 Mostre como declarar constantes dentro de uma classe.
- 18 Crie uma classe Produto com desconto aplicado no preço.
- 19 Modele uma classe Funcionario com salário e um método aumentarSalario().
- 20 Explique a diferença entre classe concreta e classe abstrata.

Parte 2 – Herança e Polimorfismo (21–40)

- 1 Crie uma classe Animal e subclasses Cachorro e Gato.
- 2 Adicione o método emitirSom() em Animal e sobrescreva em Cachorro/Gato.
- 3 Explique a diferença entre herança simples e herança múltipla.
- 4 Modele uma classe abstrata Veiculo e classes filhas Carro e Moto.
- 5 Crie uma hierarquia Funcionario → Gerente, Desenvolvedor.
- 6 Sobrescreva o método calcularSalario() em cada tipo de funcionário.
- 7 Explique o conceito de polimorfismo com exemplos práticos.
- 8 Crie um método que receba um Animal e chame emitirSom().
- 9 Modele FormaGeometrica como abstrata e implemente Quadrado e Círculo.
- 10 Crie uma lista de FormaGeometrica e percorra exibindo áreas.
- 11 Explique a diferença entre overloading e overriding.
- 12 Crie um exemplo de método sobrecarregado em Calculadora.
- 13 Explique upcasting e downcasting em OO.
- 14 Implemente casting em uma hierarquia de classes de veículos.
- 15 Crie uma interface Autenticavel e implemente em Usuario e Administrador.
- 16 Explique a diferença entre interface e classe abstrata.
- 17 Modele uma hierarquia de InstrumentoMusical com Violão, Bateria.
- 18 Crie um polimorfismo para tocarInstrumento().
- 19 Modele FormaPagamento (Dinheiro, Cartao, Pix) e sobrescreva pagar().
- 20 Crie uma lista de formas de pagamento e execute o método polimórfico.

Parte 3 – Encapsulamento e Boas Práticas (41–60)

- 1 Explique a diferença entre atributos públicos, privados e protegidos.
- 2 Refaça ContaBancaria garantindo que saldo nunca seja negativo.

- 3 Crie exceções personalizadas para SaldoInsuficienteException.
- 4 Modele uma classe Cofre com acesso apenas via senha.
- 5 Explique o princípio da responsabilidade única (SRP).
- 6 Refatore Funcionario para que só trate dados do funcionário, não da folha de pagamento.
- 7 Crie um exemplo que viole encapsulamento e corrija.
- 8 Modele uma classe Agenda com encapsulamento adequado.
- 9 Explique coesão e acoplamento em OO.
- 10 Crie um sistema de pedidos com classes Pedido, Cliente e Produto.
- 11 Aplique encapsulamento no sistema de pedidos.
- 12 Modele um carrinho de compras com lista de produtos encapsulada.
- 13 Adicione um método para calcular total do carrinho.
- 14 Implemente validação de estoque no carrinho.
- 15 Explique o conceito de imutabilidade em objetos.
- 16 Modele uma classe Data imutável.
- 17 Explique o padrão POJO (Plain Old Java Object).
- 18 Refaça Aluno como POJO.
- 19 Explique o princípio Tell, Don't Ask.
- 20 Refaça ContaBancaria aplicando esse princípio.

Parte 4 – Avançado e Padrões (61–100)

- 1 Explique o princípio de substituição de Liskov (LSP).
- 2 Dê um exemplo de violação de LSP e corrija.
- 3 Modele um sistema de notificação (Email, SMS, Push).
- 4 Aplique o padrão Strategy para envio de notificações.
- 5 Explique o padrão Singleton.
- 6 Implemente Singleton para ConfigSistema.
- 7 Explique riscos do Singleton mal utilizado.
- 8 Modele um padrão Factory para VeiculoFactory.
- 9 Explique o padrão Observer.
- 10 Implemente Observer em Estoque e Cliente.
- 11 Explique o princípio da inversão de dependência (DIP).
- 12 Aplique DIP em um sistema de pagamento.
- 13 Explique o padrão Adapter.
- 14 Modele um exemplo com sistema de tomada 110V adaptado para 220V.
- 15 Explique o padrão Decorator.
- 16 Aplique Decorator em um sistema de Bebida com adicionais.
- 17 Explique o padrão Composite.
- 18 Modele Arquivo e Pasta com Composite.
- 19 Explique o padrão Proxy.
- 20 Crie um Proxy de acesso a um banco de dados.
- 21 Explique o padrão Command.
- 22 Modele comandos para LigarLuz e DesligarLuz.
- 23 Explique o padrão Template Method.
- 24 Modele relatórios diferentes usando Template Method.
- 25 Explique o padrão Builder.
- 26 Modele PedidoBuilder para montar pedidos.
- 27 Explique o padrão Prototype.
- 28 Implemente clonagem em Documento.
- 29 Explique o padrão State.
- 30 Modele uma ContaBancaria com estados (Positivo, Negativo).
- 31 Explique o padrão Chain of Responsibility.
- 32 Modele aprovação de despesas em diferentes níveis hierárquicos.
- 33 Explique o princípio da segregação de interface (ISP).
- 34 Modele interfaces menores aplicando ISP.
- 35 Explique o conceito de DDD (Domain Driven Design).

- 36 Modele entidades e agregados para um sistema de biblioteca.
- 37 Explique a diferença entre entidade e valor em DDD.
- 38 Modele um valor-objeto CPF imutável.
- 39 Explique o padrão Repository.
- 40 Modele ClienteRepository para persistência de clientes.