

Lista de Exercícios - Programação Orientada a Objetos

Exercício 1: Paradigmas de Programação

- Defina o que é um paradigma de programação e cite 3 exemplos diferentes.
- Compare programação estruturada e orientada a objetos, destacando vantagens e desvantagens de cada uma.
- Explique por que existem tantas linguagens de programação diferentes.

Exercício 2: Fundamentos de OO

- O que é abstração em POO? Dê um exemplo prático.
- Defina encapsulamento e explique sua importância.
- Qual a diferença entre classe e objeto? Exemplifique.

Exercício 3: Diagramas de Classes

- Quais elementos compõem um diagrama de classes UML?
- Explique os tipos de relacionamentos entre classes:
 - Associação (simples, agregação, composição)
 - Generalização
 - Dependência

Exercício 4: Modificadores de Acesso

- Descreva os níveis de acesso `private`, `protected` e `public`.
- Qual a importância dos modificadores de acesso para o encapsulamento?
- Crie uma tabela comparativa mostrando o acesso para cada modificador

Exercício 5: Herança e Polimorfismo

- Defina herança e polimorfismo em POO
- Qual a diferença entre herança simples?
- Explique a regra "é um" para identificar quando usar herança
- O que é polimorfismo de inclusão?

Exercício 6: Sobrecarga

- O que é sobrecarga de métodos?
- Quais as regras para sobrecarregar métodos?
- Compare sobrecarga em C++ e Java

Exercício 7: Implementação Básica

Crie uma classe Pessoa em C++ e Java com:

- Atributos privados: nome, idade, cpf
- Construtores (padrão e parametrizado)
- Getters e setters
- Método `imprimirDados()`

Exercício 8: Encapsulamento

Implemente uma classe ContaCorrente que:

- Não permita saldo negativo
- Limite de 3 saques diários
- Registre o histórico de transações
- Use modificadores de acesso apropriados

Exercício 9: Diagrama e Implementação

Modele e implemente um sistema para Biblioteca:

Livro → ISBN, título, autor, disponível

Usuário → nome, matrícula, livrosEmprestados[]

Empréstimo → dataEmprestimo, dataDevolucao

- Desenhe o diagrama de classes
- Implemente as classes
- Crie métodos para emprestar e devolver livros

Exercício 10: Sobrecarga

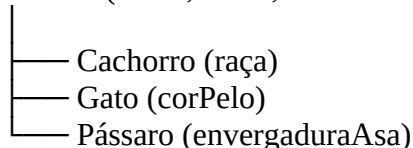
Implemente uma classe Matematica com métodos sobrecarregados:

- calcularArea() para: quadrado, retângulo, círculo, triângulo
- Use diferentes parâmetros para cada forma

Exercício 11: Herança

Crie a hierarquia:

Animal (nome, idade, fazerSom())



- Sobrescreva fazerSom() em cada subclasse
- Crie métodos específicos para cada animal

Exercício 12: Polimorfismo

Estenda o exercício anterior:

- Crie array de Animal com diferentes subclasses
- Percorra chamando fazerSom() (polimorfismo)
- Adicione método mover() com comportamentos diferentes

Exercício 13: Sistema Base

Implemente sistema de Universidade:

- Classes: Pessoa, Aluno, Professor, Disciplina, Curso
- Relacionamentos: matrícula, ministração, pré-requisitos
- Funcionalidades: matricular, atribuir nota, calcular CR

Exercício 14: C++ vs Java

Para cada item abaixo, compare a implementação em C++ e Java:

- a) Declaração de classes
- b) Gerenciamento de memória
- c) Tratamento de arquivos
- d) Modificadores de acesso
- e) Herança e polimorfismo

Exercício 15: Análise Crítica

- a) Quando usar composição em vez de herança?
- b) Qual a vantagem do polimorfismo em sistemas extensíveis?
- c) Por que é importante o encapsulamento em projetos grandes?