



Lista 5: Herança, Polimorfismo, Interface e Classes Abstratas

Monitoria de Programação Orientada a Objetos - (10/01/2025)

I. Herança e Polimorfismo

Questão 1 - [IBFC DPE-MT 2022] A programação orientada a objeto possui pilares fundamentais em sua linguagem. Dadas as definições abaixo, identifique a alternativa que corresponda respectivamente a cada pilar.

1. Mecanismo utilizado para disponibilizar métodos que operam sobre os dados e que protegem o acesso direto indevido aos atributos de uma instância fora da classe onde estes foram declarados.
 2. Mecanismo que permite criar novas classes a partir de classes já existentes, aproveitando-se das características existentes na classe a ser estendida.
 3. Situação na qual um objeto pode se comportar de maneiras diferentes ao receber uma mensagem, dependendo do seu tipo de criação.
- a) 1. Encapsulamento - 2. Polimorfismo - 3. Herança
b) 1. Herança - 2. Encapsulamento - 3. Polimorfismo
c) 1. Polimorfismo - 2. Herança - 3. Encapsulamento
d) 1. Encapsulamento - 2. Herança - 3. Polimorfismo

Questão 2 - Crie um sistema de cadastro de funcionários que utiliza herança e polimorfismo. O sistema deve conter as seguintes classes:

- **Funcionario:** com os atributos nome, idade e salarioBase e métodos públicos de acesso para esses atributos. Adicione um método calcularSalario que retorna o salário base e um método exibirDetalhes que imprime o nome, a idade e o salário base.
- **Gerente:** que herda de Funcionario, adiciona o atributo departamento e métodos públicos de acesso para este atributo. Sobrescreva o método calcularSalario para incluir um bônus de 20% sobre o salário base. Sobrescreva o método exibirDetalhes para incluir o departamento.
- **Vendedor:** que herda de Funcionario, adiciona o atributo comissao e métodos públicos de acesso para este atributo. Sobrescreva o método calcularSalario para incluir a comissão (salarioBase + comissão). Sobrescreva o método exibirDetalhes para incluir a comissão.

II. Interfaces

Questão 3 - Desenvolva um sistema para calcular o volume de diferentes sólidos geométricos.

1. Crie uma interface chamada **Solido**, que define o método `calcularVolume()`.
2. Crie duas classes que deverão implementar a interface: **Cubo** e **Cilindro**.
3. A classe Cubo deve ter um atributo `lado` (double) e a classe Cilindro deve ter atributos `raio` (double) e `altura` (double). Em ambas as classes, sobrescreva o método `calcularVolume()` para calcular e retornar o volume do sólido específico.
4. Na classe `TesteSolidos`, crie instâncias de Cubo e Cilindro. Use polimorfismo para chamar `calcularVolume()` e mostrar o volume de cada sólido.

Fórmulas de Volume Utilizadas:

- **Volume do Cubo:** $volume = lado^3$
- **Volume do Cilindro:** $volume = \pi * raio^2 * altura$

Questão 4 - Implemente um sistema para controlar dispositivos eletrônicos utilizando uma interface **ControladorDispositivo**, que define os métodos `ligarDispositivo`, `desligarDispositivo`, e `verificarStatus`. A interface deve ser implementada pela classe **ControleRemoto**, que possui um atributo `estado` do tipo boolean para indicar se o dispositivo está ligado (true) ou desligado (false). A classe `ControleRemoto` deve alterar o estado do dispositivo nos métodos `ligarDispositivo` e `desligarDispositivo`, e retornar o valor do estado no método `verificarStatus`.

Crie uma classe de teste `TesteControleRemoto` que instancia um objeto de `ControleRemoto`, altera o estado do dispositivo e verifica seu status utilizando os métodos definidos.

III. Classes Abstratas

Questão 5 - Desenvolva um sistema para gerenciar diferentes métodos de pagamento. Crie uma classe abstrata chamada **Pagamento** com os atributos `valor` (double) e `data` (String), e um método abstrato `calcularTaxa()`.

Crie duas subclasses:

- **PagamentoCredito:**
 - Atributos adicionais: `numeroParcelas` (int).
 - Sobrescreva o método `calcularTaxa()` para retornar uma taxa baseada no número de parcelas. A taxa deve ser calculada como 0,5% do valor

para cada parcela. Por exemplo, se o valor é 1000 e o número de parcelas é 5, a taxa será $1000 * 0.005 * 5 = 25$.

- **PagamentoDebito:**

- Sobrescreva o método `calcularTaxa()` para retornar uma taxa fixa de 1.5% do valor. Por exemplo, se o valor é 1000, a taxa será $1000 * 0.015 = 15$.

Na classe `TestePagamentos`, crie instâncias de `PagamentoCredito` e `PagamentoDebito`, e use polimorfismo para chamar o método `calcularTaxa()`.

Questão 6 - Crie uma classe abstrata **Estudante** com atributos `nome` (String) e `notas` (array de double), além dos métodos abstratos `calcularMedia()` e `verificarAprovacao()`. Crie subclasses **EstudanteGraduacao** e **EstudantePosGraduacao** que implementem os métodos abstratos.

- `EstudanteGraduacao` deve ter um método `calcularMedia` que retorne a média das notas e `verificarAprovacao` que retorne se o estudante foi aprovado (média ≥ 7).
- `EstudantePosGraduacao` deve ter um método `calcularMedia` que retorne a média das notas e `verificarAprovacao` que retorne se o estudante foi aprovado (média ≥ 8).