

# Exercícios sobre Polimorfismo em Java

**1. Formas Geométricas:** Crie uma classe abstrata `Forma` com o método abstrato `calcularArea()`. Crie classes concretas `Circulo`, `Retangulo` e `Triangulo` que herdam de `Forma` e implementam o método `calcularArea()` de acordo com suas fórmulas específicas.

**2. Animais:** Crie uma classe abstrata `Animal` com o método abstrato `emitirSom()`. Crie classes concretas `Cachorro`, `Gato` e `Vaca` que herdam de `Animal` e implementam o método `emitirSom()` de acordo com o som de cada animal.

**3. Funcionários:** Crie uma classe abstrata `Funcionario` com os atributos `nome` e `salario` e o método abstrato `calcularBonus()`. Crie classes concretas `Gerente`, `Vendedor` e `Desenvolvedor` que herdam de `Funcionario` e implementam o método `calcularBonus()` de acordo com as regras de cada cargo.

**4. Contas Bancárias:** Crie uma classe abstrata `ContaBancaria` com os atributos `saldo` e `numeroConta` e os métodos abstratos `depositar(double valor)` e `sacar(double valor)`. Crie classes concretas `ContaCorrente` e `ContaPoupanca` que herdam de `ContaBancaria` e implementam os métodos de depósito e saque com suas regras específicas (por exemplo, a conta corrente pode ter um limite de saque, enquanto a conta poupança pode ter um rendimento mensal).

**5. Impressora:** Crie uma interface `Impressora` com o método `imprimir(String documento)`. Implemente essa interface em classes como `ImpressoraJatoDeTinta` e `ImpressoraLaser`, cada uma com sua própria forma de simular a impressão.

**6. Meios de Transporte:** Crie uma interface MeioTransporte com o método mover(). Implemente essa interface em classes como Carro, Aviao e Navio, cada uma com sua própria forma de representar o movimento.

**7. Pagamentos:** Crie uma classe abstrata Pagamento com o método abstrato processarPagamento(double valor). Crie classes concretas PagamentoCartao, PagamentoBoleto e PagamentoPix que herdam de Pagamento e implementam o método processarPagamento() de acordo com a forma de pagamento.

**8. Instrumentos Musicais:** Crie uma interface InstrumentoMusical com o método tocar(). Implemente essa interface em classes como Guitarra, Piano e Bateria, cada uma com sua própria forma de representar o som do instrumento.

**9. Desenhos:** Crie uma interface Desenho com o método desenhar(). Implemente essa interface em classes como Circulo, Quadrado e Triangulo, cada uma com sua própria forma de desenhar a figura (pode ser uma representação textual ou gráfica, dependendo do seu conhecimento).

**10. Pessoas:** Crie uma classe abstrata Pessoa com o método abstrato apresentar(). Crie classes concretas Professor, Aluno e Funcionario que herdam de Pessoa e implementam o método apresentar() de acordo com suas informações específicas.

**11. Jogos:** Crie uma interface Jogo com os métodos iniciarJogo(), jogar() e finalizarJogo(). Implemente essa interface em classes como JogoTabuleiro, JogoCartas e JogoVideoGame.

**12. Sistema de Pedidos de Comida Online:** Crie interfaces Restaurante, Prato e FormaPagamento com métodos adequados. Implemente essas interfaces em classes concretas e utilize polimorfismo para lidar com diferentes tipos

de restaurantes, pratos e formas de pagamento em um sistema de pedidos online.

**13. Gerenciamento de Tarefas:** Crie uma interface Tarefa com métodos como obterDescricao(), obterPrioridade() e executar(). Implemente essa interface em classes como TarefaPessoal, TarefaProfissional e TarefaAcademica. Utilize polimorfismo para gerenciar diferentes tipos de tarefas em uma lista.

**14. Controle de Estoque:** Crie interfaces Produto e OperacaoEstoque com métodos adequados. Implemente essas interfaces em classes concretas e utilize polimorfismo para lidar com diferentes tipos de produtos e operações de estoque (entrada, saída, consulta).

**15. Coleções e Polimorfismo:** Crie uma lista de objetos Animal (do exercício 2) e utilize polimorfismo para chamar o método emitirSom() de cada animal na lista.

**16. Sistema de Gerenciamento de Contas Bancárias:** Crie uma interface ContaBancaria com os métodos depositar(double valor), sacar(double valor) e consultarSaldo(). Implemente essa interface em classes como ContaCorrente, ContaPoupanca e ContaInvestimento, cada uma com suas regras específicas para essas operações.

**17. Sistema de Locadora de Veículos:** Crie uma interface Veiculo com os métodos alugar(), devolver() e calcularValorAluguel(int dias). Implemente essa interface em classes como Carro, Moto e Caminhao, cada uma com sua própria lógica de cálculo de valor de aluguel.

**18. Sistema de Gerenciamento de Funcionários:** Crie uma interface Funcionario com os métodos calcularSalario() e gerarFolhaPagamento(). Implemente essa interface em classes como Gerente, Vendedor e Tecnico, cada uma com sua própria lógica de cálculo de salário e geração de folha de pagamento.

**19. Polimorfismo e Exceções:** Crie uma classe Calculadora com um método dividir(int a, int b) que lança uma exceção ArithmeticException se b for zero. Utilize polimorfismo para lidar com essa exceção em um bloco try-catch.

**20. Sistema de Gerenciamento de Escola:** Crie interfaces Pessoa, Aluno, Professor e Disciplina com métodos adequados. Implemente essas interfaces em classes concretas e utilize polimorfismo para lidar com diferentes tipos de pessoas, alunos, professores e disciplinas em um sistema de gerenciamento de escola.