**Lista de Exercícios: Interfaces e o Design de Contratos**

1. O Sistema da Receita Federal (Tributáveis)

A Receita Federal precisa de um sistema que calcule os impostos devidos sobre diferentes tipos de bens e ativos. Tanto uma ContaCorrente (que tem impostos sobre o saldo) quanto um SeguroDeVida (que pode ter tributação específica) são "tributáveis", mas são produtos financeiros completamente distintos.

- **Sua Tarefa:**
    - **Parte A:** Crie a interface Tributavel com um método: double getValorImposto().
    - **Parte B:** Crie a classe ContaCorrente, com um atributo saldo, que implements Tributavel. A lógica do imposto é retornar 2% do saldo.
    - **Parte C:** Crie a classe SeguroDeVida, que implements Tributavel. A lógica do imposto é um valor fixo de R$ 42,00.
    - **Parte D (Teste):** No main, crie um objeto de cada classe e imprima o valor do imposto de cada um para provar que ambos cumprem o contrato Tributavel, cada um à sua maneira.

2. Sistema de Automação Residencial

Você está criando um sistema para uma casa inteligente. Vários dispositivos, como uma Lampada e um ArCondicionado, compartilham a capacidade de serem ligados e desligados, mas são aparelhos completamente diferentes.

- **Sua Tarefa:** Crie uma interface DispositivoEletronico com dois métodos: ligar() e desligar(). Em seguida, crie as classes Lampada e ArCondicionado. Ambas devem implementar a interface DispositivoEletronico. A implementação dos métodos pode simplesmente imprimir no console uma mensagem como "Lâmpada ligada!" ou "Ar condicionado desligado.".

3. O Dilema do Home Office na Elotech

Na Elotech, a estrutura de Funcionario está crescendo. A empresa agora tem uma política de home office, mas nem todos os cargos são elegíveis. Um Desenvolvedor pode trabalhar de casa, assim como um ConsultorExterno, que é um prestador de serviço e não faz parte da hierarquia de Funcionario.

- **Sua Tarefa:**
    - **Parte A:** Crie uma interface TrabalhavelRemotamente com um método trabalharDeCasa().
    - **Parte B:** Modifique a classe Desenvolvedor (que já extends Funcionario) para que ela também implements TrabalhavelRemotamente. O método deve imprimir "Desenvolvedor trabalhando de casa...".
    - **Parte C:** Crie uma nova classe ConsultorExterno (que **não** herda de Funcionario) e faça com que ela também implements TrabalhavelRemotamente. O método deve imprimir "Consultor trabalhando remotamente...".
    - Este exercício prova que uma mesma capacidade pode ser adicionada a classes de hierarquias totalmente diferentes.

4. O Zoológico Virtual (Análise de Design)

Você está projetando um sistema para um zoológico. Os animais a serem modelados são: Leao, Tigre, Pato e Morcego. Você precisa representar duas coisas: 1) o fato de que todos são Animais e compartilham características como peso; 2) o fato de que Pato e Morcego podem Voar, mas Leao e Tigre não.

- **Sua Tarefa:** Em um comentário, explique qual ferramenta da POO (abstract class ou interface) você usaria para modelar o conceito de "Animal" e qual você usaria para modelar a capacidade de "Voar". Justifique sua decisão de design para ambas, baseando-se na relação "é-um-tipo-de" vs. "é-capaz-de".

5. Uma Prévia do Polimorfismo com Interfaces

Usando os objetos Tributavel do exercício 1, a Receita Federal agora precisa de uma funcionalidade para somar o imposto total de uma lista de bens de um contribuinte.

- **Sua Tarefa:** No seu método main, crie uma List<Tributavel>. Adicione a esta lista a ContaCorrente e o SeguroDeVida que você criou. Em seguida, escreva um loop for-each que percorra essa lista, chame o método t.getValorImposto() em cada item e some os resultados em uma variável de total. Por fim, imprima o imposto total a ser pago.

6. Caça ao Bug: O Contrato Quebrado

O desenvolvedor júnior da sua equipe escreveu o código abaixo, mas a classe RelatorioPDF apresenta um erro de compilação que ele não entende.

```java
interface Imprimivel {
    void imprimir();
    String getCabecalho();
}

// A classe abaixo não compila
public class RelatorioPDF implements Imprimivel {
    @Override
    public void imprimir() {
        System.out.println("Imprimindo PDF...");
    }
}
```

- **Sua Tarefa:** Em um comentário, explique ao desenvolvedor júnior por que o código não compila. Qual "cláusula" do contrato Imprimivel a classe RelatorioPDF deixou de cumprir? Como se corrige o código?

7. Sistema de Exportação de Dados

Um sistema de business intelligence precisa exportar dados de diferentes fontes (uma lista de Usuarios, um RelatorioFinanceiro, etc.) para um formato padronizado, como CSV (Comma-Separated Values).

- **Sua Tarefa:** Crie uma interface ExportavelCSV com um método String toCSV(). Em seguida, crie duas classes distintas, Usuario e Produto, e faça com que ambas implementem a interface.
    - O toCSV() de Usuario deve retornar algo como "nome_do_usuario;email_do_usuario".
    - O toCSV() de Produto deve retornar "nome_do_produto;preco_do_produto".

8. Múltiplas Capacidades: A Porta de Segurança

Você está modelando uma PortaDeSeguranca para um prédio inteligente. Esta porta tem duas capacidades distintas e independentes: ela é um dispositivo eletrônico (pode ser ligada/desligada) e também é um dispositivo de segurança (pode ser autenticada).

- **Sua Tarefa:**
    - Crie uma interface DispositivoEletronico com os métodos ligar() e desligar().
    - Crie uma interface Autenticavel com o método autenticar(String senha).
    - Crie uma única classe PortaDeSeguranca que **implementa ambas as interfaces** (implements DispositivoEletronico, Autenticavel).
    - Implemente todos os métodos exigidos pelos dois contratos dentro da classe PortaDeSeguranca.

9. Desacoplamento e Flexibilidade (Conceitual)

Na aula, foi dado o exemplo da Elotech com a interface GatewayDePagamento e as implementações PagSeguroGateway e MercadoPagoGateway.

- **Sua Tarefa:** Imagine que o sistema foi construído sem a interface, e todo o código de e-commerce chama diretamente os métodos da classe PagSeguroGateway. O que aconteceria com o projeto se a diretoria decidisse trocar o PagSeguro pelo MercadoPago? Explique com suas palavras a dificuldade dessa manutenção e como o uso da interface GatewayDePagamento resolve esse problema.

10. Desafio: Um Jogo com Habilidades

Em um jogo, você tem uma hierarquia de Personagem (classe abstrata). Heroi e Inimigo herdam de Personagem. Além disso, certos personagens e até mesmo alguns Items (que não são personagens) podem ter a capacidade de ficarem invisíveis.

- **Sua Tarefa:** Esboce (em código ou pseudocódigo) a arquitetura para essa situação.
    - Como você modelaria Personagem, Heroi e Inimigo?
    - Como você modelaria a capacidade de "ficar invisível"? (Dica: interface FicavelInvisivel { void ficarInvisivel(); }).
    - Mostre como a classe Heroi poderia herdar de Personagem e, ao mesmo tempo, implementar FicavelInvisivel. Crie também uma classe PocaoDeInvisibilidade (que não herda de Personagem) que também implementa FicavelInvisivel.