Exercícios POO

Exercício 1: Listas e Classes

1. Criar uma classe chamada Produto com as propriedades "**Nome**", "**Preco**", e "**QuantidadeEmStock**".

2. Criar uma lista de produtos para guardar os vários produtos.

3. Adicionar alguns produtos à lista.

4. Escreva um método que itere pela lista de produtos e imprima as informações de cada produto.

Exercício 2: Herança e Polimorfismo

1. Criar uma classe base chamada Animal com as propriedades como o Nome e o método "**FazerBarulho**".

2. Criar duas classes derivadas, "**Cao**" e "**Gato**", que herdam de Animal. Implemente o método "**FazerBarulho**" de forma diferente em cada classe.

3. Criar uma lista de animais e adicione as instâncias de "**Cao**" e "**Gato**".

4. Itere pela lista e chame o método "**FazerBarulho**" para cada animal.

Exercício 3: Classes e Herança

1. Criar uma classe base chamada "**Veiculo**" com propriedades "**Marca**" e "**Modelo**".

2. Criar duas classes derivadas, "**Carro**" e "**Mota**", que herdam de "**Veiculo**". Adicionar as propriedades específicas a cada uma, como "**NumeroDePortas**" para o carro e "**Cilindrada**" para a "**Mota**".

3. Criar uma lista de veículos que inclua as instâncias de **carros** e **motos**.

4. Escreva um método que itere pela lista e imprima as informações de cada veículo, incluindo as propriedades específicas de cada tipo.

Exercício 4: Interfaces

1. Criar uma interface chamada "**IPagamento**" com os seguintes métodos:

- "**CalcularPagamento()**", que calcula o valor do pagamento.
- "**ProcessarPagamento()**", que processa o pagamento.

2. Crie duas classes, "**Compra**" e "**Venda**", que representam transações comerciais. Ambas devem implementar a interface "**IPagamento**".

3. Na classe "**Compra**", inclua as propriedades como "**ValorTotal**" e o "**Fornecedor**". Implemente os métodos da interface "**IPagamento**" de acordo com o contexto de uma compra.

4. Na classe Venda, inclua as propriedades como "**ValorTotal**" e Cliente. Implemente os métodos da interface "**IPagamento**" de acordo com o contexto de uma venda.

5. Criar uma lista de transações que pode conter instâncias de Compra e Venda.

6. Utilizar um ciclo "**loop**" para iterar pela lista as transações e, para cada transação, chamar os métodos da interface "**IPagamento"** para calcular e processar o pagamento.

Exercício 5: Classes Abstratas para um Sistema de Gestão de Funcionários

Desenvolver um sistema que permita a criação e a gestão de diferentes tipos de funcionários, exemplo: funcionários a full-time ou Part-Time e Gestores. Deve utilizar as classes abstratas para definir a estrutura comum desses funcionários e, em seguida, criar classes derivadas (subclasses) para cada tipo específico de funcionário.

1. Criar uma classe abstrata chamada "**Funcionario**" que tenha os seguintes membros:

- Propriedades: "**Nome**", "**ID**", "**SalárioBase**".
- Um construtor para inicializar as propriedades.

2. Criar três classes derivadas de "**Funcionario**": "**FuncionarioFullTime**", "**FuncionarioPartTime**" e "**Gestor**". Cada uma dessas classes deve implementar um método abstrato chamado "**CalcularSalario**" que calcula o salário do funcionário com base em regras específicas para cada tipo de funcionário.

3. Adicionar as propriedades específicas para cada tipo de funcionário. Por exemplo, um "**FuncionarioFullTime**" pode ter uma propriedade para horas trabalhadas, enquanto um "Gestor" pode ter uma propriedade para bônus.

4. Implementar o método "**CalcularSalario**" nas classes derivadas de acordo com as regras específicas para cada tipo de funcionário. Por exemplo, o salário de um "**FuncionarioFullTime**" pode ser calculado com base nas horas trabalhadas e na taxa de pagamento.

5. Criar uma classe principal que mostre a criação de funcionários de diferentes tipos e o cálculo dos salários. Criar objetos de todas as classes derivadas e chamá-los para calcular o salário.