

AC2: Simulado

AC2: Simulado

- Você foi contratado para atualizar e expandir o sistema de gestão de veículos desenvolvido anteriormente no SimuladoAC1.
- O objetivo desta nova versão é tornar o código mais robusto e escalável, utilizando conceitos avançados de Programação Orientada a Objetos (POO).
- Para isso, você deverá implementar novas funcionalidades e modificar o código existente, incorporando os seguintes conceitos:

AC2 : Simulado

- Crie uma classe abstrata chamada VeiculoBase que será a base para todos os tipos de veículos (Carro, Caminhão e Motocicleta).
- Essa classe deve possuir os seguintes atributos protegidos:
 - marca (do tipo String)
 - modelo (do tipo String)
 - precoBase (do tipo double)
- A classe abstrata VeiculoBase deve declarar um método abstrato chamado calcularPreco, que será implementado de forma específica nas subclasses.

AC2 : Simulado

- As classes Carro, Caminhao e Motocicleta devem herdar de VeiculoBase e implementar suas próprias versões do método calcularPreco, usando sobrescrita de métodos para adaptar o cálculo de preço conforme as regras específicas para cada tipo de veículo:
- Carro: Preço de locação = precoBase.
- Caminhao: Preço de locação = precoBase + (precoBase * taxaCarga), onde taxaCarga é um valor fornecido.
- Motocicleta: Preço de locação = precoBase + taxaSeguro, onde taxaSeguro é um valor fixo fornecido.

AC2 : Simulado

- Adicione uma sobrecarga do método calcularPreco na classe VeiculoBase que aceita um parâmetro opcional desconto (do tipo double).
- Esta versão sobrecarregada do método deve subtrair o desconto do preço final de locação.

AC2 : Simulado

- Crie uma interface chamada Beneficios que declare os seguintes

métodos:

- `calcularDesconto(double percentual)`: método que deve calcular o desconto baseado em um percentual.
- As classes `Caminhao` e `Motocicleta` devem implementar a interface `Beneficios` e definir a lógica para calcular o desconto com base nos valores recebidos como parâmetro.

AC2 : Simulado

- Implemente uma nova classe chamada `Locadora` que possui os seguintes atributos:
- `nome` (do tipo `String`)
- `veiculos` (uma coleção do tipo `ArrayList<VeiculoBase>` para armazenar os veículos).

AC2 : Simulado

- Adicione métodos na classe `Locadora` para:
- Adicionar um veículo à locadora.
- Remover um veículo da locadora.
- Calcular a receita total da locadora, somando o preço de locação de todos os veículos na lista de veículos usando o método `calcularPreco` de cada veículo.

AC2 : Simulado

- Demonstre o uso de polimorfismo ao criar um método na classe `Locadora` chamado `imprimirPrecosVeiculos`, que percorre a lista de veículos e imprime a marca, modelo e preço de locação de cada um.
- Este método deve ser capaz de lidar com qualquer objeto que seja do tipo `VeiculoBase`, independentemente da subclasse específica (`Carro`, `Caminhao` ou `Motocicleta`), utilizando a técnica de polimorfismo.

AC2 : Simulado

- Exemplo de Uso:
- Um carro com marca "Ford", modelo "Fiesta" e preço base de 100.
- Um caminhão com marca "Volvo", modelo "FH" e preço base de 200, e taxa de carga de 10%.
- Uma motocicleta com marca "Honda", modelo "CG" e preço base de 50, com uma taxa de seguro de 20.
- O sistema deve ser capaz de calcular corretamente os preços de locação de cada tipo de veículo usando sobrecarga de métodos e encapsulamento para garantir a integridade dos dados.