

## 1. EXERCÍCIO GESTÃO DE FROTA

1. O projeto FleetManagement visa a criação de um sistema de gestão de uma frota de veículos. O objetivo principal é conceber um conjunto de classes que representem diferentes tipos de veículos, incluindo classes abstratas e concretas. O sistema deve permitir a criação de veículos específicos, geri-los numa frota e apresentar informação relevante. Após a conclusão, deverá ter um sistema FleetManagement funcional que demonstra a utilização de classes abstratas, polimorfismo e herança na gestão de uma frota diversificada de veículos.

Neste projeto, os veículos são categorizados em dois tipos principais: Car e Motorcycle. Ambos os tipos de veículos partilham atributos comuns, como o Model, o Manufacturer e o Year. No entanto, cada tipo tem propriedades únicas; os carros têm uma propriedade para o NumberOfDoors, enquanto os motociclos têm uma propriedade que indica se têm um carro lateral (HasSideCar).

Os veículos têm um método Start e Stop que são responsáveis pelo arranque/paragem do motor de um veículo. As especificidades de como o motor arranca/para podem variar entre diferentes tipos de veículos, como carros e motociclos. Neste projeto, assuma apenas que é apresentado uma *string* a especificar como é efetuado o arranque/paragem para cada veículo em específico, e.g., para o Car “Car engine started!” para identificar que um carro iniciou a marcha. Os veículos devem conter um método DisplayVehicleInfo que especifique as características destes.

Uma classe FleetManager é necessária para gerir uma coleção de veículos, permitindo a **adição** de novos veículos à frota e **exibindo** informações sobre todos os veículos da frota. Deve ser permitida a adição de, no máximo, 10 veículos.

Após a conclusão, deverá ter um sistema funcional que demonstra a utilização de classes abstratas, polimorfismo e herança na gestão de uma frota diversificada de veículos. Para testar o sistema implementado, crie vários veículos e adicione-os FleetManager. Após a adição, exiba todos os veículos adicionados.

2. De modo a verificar a sua implementação e os conhecimentos adquiridos, analise o código do disponibilizado. Preste atenção na estrutura, nas declarações de classe e nos métodos.

- a. Identifique a classe abstrata:
  - i. Que classe é declarada como abstrata no código?
  - ii. Qual é o objetivo desta classe abstrata?
- b. Identifique classes concretas:
  - i. Identifique duas classes concretas que estendem a classe abstrata.
  - ii. Liste seus nomes e descreva brevemente suas características específicas.
- c. Identifique um método na classe abstrata que está marcado como abstrato. Como o polimorfismo é demonstrado neste código? Forneça um exemplo.
- d. Como é que a herança desempenha um papel neste projeto? Forneça um exemplo de como uma classe concreta herda da classe abstrata.
- e. Imagine um novo tipo de veículo e crie uma nova classe para ele. Certifique-se de que esta classe estende a classe abstrata `Vehicle` e fornece as suas implementações.
- f. Suponha que quer adicionar o seu novo tipo de veículo à frota. Descreva os passos que daria no código para integrar a sua nova classe com o `FleetManager` existente.
- g. Explique o output:
  - i. Observe o código no método `Main` da classe `Program`.
  - ii. Preveja a saída que será exibida quando o programa for executado.
- h. Por que é benéfico ter uma classe abstrata no projeto `FleetManagement`?
- i. Como é que o polimorfismo aumenta a flexibilidade do sistema `FleetManagement`?
- j. Faça um resumo dos principais conceitos aprendidos neste exercício.
- k. Reflita sobre como a compreensão de classes abstratas, polimorfismo e herança é essencial no desenvolvimento de software do mundo real.

**Notas para os estudantes:** Reserve um tempo para ler e entender cuidadosamente cada parte do código. Sinta-se à vontade para experimentar modificações ou funcionalidades adicionais para aprofundar a sua compreensão. Esta ficha de trabalho foi concebida para encorajar os alunos a trabalhar ativamente com o código fornecido, identificar conceitos-chave e aplicar os seus conhecimentos para alargar o projeto. O seu objetivo é reforçar a compreensão das classes abstratas, do polimorfismo e da herança num contexto prático.