

UPskill – Java+.NET

Programação Orientada a Objetos – Agregação e Composição

EXERCÍCIO 1 – TAXIS

Esta ficha de trabalho tem como objetivo consolidar e aprofundar os conhecimentos adquiridos até este momento aplicados no universo dos transportes efetuados por táxis. Deverá implementar uma solução considerando que:

- a. São disponibilizadas 3 interfaces que deve utilizar no seu código:
 - i. IDisplayable – contem um método para a apresentação das informações das classes que a implementam.

```
public interface IDisplayable {  
    void DisplayInfo();  
}
```

- ii. IOperatable – contem os métodos necessários para que um veículo e motor arranque e pare, e para verificar se se encontra em andamento.

```
public interface IOperatable {  
    bool IsRunning { get; }  
    void Start();  
    void Stop();  
}
```

- iii. IVehicle – Contem as propriedades e métodos necessários para a criação de um veículo.

```
public interface IVehicle : IDisplayable, IOperatable {  
    string Color { get; }  
    string Manufacturer { get; }  
    string Model { get; }  
    string PlateNumber { get; }  
    string VIN { get; }  
    int Year { get; }  
    bool Equals(object? obj);  
    int GetHashCode();  
}
```

- b. Um veículo:
 - i. implementa a interface IVehicle.
 - ii. tem um motor.

- iii. embora `IOperatable`, por só não sabe arrancar ou parar.
 - iv. tem um `Vehicle Identification Number (VIN)` que o distingue de todos os outros.
 - c. Um motor:
 - i. implementa a interface `IDisplayable`.
 - ii. implementa a interface `IOperatable`. O método `Start` deve permitir que o motor arranque, tornando a propriedade `IsRunning` como `true`, e o método `Stop` deve parar o motor fazendo com que o `IsRunning` seja `false`.
 - iii. têm fabricante, quantidade de cavalos, tipo de combustível, número de cilindros, e número de série.
 - d. Um táxi:
 - i. é um veículo com propriedades extra como número de táxi, licença, número máximo de passageiros.
 - ii. tem um condutor. Deve ser permitido trocar de condutor se o táxi não estiver em andamento.
 - iii. pode arrancar e parar (`IOperatable`), se tiver um condutor.
 - iv. deve ser possível adicionar e remover passageiros, se o táxi não tiver em andamento.
 - v. deve ser possível adicionar passageiros, se o táxi não estiver cheio.
 - vi. deve ser possível obter o número de passageiros corrente.
 - vii. deve imprimir na consola todos os dados relativos ao táxi (`IDisplayable`).
 - e. Um passageiro é uma pessoa com um `Value Added Tax (VAT)` que o distingue de todos os outros.
 - f. Um condutor é uma pessoa com carta de condução.
 - g. Uma pessoa tem nome e implementa a interface `IDisplayable`.
 - h. Deve criar uma classe estática com um método `DisplayWithHeaderAndFooter`. Este método deve receber um `IDisplayable` e colocar um header antes do texto e um footer depois do texto a apresentar.
2. Teste a implementação das suas classes no programa principal.
 3. De modo a verificar a sua implementação e os conhecimentos adquiridos, analise o código disponibilizado. Preste atenção na estrutura, nas declarações de classe e nos métodos.

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.