

TP Garage – Partie 1

Version : 1.0

L'objectif de ce TP est de créer une mini application permettant la gestion d'un garage.

Les objectifs étant :

- Modéliser une application en utilisant les notions avancées de POO
- Mettre en place les notions de classe et méthode abstract
- Mettre en place les notions d'interface

Première étape :

On désire développer une application permettant de gérer le parc de véhicules qu'un garage pourra proposer à la vente. L'application devra permettre de gérer aussi bien des voitures, que des camions et des motos.

Une voiture sera caractérisée par son nom, son prix, sa marque, un moteur, des options, une puissance en chevaux fiscaux, un nombre de porte, un nombre de siège et une taille de coffre (exprimé en m3).

Un camion sera caractérisé par son nom, son prix, sa marque, un moteur, des options, un nombre d'essieux, un poids de chargement et un volume de chargement.

Une moto sera caractérisée par son nom, son prix, sa marque, un moteur, des options et une cylindrée.

Pour chaque véhicule, on devra pouvoir effectuer les actions suivantes :

- afficher l'ensemble des informations d'un véhicule.
- Ajouter une option au véhicule
- Connaitre la marque du véhicule
- Connaitre le moteur du véhicule
- Connaitre les options du véhicule
- Connaitre le prix du véhicule (hors options et taxe)
- Calculer la taxe sur le véhicule
- Connaitre le prix total du véhicule

Chaque véhicule disposera également d'un « Id » qui sera automatiquement généré à partir de 1. Vous pouvez utiliser pour cela une variable static permettant de faire la génération des Id avec une incrémentation de cette variable à chaque ajout d'un véhicule.

La taxe sur les véhicules se calcule de la manière suivante :

- Pour une voiture, on multiplie la puissance en chevaux fiscaux par 10 €,
- Pour un camion, on prend le nombre d'essieux et on multiplie par 50 €,
- Pour une moto, on multiplie la cylindrée par 0,3 euros et on prendra la partie entière.

Un moteur sera caractérisé par un type (diesel, essence, hybride, électrique) et d'une puissance. On devra pouvoir afficher l'ensemble des informations d'un moteur.

Une option sera caractérisée par un nom et un prix. On devra pouvoir afficher l'ensemble des informations d'une option.

On devra pouvoir ajouter aux garages des véhicules, trier les véhicules suivant leur prix, et afficher l'ensemble des informations concernant les véhicules du garage.

Le tri sera mis en place avec l'implémentation de la classe `IComparable`. Vous trouverez un exemple d'implémentation de cette classe dans le support de cours « Programmation orientée objet C# Notions avancées ».

Travail à effectuer :

1. Modéliser la gestion du garage par un diagramme de classes. Vous devrez utiliser les notions avancées de POO avec l'utilisation de class abstraites et d'interface.
2. Développer l'ensemble des classes de l'application
3. Créer un jeu de tests de véhicules afin de les ajouter au garage et afficher ensuite tous les véhicules de ce garage. Tester également le tri et afficher les véhicules une fois trié.