Voiture On Line

# Objectifs

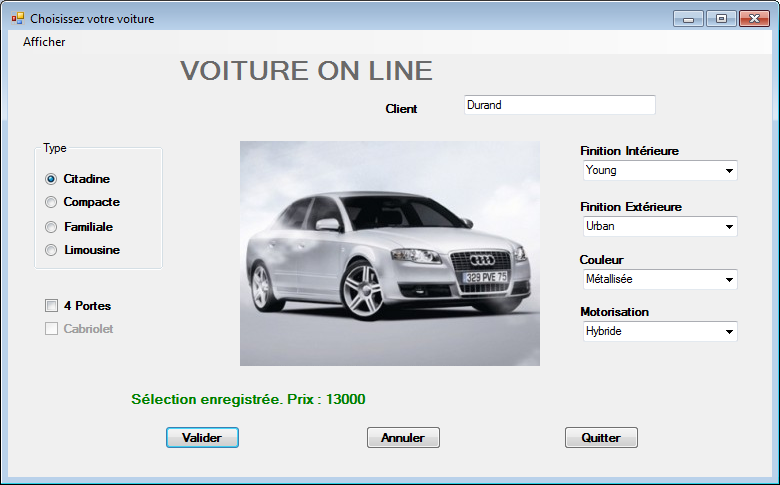
* Créer et utiliser des classes dérivées avec C#
* Créer et utiliser des méthodes virtuelles
* Faire collaborer des objets
* Créer et utiliser une collection d’objets.

# Contexte

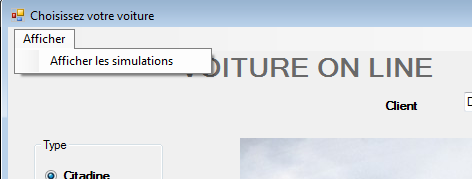
"Voiture on Line" est une entreprise qui se charge de rechercher des voitures pour ses clients. Il vous est demandé de réaliser une application à l'usage des commerciaux, permettant de regrouper et de visualiser les demandes des clients.

# Travail à faire

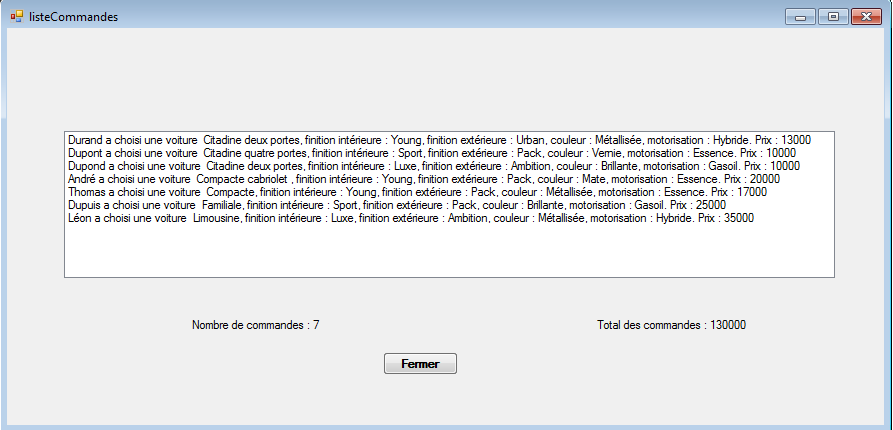
Cette application est très similaire au cas "IKEBANA" que vous avez déjà réalisé. Vous pourrez sans problème vous en servir, et même faire des "copier-coller" pour gagner en efficacité.



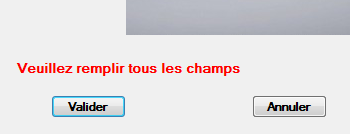
*Fig. 1 Aperçu du formulaire d'entrée dans l'application*



*fig. 2 Détail du menu "Afficher"*



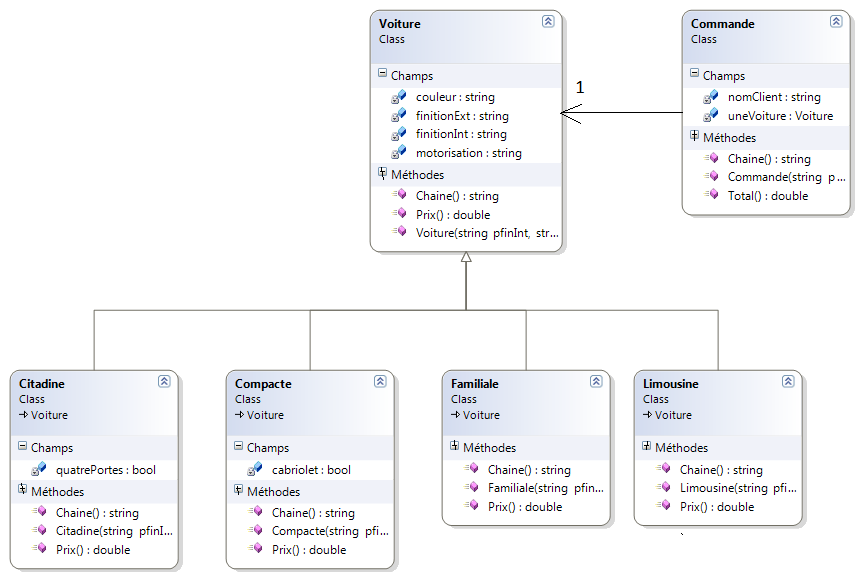
*Fig 3. Formulaire affichant les commandes saisies*

**

*Fig 4. Détail du contrôle de saisie du formulaire 1*

# Conception

## Diagramme de classes édité par Visual Studio



## Objectifs des méthodes

La méthode Chaine est une méthode virtuelle dont l’implémentation dans les classes doit permettre d’obtenir l’affichage donné par la figure 3.

De même la méthode Prix est une méthode virtuelle à implémenter selon les règles suivantes :

* Le prix de base d'une voiture est de 10 000 €.
* La Citadine est affichée au prix de base. En version, 2 portes, elle coûtera 2000 € de moins.
* La Compacte coûtera 7 000 € de plus que le prix de base. En version cabriolet, son prix sera encore augmenté de 3000 €.
* La Familiale coûtera 13 000 € de plus que le prix de base.
* La Limousine coûtera 20 000 € de plus que le prix de base.
* Pour toutes les versions, en motorisation Hybride, le prix de base sera augmenté de 5 000 €.
* Pour toutes les versions, en motorisation Gasoil, le prix de base sera augmenté de 2 000 €.

## Etapes de travail et barème

Il est important, à chaque étape et **très régulièrement de compiler puis d'exécuter** vos programmes.

# Test

Vous pourrez réaliser le test suivant pour valider la méthode Prix() :

[TestMethod]

public void TestPrixVoiture()

{

Citadine uneC = new Citadine("Young", "Urban", "Métallisée", "Hybride", false);

Citadine uneC2 = new Citadine("Young", "Urban", "Métallisée", "Essence", true);

Citadine uneC3 = new Citadine("Young", "Urban", "Métallisée", "Gasoil", false);

Compacte uneCo1 = new Compacte("Young", "Urban", "Métallisée", "Essence", true);

Compacte uneCo2 = new Compacte("Young", "Urban", "Métallisée", "Essence", false);

Familiale uneF = new Familiale("Young", "Urban", "Métallisée", "Gasoil");

Limousine uneL = new Limousine("Young", "Urban", "Métallisée", "Hybride");

Assert.AreEqual(13000, uneC.Prix());

Assert.AreEqual(10000, uneC2.Prix());

Assert.AreEqual(10000, uneC3.Prix());

Assert.AreEqual(20000, uneCo1.Prix());

Assert.AreEqual(17000, uneCo2.Prix());

Assert.AreEqual(25000, uneF.Prix());

Assert.AreEqual(35000, uneL.Prix());

}

|  |  |  |
| --- | --- | --- |
| Barème Voitures on Line | |  |
|  | |  |
| formulaire choix de voiture, création et nommage des contrôles, boutons fermer et annuler | 4 | |
| classe Voiture, attributs et constructeur, | 2 | |
| classes filles, attributs et constructeur | 2 | |
| créer un attribut privé qui est une collection de Commande dans la classe "formulaire choix de voiture" | 1 | |
| le bouton Valider : 1. instancier selon le radioButton sélectionné un objet Citadine, ou Compacte ou… | 2 | |
| le bouton Valider : 2 instancier un objet Commande | 1 | |
| le bouton Valider : et rajouter cet objet commande dans la collection de commande | 1 | |
| Créer le formulaire "affichage des commandes" , bouton Fermer | 3 | |
| et modifier le constructeur en lui ajoutant la collection en paramètre | 1 | |
| instancier et afficher le formulaire "affichage des commandes" sur clic sur le menu du formulaire 1. (fig 2) | 1 | |
| implémenter les méthodes Chaine() des différentes classes | 2 | |
| remplir la listBox du formulaire 2. | 2 | |
| Vérifier que le formulaire 2 s'ouvre et affiche des informations |  | |
| implémenter les méthodes Prix() des différentes classes | 3 | |
| Vérifier que les informations données dans la ListBox correspondent à la figure 3. |  | |
| **Finitions dans le formulaire choix de voiture** : chekBox actifs selon les cas, | 1 | |
| Afficher le message en vert : commande enregistrée | 1 | |
| Afficher le message en rouge si tous les champs ne sont pas remplis. | 1 | |
| Vider les champs lorsque la commande est remplie | 1 | |
| **Finitions dans le formulaire liste des commandes** : labels affichant le nombre de commandes et le total des commandes | 1 | |
|  | 30 | |