


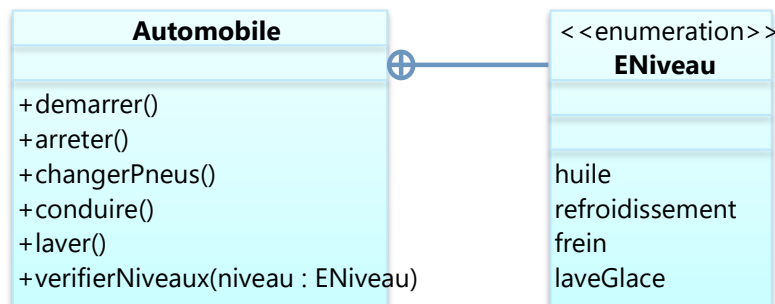
TD 09 – Premiers Principes de Conception Orientée Objet

Exercice 1 Une automobile

1.1 Version 1


Nous voulons modéliser une application centrée sur les automobiles. Dans ce logiciel, il est possible de réaliser un certain nombre d'actions sur les automobiles. Notamment, l'utilisateur peut démarrer ou arrêter le moteur, changer les pneus, laver la carrosserie, la conduire et vérifier les niveaux de liquides (huile moteur, liquide de refroidissement, liquide de frein, lave glace).

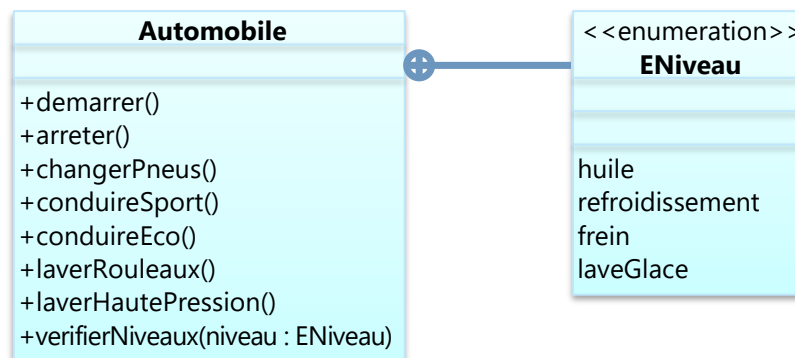
 Modélisez cette classe « automobile » à l'aide d'un diagramme UML.



1.2 Version 2

Votre client est super content de votre logiciel et vous demande quelques fonctionnalités supplémentaires. A présent, il doit être possible de choisir le mode de conduite (sportive ou économique) et de laver la voiture de deux façons différentes : un lavage automatique à rouleaux et un lavage manuel à haute pression.

 Modifier votre modélisation pour prendre en compte ces changements.



 Quelles sont les implications sur le code source ?

- ➡ Nécessité de casser les fonctions conduire et laver, de modifier la classe Automobile avec beaucoup de modifications et des répercussions potentielles sur les autres méthodes.
- ➡ Qu'aurions-nous du faire pour éviter tout cela ? → Appliquer un principe de conception : le Principe de Responsabilité Unique (*Single Responsibility Principle*) {NB : le terme anglais est important}. Distribuer alors la fiche suivante.