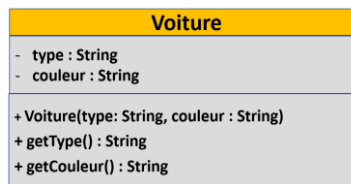


## TD 6 Les Interfaces

Dans ce TD nous allons nous concentrer sur les interfaces.

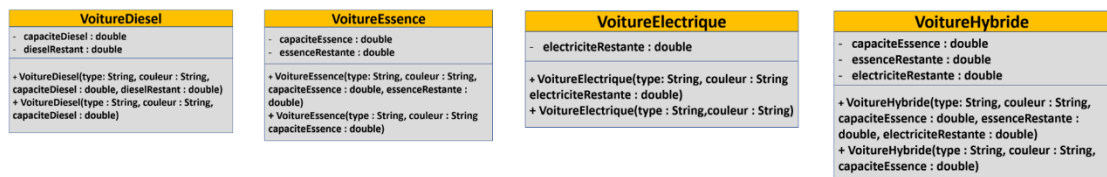
Dans un TD précédent, nous avons vu des notions de l'héritage appliquées aux véhicules. Aujourd'hui nous allons reprendre quelques éléments de ce TD, mais en ajoutant la notion d'interfaces.

Nous allons commencer sur une classe Voiture, avec ce diagramme de classe :



Nous allons considérer trois types de voiture : les voitures Diesel (qui vont prendre du Diesel comme carburant), les voitures Essence, les voitures électriques et les voitures hybrides (qui prennent de l'essence ET de l'électricité). Pour les voitures thermiques (Diesel, Essence), nous allons pouvoir faire le plein en remplissant leur réservoir avec le carburant utilisé. De même pour le réservoir essence des voiture hybrides. Pour les voitures électriques et hybrides, nous allons considérer leur charge électrique comme un pourcentage de la valeur maximale pour la batterie : par exemple 100% (maximum), 90%, 18.3%, etc. D'habitude, on nous conseille de ne jamais recharger au maximum une voiture électrique, sinon de la recharger à 90% tous les jours.

1. Voici des diagrammes de classe pour les classes VoitureDiesel, VoitureEssence, VoitureElectrique, VoitureHybride, qui héritent de la classe Voiture.

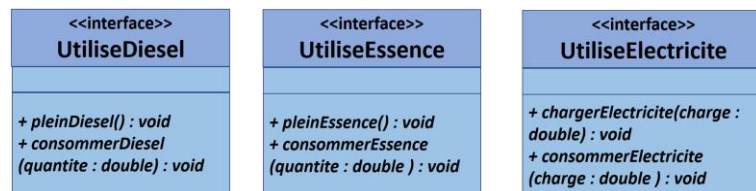


- Représentez la relation d'héritage dans un diagramme.
- Ecrivez la classe Voiture.
- Pour chacune des classes VoitureDiesel, VoitureEssence, VoitureElectrique et VoitureHybride, écrivez leur premier constructeur.
- Pour le deuxième constructeur des classes VoitureDiesel et VoitureEssence, la quantité de carburant restant (Diesel ou Essence) sera mise à la capacité maximale du réservoir. Pour la classe VoitureElectrique, l'électricité restante est mise à la capacité pleine, c.à.d. 100%. Finalement, pour les voitures hybrides on combine les caractéristiques d'une

voitureEssence et une voitureElectrique. Ecrivez ce deuxième constructeur pour les classes VoitureEssence, VoitureElectrique, VoitureHybride.

Finalement, nous prenons 4 voitures spécifiques : une Renault Clio (Diesel, capacité réservoir 55 l), une Tesla Model S (Electrique), une Peugeot 208 (Essence, capacité réservoir 50 l) et une KIA Optima Hybride (hybride, capacité réservoir 55l). Vous pouvez choisir les couleurs de votre choix pour ces voitures.

- Dans une classe MonApplication, dans une méthode main, créez 4 objets qui représentent les quatre voitures ci-dessus. Les quantités de carburant/électricité restant(e) seront mises à 10.
  - Affichez les couleurs de toutes les quatre voitures dans la méthode main.
2. Mettre le mauvais carburant dans une voiture peut causer des problèmes sérieux pour une voiture. Il est essentiel alors de ne jamais le faire.  
Nous allons s’imaginer ces trois interfaces : UtiliseEssence, UtiliseDiesel, UtiliseElectricite, dont les diagrammes de classe sont donnés ci-dessous.



- Est-ce qu’une classe peut hériter de plusieurs classes ?
- Est-ce qu’une classe peut implémenter plusieurs interfaces ?
- Pour les classes données à l’exercice 1, indiquez quelle(s) classes implémentent quelle(s) interfaces. Comment allez-vous indiquer ceci ?
- Mettez à jour vos diagrammes de classe par rapport aux méthodes ajoutées
- Ecrivez chacune des interfaces données.
- Pour les classes VoitureDiesel, VoitureEssence, VoitureElectrique, VoitureHybride, écrivez les méthodes qui concrétisent les méthodes des interfaces que ces classes implémentent. Le paramètre de la méthode chargerElectricite représente la charge maximale jusqu’à laquelle on veut charger la voiture. Au contraire, le paramètre de la méthode consommerElectricite représente la quantité d’électricité qu’on a dépensé.
- Dans la classe MonApplication, écrivez du code permettant à chaque voiture donnée dans l’exercice précédent de faire son plein en carburant, et de se recharger jusqu’à une charge maximale de 90. La voiture hybride fera son plein en carburant ET se rechargera jusqu’à 90%.

- Soit le code suivant, qui sera intégré dans votre méthode main :

```
VoitureDiesel fiatPunto = new VoitureDiesel("Fiat Punto", "rouge", 45, 30);
fiatPunto.pleinDiesel();
fiatPunto.recharge(90);
```

Est-ce que ce code compile ? Pourquoi (pas) ?

- Même question pour le code suivant :

```
Voiture seatIbiza = new VoitureDiesel("SEAT Ibiza", "bleue", 55, 20);
seatIbiza.pleinDiesel();
```

3. Pensons maintenant à une station d'essence, qui permet la recharge de tous ces types de voitures et de carburants. Cette station a une pompe à Essence, une à Diesel, et une station de recharge électrique.

Dans votre classe MonApplication, nous parlerons des voitures suivantes (couleurs de votre choix pour chacune) :

- Une Renault Clio (Diesel, capacité 55, actuellement à 10)
- Une SEAT Ibiza (Essence, capacité 55, actuellement à 10)
- Une Ford Mondeo (Essence, capacité 65, actuellement à 10)
- Une Peugeot 208 (Essence, capacité 50, actuellement à 10)
- Une Tesla Model S (Electrique, actuellement à 10% charge)
- Une Tesla Model X (Electrique, actuellement à 15% charge)
- Une Toyota Yaris Hybride (Hybride, capacité 50l, actuellement à 20, charge actuelle 10%)
- Une KIA Optima Hybride (Hybride, capacité 55, actuellement à 10, charge actuelle 10%)
- Nous allons avoir trois tableaux qui stockeront les voitures compatibles pour chacune des pompes/stations de recharge de notre station. Le premier tableau s'appellera pompeDiesel, le deuxième s'appellera pompeEssence, le troisième, stationElectrique. Les éléments de ces trois tableaux seront du type de leurs interfaces (pompeDiesel -- UtiliseDiesel[], pompeEssence -- UtiliseEssence[], stationElectrique -- UtiliseElectricité). Déclarez et initialisez ces trois tableaux. Les hybrides feront leur plein en carburant ET se rechargeront.
- Quels sont les éléments de chaque tableau ?
- Pour chaque élément de chaque tableau faites son plein en carburant et/ou rechargez-le à 90%.