

## Cas Pratique : Gestion d'un Parc de Véhicules

### Objectif :

Créer un système de gestion de véhicules utilisant la programmation orientée objet (POO) avec JavaScript. Ce cas pratique met en œuvre l'héritage et réutilise les notions de propriétés publiques et privées, méthodes, getters, setters, et méthodes statiques.

### Étape 1 : Création de la Classe de Base Vehicule

1. **Créez une classe Vehicule avec les propriétés suivantes :**
  - marque (publique)
  - #kilometrage (privée)
  - annee (publique)
2. **Ajoutez un constructeur pour initialiser les propriétés marque, #kilometrage, et annee.**
3. **Ajoutez les méthodes suivantes à la classe Vehicule :**
  - afficherInfos() : affiche les informations du véhicule (marque, kilometrage, annee).
  - Getter et setter pour kilometrage :
    - get kilometrage() : retourne le kilometrage.
    - set kilometrage(nouveauKilometrage) : modifie le kilometrage.
  - ajouterKilometres(km) : ajoute les kilomètres spécifiés au kilométrage actuel.
4. **Ajoutez une méthode statique creerVehiculeDeBase qui crée et retourne une instance de Vehicule avec les valeurs suivantes : marque = "Generic", kilometrage = 0, annee = 2022.**
5. **Testez la classe Vehicule en créant des instances et en utilisant les méthodes et propriétés.**

### Étape 2 : Création des Classes Dérivées

1. **Créez une classe Voiture qui hérite de la classe Vehicule.**
  - Ajoutez une propriété publique nombrePortes.
2. **Ajoutez un constructeur à la classe Voiture pour initialiser les propriétés marque, kilometrage, annee, et nombrePortes.**
  - Utilisez super() pour appeler le constructeur de la classe parente Vehicule.
3. **Ajoutez une méthode afficherInfos() à la classe Voiture qui surcharge celle de la classe Vehicule pour inclure le nombrePortes.**

4. **Testez la classe Voiture en créant des instances et en utilisant les méthodes et propriétés.**
5. **Créez une classe Moto qui hérite de la classe Vehicule.**
  - Ajoutez une propriété publique typeGuidon.
6. **Ajoutez un constructeur à la classe Moto pour initialiser les propriétés marque, kilometrage, annee, et typeGuidon.**
  - Utilisez super() pour appeler le constructeur de la classe parente Vehicule.
7. **Ajoutez une méthode afficherInfos() à la classe Moto qui surcharge celle de la classe Vehicule pour inclure le typeGuidon.**
8. **Testez la classe Moto en créant des instances et en utilisant les méthodes et propriétés.**

### **Étape 3 : Gestion du Parc de Véhicules**

1. **Créez une classe Parc pour gérer une collection de véhicules.**
  - Ajoutez une propriété publique vehicules qui est un tableau vide.
2. **Ajoutez une méthode ajouterVehicule(vehicule) qui ajoute un véhicule au tableau vehicules.**
3. **Ajoutez une méthode afficherTousLesVehicules() qui affiche les informations de tous les véhicules dans le parc.**
4. **Testez la classe Parc en ajoutant des instances de Voiture et de Moto et en affichant les informations de tous les véhicules.**