

 OFPPT	مكتب التكوين المهني وإنعاش الشغل
	Office de la Formation Professionnelle et de la Promotion du Travail

**Filière: Dev 101/102**

**Module: Programmation Orientée Objet**

**TP 9**

**Exercice 1:**

Créer un nouveau projet **Gestion-vehicules**

Un **Véhicule** est caractérisée par un code et dateService

La classe **voiture** hérite de la classe véhicule. La classe voiture a pour caractéristiques, la marque et puissance

1. Créer la classe véhicule
2. Les attributs de la classe véhicule sont protected
3. Ajouter les propriétés relatives à chaque attribut
3. Créer le constructeur d'initialisation à la classe **véhicule**.
4. Ajouter la méthode **Afficher()** à la classe **véhicule** qui **retourne** toutes les informations séparées par tabulation
5. Créer la classe voiture, les attributs sont privés
6. Créer le constructeur d'initialisation (qui initialise tous les attributs) qui hérite du constructeur de la classe mère
7. Ajouter les propriétés correspondant à chaque attribut en lecture seule
8. Redéfinir la méthode **Afficher()** dans la classe **voiture** de manière à ce qu'elle affiche aussi les données d'une voiture.
9. Créer un objet véhicule, et appeler les méthodes déjà définies pour des éventuels tests.  
Faire la même chose pour la classe voiture

**Exercice 2:**

Créer un nouveau dossier **GestionPersonne**

- 1) Créer la classe **personne** caractérisé par nom, prénom et adresse (protected)
- 2) ajouter le constructeur d'initialisation
- 3) ajouter les propriétés nécessaires
- 4) ajouter la méthode **présentation()** qui **affiche** les données de la personne séparés par des espaces
- 5) Créer la classe **formateur** qui **hérite** de la classe **personne** qui est caractérisé par une matricule, secteur et groupes (listes de groupes). Les attribus sont privés
- 6) ajouter le constructeur d'initialisation en héritant du constructeur de la classe mère
- 7) ajouter les getters et setters nécessaires
- 8) ajouter une méthode **ListeGroupes()** permettant d'afficher tous les groupes assurés par le formateur
- 9) redéfinir la méthode **présentation()** de manière à ce qu'elle affiche aussi la matricule, le secteur et les groupes
- 10) créer la classe **Etudiant** qui **hérite** de la classe **personne**. Un étudiant est caractérisé par un numéro d'inscription, une filière et une moyenne (les attributs sont privés)
- 11) ajouter le constructeur d'initialisation
- 12) ajouter les getters nécessaires
- 13) ajouter la méthode **mention()** permettant de retourner la mention du stagiaire selon sa moyenne (T Bien si moyenne > 16, Bien si moyenne entre 14 et 16 , Assez Bien si moyenne entre 12 et 14, passable si moyenne entre 10 et 12)
- 14) redéfinir la méthode **présentation()** pour qu'elle affiche toutes les informations du stagiaire ainsi que son numéro d'inscription, sa filière , sa moyenne et sa mention.