

---

# Programmation Orientée Objet

TP2 : Héritage et Polymorphisme (Partie 1)

2022-2023

---

## Exercice 1 :

Dans cet exercice, nous allons implémenter la gestion d'une simple bibliothèque contenant des livres où chaque livre doit avoir un auteur. Pour cela on commencera à créer la classe `Personne` (qui représente les auteurs et aussi les acheteurs des livres). Cette classe est caractérisée par un nom un prénom et l'année de naissance.

Par la suite créer la classe `Document` de sorte à ce que chaque document doit posséder un nom et au moins un auteur (au maximum 5 auteurs).

- Ajouter la méthode `NbrAuteur()` dans la classe `Document` qui retourne le nombre des auteurs d'un document.
- Ajouter aussi la méthode `toString()` pour afficher le nom et les auteurs d'un document.

Maintenant, créer la classe `Livre` qui sera défini comme une sous classe de la classe `Document`. La classe `Livre` possède le ISBN, le Prix, et l'Année comme attributs.

- Redéfinissez la méthode `toString()` dans la classe `Livre` pour afficher tous les informations d'un Livre à l'exception de son prix.
- Ajouter une méthode appelée `PrixVente(Personne p)` qui affiche le prix de vente de chaque Livre, à savoir que le prix sera réduit de 20% pour les personnes moins de 20 ans. De plus, il faut tester si la personne passée en paramètre est l'auteur du livre, le prix sera affiché 0 de sorte que le livre est toujours gratuit pour son auteur(s).

## Exercice 2 :

L'objectif de cet exercice est de programmer le calcul des salaires hebdomadaires des employés d'une entreprise. Cette entreprise comporte plusieurs types d'employés :

- Des **employés développeurs** qui sont payés suivant le nombre d'heures qu'ils ont travaillé dans la semaine (40 heures). Le prix de l'heure est fixé à 500 DA, et le prix des heures supplémentaires et 700 DA pour chaque heure.
- De l'autre côté il y a les **employés de commerciaux**, payés de la même façon, mais le prix de l'heure est fixé à 350 DA et l'heure supplémentaire à 500 DA.

Modélisez cette situation à l'aide de classes qu'il faut en suivant les étapes suivantes :

- Créez la classe **Employé** qui possède le nom, le prénom et l'année de recrutement comme attributs ainsi que le `NbrHeures` et `NbrHeuresSupp`.
- Créez la méthode `toString()` pour afficher les informations d'un employé.
- Créez la classe **Développeur** qui hérite de la classe `Employé`.
- Ajoutez la méthode `CalculPaie()` qui calcule la première paie (sans les heures supp).
- Ajoutez la méthode `CalculHeuresSupp()` qui calcule le bénéfice de l'employé par rapport à ses heures supplémentaire de travail.
- Après, ajoutez la méthode `CalculPaieFinal()` qui calcule la paie finale de l'employé en ajoutant la première paie et les heures supplémentaires. De plus il faut considérer que l'employé bénéficie de 2000 DA supplémentaire sur sa paie pour chaque année d'ancienneté dans l'entreprise.
- Faites la même chose avec la classe **Commerciaux**.
- Redéfinissez la méthode `toString()` dans `Développeurs` et `Commerciaux` pour afficher le nom, le prénom et la paie finale des employés.

### Exercice 3 :

Définissez une classe **Véhicule** qui a pour attributs des informations valables pour tout type de véhicule à savoir : la marque, la date d'achat, et le prix d'achat du véhicule.

Définissez une méthode publique afficher() qui affiche les valeurs des attributs.

Définissez deux classes **Voiture** et **Avion**, héritant de la classe Véhicule.

La classe Voiture possède NbrPortes, Puissances et Kilométrage comme attributs.

De l'autre côté la classe Avion est caractérisée par les heures de vol.

- Ajoutez une méthode calculprix() dans la classe véhicule, qui calcule le prix d'une voiture d'occasion où le prix sera réduit de 5% pour chaque année d'ancienneté du véhicule.
- Redéfinissez cette méthode dans la classe Voiture pour considérer le kilométrage, de sorte que le prix du véhicule réduit de 10% pour chaque 100 mille kilomètres que la voiture a parcourus.
- Redéfinissez aussi la méthode calculeprix() dans Avion, où le prix de l'Avion sera réduit de 8% pour chaque 5000 heures de vol.
- Redéfinissez la méthode afficher() dans Voiture et Avion afin de pouvoir afficher leurs informations.
- Redéfinissez la méthode equals() dans la classe Véhicule, afin de comparer si deux véhicules sont identiques.

