

## Programmation orientée objet Java

### TP2 : héritage et polymorphisme

#### Exercice 1.

Créer et tester la classe `Pointcol` vue en cours. Redéfinir les deux méthodes `affiche()` et `toString()`.

#### Exercice 2.

Une entreprise donnée dispose de deux catégories d'employés :

- Catégorie A (`CommissionEmployee`) : des employés payés en pourcentage (ratio) des ventes mensuelles
- Catégorie B (`BasePlusCommissionEmployee`) : des employés qui sont payés avec un salaire de base auquel est ajouté un pourcentage des ventes mensuelles.

Nous retenons les informations suivantes pour les employés.

Catégorie A :

- NNI, prénom, nom, ratio et montant de ventes d'un mois donné

Catégorie B :

- NNI, prénom, nom, ratio, montant de ventes d'un mois donné et salaire de base

Réaliser les deux classes correspondantes aux deux types d'employés en ajoutant les attributs et les méthodes nécessaires. Pour chacune de deux classes, ajouter une méthode appelée `revenu` qui calcule le revenu mensuel d'un employé.

Créer une classe `TestEmployee` permettant de tester les deux classes.

#### Exercice 3.

Nous proposons les modifications suivantes à l'énoncé de l'exercice précédent. En plus de deux catégories précédentes, on ajoute :

- Catégorie C (`SalariedEmployee`) : employés avec un salaire mensuel fixe.
  - Catégorie D (`HourlyEmployee`) : employés payés selon le nombre d'heures de travail avec une majoration 50% si le nombre dépasse 160.
1. Préciser les attributs et méthodes des différentes classes.
  2. Donner le diagramme de classe.
  3. Créer les différentes classes en redéfinissant les méthodes `toString()` et `revneu()`
  4. Créer une classe `TestEmployee` permettant de créer et afficher un tableau contenant des employés de différentes catégories.

#### Exercice 4.

On veut gérer le paiement d'employés (définis dans les exercices 1 et 2) et factures d'une entreprise. Pour une facture, nous retenons comme information, un identifiant, le prix unitaire d'un article, le nombre d'article.

1. Créer la classe `Invoice` (facture)
2. Créer une interface nommée `Payable` qui contient la méthode `getPaymentAmount()`.
3. Implémenter cette interface dans les classes `Invoice` et `SalariedEmployee`.
4. Ecrire une classe `TestPayment` qui crée un tableau hétérogène de factures et d'employés et affiche le résultat de la méthode `getPaymentAmount()`.