

**Travaux Dirigés n° 3 : Héritage**

*Objectifs : Comprendre le concept et l'utilité du mécanisme d'héritage. Savoir manipuler des références à des objets ayant un ancêtre commun grâce au polymorphisme de JAVA.*

## 1 Exercice 1 : villes et capitales

On souhaite écrire une classe `Ville` destinée à modéliser une ville. Les attributs d'une ville sont son nom (qui ne peut pas varier) et son nombre d'habitants (qui peut varier). Les villes sont de plus classées en trois catégories : "grande" pour les plus de 500 000 habitants, "moyenne" pour les plus de 100 000 mais moins de 500 000 habitants et enfin "petite" pour les autres.

1. Écrire une classe `Ville` conforme à la spécification. Les informations de nom et de nombre d'habitants devront impérativement être connues lors de la création d'instances de `Ville`. La classe devra respecter le principe d'encapsulation et proposer une méthode de signature `String afficheInformations()` qui renvoie une chaîne décrivant les informations d'une ville, par exemple `Orsay, 16236 habitants, petite ville`.
2. Une capitale est une ville contenant l'information du pays dans lequel elle se situe. Écrire une classe `Capitale` destinée à modéliser une capitale. La classe devra proposer une méthode de signature `String chaineInformations()` qui renvoie une chaîne décrivant les informations d'une capitale, par exemple `"Paris, 2101816 habitants, grande ville, capitale de France"`.
3. Écrire une classe `Principale` contenant seulement la méthode `main` où vous créerez deux objets villes de catégories différentes et un objet capitale, puis afficherez les informations de ces trois villes.

## 2 Exercice 2 : gestion d'employés

Une entreprise possède plusieurs types de collaborateurs :

- Les *employés* qui sont payés en fonction du nombre d'heures qu'ils ont travaillé dans la semaine. Ils sont payés à un certain tarif horaire et leurs heures supplémentaires (au delà de 35 heures) sont payées 25% de plus.
- Les *managers* qui sont payés de la même façon que les employés mais dont les heures supplémentaires sont payées 50% de plus que les heures normales.
- Les *commerciaux* qui sont payés une somme fixe, équivalente à 35 heures au tarif horaire des employés, à laquelle on ajoute 1% du chiffre d'affaire qu'ils ont fait dans la semaine.

Le travail demandé est de modéliser cette situation à l'aide de classes JAVA.

1. Définissez et décrivez une hiérarchie de classes sur papier (faites-la vérifier par l'enseignant avant de continuer).

2. Écrivez les trois classes `Employe`, `Manager` et `Commercial`, modélisant respectivement les employés, les managers et les commerciaux, sachant que :
  - Chaque collaborateur a un nom qui ne doit pas pouvoir être modifié.
  - Chaque classe doit disposer de deux constructeurs. L'un prend seulement en paramètre le nom du collaborateur, l'autre prend le nom et toutes les informations nécessaires au calcul du salaire de la semaine.
  - Dans chaque classe, le calcul du salaire se fait *via* une méthode dont la signature est *double calculSalaire()*.
3. Écrivez une classe `Gestion` comportant uniquement la méthode `main`. Dans cette méthode, vous créez plusieurs collaborateurs de plusieurs types et vous les enregistrerez tous dans un même tableau. Vous veillerez à utiliser les différents types de constructeur. La méthode `main` affichera le salaire hebdomadaire de tout le personnel dans une boucle parcourant le tableau des collaborateurs. Pour un collaborateur donné, l'affichage aura la forme : Salaire de Gates : 2008 euros.