

## TP 1 - Employés

Programmation Java : Restrictions d'accès, héritage et polymorphisme.

---

L'objectif de ce TP est d'implémenter les classes **Employe** et **Cadre** vues en TD.

Vous pouvez au choix travailler en ligne de commande, en utilisant les commandes `javac` et `java`, ou travailler en utilisant le logiciel **NetBeans** :

- commencez par créer un dossier personnel où vous mettrez l'ensemble de vos TP Java ;
- si vous travaillez en ligne de commande, créez un sous-dossier **TPemploye** ;
- si vous utilisez **NetBeans**, créez un projet **tpemploye** dans votre dossier personnel ;
- dans les deux cas, créez une classe **Principale** contenant la fonction `main` dans laquelle vous testerez vos différentes méthodes au fur et à mesure ;
- assurez-vous à la fin du TP d'avoir enregistré vos fichiers sources **.java** dans un répertoire non temporaire (c.à.d. personnel) et/ou sur une clé USB personnelle.

Les classes **Employe** et **Cadre** auront les structures de bases suivantes :

```
public class Employe{
    public static final boolean _HOMME = false;
    public static final boolean _FEMME = true;

    private boolean genre;
    private String nom;
    private Cadre superieur;

    ...
}
```

```
public class Cadre extends Employe {
    private Employe[] subordonnes;

    ...
}
```

Tous les employés doivent avoir les fonctionnalités suivantes :

- Q.1)** - Deux constructeurs publics suivants s'il y a ou non un supérieur.
- Q.2)** - Des accesseurs publics aux attributs `genre`, `nom` et `superieur`.
- Q.3)** - Une redéfinition de la méthode `public String toString()` pour qu'elle retourne le nom de l'employé. Pour un cadre, le nom sera précédé la formule de politesse *Mr* ou *Mme* suivant le genre (mais pas pour un simple employé).
- Q.4)** - Une méthode `boolean equals(Employe e)` pour tester l'égalité de deux objets **Employe**. Ces derniers sont considérés égaux s'ils ont même nom et même genre.
- Q.5)** - Dans la classe cadre, une méthode `void ajouteSubordonne(Employe e)` qui permet d'ajouter un nouvel employé au service de ce cadre. Attention, elle doit vérifier qu'il n'est pas déjà présent. Cette méthode doit être utilisée pour compléter le constructeur de la classe **Employe** ayant pour argument un supérieur.

- Q.6)** - Dans la classe cadre, une méthode `public void afficheSubordonnes ()` qui affiche sur la sortie standard la liste des subordonnés directs de ce cadre.
- Q.7)** - Une variable qui représente le salaire de base des employés. Celui-ci sera initialisé à 1457,52€ (smic mensuel brut au 1/01/2015).
- Q.8)** - Une méthode pour revaloriser ce salaire de base.
- Q.9)** - Une méthode `public double getSalaire ()` retournant le salaire mensuel brut d'un employé :
- pour les simples employés, il sera égal au salaire de base ;
  - pour les cadres il sera égal au salaire d'un simple employé plus 40% du salaire de chacun de ses subordonnés directs.

N'oubliez pas de tester au fur et à mesure les méthodes précédentes. Puis,

- Q.10)** - dans la fonction `main` de la classe **Principale** :
- Créer un cadre A qui a pour subordonnés directs un cadre B et deux simples employés.
  - Ajouter quatre simples employés au service du cadre B.
  - Tester vos méthodes avec ces deux cadres et six simples employés.

#### Questions facultatives.

- Q.11)** - Ajouter à la classe **Employe** :
- une méthode `public boolean aPourSuperieur (Employe e)` qui teste si l'employé a pour supérieur direct ou indirect l'employé `e`.
  - une méthode `public void changementAffectation (Cadre c)` qui change l'affectation de l'employé vers le service du cadre `c`. Ce changement est interdit si l'employé est un supérieur hiérarchique (direct ou indirect) du cadre `c`. Si l'employé était précédemment affecté à un autre cadre, il faut l'enlever des subordonnés de ce cadre.
- Q.12)** - Ajouter à la classe **Cadre** une méthode qui décrit complètement le service de celui-ci sur la sortie standard. La description comprendra la liste complète de tous les subordonnés directs et indirects de ce cadre avec le salaire de chacun.
- Q.13)** - Enlever la limite de 10 subordonnés directs par cadre.