

## TD 3 - Héritage

### Programmation Orientée Objet

### Objectif

- Comprendre le principe de l'héritage
- Comprendre comment créer un constructeur à la classe dérivée.
- Comprendre comment créer une méthode avec des arguments optionnels.
- Comprendre comment faire appel à une méthode de la super classe.

## 1 Héritage

On vous demande d'écrire un programme plus complexe pour la DRH d'une entreprise. Vous devrez prendre en considération plusieurs types de métier. Il existe deux catégories de personne dans l'entreprise : les employés et les actionnaires. Les employés sont les personnes qui travaillent dans l'entreprise. Les actionnaires sont les personnes extérieures à l'entreprise qui donnent des ordres.

- 1) créez la classe **Employe** qui hérite de la classe **Personne**. Que remarquez-vous ?
- 2) Ajoutez à la classe **Employe** un attribut privé contenant le métier de l'employé, ainsi qu'un numéro d'identifiant. Ce numéro sera égal à -1 si aucun identifiant n'a encore été affecté à l'employé. Vérifier que ces attributs puissent être donnés à la construction de manière optionnelle. Créez les accesseurs de ces attributs.
- 3) Créez une méthode publique **travailler** qui fait appel à la méthode **affiche()** de la classe **Personne** puis qui affiche le message suivant : "Mon identifiant est : <sMetier>. Je suis employé et mon métier est : <sMetier>".
- 4) Surchargez l'opérateur d'égalité et de différence pour la classe **Employe**.

## 2 Héritage multiple

Seule une personne peut être à la fois employé et actionnaire. Il s'agit du PDG de l'entreprise.

- 1) Dessinez le graphe de dépendance des différentes classes.
- 2) Créez la classe **Actionnaire**, ainsi que la classe **PDG**. La classe **Actionnaire** aura une méthode **ordonner** (semblable à la méthode **travailler** de la classe). Donnez un **main** illustratif de l'utilisation de ces classes.