



TD **Heritage**

Exercice 1

On dispose de la classe suivante :

class docteur

```
{    private String nom, hopital ;

    public docteur (String n,String m) { nom = n ; hopital = m ; }

    public void affiche() { System.out.println ("nom : " + nom + " à l' hôpital " + hopital) ;}

}
```

Réaliser une classe **Specialiste**, dérivée de **docteur** définis par le nom, l'hôpital et la spécialité. On y ajoutera les méthodes suivantes pour la classe **Specialiste** :

- **constructeur** à 3 arguments
- **constructeur** sans argument
- **affiche** pour afficher le nom, l'hôpital et la spécialité

Exercice 2

Un Personnel est défini par un nom et un numéro de sécurité social. Chacune associée à un getter et un setter

1. Créer la classe **Personnel**
2. Ajouter les méthodes suivantes :
 - a. Un constructeur avec deux paramètres

Un Chercheur est un Personnel. Donc celui-ci dispose en plus des caractéristiques du personnel le domaine de recherches et le nombre de publication

1. Créer la classe **Chercheur**
2. Ajouter un attribut qui permet de compter le nombre de chercheur
3. Ajouter les méthodes suivantes :
 - a. Un constructeur avec 4 paramètres
 - b. Une méthode **set_domaine_numero (String d, int n)** qui modifie le domaine de recherche et le numéro de sécurité social



Introduction à la Programmation Orientée Objet

Héritage

Objectif : Comprendre la construction d'un objet dérivé

1. Réaliser une classe **Personne**, qui possède les attributs suivants:

- **nom**, (chaîne des caractères)
- **année_de_naissance**. (int)

La classe **Personne** doit disposer

- d'un **constructeur** avec 2 arguments (nom, annee)
- les accesseurs (**getnom** et **getAnnee**) et les mutateurs (**setnom** et **setannee**)
- **AfficherPer ()**: qui affiche tous les attributs de la Personne

2. Réaliser une classe **Etudiant**, dérivée de la classe **Personne** qui possède les attributs suivants:

- **notes**, (ensembles des reels)
- **numero** (int)
- **filière**, (chaîne des caractères)

Rajouter un attribut statique **nombre_etudiants** dans cette classe.

Le numéro de l'étudiant sera égal au nombre d'étudiant à sa création.

La classe **Etudiant** doit disposer

- d'un **constructeur** avec 3 arguments (nom, annee, filiere)
- des accesseurs (**get**) pour les différents attributs et des mutateurs (**set**)
- **AfficherEtu ()**: qui affiche tous les attributs de l'Etudiant
- **AjouterNote(float a)** qui ajoute une note à l'étudiant
- **nombre_etudiants ()** : une méthode statique qui fournit le nombre d'objets de type Etudiant créés ;
- **float calculer_moy ()**

3. Ecrire une classe **TstEtudiant** afin de tester la classe Personne.

- Créer un étudiant
- Afficher le numéro de cet étudiant
- Afficher le nom de cet étudiant
- Modifier son année de naissance
- Ajouter plusieurs notes a cet étudiant
- Afficher les notes de cet étudiant
- Afficher sa moyenne
- affiche tous les attributs de l'Etudiant