



Série N 3 - Module 14  
Programmation Orienté Objet

**FILIERE** : TDI  
**NIVEAU** : 2<sup>ème</sup> année

**Exercices 1 (Gestion des Etudiants et leur Notes):**

On doit gérer des notes d'étudiants dans diverses matières. Un étudiant possède un nom et une adresse.

Chaque étudiant est représenté par un objet unique, maintenu dans une liste, la liste de tous les étudiants.

On doit pouvoir ajouter un étudiant à la liste des étudiants, modifier l'adresse d'un étudiant, obtenir un étudiant à partir de son nom.

Pour chaque Matière(Num, Coeff, Libel), certains étudiants possèdent une note. Pour une Etudiant donnée, on doit pouvoir ajouter une Matière et sa Note, demander la note d'un étudiant, modifier la note d'un étudiant, calculer la moyenne des notes, imprimer la liste des notes avec le Libel et Coefficient des Matière.

Définir les classes qui vous semblent judicieuses (Etudiant, Matière, etc.) pour cette application. Donner les profils des méthodes, choisir une représentation pour ces classes et programmer les méthodes.

**Exercices 2 (La classe Personne):**

**Précisons les propriétés des objets Personne :**

- 1) Une personne est décrite par trois informations : son nom, la société où elle travaille et son numéro de sécurité.
- 2) Une personne a toujours un nom. Ce nom ne peut pas changer. C'est une chaîne de caractères, dans laquelle les lettres sont en majuscules.
- 3) Une personne n'est pas forcément associée à une société. Dans ce cas, elle est considérée comme un travailleur indépendant ou comme une personne sans activité.
- 4) Une personne doit pouvoir indiquer si elle est ou non salariée (c'est-à-dire si sa société est identifiée).
- 5) Lorsqu'une personne est associée à une société, cette société est identifiée par une chaîne ne comportant pas plus de 32 caractères, dans laquelle les lettres sont en majuscules.
- 6) Une personne peut changer de société ou même perdre toute association à une société.
- 7) Une personne est capable de s'identifier, en affichant les informations qui la caractérisent.
- 8) Le numéro de sécurité comporte 13 numéros. Par exemple : "1681046068948". On s'intéresse aux 5 premiers numéros qui correspondent respectivement :
  - a. au sexe de la personne : 1: sexe masculin, 2 : sexe féminin
  - b. à l'année de naissance : 68 : 1968,
  - c. et au mois de naissance 10 : Octobre
- 9) L'accès aux données de la classe Personne doit être contrôlé.

**Une proposition de définition de la classe Personne :**

```
private String nom ;  
private String societe ;  
private String numSecu;  
private static final String pasDeSociete = "?";
```

```
public Personne (String nom)  
public Personne (String nom, String Entreprise)  
public Personne (String nom, String Entreprise, String NumSecu);  
public int getAge();  
public boolean isMasculin();  
public String getDateNaissance();  
public String getInfo();  
public String getNom();  
public void setSociete(String societe);  
public String getSociete();  
public void quitterSociete()  
public void setNumSecu(String numSecu);
```

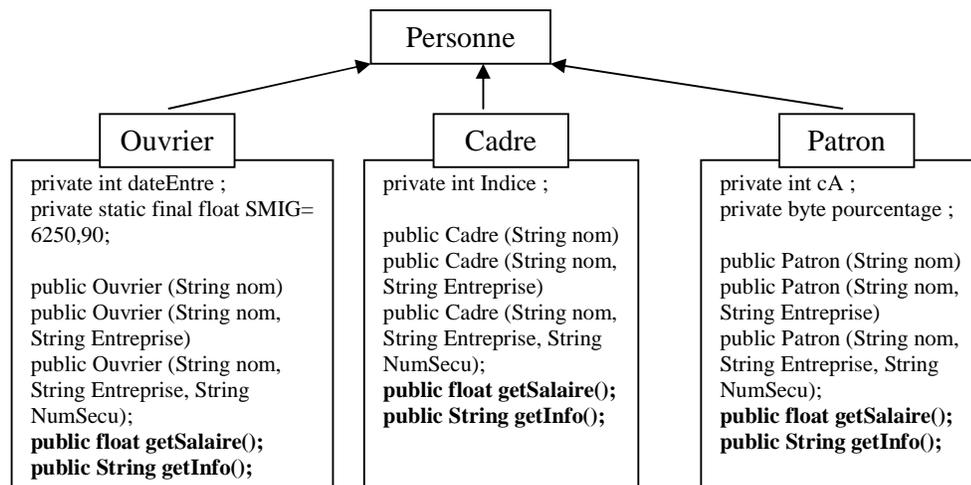


```
public String getNumSecu();  
public boolean etreSalarie() {  
private String validerSociete(String sNom);  
private String validerNumSecu( String numSecu);
```

**Travail à faire :**

- Implémenter la classe Personne.
- La méthode *validerNumSecu(String num)* permet de tester la validité du numéro de sécu entré (13 numéros entrés, premier chiffre égal à 1 ou 2 et mois de naissance compris entre 1 et 12). Pour cela on utilisera les méthodes de la classe String.
- Ecrire les accesseurs boolean *isMasculin()* et *int getAge()* et *setNumSecu(String num)*.
- Ecrire la méthode String *getInfo()* qui renvoie le nom, le sexe, l'âge, la société.

**Exercices 3 (Héritage de la Classe Personne):**



Créer la classe Ouvrier, la classe Cadre et la classe Patron qui héritent de la classe Personne, et prévoir les constructeurs (à 3 et à 0 arguments) de chacune des 3 classes.

Le patron a un salaire qui est égal à x% du chiffre d'affaire :  $\text{salaire} = \text{cA} * \text{pourcentage} / 100$

Le cadre a un salaire qui dépend de son indice :

- E1 : salaire annuel brut 130.000,00 F,
- E2 : salaire annuel brut 150.000,00 F,
- E3 : salaire annuel brut 170.000,00 F,
- E4 : salaire annuel brut 200.000,00 F,

L'ouvrier a un salaire qui est  $\text{salaire} = \text{SMIG} + (\text{age} - 18) * 100 + (\text{dateCourante} - \text{dateEntree}) * 150$ . De plus, le salaire ne doit pas dépasser  $\text{SMIG} * 2$ .

Ecrire la méthode float *getSalaire()* qui permet de calculer le salaire pour chacune des classes.

Ecrire la méthode String *getInfo()* qui renvoie le nom, le sexe, l'âge, la société, le salaire et le poste occupé (On pourra utiliser la méthode *getInfo()* de classe mère et rajouter l'information du poste occupé).

Le programme principal devra créer 1 patron et 2 cadres aux indices E1 et E3 et 5 ouvriers. Tester les différentes fonctions. Penser à utiliser un tableau de Personne.

Afficher les informations concernant les différents employés. Peut-on utiliser une boucle for ?