

TP1

Ex 1 :

Résoudre le problème suivant en utilisant les concepts d'objet et traduisez-le dans un nouveau projet Java.

Problème :

Vous devez créer un programme permettant d'enregistrer les informations sur les étudiants et les professeurs d'une université.

Consignes :

Créer les classes avec leurs méthodes, leurs accesseurs/mutateurs, leurs variables grâce aux informations suivantes :

- Un étudiant a un identifiant, un nom, un prénom, un numéro de sécurité sociale, son année d'étude, ses frais d'inscription qui sont calculés comme suit :
 $\text{frais inscription} = \text{année d'étude} * 150$
- Un professeur a un identifiant, un nom, un prénom, un numéro de sécurité sociale, son ancienneté, un salaire calculé comme suit :
 $\text{salaire} = \text{année d'ancienneté} * 50 + 1600$
- Créer un constructeur dans la classe étudiant qui donne des valeurs à chaque attribut de l'objet étudiant.
- Créer la classe principale qui contient la procédure main qui vous permet de créer deux objets étudiants et deux objets professeurs. Les deuxièmes objets étudiant et professeur sont créés en utilisant le constructeur qui permet de saisir directement les valeurs de ces attributs sans passer par les mutateurs (setters).
- Comparer les deux étudiants pour savoir lequel paie le plus de frais d'inscription et afficher ses informations.
- Comparer les deux professeurs pour les afficher par ordre alphabétique du nom.

Ex 2 : Les articles du supermarché

A. Écrivez une classe `Article` pour représenter les articles vendus dans un supermarché. Chacun comporte quatre attributs

- **long reference** – un numéro qui caractérise l'article de manière unique
- **String intitule** – la description de l'article sous forme de texte
- **float prixHT** – le prix unitaire hors taxes de l'article
- **int quantiteEnStock** – le nombre d'unités de l'article disponibles

et dispose d'un certain nombre de méthodes

- **public Article(long reference, String intitule, float prixHT, int quantiteEnStock)** – constructeur «immédiat» qui donne une valeur à chacune des variables d'instance,
- **public long getReference(), public String getIntitule(), public int getQuantiteEnStock()** – les « accesseurs » permettant d'obtenir la valeur de chaque attribut
- **public void approvisionner(int nombreUnites)** – méthode pour augmenter la quantité disponible de l'article
- **public boolean vendre(int nombreUnites)** – cette méthode enregistre la vente d'un certain nombre d'unités de l'article, dont il faut donc diminuer en conséquence la quantité disponible en stock.
Si **nombreUnites** est supérieur à la quantité disponible alors le stock n'est pas modifié et la méthode renvoie **false** ; autrement elle renvoie **true**.
- **public float prixHT()** – cette méthode calcule et renvoie le prix HT de l'article
- **public float prixTTC()** – cette méthode calcule et renvoie le prix TTC de l'article
- **public String toString()** – chaîne de caractères exprimant la référence, l'intitulé et le prix de l'article.
- **public boolean equals(Article unArticle)** – **a.equals(b)** est true si et seulement si a et b représentent le même article (c'est-à-dire s'ils ont la même référence).

B. Pour tester cette classe écrivez une méthode **main** qui crée trois articles (au moins) et qui essaie toutes ces méthodes.