#TP1 : (Evaluation formative)

But : Rappel sur la conception d'une hiérarchie de classes utilisant la notion d'interface.

Exercice1 :

Cet exercice vous permettra de concevoir une hiérarchie de classes utilisant la notion d'interface. Il vous servira également de révision pour les notions d'héritage, de classes abstraites et de polymorphisme.

Le directeur d'une entreprise de produits alimentaires souhaite gérer les salaires et primes de ses employés au moyen d'un programme Java.

Un employé est caractérisé par son nom, son prénom, son âge et sa date d'entrée en service dans l'entreprise.

1. Dans un fichier **Salaires.java**(*MAIN*), codez une classe **Employe** dotée des attributs nécessaires, d'une méthode abstraite calculerSalaire (ce calcul dépendra en effet du type de l'employé) et d'une méthode getNom retournant une chaine de caractère obtenue en concaténant la chaine de caractères "L'employé " avec le prénom et le nom.
2. Dotez également votre classe d'un *constructeur* prenant en paramètre l'ensemble des attributs nécessaires.
3. Calcul du salaire  
   Le calcul du salaire mensuel dépend du type de l'employé. On distingue les types d'employés suivants :
   1. **Vendeurs**. Leur salaire mensuel est le 20 % du chiffre d'affaire qu'ils réalisent mensuellement, plus 1000$.
   2. **Représentants**. Leur salaire mensuel est également le 20 % du chiffre d'affaire qu'ils réalisent mensuellement, plus 1400 $.
   3. **Producteurs**. Leur salaire vaut le nombre d'unités produites mensuellement multipliées par 5$.
   4. Ceux affectés à la **Manutention**. Leur salaire vaut leur *nombre d'heures* de travail mensuel multipliées par 100$.
4. Codez dans votre fichier Salaires.java une hiérarchie de classes pour les employés en respectant les conditions suivantes :
   1. La super-classe de la hiérarchie doit être la classe Employe.
   2. Les nouvelles classes doivent contenir les attributs qui leur sont spécifiques ainsi que le codage approprié des méthodes calculerSalaire et getNom, en changeant le mot "employé" par la catégorie correspondante.
   3. Chaque sous classe est dotée de constructeur prenant en argument l'ensemble des attributs nécessaires.

Exercice2 :

Employés à risques

Certains employés des secteurs ***production*** et ***manutention*** sont appelés à fabriquer et manipuler des produits dangereux.  
Après plusieurs négociations syndicales, ces derniers parviennent à obtenir une prime de risque mensuelle.

1) Complétez votre programme en introduisant deux nouvelles **sous-classes d'employés**. Ces sous-classes désigneront les employés des secteurs *production* et *manutention* travaillant avec des produits dangereux.  
  
2) Ajouter également à votre programme une **entité** Java pour les *employés à risque* permettant de leur associer une *prime mensuelle* fixe de 200.

3) Créer des instances des classes implémentées et afficher leurs salaires.