

## POO Java 1

### Exercice de révision

#### Sujet :

Le directeur d'un laboratoire de recherche scientifique souhaite gérer ses employés et leurs salaires à l'aide d'un programme Java. Alors, il propose de développer un programme pour gérer les bureaux de son laboratoire.

Ce laboratoire dispose de 50 bureaux au maximum et que chaque bureau peut contenir jusqu'à 5 chercheurs et un seul chef de bureau.

Un **laboratoire** est caractérisé par son nom(String), sa spécialité(String) et son nombre total de bureaux (tableau des objets de type Bureau).

Un **bureau** est identifié par son code (int), son nom(String) et il contient un ensemble de chercheurs. Les codes de bureau.

#### **Travail à faire :**

1. Codez une classe abstraite **Employe** dotée :

- Des attributs nécessaires (nom (String), matricule (int) et codeBureau(int)),
- Deux constructeurs (par défaut et avec arguments),
- D'une méthode abstraite `String getPostEmploye ()`, qui retourne le type d'employé (chercheur ou chef de bureau),
- D'une méthode abstraite `float calculeSalaire ()`, qui retourne le salaire net de chaque type d'employé,

Une méthode `float calculePrime ()`, qui retourne le taux de prime pour chaque employé.

- Une méthode `String toString ()`

2. Codez une hiérarchie de classes pour les employés en respectant les conditions suivantes :

- La super-classe (la classe mère) de la hiérarchie doit être la classe **Employe**.
- Une sous-classe **ChefDeBureau** qui est caractérisé par les attributs qui leur sont spécifiques dans la classe mère, ainsi une salaire de base fixe de 845 Dinars et un pourcentage de prime fixé à 20%.
- Une classe **Chercheur** qui est caractérisé par les attributs qui leur sont spécifiques dans la classe mère, et un nombre des heures supplémentaires.
- Les deux nouvelles sous-classes doivent contenir le codage approprié de la méthode `toString ()`, en redéfinir la contenu de la méthode `getPostEmployer ()` par la catégorie correspondante, ainsi la définition de la méthode `calculeSalaire ()` et `calculePrime ()`

Certains employés de type chercheur sont appelés à réaliser deux taches complètement délégués. Après plusieurs négociations entre les chefs des bureaux et le directeur de laboratoire, ils partagent les taches entre deux types de chercheurs.

- Une classe **ChercheurVacataire** qui est caractérisé par les attributs qui leur sont spécifiques dans la classe mère.
- Une classe **ChercheurContractuel** qui est caractérisé par les attributs qui leur sont spécifiques dans la classe mère, ainsi une salaire de base fixe de 845 Dinars.
- Les deux nouvelles sous-classes doivent contenir le codage approprié de la méthode `toString ()`, en redéfinir la contenu de la méthode `calculeSalaire ()`

⇒ Calcul du salaire

Le calcul du salaire mensuel dépend du type de l'employé:

- Ceux affectés à la recherche et développement scientifique (**Chercheur**). Leur salaire mensuel est dépend du type de contrat, on distingue alors deux type de chercheur :
  - Ceux affectés à la notion de recherche(**ChercheurContractuel**), leur salaire mensuel vaut le salaire de base plus nombre d'heures de travail mensuel supplémentaire multipliées par 25 dinars fixe et non modifiable.
  - Ceux affectés au développement et édition de la contenue(**CercheurVacataire**), leur salaire vaut le nombre d'heures supplémentaire multipliées par 20 dinars fixe et non modifiable.
- Ceux affectés à la direction et les affaires administrative(**ChefDeBureau**), leur salaire vaut le salaire de base plus un prime, appelé un prime de chef de bureau.

3. Ecrire une classe **Bureau** contenant un « tableau » des **Employes**

- Ses attributs ;
- Deux constructeurs (par défaut et avec arguments),
- La méthode `void ajouterEmploye(Employe employe)`, permet d'ajouter un employé au bureau en cours selon leur poste occupé, tout en vérifiant le respect du nombre max de chercheurs et les chefs par bureau.
- La méthode `String toString ()` qui retourne les informations concernant les déférents types d'employés de ce bureau.

4. Ecrire une classe **Laboratoire** contenant une liste "tableau" des **Bureaux**

5. Définissez ensuite les méthodes suivantes à la classe **Laboratoire** :

- `void ajouterBureau (Bureau bureau)`, qui ajoute un bureau à la collection.
- `void afficheBureaux ()`, qui affiche la liste des informations détaillés des bureaux dans ce laboratoire.
- `public static void main (String args[])`, c'est la méthode principale.