

Exercice1 : Calcul des impôts locaux

Dans le cadre de l'informatisation d'une mairie, on veut automatiser le calcul des impôts locaux. On distingue deux catégories d'habitation : les habitations à usage professionnel et les maisons individuelles, l'impôt se calculant différemment selon le type d'habitation. Pour cela, on définit les classes **HabitationProfessionnelle** et **HabitationIndividuelle** et les caractéristiques communes à ces deux classes sont regroupées dans la classe **Habitation**.

L'objectif du problème est d'implémenter ce schéma d'héritage et de mettre en œuvre le mécanisme de liaison dynamique.

A) Définition de la classe **Habitation** :

La classe **Habitation** comprend les attributs :

- propriétaire du type chaîne de caractères et qui correspond au nom de propriétaire,
- adresse du type chaîne de caractères et qui correspond à l'adresse de l'habitation,
- surface du double et qui correspond à la surface de l'habitation,

Les méthodes :

- Impot () qui permet de calculer le montant de l'impôt que doit payer le propriétaire de l'habitation, à raison de 10DH par mètre carré.
- Affiche () qui permet d'afficher les trois attributs de la classe **Habitation**.
- Un constructeur à trois paramètres permettant d'initialiser une instance de la classe **Habitation**.

B) Définition des classes **HabitationProfessionnelle** et **HabitationIndividuelle** :

- Le calcul de l'impôt d'une maison individuelle est différent de celui d'une habitation, il se calcule en fonction de la surface habitable, du nombre de pièces et de la présence ou non d'une piscine. On compte 100DH par pièce et 400DH supplémentaires en cas de présence d'une piscine
- Ajouter les attributs **NbPieces** de type entier et **Piscine** de type booléen.
- Redéfinir les méthodes afficher et impot.
- Le calcul d'une habitation à usage professionnel est également différent de celui d'une habitation. Il se calcule en fonction de la surface occupée et le nombre d'employés. On compte 60DH supplémentaires pour chaque employé.
- Ajouter l'attribut **NbEmployes** de type entier.
- Redéfinir les méthodes afficher et impot.

C) Gestion des habitations d'une commune

On désire à présent calculer l'impôt local des habitations (individuelles ou professionnelles) d'une commune. Pour cela, on utilise une collection d'objets représentée par un tableau où chaque élément désigne une habitation individuelle ou professionnelle.

- Définir la classe **Commune**.

- Définir une méthode **add** permettant d'ajouter une habitation à une commune

- Définir une méthode **show** pour afficher toutes les habitations de la commune.

Exercice2 : Ecrivez une interface **Forme** avec les méthodes abstraites suivantes:

- **perimetre()**: renvoie le périmètre de la forme,

- **aire()**: renvoie l'aire de la forme.

Ecrivez une classe **Carre** et **Rectangle** implémentant l'interface **Forme** avec les attributs suivants: **carre(cote:le coté du carré)** et **Rectangle (longueur et largeur)**

Ces deux classes doivent disposer des constructeurs suivants:

- **Carre(),Carre(cot)**.

- **Rectangle(),Rectangle(largeur,longueur)**.

Elles doivent contenir des accesseurs et mutateurs pour leurs différents attributs, et les méthodes suivantes:

- **perimetre()**: Donne le périmètre de la forme,

- **aire()**: Donne l'aire de la forme,

- **toString()**: Donne une représentation de la forme

Ecrivez aussi une classe de test **Forme** afin de tester les classes.

Exercice3 : Dans le cadre de l'informatisation d'une entreprise, un directeur souhaite automatiser la gestion des salaires de ses employés pour cela on aura :

-Une classe Employe : Un employé est caractérisé par :

+ *ces attributs* : **nom**(chaîne de caractères), **son prénom**(chaîne de caractères) et **son âge**(entier), **son année de recrutement**(entier)

+ *Ces méthodes* : -Un constructeur a quatre paramètres permettant d'initialiser une instance de la classe **Employe**.

+ **Affiche()** : qui permet d'afficher les quatre attributs de la classe **Employe**.

+ **CalculSalaire()** : permet de renvoyer le salaire mensuel d'un employé mais ce calcul dépend du type de l'employé. On distingue les types d'employés suivants qui seront définies dans les classes suivantes:

-Producteur : Leur salaire vaut le nombre d'unités produites mensuellement multipliées par 5 :

Ajouter l'attribut **NbUnités** de type entier ; Définir la méthode **CalculSalaire()** ; Redéfinir la méthode **Affiche()**

-Commercial : Leur salaire mensuel se calcule en fonction du chiffre d'affaire qu'ils réalisent :

Ajouter l'attribut **ChiffreAffaire** de type double ; Redéfinir la méthode **Affiche()**

- La classe Commercial dérive la classe **Vendeur** et la classe **Représentant** tel que leur salaire est le 20 % du chiffre d'affaire qu'ils réalisent mensuellement, plus respectivement 400DH et 800DH : Définir la méthode CalculSalaire() .

-Ajoutez une classe **Personnel** contenant un tableau d'employés. Il s'agira d'une collection polymorphique d'Employe contenant ces méthodes :

+void **ajouterEmploye(Employe)** : qui ajoute un employé au tableau

+void **show()** : qui affiche tous les employes du Personnel.

+double **salaireMoyen()** :qui affiche le salaire moyen des employés de l'entreprise.

+void **NbEmployes()** :qui affiche le nombre de employés instanciées pour chaque d'employées

Exemple d'exécution de cette méthode :Dans votre entreprise vous avez 3 représentant,2vendeur et 2 producteur. Pour cela il faut ajouter les attributs nécessaires dans les classes précédentes.