TP N°2 : L'héritage

Objectif :

- **Créer une classe dérivée.**
- **Ajouter des méthodes à une classe dérivée.**
- **Redéfinir des méthodes dans une classe dérivée.**
- **Créer une classe abstraite.**
- **Dériver une classe abstraite**
- **Implémenter une méthode abstraite**

Exercice 1 :

Ecrivez une classe Bâtiment avec les attributs suivants:
- adresse.

La classe Bâtiment doit disposer des constructeurs suivants:
- Batiment(),
- Batiment (adresse).

La classe Bâtiment doit contenir des accesseurs et mutateurs (ou propriétés) pour les différents attributs. La classe Bâtiment doit contenir une méthode *ToString ()* donnant une représentation du Bâtiment.

Ecrivez une classe Maison héritant de Bâtiment avec les attributs suivants:
- NbPieces: Le nombre de pièces de la maison.

La classe Maison doit disposer des constructeurs suivants:
- Maison(),
- Maison(adresse, nbPieces).

La classe Maison doit contenir des accesseurs et mutateurs (ou des propriétés) pour les différents attributs. La classe Maison doit contenir une méthode *ToString ()* donnant une représentation de la Maison.

Ecrivez aussi un programme afin de tester ces deux classes.

Exercice 2 :

- Un compte bancaire possède à tout moment une donnée : son solde. Ce solde peut être positif (compte créditeur) ou négatif (compte débiteur).
- Chaque compte est caractérisé par un code incrémenté automatiquement.

- A sa création, un compte bancaire a un solde nul et un code incrémenté.
- Il est aussi possible de créer un compte en précisant son solde initial.
- Utiliser son compte consiste à pouvoir y faire des dépôts et des retraits. Pour ces deux opérations, il faut connaître le montant de l'opération.
- L'utilisateur peut aussi consulter le solde de son compte par la méthode ToString().
- Un compte Epargne est un compte bancaire qui possède en plus un champ « TauxInterêt=6 » et une méthode calculIntérêt() qui permet de mettre à jour le solde en tenant compte des interêts.
- Un ComptePayant est un compte bancaire pour lequel chaque opération de retrait et de varsement est payante et vaut 5 dh

Questions :

- Définir la classe CompteBancaire.
- Définir la classe CompteEpargne.
- Définir la classe ComptePayant.
- Créer un programme permettant de tester les classes CompteBancaire et CompteEpargne avec les actions suivantes:
    - Créer une instance de la classe CompteBancaire , une autre de la classe CompteEpargne et une instance de la classe ComptePayant
    - Faire appel à la méthode deposer() de chaque instance pour déposer une somme quelconque dans ces comptes.
    - Faire appel à la méthode retirer() de chaque instance pour retirer une somme quelconque de ces comptes.
    - Faire appel à la méthode calculInterêt() du compte Epargne.
    - Afficher le solde des 3 comptes.

Exercice 3 :

Définir une classe Vecteurs2D caractérisée par l'abscisse X et l'ordonné Y, ainsi qu'un attribut qui renseigne sur le nombre de vecteurs créés lors de l'exécution du programme.

Définir les constructeurs (par défaut, d'initialisation et de recopie), et les accesseurs aux différents attributs.

Définir les méthodes ToString et Equals, la première retourne une chaine de caractères représentant l'abscisse et l'ordonné du vecteur comme suit : X = 1.5 - Y=2, la deuxième retourne True lorsque l'abscisse et l'ordonné des deux vecteurs sont égaux.

Définir une méthode Norme qui retourne la norme d'un vecteur, cette méthode peut être redéfinie dans les classes dérivées.

Définir une classe Vecteurs3D dérivée de la classe Vecteur2D qui contient, en plus des propriétés de la classe de base, la propriété Z symbolisant la troisième dimension.

Définir les constructeurs (par défaut, d'initialisation et de recopie) de cette classe, ainsi que les accesseurs des propriétés.

Redéfinir les méthodes ToString et Equals pour intégrer la troisième propriété.

Redéfinir la méthode Norme pour qu'elle retourne la norme d'un vecteur dans l'espace.

Définir un programme de test permettant de créer trois objets de type Vecteurs2D et trois autres de type Vecteurs3D en utilisant les trois constructeurs. Afficher les informations de tous les objets, et tester les méthodes Equals et Norme. Afficher le nombre d'objet créés.

Exercice 4 :

Un parc auto se compose des voitures et des camions qui ont des caractéristiques communes regroupées dans la classe Vehicule.

- Chaque véhicule est caractérisé par son matricule, l'année de son modèle, son prix.
- Lors de la création d'un véhicule, son matricule est incrémenté selon le nombre de véhicules créés.
- Tous les attributs de la classe véhicule sont supposés privés. ce qui oblige la création des accesseurs (get…) et des mutateurs (set….) ou les propriétés.
- La classe Vehicules possède également deux méthodes abstraites demarrer() et accelerer() qui seront définies dans les classes dérivées et qui afficheront des méssages personnalisés.
- La méthode toString() de la classe Vehicule retourne une chaîne de caractères qui contient les valeurs du matricule, de l'année du modèle et du prix.
- Les classes Voiture et Camion étendent la classe Vehicule en définissant concrètement les méthodes accelerer() et demarrer() en affichant des messages personnalisés.

1- Créer la classe Vehicule.
2- Créer les classes Camion et Voiture.
3- Créer une classe Test qui permet de tester la classe Voiture et la classe Camion.

Exercice 5 :

Ecrivez une classe abstraite Employé avec les attributs suivants :

- Matricule
- Nom
- Prénom

- Date de naissance

La classe Employé doit disposer des méthodes suivantes :

- un constructeur d'initialisation
- des propriétés pour les différents attributs
- la méthode Tostring
- une méthode abstraite GetSalaire.

Un ouvrier est **un employé** qui se caractérise en plus par sa date d'entrée à la société.
Tous les ouvriers ont une valeur commune appelée SMIG=2500 DH

L'ouvrier a un salaire mensuel qui est :
Salaire = SMIG + (Ancienneté en année)*100.
De plus, le salaire ne doit pas dépasser SMIG*2.

Un cadre est un employé qui se caractérise par un indice.
Le cadre a un salaire qui dépend de son indice :
1 : salaire annuel brut 13000 DH
2 : salaire annuel brut 150000 DH
3 : salaire annuel brut 17000 DH
4 : salaire annuel brut 20000 DH

Un patron est un employé qui se caractérise par un chiffre d'affaire et un pourcentage.

Le chiffre d'affaire est commun entre les patrons.

Le patron a un salaire annuel qui est égal à x% du chiffre d'affaire :

Salaire = CA*pourcentage/100

Créer la classe Ouvrier, la classe Cadre et la classe Patron qui héritent de la classe Personne, et prévoir les Constructeurs de chacune des 3 classes.
Implémenter la méthode getSalaire() qui permet de calculer le salaire pour chacune des classes.