TP N°1 : Classes et Objets

Objectif :

- Définir les propriétés et méthodes d'une classe
- Définir des constructeurs
- Créer une instance de classe
- Accéder par les accesseurs aux propriétés en lecture et écriture d'un objet
- Appliquer des méthodes
- Définir des attributs et méthodes statiques

Exercice 1 :

1. Définir une classe **Point** caractérisée par son abscisse et son ordonné.
2. Définir à l'aide des **getters** et les **setters** les méthodes d'accès aux attributs de la classe.
3. Définir le **constructeur** par défaut et d'initialisation de la classe. Le constructeur par défaut affecte 0 aux attributs.
4. Définir la méthode **Norme ( )** qui retourne la distance entre l'origine du repère et le point en cours.
5. Écrire un programme permettant de tester la classe.

Exercice 2 :

1. Définir une classe Livre avec les attributs suivants : Titre, Auteur (Nom complet), Prix.
2. Définir à l'aide des propriétés les méthodes d'accès aux différents attributs de la classe.
3. Définir un constructeur permettant d'initialiser les attributs de la méthode par des valeurs saisies par l'utilisateur.
4. Définir la méthode Afficher ( ) permettant d'afficher les informations du livre en cours.
5. Écrire un programme testant la classe Livre.

Exercice 3 :

1. Définir une classe Rectangle ayant les attributs suivants : Longueur et Largeur.
2. Définir à l'aide des propriétés les méthodes d'accès aux attributs de la classe.
3. Ajouter un constructeur d'initialisation.
4. Ajouter les méthodes suivantes :
    - Périmètre ( ) : retourne le périmètre du rectangle.
    - Aire( ) : retourne l'aire du rectangle.

- EstCarre( ) : vérifie si le rectangle est un carré.
- AfficherRectangle( ) : expose les caractéristiques d'un rectangle comme suit : Longueur : […] - Largeur : […] - Périmètre : […] - Aire : […] - Il s'agit d'un carré / Il ne s'agit pas d'un carré

Exercice 4 :

Écrire une classe Complexes permettant de représenter des nombres complexes. Un nombre complexe Z comporte une partie réelle et une partie imaginaire :

Z = PartieRéelle + PartieImaginaire * i

Définir à l'aide des propriétés les méthodes d'accès aux attributs de la classe.

Définir un **constructeur** par défaut permettant d'initialiser les deux parties du nombre à 0.

Définir un **constructeur** d'initialisation pour la classe.

Ajouter les méthodes suivantes :

- **getModule**() : retourne le module du nombre complexe.
-
- Plus(Complexe) : Elle permet de retourner le nombre complexe obtenu en ajoutant au nombre en cours un nombre complexe passé en argument.
- **Moins(Complexe) :** Elle permet de retourner le nombre complexe obtenu en soustrayant au nombre en cours un nombre complexe passé en argument.
- **Afficher ( ) :** Elle donne une représentation d'un nombre complexe comme suit : a+b*i.

Écrire un programme permettant de tester la classe Complexe.

Exercice 5 :

Un cercle est défini par :

- Un point qui représente son centre
- Son rayon r

On peut créer un cercle en précisant son centre et son rayon.

Dans ce problème, nous allons commencer tout d'abord par définir la classe Point définie par :

- Les **attributs** : x et y de type int
- Un **constructeur** qui permet de définir les valeurs de x et de y.
- Une méthode **Afficher ()** qui affiche une chaîne de caractères POINT(x,y).

Les opérations que l'on souhaite exécuter sur un cercle sont :

- **getPerimetre()** : retourne le périmètre du cercle
- **getSurface()** : retourne la surface du cercle.
- **appartient (Point p)** : retourne si le point p appartient ou non au cercle.
- **Afficher ()** : Affiche une chaîne de caractères de type CERCLE(x,y,R)

Ecrire la classe Cercle.

Écrire un programme permettant de tester la classe Cercle.

Exercice 6 :

1. Définir une classe Employé caractérisée par les attributs : **Matricule**, **Nom**, **Prénom**, **DateNaissance**, **DateEmbauche**, **Salaire**.
2. Définir à l'aide des propriétés les méthodes d'accès aux différents attributs de la classe.
3. Définir un constructeur permettant d'initialiser les attributs de la méthode par des valeurs saisies par l'utilisateur.
4. Ajouter à la classe la méthode **Age( )** qui retourne l'âge de l'employé.
5. Ajouter à la classe la méthode **Anciennete( )** qui retourne le nombre d'années d'ancienneté de l'employé.
6. Ajouter à la classe la méthode **AugmentationDuSalaire( )** qui augmente le salaire de l'employé en prenant en considération l'ancienneté.

   Si Ancienneté < 5 ans, alors on ajoute 2%. - Si Ancienneté < 10 ans, alors on ajoute 5%. - Sinon, on ajoute 10%.

7. Ajouter la méthode **AfficherEmployé()** qui affiche les informations de l'employé comme suit :

   - Matricule : […]

   - Nom complet : [NOM Prénom]

   - Age : […]

   - Ancienneté : […]

   - Salaire : […]

Le nom doit être affiché en majuscule. Pour le prénom, la première lettre doit être en majuscule, les autres en minuscule.

8. Ecrire un programme de test pour la classe Employé.

Exercice 7 :

1. Créer la classe Article caractérisée par 5 attributs : Référence, Désignation, PrixHT, TauxTVA.

   Ces attributs doivent seulement être accessibles par le biais des accesseurs (get / set) en lecture/écriture mis en œuvre par les propriétés.

2. Ajouter les constructeurs suivants :

   - Un constructeur par défaut
   - Un constructeur initialisant toutes les propriétés.
   - Un Constructeur qui permet de renseigner la référence et la désignation lors de l'instanciation
   - Un constructeur de recopie

3. Implémentez la méthode CalculerPrixTTC() ;

   Cette méthode doit calculer le prix TTC d'un article qui équivaut à : PrixHT + (PrixHT*TauxTVA/100) et retournera la valeur calculée.

4. Ajouter la méthode **AfficherArticle()** qui affiche les informations de l'article.

5. Créer un programme de test où il faut créer des objets (en utilisant les différents constructeurs) et leur calculer le prix TTC.

6. Le taux de TVA est en fait commun à tous les articles. Pour éviter toute redondance de cet attribut, vous devriez donc la déclarer comme partagée au niveau de la classe Article et non comme un attribut spécifique des objets instanciés à partir de la classe. Proposer une solution et tester de nouveau.

Exercice 8 :

1. Définir une classe Client avec les attributs suivants : CIN, Nom, Prénom, Tél.
2. Définir à l'aide des propriétés les méthodes d'accès aux différents attributs de la classe.
3. Définir un constructeur permettant d'initialiser tous les attributs.
4. Définir un constructeur permettant d'initialiser le CIN, le nom et le prénom.
5. Définir la méthode Afficher ( ) permettant d'afficher les informations du Client en cours.
6. Créer Une classe Compte caractérisée par son solde et un code qui est incrémenté lors de sa création ainsi que son propriétaire qui représente un client.
7. Définir à l'aide des propriétés les méthodes d'accès aux différents attributs de la classe (le numéro de compte et le solde sont en lecture seule)
8. Définir un constructeur permettant de créer un compte en indiquant son propriétaire.
9. Ajouter à la classe Compte les méthodes suivantes :
    a. Une méthode permettant de Crediter() le compte, prenant une somme en paramètre.
    b. Une méthode permettant de Crediter() le compte, prenant une somme et un compte en paramètres, créditant le compte et débitant le compte passé en paramètres.
    c. Une méthode permettant de Debiter() le compte, prenant une somme en paramètre
    d. Une méthode permettant de Débiter() le compte, prenant une somme et un compte bancaire en paramètres, débitant le compte et créditant le compte passé en paramètres
    e. Une méthode qui permet d'afficher le résumé d'un compte.
    f. Une méthode qui permet d'afficher le nombre des comptes crées.
10. Créer un programme de test pour la classe Compte.