

TP N°2 : Classes et Objets**Exercice 1 :**

Créez une classe Livre avec les caractéristiques suivantes :

- Des attributs privés __titre, __auteur et __prix qui représentent respectivement le titre, l'auteur et le prix du livre.
- Un constructeur __init__ pour initialiser le titre, l'auteur et le prix du livre.
- Des méthodes d'accès (getters et setters) pour les attributs titre, auteur et prix. Assurez-vous que le prix est toujours positif.
- Une méthode afficher_details qui affiche les détails du livre (titre, auteur et prix).

Créez quelques instances de la classe Livre et testez votre implémentation.

Exercice 2 :

Créez une classe Vehicule avec les caractéristiques suivantes :

- Des attributs privés __marque, __modele et __annee qui représentent respectivement la marque, le modèle et l'année de fabrication du véhicule.
- Un constructeur __init__ pour initialiser la marque, le modèle et l'année du véhicule.
- Utilisez des propriétés pour les attributs marque, modele et annee afin d'accéder et de modifier ces attributs de manière sécurisée. Assurez-vous que l'année est positive.
- Une méthode afficher_details qui affiche les détails du véhicule (marque, modèle et année).

Créez quelques instances de la classe Vehicule et testez votre implémentation.

Exercice 3 :

Créez une classe CompteBancaire avec les caractéristiques suivantes :

- Un attribut statique taux_interet qui représente le taux d'intérêt commun à tous les comptes.
- Des attributs privés __titulaire et __solde qui représentent respectivement le titulaire et le solde du compte.
- Un constructeur __init__ pour initialiser le titulaire et le solde du compte.
- Une méthode statique definir_taux_interet qui permet de définir le taux d'intérêt pour tous les comptes.
- Des méthodes d'accès (getters et setters) pour les attributs titulaire et solde.
- Une méthode calculer_interets qui calcule et affiche les intérêts en fonction du solde et du taux d'intérêt.

Créez quelques instances de la classe CompteBancaire et testez votre implémentation.

Exercice 4 :

Créez une classe **Produit** pour représenter les produits dans un magasin avec les caractéristiques suivantes :

- Des attributs privés `__nom`, `__prix` et `__quantite_stock` qui représentent respectivement le nom du produit, son prix et la quantité en stock.
- Un attribut statique `total_produits_vendus` qui représente le nombre total de produits vendus dans le magasin.
- Un constructeur `__init__` pour initialiser le nom, le prix et la quantité en stock du produit.
- Des méthodes d'accès (getters et setters) pour les attributs `nom`, `prix` et `quantite_stock`.
- Une méthode `vendre_produit` qui permet de vendre une quantité spécifiée d'un produit et met à jour la quantité en stock et le total des produits vendus.
- Une méthode statique `afficher_total_vendus` qui affiche le nombre total de produits vendus dans le magasin.

Créez quelques instances de la classe **Produit** et testez votre implémentation.

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 réel.
- 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 Client avec les attributs suivants : CIN, Nom, Prénom, Tél.
2. Définir 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 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 (client).
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 Debiter() le compte, prenant une somme en paramètre
 - c. Une méthode qui permet d'afficher le résumé d'un compte.
 - d. Une méthode qui permet d'afficher le nombre des comptes créés.
10. Créer un programme de test pour la classe Compte.