Formation Python : AJC Classroom

TP1 : Initiation vers le langage de programmation Python : Dictionnaires

Exercice 1 :

Ecrire une classe Rectangle en langage Python, permettant de construire un rectangle dotée d'attributs **longueur** et **largeur.**
Créer une méthode Perimetre() permettant de calculer le périmètre du rectangle et une méthode Surface() permettant de calculer la surface du rectangle
Créer les getters et setters.
Créer une **classe fille Parallelepipede héritant** de la classe Rectangle et dotée en plus d'un **attribut hauteu**r et d'une autre **méthode Volume()** permettant de calculer le volume du Parallélépipède.

Exercice 2 :

Créer une **classe Python** nommée **CompteBancaire** qui représente un compte bancaire, ayant pour attributs : **numeroCompte** (type numérique) , **nom** (nom du propriétaire du compte du type chaîne), & **solde**.
Créer un constructeur ayant comme paramètres : **numeroCompte, nom, solde**.
Créer une méthode **Versement()** qui gère les versements.
Créer une méthode **Retrait()** qui gère les retraits.
Créer une méthode **Agios()** permettant d'appliquer les agios à un pourcentage de 5 % du solde
Créer une méthode **afficher()** permettant d'afficher les détails sur le compte
Donner le code complet de la c**lasse CompteBancaire.**

Exercice 3 :

Définir une classe Cercle permettant de créer un cercle C(O,r) de centre O(a,b) et de rayon r à l'aide du constructeur :

Définir une méthode Surface() de la classe qui permet de calculer la surface du cercle
Définir une méthode Perimetre() de la classe qui permet de calculer le périmètre du cercle
Définir une méthode testAppartenance() de la classe qui permet de tester si un point A(x,y) appartient ou non au cercle C(O,r)

Exercice 4:

Créer une **classe Calcul** ayant un **constructeur par défau**t (sans paramètres) permettant d'effectuer différents calculs sur les nombres entiers.
Créer au sein de la classe Calcul une **méthode** nommée **Factorielle()** qui permet de calculer le factorielle d'un entier. Tester la méthode en faisant une instanciation sur la classe.
Créer au sein de la classe Calcul une **méthode** nommée **Somme()** permettant de calculer la **somme des n premiers entiers: 1 + 2 + 3 + .. + n.** Tester la méthode.
Créer au sein de la classe Calcul une **méthode** nommée **testPrim()** permettant de tester la **primalité d'un entier** donné. Tester la méthode.
Créer au sein de la classe Calcul une **méthode** nommée **testPrims()** permettant de tester si deux nombres sont premier entre eux.
Créer une **méthode tableMult()** qui crée et affiche la table de multiplication d'un entier donné. Créer ensuite une méthode **allTablesMult()** permettant d'afficher toutes les tables de multiplications des entiers 1, 2, 3, ..., 9.
Créer une **méthode listDiv()** qui récupère tous **les diviseurs d'un entier** donné sur une **liste Ldiv**. Créer une autre méthode **listDivPrim()** qui récupère tous **les diviseurs premiers** d'un entier donné

Exercice 5 :

Coder une classe **myString** permettant de doter les chaines de caractères des méthodes **append()** et **pop()** faisant les mêmes opérations que celles des listes.
Exemple si on crée des chaines via l'instanciation s1 = myString("Hello") et s2 = "bonjour", et on lui applique les méthodes :

```
print(s1.append(" world !")) # affiche  'Hello world !'
print(s2.pop(2))  # affiche 'bojour
```

Exercice 6:

1. Définir une **classe Book** avec les attributs suivants : T**itle, Author (Nom complet), Price.**
2. Définir un **constructeur** ayant comme attributs: **Title, Author, Price.**
3. Définir la **méthode View()** pour afficher les informations d'une instance object Book.
4. Ecrire un programme pour **tester la classe Book**.

Exercice 7 :

Ecrire une classe Python nommée **Geometry** avec un **constructeur** par défaut **sans paramètres**.

1. Ajouter une **méthode** nommée **distance()** à la classe geometry qui permet de calculer la distance entre deux points
A = (a1, a2), B = (b1, b2) (avec la convention: un point est est identifié à ses coordonnées **M = (x_M , y_M))**
2. Ajouter une **méthode** nommée **middle()** à la classe geometry qui permet de déterminer le milieu d'un bipoint (A , B).
3. Ajouter une **méthode** nommée **trianglePerimeter()** à la classe geometry qui permet de calculer le périmètre d'un triangle ABC.
4. Ajouter une **méthode** nommée **triangleIsoscel()** qui renvoie True si le triangle est isoscel et False sinon.