

 <p>OFPPT</p>	<p>مكتب التكوين المهني وإنعاش الشغل</p>
	<p><i>Office de la Formation Professionnelle et de la Promotion du Travail</i></p>

**Filière: Dev Digital 101/102**

**Module: Programmation Orientée Objet**

**TP 6**

**Exercice 1:**

- Copier une classe stagiaire dans un nouveau fichier.
- Le numéro d'inscription doit être une valeur positive, faire les modifications nécessaires dans le setter du numéro d'inscription. Une valeur négative ne doit pas être affecté à l'attribut numéro d'inscription.
- Les notes doivent être comprises entre 0 et 20, faire les modifications nécessaires dans les setters des notes 1, 2 et 3. Une note incorrecte ne doit pas être affectée aux attributs des notes.
- Créer les propriétés **NumeroInscription, Nom, Prenom, Note1, Note2, Note3** en utilisant la fonction **property()**
- Créer ensuite un objet **stg** en utilisant le constructeur par défaut
- Demander à l'utilisateur de saisir les informations du stagiaire à les stocker dans les attributs à l'aide des propriétés.  
Essayer votre code en donnant des valeurs incorrecte pour le numéro d'inscription et pour les notes pour vérifier que les setters fonctionnent correctement.
- Afficher les informations à l'aide des propriétés déjà définits.

**Exercice 2:**

Copier la classe stagiaire dans un autre fichier et faire les actions demandées dans les questions suivantes:

- 1) Commenter les propriétés déjà créés dans l'exercice 1 à l'aide de la fonction **property()**
- 2) Utiliser le décorateur **@property** pour créer les propriétés pour chaque attribut

- 3) Utiliser le décorateur **@propriete.setter**, pour créer un setter de chaque attribut, pour les setter des notes, on doit d'abord vérifier que la note est correcte (entre 0 et 20) avant de faire l'affectation, sinon attribuer la valeur 0 à cette note
- 4) Créer un objet **stg** en utilisant le constructeur déjà définit
- 5) Demander à l'utilisateur de saisir les informations du stagiaire en appelant les propriétés convenables
- 6) Afficher les informations en utilisant les propriétés convenables aussi

### **Exercice 3:**

Soit la classe **Point** définie par ses coordonnées (**abscisse** et **ordonnée**)

- 1) Définir la classe **point**.
- 2) Ajouter un constructeur d'initialisation (les attributs privés).
- 3) Ajouter les propriétés nécessaires de la classe point. (soit en utilisant la méthode property ou le decorateur @property)
- 4) Ecrire une méthode déplacer qui admet comme paramètres deux variables de type entier **Deplacer (nx, ny)**, qui permet de déplacer le point en lui attribuant les nouvelles valeurs.
- 5) Ajouter une méthode **distance** qui calcule et qui renvoie la distance entre les deux points.

Par exemple la distance entre deux points A et B est :

$$AB = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}$$

- 6) Ecrire un programme de teste qui permet de créer 2 objet points en utilisant les deux constructeurs et en appelant les méthodes créées.

### **Exercice 4:**

- 1) Créer un nouveau fichier python nommé **GestionArticle**.
- 2) Créer une classe **Article**, caractérisée par : un nom (chaîne), une référence (chaîne) et un prix unitaire (double) et TauxTVA (double)
- 3) Ajouter un constructeur paramétrique initialisant tous les attributs (doivent être privés)
- 4) Implémenter les propriétés nécessaires.
- 5) Implémenter les getters et setters
- 6) Implémentez la méthode **CalculerPrixTTC()**

Cette méthode doit calculer le prix TTC d'un article qui équivaut à :

$$\text{PrixHT} + (\text{PrixHT} * \text{TauxTVA} / 100)$$

et retournera la valeur calculée.

- 7) Implémenter la méthode **Afficher()** permettant d'afficher les information d'un article comme suit:  
Reference :.....  
Nom :.....  
Prix:.....
- 8) Définir la méthode **\_\_str\_\_(self)** qui retourne les informations de l'article sous forme de chaine de caractère
- 9) Créer un objet Article à l'aide du constructeur d'initialisation
- 10) Demander à l'utilisateur de saisir les informations, en se servant des propriétés
- 11) Afficher les informations en utilisant les propriétés.
- 12) Afficher les informations en utilisant la méthode **Afficher()**
- 13) 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.
- 14) Modifier la classe Article afin d'avoir la référence un entier et qui s'incrémente automatiquement.
- 15) Les propriétés relative à la référence doit être en lecture seule.
- 16) Faire les modifications nécessaires.