

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

Filière: Dev Digital 101/102

Module : Programmation Orientée Objet

TP 8

I) Créer une 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 des attributs en utilisant le décorateur **@property**.
- 4) Définissez la méthode **__str__()** qui retourne la représentation mathématique d'un point(abs , ord)
- 5) Ajouter une méthode **distance(self)** qui calcule et qui renvoie la distance entre le point courant(self) et un autre point donnée en paramètre.

En mathématique la distance entre deux points A et B est :

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

- 6) Ajouter la méthode **calculermilieu(self)** qui permet de calculer et de retourner **un objet** correspondant au milieu du segment défini par le point de l'objet courant (self) et l'objet Point p passé en paramètre.

Nous rappelons que les coordonnées d'un point M(xM,yM) milieu de A(x1,y1) et B(x2,y2), en mathématiques, sont : $x_M = (x_1 + x_2) / 2$ et $y_M = (y_1 + y_2) / 2$

- 7) Créer une méthode de classe permettant de retourner le nombre de points créés

II) Créer une deuxième classe appelée **TroisPoints** ayant les attributs **__point1**, **__point2** et **__point3** qui sont de type **Point**.

- 1) Créer un constructeur acceptant trois paramètres obligatoirement.
- 2) Définir les getters et les setters nécessaires

- 3) Définir une méthode **sontalignes(self)** qui retourne True si les trois points point1, point2 et point3 sont alignés, False sinon.
Nous rappelons que trois points A, B et C sont alignés si
 $AB = AC + BC$ ou
 $AC = AB + BC$ ou
 $BC = AC + AB$
(AB désignant la distance séparant le point A du point B, pareillement pour AC et BC).
- 4) Écrivez une méthode **estisocèle(self)** qui retourne True si les trois points point1, point2 et point3 forment un triangle isocèle, False sinon.
Nous rappelons qu'un triangle ABC est isocèle si
 $AB = AC$ ou $AB = BC$ ou $BC = AC$.
- 5) Implémentez une version statique (méthode décorée par **@staticmethod**) des deux méthodes calculant la distance et le milieu (dans la classe point) et nommez ces méthodes distance2() et milieux2().
- 6) Créer un fichier main.py, dans lequel vous importez les deux classes Point et TroisPoints
- 7) Créer au moins trois objets Point(par exemple point A, point B et point C) avec des coordonnées de votre choix
- 8) Afficher le nombre de Points créés
- 9) Afficher la distance entre deux points (par exemple entre le point B et le point C)
- 10) Afficher l'abscisse et l'ordonnée du **milieu** de deux points A et B
- 11) Définir une instance TR de la classe **TroisPoints** à l'aide des points déjà créés
- 12) Vérifier si la TR est aligné ou non et afficher un message, "les trois points sont alignés" ou "les trois points ne sont pas alignés"
- 13) Vérifier si TR est isocèle ou non et afficher un message convenable