

Activité1

Coder une solution orientée objet en Python

Considérons une classe appelée Point ayant deux attributs privés **__abs** (abscisse du point) et **__ord** (ordonnée du point) et un attribut statique **nb** qui représente le nombre de point créés

1. Définissez la classe Point et un constructeur à deux paramètres.
2. Définissez les getters et setters pour les deux attributs en utilisant le décorateur @property
3. Définissez la méthode **__str()** qui retourne la représentation mathématique d'un point : (abs,ord).
4. Proposer une surcharge de l'opérateur == permettant de vérifier si deux point p1(x1,y1) et p2(x2,y2) sont égaux ou non (x1=x2 et y1=y2)
5. Ecrivez la méthode **calculerdistance(self)** qui permet de calculer la distance entre le point de l'objet courant (self) et l'objet p passé en paramètre. Nous rappelons que la distance entre deux points A(x1,y1) et B(x2,y2), en mathématiques, est égale à : $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$. Vous pouvez utiliser la fonction `math.sqrt(a)` pour calculer la racine carrée de a et `math.pow(x, y)` pour calculer x puissance y
6. Ecrivez la fonction **calculermilieu(self)** qui permet de calculer et de retourner un objet correspondant au milieu du segment défini par le point de l'objet courant (this) 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 :

$$x_M = (x_1 + x_2) / 2, y_M = (y_1 + y_2) / 2$$

Considérons maintenant une deuxième classe appelée **TroisPoints** ayant les attributs **__point1**, **__point2** et **__point3** qui sont de type Point

7. Définissez les getters/setters (avec le décorateur @property) et un constructeur acceptant trois paramètres.
8. Ecrivez 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 $AB = AC + BC$, $AC = AB + BC$ ou $BC = AC + AB$ (AB désignant la distance séparant le point A du point B, pareillement pour AC et BC).
9. Ecrivez 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 $BC = AC$.

Activité1

Coder une solution orientée objet en Python

10. Implémentez une version statique (méthode décorée par `@staticmethod`) des deux méthodes calculant la distance et le milieu.
11. Dans un fichier `main.py`, testez toutes les classes et méthodes que vous avez implémentées.