

Série 4 : La programmation orientée objet en Python

Objectifs

Les objectifs de ce TP sont de découvrir la programmation orientée objet en Python et plus particulièrement

- La notion de classe.
- La notion Héritage
- La notion de polymorphisme.

Exercice 1

1- Ecrire une classe **Point2D** avec :

- Deux attributs : x et y
- Une méthode **afficher** () qui permet d'afficher un point2D
- Une méthode qui permet de déplacer le point de (dx,dy)
- Une méthode **distance(Point)** qui permet de calculer la distance entre deux point2D

2- Ecrire une classe **Point3D** qui hérite de la classe **Point** avec :

- L'attribut z
- Une méthode **afficher** () qui permet d'afficher un point3D
- Une méthode qui permet de déplacer le point de (dx, dy, dz)
- Une méthode **distance(Point3D)** qui permet de calculer la distance entre deux point3D

Exercice 2

Ecrire une classe « vecteur » avec :

- Deux attributs de classe « Point » p1 et p2.
- Une méthode qui permet de renvoyer les coordonnées du vecteur sous forme de dictionnaire
- Une méthode qui permet d'inverser le sens du vecteur
- Une méthode qui permet de calculer la longueur d'un vecteur
- Une méthode qui permet de calculer l'angle entre le vecteur et l'axe X
- Une méthode qui permet d'étaler la longueur du vecteur d'une valeur 'd'
- Une méthode qui permet de tourner le vecteur d'un angle « a »

Exercice 3

Ecrire une classe Polygone avec :

- Un attribut « points » de type Liste qui contient les points
- Une méthode qui permet de calculer la longueur du contour du Polygone
- Une méthode « déplacer » qui permet de déplacer tous les points du Polygone

Ecrire une classe Rectangle avec :

- Deux attributs de classe Point p1 et p2
- La méthode Surface qui calcule la surface du rectangle
- La méthode contient(Point) qui permet de vérifier si un point est à l'intérieur d'un rectangle
- La méthode « show » qui affiche les coordonnées des quatre points formant le rectangle