

TRAVAUX PRATIQUES

LES FORMES : MES PREMIÈRES CLASSES

Ce TP va permettre de découvrir la notion de classe Python à travers le codage de différentes formes géométriques classiques. Il est important à chaque fois de préparer le schéma UML de votre classe avant d'entamer toute écriture de code. Vous commenterez votre code en utilisant les éléments suivants

- le
 #
est un commentaire programmeur¹ sur une seule ligne
- le commentaire sur plusieurs lignes
""" ... """
est un commentaire programmeur qui peut être sur plusieurs lignes, tout ce qui se situe entre les deux balises étant ignoré par le compilateur

1. LE POINT

Vous allez créer la classe **Point**. Cette Classe possède deux attributs correspondants aux coordonnées du point dans un repère cartésien.

- x l'abscisse
- y l'ordonnée

Vous définirez pour cette classe, deux constructeurs, l'un par défaut initialisant les coordonnées au point origine (0,0), l'autre permettant d'instancier un point particulier dont les coordonnées seront précisées au constructeur.

- `Point()` le constructeur par défaut
- `Point(double a, double b)` le second constructeur

Enfin, cette classe possédera deux méthodes de calcul de distance par rapport à un autre point. La première nécessitant les coordonnées de l'autre point, la seconde utilisant l'objet **Point** associé à cet autre point.

- `double distance(double a, double b)`
- `double distance(Point camarade)`

Avant de coder cette classe, vous prendrez le temps de rédiger son schéma UML afin de bien visualiser les différents éléments de cette classe et leur type.

2. LE CERCLE

la seconde classe à créer est la classe **Cercle**. Avant de coder cette classe il sera important de concevoir et d'écrire sur feuille le schéma UML de la classe. Un cercle possède deux attributs, son centre qui est un **Point** et son rayon. Elle possédera deux constructeurs, le premier permettant de créer un cercle centre sur l'origine du repère et dont le rayon sera précisé. Le second précisera aussi le **Point** centre à utiliser. Enfin, cette classe proposera les 5 méthodes suivantes

- (1) une méthode permettant de calculer le diamètre du cercle
- (2) une méthode de calcul du périmètre du cercle
- (3) une méthode de calcul de la surface du cercle
- (4) une méthode permettant de dire si un autre cercle est en intersection avec celui ci.

- (5) une méthode permettant de dire si un **Point** A fait parti du cercle

De même, vous utiliserez cette classe et ces méthodes dans la méthode main de la classe principale

3. LE RECTANGLE

Cette classe permet de caractériser un rectangle. Avant de coder cette classe il sera important de concevoir et d'écrire sur feuille le schéma UML de la classe. Un rectangle comporte 3 attributs, le **Point** initial en bas à gauche, sa longueur parallèle à l'axe des abscisses et sa hauteur parallèle à l'axe des ordonnées. Vous créerez trois constructeurs, l'un par défaut avec un **Point** situé à l'origine et une longueur et hauteur de 1, un autre permettant de spécifier tous les attributs et le dernier enfin fournissant les **Points** bas-gauche et haut-droit du rectangle. Pour finir, vous coderez les méthodes suivantes

- (1) le calcul de la surface,
- (2) le calcul du périmètre,
- (3) la position de chacun des **Points** du rectangle, une méthode par **Point**
- (4) une méthode vérifiant si un **Point** est situé dans le rectangle.

De même, vous utiliserez cette classe et ces méthodes dans la méthode main de la classe principale

4. LE TRIANGLE RECTANGLE

Cette permet de décrire un triangle rectangle. Avant de coder cette classe il sera important de concevoir et d'écrire sur feuille le schéma UML de la classe. Les attributs du triangle rectangle seront les deux petits cotés accolés à l'angle droit et le **Point** de l'angle droit. Vous créerez deux constructeurs : le premier par précisera uniquement les valeurs des cotés, le **Point** étant initialisé à l'origine (0,0). Le second fournira les dimensions des deux cotés et le **Point** de l'angle droit. Vous donnerez ensuite les méthodes suivantes :

- la valeur de l'hypothénuse
- le périmètre du triangle
- la surface du triangle
- une méthode permettant de dire si le triangle est isocèle.

1. il faut consulter le code pour voir ces commentaires programmeurs