

Méthologie de la Programmation

TP4 : Programmation objet

am@up8.edu

Octobre 2022

Dans ce TP :

- Définition d'une classe
- Utilisation de classes dans les fonctions
- Implémentation de nouvelles méthodes
 - Fichiers à rendre : `test.py`, `geom.py`
 - `zip L1_MdP_TP4_VOTRENOM.zip test.py geom.py` permet de créer une archive contenant vos fichiers.
- **notation** :
 - le **nommage** des fichiers, des fonctions, et des variables sera pris en compte dans la notation.
 - votre programme doit comporter une fonction `main` (voir exemple) dans lequel vous effectuez des appels aux fonctions développées pour **tester et montrer leur bon fonctionnement**. Pensez à tester les cas limites.
 - rendre le TP avant la date spécifiée sur le moodle ne donne pas de bonus.
 - pas de rendu sur moodle \Rightarrow 0.

Placez-vous dans un dossier `tp4` situé à un emplacement approprié de votre hiérarchie de fichiers.

1 Exercice 1 : première classe en python

Cette exercice reproduit en partie ce qui a été vu dans le cours. Pour chaque fichier, ne **pas** copier/coller depuis le cours, le code doit être ré-écrit intégralement.

1. Créer un fichier `geom.py` dans lequel vous définissez une classe `Point` et son constructeur qui définit ses deux attributs : `abs` et `ord`.
2. Écrivez la méthode `__str__()` permettant d'afficher la représentation mathématique d'un point (`abs`, `ord`) comme indiqué ci-dessous. Contrairement à ce qui a été fait en cours, utilisez la fonction `format` pour faire l'affichage.

```
1 a = Point(12, 24)
2 print(a) # (12, 24) doit s'afficher.
```

3. Testez cette première méthode dans un fichier `test_1.py` dans lequel vous importez le module `point`, définissez un objet `Point` et l'affichez. Comme dans les TPs précédents, vos appels de fonction doivent être réalisés dans un bloc `main` :

```
1 if __name__ == "__main__":
```

4. Ajoutez à votre module `geom.py` :
 - (a) une méthode qui calcule la distance euclidienne entre deux points ;
 - (b) une méthode qui renvoie un objet **Point** dont les coordonnées correspondent au milieu du segment défini par l'objet **Point** courant et un autre **Point** passé en paramètre.
5. **Testez** ces méthodes grâce à des instanciations de la classe **Point** et à des appels de fonction dans le script `test_1.py`.

2 Exercice 2 : classe TroisPoints

1. Dans le fichier `geom.py`, créez une classe `TroisPoints` et son constructeur qui définit ses trois attributs :
 - A
 - B
 - C
2. Ajoutez à la classe `TroisPoints` la méthode `__str__()` qui permet par exemple avec le code suivant :

```
1 P = Point(0,1)
2 Q = Point(1,0)
3 R = Point(0,0)
4 TP=TroisPoints(P,Q,R)
5 print(TP) # appel de print sur l'objet de la classe TroisPoints
```

l'affichage :

```
1 premier : (0,1)
2 deuxieme : (1,0)
3 troisieme : (0,0)
```

Il y a plusieurs manières de faire : pour limiter la réécriture de code, vous pouvez notamment utiliser la méthode `__str__()` sur les différents points.

3. Écrivez la méthode `sont_alignes()` qui renvoie `True` si A, B, et C sont alignés, et `False` sinon. Faites des appels à la méthode permettant de tester son fonctionnement.
4. Écrivez la méthode `est_isocele()` qui renvoie `True` si A, B, et C forment un triangle isocèle, et `False` sinon. Faites des appels à la méthode permettant de tester son fonctionnement.
5. Vérifiez que votre code est propre, lisible, et optimisé (si des morceaux de code similaires se répètent, c'est sans doute qu'il vous manque une fonction). Les fonctions permettent aussi d'améliorer la lisibilité.

Aide

1. trois points A, B et C sont **alignés** si les vecteurs \overrightarrow{AB} et \overrightarrow{BC} sont colinéaires. Les vecteurs colinéaires présentent la propriété suivante :

$$x\overrightarrow{AB} * y\overrightarrow{BC} - y\overrightarrow{AB} * x\overrightarrow{BC} = 0$$

3 Exercice 3 (Bonus) : classe Cercle

- Définissez une classe `Cercle`, ayant deux attributs : **centre** et **rayon**, et écrivez la méthode `intersection()` qui vérifie si le cercle `Cercle` courant et un autre `Cercle` se recouvrent partiellement ou pas.