

Algorithmique / Python

Introduction aux objets en Python

Manuel Munier

Version 15 mars 2022 (10:38)

1 Classe Point

Ce premier exercice est (très) simple : il s'agit de vous initier à la syntaxe Python pour la définition de classes, d'attributs, de méthodes, l'instanciation d'objets et l'appel de méthodes. Vous vous baserez pour cela sur la classe `Point` présentée en cours en étant vigilant à prendre, dès le départ, de "bonnes" habitudes de programmation OO, à savoir :

- ce sont les constructeurs qui créent les attributs des objets
- tous les attributs doivent être privés (sauf mention contraire)
- ce qui vous obligera à définir et à utiliser des méthodes d'accès

Votre classe `Point` devra disposer de 2 attributs `x` et `y` pour stocker les coordonnées de ce point. Pour les méthodes d'accès, les accesseurs (*getters* en anglais) se nommeront `get_x()` et `get_y()`. Les mutateurs (*setters* en anglais) se nommeront `set_x(val)` et `set_y(val)`.

Vous écrirez bien entendu un programme de test pour créer des objets à partir de cette classe et y invoquer leurs méthodes, ceci afin de vérifier que votre classe `Point` est bien opérationnelle.

2 Classe Segment

Cette classe disposera de 2 attributs de type `Point` nommés `p1` et `p2`. Outre les constructeurs, les accesseurs et les mutateurs, vous définirez les méthodes suivantes :

- `longueur()` qui retourne la longueur du segment
- `centre()` qui retourne les coordonnées du centre du segment ; le résultat sera donc de type `Point`

Et, comme d'habitude, vous écrirez bien évidemment un programme de test afin de vérifier le bon comportement de votre classe.

NB : Pour information, voici comment calculer en Python le carré et la racine carrée d'une valeur :

```
Python 3.8.10 (default, Nov 26 2021, 20:14:08)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> a=4
>>> a**2
16
>>> import math
>>> math.sqrt(a)
2.0
>>> exit()
```

3 Classe Polygone

L'objectif de cette classe est de représenter un polygone irrégulier. Cette classe stockera donc dans ses attributs une liste de points (un tableau par exemple). Le constructeurinstanciera un objet `Polygone` "vide", c'est-à-dire sans aucun point. Il faudra alors invoquer la méthode `ajouterPoint(pt)` de ce polygone pour ajouter les différents points un à un.

Vous définirez ensuite la méthode `perimetre()` de cette classe `Polygone` qui calculera, comme son nom l'indique si bien, le périmètre d'un objet `Polygone`. Vous terminerez cet exercice en écrivant un programme de test afin de vérifier le bon comportement de votre classe.