

INITIATION À LA PROGRAMMATION OBJET

PRINCIPES ET VOCABULAIRE DE LA PROGRAMMATION ORIENTÉE OBJET

2.1 Les classes et les objets

En programmation structurée, un programme est l'ensemble de différentes fonctions et de différentes structures de données. En programmation orientée objet, un programme repose sur différents *objets*. Chaque objet associe des données et des *méthodes* agissant exclusivement sur les données de l'objet.

Notez que la programmation orientée objet a son vocabulaire propre qui diffère quelque peu de la programmation structurée vue jusqu'ici.

On parlera de *méthodes* plutôt que de fonctions ou procédures. En effet en programmation objet il n'est pas possible d'interagir directement avec les données (ou *attributs*) d'un objet car elles sont encapsulées dans l'objet. Il est nécessaire de passer par les méthodes qui composent une interface entre l'utilisateur et les données de l'objet.

Tout comme pour les structures de données abstraites, peu importe la manière dont sont implémentées les méthodes, seule leur spécification importe.

Le concept de classe n'est rien d'autre que la description d'un ensemble d'objets ayant une structure de données commune et disposant des même méthodes.

Les objets sont alors des variables d'un type de classe, on dit qu'un objet est une instance de sa classe.

Voici un exemple permettant d'expliciter la notion de classe et de fixer le vocabulaire spécifique à la programmation objet.

2.2 Élaboration d'une classe Point

Considérons, à titre d'exemple, dans le plan muni d'un repère orthonormé, la classe *Point*, destinée à manipuler des points d'un plan.

a. Définition des attributs et méthodes

Un *point* sera représenté par ses deux coordonnées réelles, abscisse et ordonnée. La classe *Point* aura deux attributs *abscisse* et *ordonnee*.

Supposons que l'on souhaite disposer des trois méthodes ci-dessous :

- *Initialiser* pour créer un point et initialiser ses coordonnées
- *deplacer* pour faire subir au point une translation, c'est-à-dire modifier ses coordonnées en ajoutant un réel *a* à son abscisse et un réel *b* à son ordonnée
- *afficher* pour avoir une représentation de l'objet sous forme d'une chaîne de caractères affichable.

Voici ce que cela pourrait donner en Python :

```

class Point :
    """Un point du plan muni d'un repère"""
    # Méthode Constructeur
    def __init__(self, x, y):
        self.abcisse = x
        self.ordonne = y
    # Méthode deplace qui translate le point de (dx,dy)
    def deplace(self, dx, dy):
        self.abcisse += dx
        self.ordonne += dy
    # Méthode pour représenter le point affichage
    def __repr__(self):
        return f'({self.abcisse!r} ; {self.ordonne!r})'

```

Remarques :

- en python les méthodes dites « double underscore » sont des méthodes « spéciales », des noms réservés qui s'intégreront aux autres fonctionnalités du langage.
`__init__` est un constructeur, il sera invoqué par le langage Python lors de la création d'un objet de classe Point.
`__repr__` est une méthode qui sera invoquée lorsque l'on voudra afficher un objet de la classe Point, dans une console Python en saisissant le nom de l'objet ou à l'aide de l'instruction print.
- la variable `self`, dans les méthodes d'un objet, désigne l'objet lui-même à qui s'appliquera la méthode. Elle représente une référence à l'objet dans la méthode dont on ne connaît pas encore le nom puisqu'il n'a pas encore été créé.

Illustration dans une console Python :

```

>>> A = Point(3,4)
>>> type(A)
<class 'point.Point'>
>>> A
(3 ; 4)
>>> A.deplace(0, 10)
>>> A
(3 ; 14)

```

Remarque : comme on peut le constater, pour appliquer une méthode à un objet on utilise ce que l'on appelle la notation « pointée » objet.methode(avec ou sans argument).

Exercice 1 :

Écrire une méthode `rotation180` permettant de transformer un point en son symétrique par rapport à l'origine du repère.

Exercice 2 :

Ajouter un attribut `couleur` à la classe `Point` permettant de définir un point par ses

coordonnées que qu'une couleur au format RVB codé sur 24 bits (#000000 à #FFFFFF) et modifier les méthodes en conséquence.

2.3 Récapitulatif vocabulaire de la programmation objet

Classe, Attributs, Méthodes

- Le type de données avec ses caractéristiques et ses actions possibles s'appelle classe.
- Les caractéristiques (ou variables) de la classe s'appellent les attributs.
- Les actions possibles à effectuer avec la classe s'appellent les méthodes.
- La classe définit donc les attributs et les actions possibles sur ces attributs, les méthodes.
- Un constructeur est une méthode obligatoirement présente dans une classe elle permet la création et l'initialisation d'un objet de cette classe.