

Programmation Orientée Objet

Cours

Objet

Un objet est une variable contenant à la fois des données et des comportements.
En d'autres termes, on peut dire qu'à l'intérieur d'un objet, il y a des variables, qu'on appelle attributs, et des fonctions, qu'on appelle méthodes.

Classe

Une classe est une sorte de plan, de manuel d'utilisation.
Une classe définit un type d'objet. Lorsque l'on va créer un objet, on se servira d'une classe afin que l'objet puisse lui être conforme. Lorsqu'on utilisera l'objet, on se référera à sa classe pour savoir quels sont ses attributs, quelles sont ses méthodes.

Constructeur

Le constructeur d'une classe est une méthode particulière définie par cette classe. C'est la méthode qui sera exécutée lorsqu'on aura besoin de créer un nouvel objet de cette classe.

Attributs privés

Les méthodes servent à définir des comportements de l'objet. Ces comportements ne doivent pas être court-circuités. C'est pour cela qu'on prend l'habitude de définir les attributs de façon privée : ils ne pourront être accédés que depuis l'intérieur de la classe.

Cours et TD d'algorithmique

Et en python ?

```
class Personnage:
    #on déclare d'abord une méthode "__init__" qui est le constructeur
    def __init__(self,name):#le 1er paramètre est obligatoire,
        #ensuite il y en a autant que vous voulez
        self.nom = name # 1ere donnée : le nom du personnage
        self.__pv = 100 # 2nde donnée : son nombre actuel de points de vie
    def perdPV(self,nombre):
        self.__pv = self.__pv - nombre
        print(self.nom + " perd " + str(nombre) + " points de vie !")
        if(self.__pv <= 0):
            self.__pv = 0
            print(self.nom + " trépasse...")
```

Une classe qui a été déclarée telle que ci-dessus pourra être utilisée ainsi dans le programme principal.

```
zagreus = Personnage("Zagreus") #appelle le constructeur de Personnage
celeste = Personnage("Celeste") #idem

zagreus.perdPV(12) # zagreus va perdre 12 points de vie
```

Dans cet exemple, on crée deux objets, zagreus et celeste, grâce à la classe Personnage. Ensuite, on utilise la méthode perdPV de la classe Personnage sur l'objet zagreus.

Travaux Dirigés

Gestion d'inventaire

De nombreux jeux autorisent la gestion d'un inventaire, souvent accessible via un menu.

1/ On souhaite modéliser une classe Item. Dans une première version, un Item possède un nom et un nombre (ce sont des données). Il possède aussi une méthode "affiche()" qui lui permet de s'afficher lui-même à l'écran. Implémentez cette classe, et testez-la en considérant que votre personnage aura trois items (item1 trois potions de soin, item2 une épée, item3 deux fléchettes, par exemple).

2/ On souhaite améliorer la classe Item afin que chaque item possède également une puissance. Si cette puissance est positive, c'est un item curatif, si elle est négative alors il est considéré comme un item consommable agressif (qu'on lance sur un ennemi), si elle est nulle alors c'est un équipement qui ne s'utilise pas directement.

Améliorez le constructeur pour prendre en compte cette nouvelle donnée.

Ajoutez une méthode "utilise()", qui affiche un message spécifique (disant que cela occasionne des dégâts, ou soigne, par exemple), et qui réduit la quantité de 1. Cette méthode n'aura néanmoins pas d'autre effet dans pour le moment.

3/ Cela doit-il avoir un impact sur la méthode "affiche" ? Dans quelle mesure ? Appliquez vos conclusions.

4/ Comment pourrait-on gérer l'achat d'un nouvel item dans un inventaire ? Peut-on considérer que cet inventaire est un tableau (une liste python) ? Comment ?