TD8: Programmation Orientée Objet en python

Notions de base de la POO :

- *Classe :* une classe est un "modèle" , ou un type qu'on crée afin de catégoriser des objets structurellement identiques. Une classe est définie par des attributs et peux comporter des constructeurs et des méthodes.
- *Constructeur :* est une méthode spécial et unique (sous python) par classe, elle permet de créer, donc réserver de la mémoire, et initialiser un objet. Le nom de cette méthode est « __init(self,arg1,arg2,…)__ ». cette méthode doit contenir au minimum un argument appelé ici « self » (peut prendre n'importe quelle nom). « self » fais référence au mot clé « this » en Java. Ce mot représente l'objet courant et permet de différencier les variables d'instances des arguments quand ces derniers ont le même nom.
- *Objet (ou instance) d'une classe :* un objet ou instance d'une classe est une réalisation de la classe pour un élément spécifique. L'objet est défini par son état et les opérations qui lui sont applicable.
- *Variables d'instance :* ce sont les variables (ou attributs) associés à une instance d'une classe.
- *Variables de classe :* ce sont des variables (ou attributs) défini au niveau de la classe. Ces variables peuvent être accessibles uniquement avec le nom de la classe.
- *Méthodes d'instances :* ce sont des méthodes qui ont besoin d'instances pour s'exécuter. Attention au mot clé « self » qui représente l'instance qui appel la méthode.
- *Méthodes de classes :* les méthodes de classes sont différentes des méthodes d'instances en ce qu'elles prennent comme premier argument la classe elle-même. De plus, une méthode de classe est liée à la classe, mais pas à l'objet de cette classe (« self » n'est pas reconnu dans une méthode de classe).

Exo1 : familiarisation avec les notions de la POO

1) On cherche à représenter un pokemon comme dans la kholle. Créer une classe « Pokemon » qui contient les attributs 'nom' et 'pv'. Instanciez deux pokemons différents à partir de cette première version de la classe.

2) Notre Pokemon a besoin d'attaquer pour pouvoir faire un combat. Créez une deuxième classe « Attaque » qui contient les attributs 'nom', 'degats', 'usages' et 'usages_max'. Créez plusieurs attaques différentes à partir de cette nouvelle classe. Ajoutez un attribut 'attaques' à la classe Pokemon qui contient une liste vide.

3) Passons aux méthodes d'instances : Dans la classe Pokemon, écrire une méthode « Ajouter_attaque » qui prend une instance de la classe Attaque en argument et qui l'ajoute au pokemon courant. L'attaque ne peut être ajoutée que si cette dernière n'a été déjà apprise et que le nombre d'attaques est inférieur à 4. Écrire une seconde méthode 'soigner' qui prend en paramètre le nombre de 'pv' à restaurer pour un Pokemon donné et qui restaure les attributs 'usages' de chacune de ses attaques.

4) Dans la classe Attaque, écrire une méthode « Afficher_attaque » qui affiche les détails d'une attaque. Ensuite dans la classe Pokemon, écrivez une méthode « Afficher_Pokemon » qui affiche toutes les variables décrivant le pokemon courant.

5) Dans la classe Pokemon, écrire une méthode « Attaque_Pokemon » dans laquelle le pokemon courant attaque celui passé en deuxième argument. Le pokemon attaquant choisi une attaque aléatoire de sa liste d'attaque, si celle-ci n'est pas complètement usée alors il applique cette attaque sur le 2ème Pokemon et lui fais baisser son pv selon les 'degats ' de cette dernière.

6) Créer deux Pokemons « p1 » et « p2 », chaque pokemon doit posséder deux (ou trois) attaques. Afficher ensuite les deux pokemons.
Scénario1 : « p1 » attaque deux fois « p2 », si « p2 » est toujours en vie alors « p2 » attaque de son tour trois fois « p1 ». Afficher l'état des deux pokemons après chaque attaque.
Scénario2 : D'abord, soignez les deux pokemons. Les deux pokemons vont se battre jusqu'à ce que l'un des deux ne peut plus se relever (i.e son pv <= 0). à chaque tour on tire un nombre aléatoire dans {0,1}.Si on tire 0 (resp. 1) alors c'est p1 (resp. p2) qui attaque. Faites tourner ce scénario en affichant l'état des pokemons à chaque tour, puis le nom du vainqueur.

7) Créer une Variable de classe « compteur » qui va compter le nombre d'objets de type « Pokemon » créés. Mettez à jour le constructeur.