TD9: Programmation Orientée Objet en python
(Suite du TD8)

Nouvelles notions dans la POO :

- *Méthode d'instance :* est une méthode liée à un objet (ou instance) d'une classe. Cette méthode donc s 'applique à l'objet qui l'évoque. Synthax : « Methode_Instance(self, ...) ». Par définition, elle prend au minimum l'argument « self ».
- *Méthode de classe :* est liée à la classe et non à l'objet de la classe. Elle prend « cls » comme premier argument. « cls » permet d'accéder aux attributs de classe. Il faut annoter une méthode de classe avec « @classmethod » juste avant la déclaration de celle-ci. Synthax : « Methode_Instance(cls, ...) ».
- *Méthode statique :* n'a accès à aucun des deux états (ni « self » ni « cls »), c'est à dire qu'elle ne prend comme paramètre ni l'instance (self) ni la classe (cls). Les méthodes statiques sont autonomes et ne dépendent que des arguments passer en paramètre. Ce type de méthode est annotée « @staticmethod » avant sa déclaration. Synthax : « Methode_Statique(arg1,arg2, ...) ».

Exo1 : Modifications du fichier « TD8.py »

1) Reprenez votre fichier python du TD8. Ajoutez l'attribut « pvmax » dans la classe « Pokemon ». Modifiez la méthode « Soigner_Pokemon(self) » qui met le pv à pvmax. Modifiez aussi la méthode « Afficher_Pokemon » en rajoutant l'affichage du pvmax.
Le Scenario1 (resp. Scenario2) doivent être des méthodes d'instances dans votre classe « Pokemon ».

2) « TD8.y » ne doit contenir que les deux classes (constructeurs et méthodes y compris).

Exo2 : Création d'un nouveau fichier « TD9.py »

La suite est à implémenter dans « TD9.py ».

1) Créez un fichier « TD9.py » dans le même dossier que « TD8.py ». Importez « TD8.y » dans « TD9.y » en utilisant l'instruction « from TD8 import * ». Ensuite dans « TD9.y », créez trois pokemons avec environ trois attaques chacun. Affichez les pokemons.

2) Créez une classe « Dresseur » avec les attributs « nom », « xp », « niveau » et « pokemons ». Les attributs « xp » et « niveau » sont initialisés à zéro, alors que « pokemons » est initialisé par une liste vide. Cet attribut fait référence à la liste des pokemons que le dresseur possède.

3) Créez une méthode « Ajout_Pokemon(self,p) » qui ajoute le pokemon « p » dans la liste du dresseur courant « self ». Le pokemon n'est ajouté que s'il n'est pas déjà dans la liste.

4) Créez une méthode « Afficher_Dresseur(self) » qui affiche tous les attributs d'un dresseur. Notez que vous disposez déjà de la fonction affichant un pokemon.

5) Faites un premier test : créez deux dresseurs, affectez leur des pokemons, puis affichez les dresseurs.

6) Créez une méthode « Taverne(self) ». Cette méthode soigne tous les pokemons d'un dresseur.

7) Créez une méthode « Update(self) » qui incrémente le « niveau » de 1 et met « xp » à zéro si l'expérience est égale à 10.

8) Créez une méthode «Defi_Aleatoire(self,dresseur2) » qui (a) fait passer les deux dresseurs à la taverne (b) choisi un pokemon de façon aléatoire dans la liste des pokemons de chaque dresseur (c) lance un défi entre les deux pokemons choisis. Le défi correspond au Scenario2 (c'est à dire jusqu'à ce que l'un des deux pokemons épuise toute son énergie vitale (pv<= 0)). (d) Le pokemon vainqueur se voit augmenter son expérience de 1. (e) Mettez à jour les niveaux.

9) Créez une méthode « Arene1(self,dresseur2) » qui lance 100 défis entre les 2 dresseurs. Le vainqueur est celui avec le plus grand niveau (si même niveau alors celui avec la plus grande expérience).

10) Créez une méthode statique « Defi_Deterministe(dresseur1,dresseur2) ».(a) Chaque dresseur choisi son pokemon avec le plus grand « pv ».(b) Ensuite un défi de type Scenario2 est lancé. (c) Mettez à jour l' « xp » et le « niveau » de chaque dresseur. Notez que dans cette méthode les dresseurs ne passent pas à la taverne.

11) Créez une méthode statique « Arene2(dresseur1,dresseur2) » qui (a) fait passer les deux dresseurs à la taverne (qu'une seule fois, au début) (b) applique 100 fois un défi déterministe (à la fin d'un défi, les dresseurs ne passent pas à la taverne, mais enchaînent les défis) (c) la méthode s'arrête lorsque tous les pokemons d'un dresseur sont mort (pv <= 0). (d) Affichez le vainqueur de cette arène.