

Méthologie de la Programmation

TP7 (noté) : Programmation objet

am@up8.edu

Novembre 2022

Dans ce TP :

- Rendre sur papier la première partie
- Rendre sur le moodle :
 - l'implémentation des classes dans des modules (`module_1.py` et `module_2.py`)
 - les programmes qui permettent de tester les classes et leurs méthodes (`etape_1.py` et `etape_2.py`)

1 Exercice sur papier (à rendre)

Pour cette première partie, forcez-vous à ne pas utiliser la machine. À chacune des étapes, décrivez la classe, ses attributs et (toutes) ses méthodes nécessaires. Pour les attributs, donnez le type. Pour les méthodes, donnez les paramètres et le type de valeur renvoyée.

1. Première étape (**une** classe) :

On souhaite décrire l'état des parcs naturels d'Île de France. Un(e) observateur(trice) est désigné(e) à une date donnée pour observer et noter les animaux et végétaux rencontrés au cours de la journée.

Par exemple, le 23 janvier, Nadia Brea recense les animaux et végétaux du parc du Vexin.

Pour l'instant, on veut juste savoir à la fin de la journée quels sont les animaux et plantes que l'observateur(trice) a croisés et produire un affichage du type :

Nom du parc : Parc Naturel du Vexin

Date : 23/11

Observateur(trice) : Nadia Brea

Animaux :

- sanglier

Vegetaux :

- fougere

- tulipe

2. Seconde étape :

Pour chaque spécimen observé (animal ou végétal), l'observateur(trice) renseigne le nom de l'espèce, le lieu d'observation (par exemple "forêt", "prairie", "lac", "chemin") où il(elle) l'a observé. Pour les végétaux, l'observateur(trice) renseigne si le végétal est comestible. Parmi les animaux rencontrés, il y a des mammifères et des insectes. Si l'animal est un mammifère, alors il est pesé et son poids (valeur entière) est renseigné. Si l'animal est un insecte, l'observateur(trice) tente de le capturer, et note si oui ou non il(elle) a pu le faire.

La modélisation permettra un affichage du type :

```
Nom du parc : Parc Naturel du Vexin
Date : 11/12
Observateur(trice) : Nadia Brea
animaux :
- ecureuil (lieu : foret, 4 kg)
- sanglier (lieu : chemin, 243 kg)
- sanglier (lieu : foret, 221 kg)
- papillon (lieu : foret, capture : False)
vegetaux :
- bolet (lieu : chemin, comestible : True)
- bolet (lieu : chemin, comestible : True)
```

Vous devez ici définir de nouvelles classes et modifier la classe précédente. **Faites le dans un nouveau fichier.**

2 Implémentation et test

1. Première étape : écrivez les classes correspondantes dans un module `module_1.py` ainsi qu'un programme `etape_1.py` qui permet de les tester. Créez les objets nécessaires pour produire l'affichage présenté en section 1.1.
2. Deuxième étape : écrivez les classes correspondantes dans un module `module_2.py` ainsi qu'un programme `etape_2.py` qui permet de tester. Créez les objets nécessaires pour produire l'affichage présenté en section 1.2.

3 Améliorations

1. Pour éviter d'avoir à ajouter un par un les animaux (ou végétaux) rencontrés, ajoutez une méthode permettant d'ajouter une liste d'animaux (ou de végétaux) directement (utilisez la fonction `extend()`).
2. Si ce n'est pas déjà fait, écrivez des fonctions `__str().__` lorsque c'est utile afin de produire des affichages propres. Utilisez la fonction `format()`.
3. Ajoutez la possibilité de capturer un insecte après l'avoir renseigné.
4. Numérotez les animaux à l'affichage (vous pouvez vous servir par exemple de la fonction `enumerate` pour accéder à l'index d'un élément d'une liste). Exemple d'utilisation :

```
1 L = ['a', 'b', 'c']
2 for id, x in enumerate(L):
3     print(id, x) // Affiche : (0, 'a') (1, 'b') (2, 'c')
```

5. affichage conditionnel (vous pouvez utiliser la compréhension de liste)
- (a) Affichez tous les animaux d'un parc dont le poids est supérieur à 100kg.
 - (b) Affichez tous les animaux et végétaux observés dans un certain lieu.