

### **Exercice 1 :**

Écrivez une classe city représentant une ville. Elle doit avoir les attributs suivants :

- nom : le nom de la ville en question (par exemple Paris)
- département : le département de la ville sous forme d'un integer

La valeur des attributs est déterminée dans un constructeur

A cette classe, vous rajouterez, les méthodes suivantes :

- une méthode show\_location() : elle affiche une string dans le terminal nous indiquant le nom de la ville et sa localisation, par exemple elle affiche : « la ville X est dans le département Y »
- une méthode change\_location() : elle attend deux paramètres, un nom sous forme de string et un département sous forme d'integer. Elle assigne à l'objet les nouvelles valeurs du nom et du département

Créez au moins 5 objets ville, affectez leurs propriétés, et utilisez la méthode d'affichage. Ensuite appelez la méthode change\_location sur l'un d'elle et vérifiez que ses coordonnées ont été modifiées.

Attention votre classe city est stockée dans un fichier séparé et appelée dans un fichier main.py.

### **Exercice 2 :**

Vous allez reprendre la classe précédemment créée et l'améliorer un peu. En effet, que se passerait-il si notre classe city avait des dizaines d'attributs ? Cela deviendrait vite chaotique. Impossible d'avoir autant d'attributs ? Réfléchissez 5 minutes à toutes les caractéristiques d'une ville : nom, département, population, pays, code postal, nom des habitants, date de création, dirigeant, personnages célèbres etc...

Vous trouverez dans le dossier qui vous a été donné un fichier cities.py qui contient une liste de dictionnaires. Chaque dictionnaire représente une ville.

A vous de modifier votre classe ville pour que son constructeur accepte un dictionnaire en argument et hydrate tous les attributs d'un coup.

Vous rajouterez une méthode show\_information() qui affiche la carte d'identité complète de la ville, c'est à dire l'ensemble de ses informations.

Dans votre fichier main.py réalisez ensuite une boucle pour instancier toutes les villes en une fois et afficher leur carte d'identité

Pour aller plus loin, faites en sorte que si un de vos dictionnaires contient des clés qui ne doivent pas être présentes dans votre objet alors l'attribut correspondant n'est pas créé. Un indice, recherchez du côté des méthodes setattr et getattr.

### **Exercice 3 :**

Cet exercice est un peu plus long. Il va vous montrer comment la programmation orientée objet nous permet de modéliser les processus physiques du monde qui nous entoure.

Dans cet exercice, vous allez modéliser une bibliothèque très simple et un lecteur qui vient emprunter un livre et le lit. Pour vous aider, les fichiers vides des classes vous ont été donnés ainsi qu'un fichier contenant une liste de livres. Vous trouverez également un fichier main.py avec un script d'exécution du code pré-écrit. Votre objectif est de faire en sorte que ce code soit fonctionnel sans modifications.

Votre programme sera composé des objets suivants :

- Le livre, il est composé d'un titre, du numéro de page par défaut sur lequel on se trouve et d'une liste de pages. Par simplicité, dans le fichier books\_list, vous verrez qu'il est simplement écrit 'page1', 'page2' etc. Dans le cadre d'une vraie application ces chaînes de caractères seraient beaucoup plus longues et contiendraient aussi la mise en forme et les retour à la ligne. Attention le fichier books\_list ne contient pas les objets livre, il contient les données des livres à faire rentrer dans des objets, à vous de coder la classe Book en conséquence.
- La bibliothèque, elle contient simplement la liste des objets livre, tout comme une bibliothèque dans le monde réel n'est qu'un entrepôt pour livre. Elle peut rajouter des livres à son catalogue et renvoyer l'objet livre correspondant sur la base d'un titre.
- Le lecteur, il ne contient que le livre qu'il a emprunté. Il peut réaliser plusieurs actions. Il peut emprunter un livre, aller à une page spécifique du livre, aller à la page suivante, aller à la précédente, lire une page c'est à dire afficher son contenu à l'écran et enfin lire l'entièreté du livre.