

I - TP 2

Introduction à la POO

1 Objets : attributs et création

Le libraire du coin de votre rue décide de faire appel à vous pour informatiser son système de gestion de livres : l'objectif de ce tp est de comprendre comment on peut rassembler les informations se rapportant à un livre, ainsi que les opérations que l'on peut effectuer sur les livres (en calculer le prix, par exemple) dans une structure cohérente.

En python, le mot-clé `class` permet de définir un nouveau type : c'est une nouvelle structure de donnée qui permet de représenter, structurer et manipuler des objets.

Code python

```
1 class Livre:
2     pass
3
4 livre1 = Livre()
```

Le code ci-dessus permet de créer un nouvel objet de type `Livre` : c'est le rôle de l'instruction `Livre()` (**avec des parenthèses**). Lorsque l'on exécute l'instruction `print(livre1)`, cela indique l'adresse mémoire où est stocké l'objet désigné par `livre1`.

Code python

```
1 # Un nouvel objet est créé et affecté à la variable livre1
2 print(livre1)
3 print(type(livre1))
```

 Résultat

```
<__main__.Livre object at 0x79cc39f01070>
<class '__main__.Livre'>
```

1.1 Attributs d'un objet

Le code ci-dessous permet d'ajouter différents **attributs** au livre : son titre (*La peste*), son auteur (Albert Camus), et son nombre total de page (59).

Code python

```
1 livre1.titre = "La peste"
2 livre1.auteur = "Camus Albert"
3 livre1.nb_pages = 59
```

Question 1. 1. Utiliser Thonny pour inspecter l'objet `livre1` : dans le menu *Affichage*, faire afficher les panneaux *Variables* et *Inspecteur d'objets* en cliquant sur ces entrées. Une fois le panneau *Inspecteur d'objets* ouvert, cliquer sur le bouton *Attributs*. Commenter.

2. Écrire le code python permettant de représenter le livre Fondation (256 pages), écrit par Isaac Asimov en 1951.

Code python

```
1 livre2 = Livre()
2 # À compléter
3 livre2.titre = ...
4 livre2.auteur = ...
5 livre2.nb_pages = ...
```

1.2 Création d'un nouveau livre

Écrire une fonction `creer_livre` qui prend en argument une chaîne de caractère `t`, une chaîne de caractères `a` et un entier positif `n` et qui renvoie l'objet de type `Livre` de titre `t`, d'auteur `a` et constitué de `n` pages.

Code python

```
1 def creer_livre(t, a, n):
2     """ str, str, int -> Livre
3     Renvoie un nouveau livre """
4     pass
```

Code python

```
1 print(livre3)
```

 Résultat

```
<__main__.Livre object at 0x79cc39a70dd0>
```

Question 2. À l'aide de la fonction `creer_livre`, instancier une variable `livre3` de type `Livre` qui représente le livre Liens de sang, écrit par Octavia Butler et publié pour la première fois en 1979 (467 pages dans l'édition française par la maison Au diable vauvert).

1.3 Fonction `__init__`

Commenter tout le code précédemment écrit. On remplace la définition précédente de la classe `Livre` par la définition suivante.

Code python

```
1 class Livre:
2     def __init__(self, t, a, n):
3         self.titre = t
4         self.auteur = a
5         self.nb_pages = n
```

Question 3. 1. Exécuter ce code, puis exécuter l'instruction dans la console :

Code python

```
1 Livre("Les misérables", "Hugo Victor", 2135)
```

Observer l'effet sur l'état de la mémoire. Commenter.

2. À quoi peut servir la fonction `__init__` placée dans le corps de la classe `Livre` ?
3.
 - a. Peut-on encore instancier une variable de type `Livre` avec l'instruction `Livre()` ?
 - b. Écrire les instructions permettant d'instancier les variables `livre1` (La peste, Albert Camus, 59 pages), `livre2` (Fondation, Isaac Asimov, 256 pages), `livre3` (Liens de sang, Octavia Butler, 467 pages) et `livre4` (Les misérables, Victor Hugo, 2135 pages) avec cette nouvelle définition de la classe `Livre`.

2 Premières fonctions

2.1 Afficher un livre

Le libraire est un peu perturbé par l'effet de l'instruction `print(livre1)` : il préférerait un affichage plus lisible. La fonction `affiche` prend en argument un objet `lvr` de type `Livre` et réalise un affichage partiel des informations contenues dans l'objet `lvr`.

Question 4. Modifier la fonction `affiche` afin que l'appel `affiche(livre1)` indique également le nom de l'auteur du livre dans la description affichée. Tester votre modification sur les variables de type `Livre` déjà définies.

Code python

```
1 def affiche(lvr):
2     """ Livre -> None
3     Affiche le livre lvr """
4     titre_livre = lvr.titre
5     pages_tot = lvr.nb_pages
6     # À modifier
7     print(f"< Livre '{titre_livre}', nombre de pages {pages_tot} >")
```

Code python

```
1 affiche(livre1)
```

 Résultat

< Livre 'La peste', auteur Camus Albert, nombre de pages 59 >

2.2 Prix d'un livre

2.2.1 Prix de base

Ce libraire peu adepte de la qualité intrinsèque des livres calcule le prix d'un livre de la manière suivante :

- le livre est égal à la somme des prix de ses pages ;
- chaque page avant la 100^è (inclue) coûte 10 centimes ;
- chaque page entre la 101^è (inclue) et la 1000^è (inclue) coûte 2 centimes ;
- chaque page après la 1001^è (inclue) coûte 1 centime.

Écrire une fonction `prix` qui étant donné un livre `lvr` et un flotant `prix_page` renvoie le prix du livre.

Code python

```
1 def prix(lvr):
2     """ Livre -> float
3     Calcule le prix d'un livre """
```

Code python

```
1 print(prix(livre1))
```

 Résultat

8.85

2.2.2 Prix réduit

Il arrive au libraire de faire occasionnellement des réductions sur le prix des livres. Comme il oublie systématiquement quel calcul il doit effectuer sur sa calculatrice, il aimerait disposer d'une fonction toute prête qui fasse le travail à sa place.

Écrire une fonction `prix_reduit` qui étant donné un livre `lvr`, et un pourcentage de réduction `reduc` (un entier compris entre 0 et 100) calcule le prix du livre, après une réduction de `reduc` pourcent.

Code python

```
1 def prix_reduit(lvr, reduc):
2     """ Livre, int -> float
3     Renvoie le prix du livre après réduction """
4     pass
```

Code python

```
1 print(prix_reduit(livre1, 50))
```

 Résultat

4.425

2.3 Méthodes d'un objet

Sur ce même modèle, on peut imaginer :

- d'autres **attributs** pour la classe Livre. Ils permettent d'ajouter des informations qui permettent de **décrire** l'objet.
- d'autres fonctions (ou **méthodes**) opérant sur les objets de type livre. Elles permettent de **réaliser des opérations** sur les objets de type Livre.

Afin de structurer efficacement notre code, nous allons *encapsuler* toutes ces informations dans la classe qui représente notre objet.

Remplacer le code déjà existant de la classe Librairie par le code ci-dessous.

Code python

```
1 class Livre:
2     def __init__(self, t, a, n):
3         self.titre = t
4         self.auteur = a
5         self.nb_pages = n
6
7     def affiche(self):
8         """ Livre -> None """
9         titre_livre = self.titre
10        auteur_livre = self.auteur
11        pages_tot = self.nb_pages
12        print(f"< Livre '{titre_livre}', auteur {auteur_livre}, nombre de
13             ↪ pages {pages_tot} >")
14
15    def prix(self):
16        """ Livre -> float
17        Calcule le prix d'un livre """
18        if self.nb_pages <= 100:
19            return self.nb_pages * 0.15
20        elif 100 <= self.nb_pages <= 1000:
21            return 100*0.10 + (self.nb_pages - 100)*0.01
22        else:
23            return 100*0.10 + 900*0.02 + (self.nb_pages - 1000)*0.01
24
25    # Initialisation des variables
26    livre1 = Livre("La peste", "Camus Albert", 59)
27    livre2 = Livre("Fondation", "Asimov Isaac", 265)
28    livre3 = Livre("Liens de sang", "Octavia Butler", 467)
29    livre4 = Livre("Les misérables", "Victor Hugo", 2135)
```

2.3.1 Méthodes d'un objet : utilisation

Exécuter dans la console les lignes suivantes.

```
Code python
1 livre1.affiche()
2 livre1.prix()
```

Question 5. 1. Quelle est la différence entre cette syntaxe en **notation objet** et la syntaxe `affiche(livre1)` et `prix(livre1)` ?

2. Copier coller le code de la fonction `prix_reduit` dans la classe `Livre`, au même degré d'indentation que celui de la fonction `prix`.

a. Exécuter dans la console l'instruction `livre1.prix_reduit(50)`.

Commenter et corriger l'erreur soulevée par python.

À quoi ce résultat correspond-t-il ?

b. Quelle instruction python permet de calculer le prix du roman Les misérables, si on lui applique une réduction de 42% ?

2.3.2 Ajout d'un attribut

Modifier la méthode `__init__` de la classe `Livre` afin d'ajouter un attribut disponible à la classe `Livre` : il s'agit d'un entier positif ou nul qui représente combien de livres de ce type sont disponibles chez le libraire. L'attribut disponible d'un livre aura pour valeur par défaut 1.

```
Code python
1 print(livre1.disponible)
2 print(livre2.disponible)
```

⚙️ ➤ Résultat

```
1
1
```

2.3.3 Reception d'un livre

Écrire une méthode `recevoir` de la classe `Livre` : cette méthode prend comme argument un nombre entier `n` correspondant au nombre de copies supplémentaires reçues du livre `self`. Celles-ci seront ajoutées aux copies disponibles du livre : on modifiera pour cela l'attribut `disponible`. **Attention.** On rappelle que le code de cette méthode devra être écrit *dans* la classe `Livre` (c'est à dire au bon niveau d'indentation).

```
Code python
1 def recevoir(self, n):
2     """ Livre, int -> None
3     n nouveaux exemplaires du livre sont disponibles """
4     pass
```

```

Code python
1 livre1.recevoir(1)
2 print(livre1.disponible)
3 livre2.recevoir(3)
4 print(livre2.disponible)

```

⚙️ ➤ Résultat

2
4

2.3.4 Vente d'un livre

Écrire une méthode vendre de la classe Livre qui prend en argument un entier reduc et qui simule la vente du livre self :

- si aucun livre de ce type n'est disponible, la méthode renvoie 0.
- sinon : on calcule le prix du livre self en appliquant la réduction de reduc pourcent, et on décrémente de 1 l'attribut disponible de self.

Enfin la méthode renvoie le prix du livre en euros, arrondi au centime.

```

Code python
1 def vendre(self, prix_page, reduc):
2     """ Vend le livre, renvoie l'argent généré par la vente. """
3     pass

```

⚙️ ➤ Résultat

```

Code python
1 print(livre4.vendre(42))

```

22.82

3 Gestion d'une librairie

3.1 Introduction et motivation

Notre libraire cherche maintenant à gérer une collection de livre. Son stock de livre ne faisant qu'augmenter, il souhaite pouvoir répondre aux clients lui demandant s'il possède ou non un livre dont on lui donne le titre.

Écrire une fonction recherche qui étant donné une liste de livres collection et un titre, renvoie le nombre d'exemplaires disponibles du livre intitulé titre que possède le libraire. On supposera que chaque titre de la liste collection est unique.

```

Code python
1 def recherche(collection, titre):
2     """ [Livre], str -> int
3     Nombre de livres disponible avec le titre indiqué dans la collection.
4     ↪ """
5     pass

```

⚙️ ➤ Résultat

```

Code python
1 print(recherche([livre1, livre2], "La peste"))
2 print(recherche([livre1, livre2], "Le briocolage
↪ pour les nuls"))

```

2
0

Question 6. Que renvoie l'instruction ci-dessous ? Commenter.

```

Code python
1 recherche([livre1, livre2, livre3, livre4], "Les misérables")

```

3.2 La classe Librairie

Afin de structurer efficacement notre code, on décide de regrouper toutes les fonctions opérant sur une collection de livres dans une classe intitulée `Librairie`. Ainsi, à l'aide de cette classe, le libraire pourra efficacement gérer l'ensemble des livres de sa librairie. Dans le code ci-dessous, l'attribut `collection` est un objet de type `list` constitué d'objets de type `Livre`.

Code python

```
1 class Librairie:
2     def __init__(self, c):
3         self.collection = c
```

- Question 7.**
1. Avec quelle instruction instancie-t-on une librairie vide (ne contenant aucun livre) ?
 2. Instancier une variable `lib` de type `Librairie` représentant une librairie contenant les livres suivants :
 - 8 exemplaires du livre La peste, d'Albert Camus (59 pages) ;
 - 2 exemplaires du livre Fondation, d'Isaac Asimov (256 pages) ;
 - 4 exemplaires du livre Liens de sang, d'Octavia Butler (467 pages)
 - 3 exemplaires du livre Les misérables, de Victor Hugo (2135 pages) ;
 - 5 exemplaires du livre Harry Potter à l'école des sorciers, de J. K. Rowling (132 pages)
 3. Transformer la fonction recherche écrite précédemment en une méthode recherche de la classe `Librairie`.

3.2.1 Ajouter un livre dans une librairie

- Question 8.**
1. Écrire une méthode `ajoute` de la classe `Librairie` qui prend en argument un objet `lvr` de type `Livre` et qui l'ajoute à la librairie représentée par `self`.

Code python

```
1 def ajoute_livre(self, lvr):
2     """ Librairie, Livre -> None
3     Ajoute le livre lvr à la librairie self """
4     pass
```

2. Écrire le code python permettant d'ajouter à la variable `lib` définie précédemment 11 exemplaires du livre Le roi Babar (15 pages), écrit par Jean de Brunhoff.

Code python

```
1 print(len(lib.collection)) #après ajout du livre Le roi Babar
```

 > Résultat

6

3.2.2 Afficher l'ensemble des livres de la librairie

Écrire une méthode `affiche` de la classe `Librairie` qui affiche les informations de tous les livres présents dans la librairie `self`.

On utilisera pour cela un parcours de liste de telle sorte que pour chaque élément `lvr` de l'attribut `collection` de la variable `self`, on fasse appel à la méthode `affiche` de l'objet de `lvr` afin d'afficher le livre en question.

Code python

```
1 def affiche(self):  
2     """ Librairie -> None  
3     Affiche les livres de la librairie """  
4     pass
```

Code python

```
1 lib.affiche()
```

 Résultat

```
< Livre 'La peste', auteur Camus Albert, nombre de pages 59 >  
< Livre 'Fondation', auteur Asimov Isaac, nombre de pages 265 >  
< Livre 'Liens de sang', auteur Octavia Butler, nombre de pages 467 >  
< Livre 'Les misérables', auteur Hugo Victor, nombre de pages 2135 >  
< Livre 'Harry Potter à l'école des sorciers', auteur J. K. Rowling,  
  ↳ nombre de pages 132 >  
< Livre 'Babar', auteur Jean de Brunhoff, nombre de pages 15 >
```

3.2.3 À vous de jouer

1. Ajouter un nouvel attribut de votre choix à la classe Livre.

Écrire une méthode de la classe Livre qui ajoute une fonctionnalité aux objets de type Livre (à vous de l'inventer!).

2. Ajouter un nouvel attribut de votre choix à la classe Librairie.

Écrire une méthode de la classe Librairie qui ajoute une fonctionnalité aux objets de type Librairie (à vous de l'inventer!).