



Partie 1 : épreuve écrite

Exercice 1. Une Bibliothèque

Thème abordé : programmation objet en langage Python

On s'intéresse à la gestion d'une bibliothèque. Une bibliothèque possède des livres et des lecteurs qui peuvent emprunter des livres. On souhaite modéliser ce système en utilisant la programmation orientée objet en Python.

Partie 1 : Classe Livre

On considère la classe suivante :

```
class Livre:
    """Modélise un livre de la bibliothèque"""
    def __init__(self, titre, auteur):
        """
        Initialise un livre
        :param titre (str) le titre du livre
        :param auteur (str) l'auteur du livre
        """
        self.titre = titre
        self.auteur = auteur
        self.disponible = True
```

1. Déterminer les types de : Livre, titre, auteur, disponible et __init__ (classe, attribut, méthode ou objet ...).
2. Indiquer le type attendu pour le paramètre titre.
3. Comment définir la variable self.
4. Écrire une instruction permettant de créer une instance livre1 représentant « Le Petit Prince » d'Antoine de Saint-Exupéry.
5. Compléter la méthode suivante :

```
def est_disponible(self):
    """
    Renvoie True si le livre est disponible, False sinon.
    """
    ...
```

Partie 2 : Classe Lecteur

On modélise maintenant un lecteur :

```
class Lecteur:
    """Modélise un lecteur de la bibliothèque"""
    def __init__(self, nom):
        self.nom = nom
        self.emprunts = [] # liste des livres empruntés
```

5. Compléter la méthode emprunter :

```
def emprunter(self, livre):
    """
    Ajoute le livre à la liste des emprunts si le livre est disponible,
    et marque le livre comme non disponible.
    Renvoie True si l'emprunt réussit, False sinon.
    """
    ...
```

6. Compléter la méthode rendre :

```
def rendre(self, livre):
    """
    Retire le livre de la liste des emprunts et marque le livre
    comme disponible.
    Renvoie True si la restitution réussit, False sinon.
    """
    ...
```

Partie 3 : Classe Bibliotheque

Une bibliothèque contient plusieurs livres :

```
class Bibliotheque:
    """Modélise une bibliothèque"""
    def __init__(self, nom):
        self.nom = nom
        self.livres = [] # liste d'objets Livre
```

7. Compléter la méthode ajouter_livre :

```
def ajouter_livre(self, livre):
    """
    Ajoute le livre à la bibliothèque.
    """
    ...
```

8. Compléter la méthode livres_disponibles :

```
def livres_disponibles(self):
    """
    Renvoie la liste des titres des livres encore disponibles.
    """
    ...
```

9. Compléter la méthode chercher_auteur :

```
def chercher_auteur(self, auteur):
    """
    Renvoie la liste des titres de livres écrits par l'auteur
    passé en paramètre.
    """
    ...
```

Partie 4 : Algorithmique

10. On souhaite savoir quel lecteur a emprunté le plus grand nombre de livres. On dispose d'une liste tab_lecteurs d'instances de Lecteur. Écrire la fonction lecteur_max :

```
def lecteur_max(tab_lecteurs):
    """
    Renvoie le lecteur ayant emprunté le plus de livres.
    Si plusieurs lecteurs ont le même maximum, renvoie l'un d'entre eux.
    Si la liste est vide, renvoie None.
    """
    ...
```

↩ **Fin du devoir** ↪



Remarque

| Rendez votre copie avec le sujet au professeur puis travailler sur votre TD.