



ÉCOLE CENTRALE LYON

UE INF
BIBLIOTHEQUE
RAPPORT

Rapport

Élèves :

Luc GUIDI
Hugo PUYBAREAU

Enseignant :

Daniel MULLER

17 mars 2023

Table des matières

1	Diagramme des classes	2
2	Ajout des nouvelles classes	3
2.1	Programmation de la classe 'Bibliothecaire'	3
2.2	Programmation de la classe 'Conservateur'	3
3	Programmation de la bibliothèque	4
3.1	Modification de la classe 'Emprunt'	4
3.2	Modification de la classe 'Bibliotheque'	4
4	Tests	5

1 Diagramme des classes

La première étape de notre travail fut de tracer le diagramme des classes. Dans ce dernier, nous retrouvons les liens entre les classes, les attributs et les méthodes de ces dernières ainsi que les cardinalités. Le diagramme est le suivant :

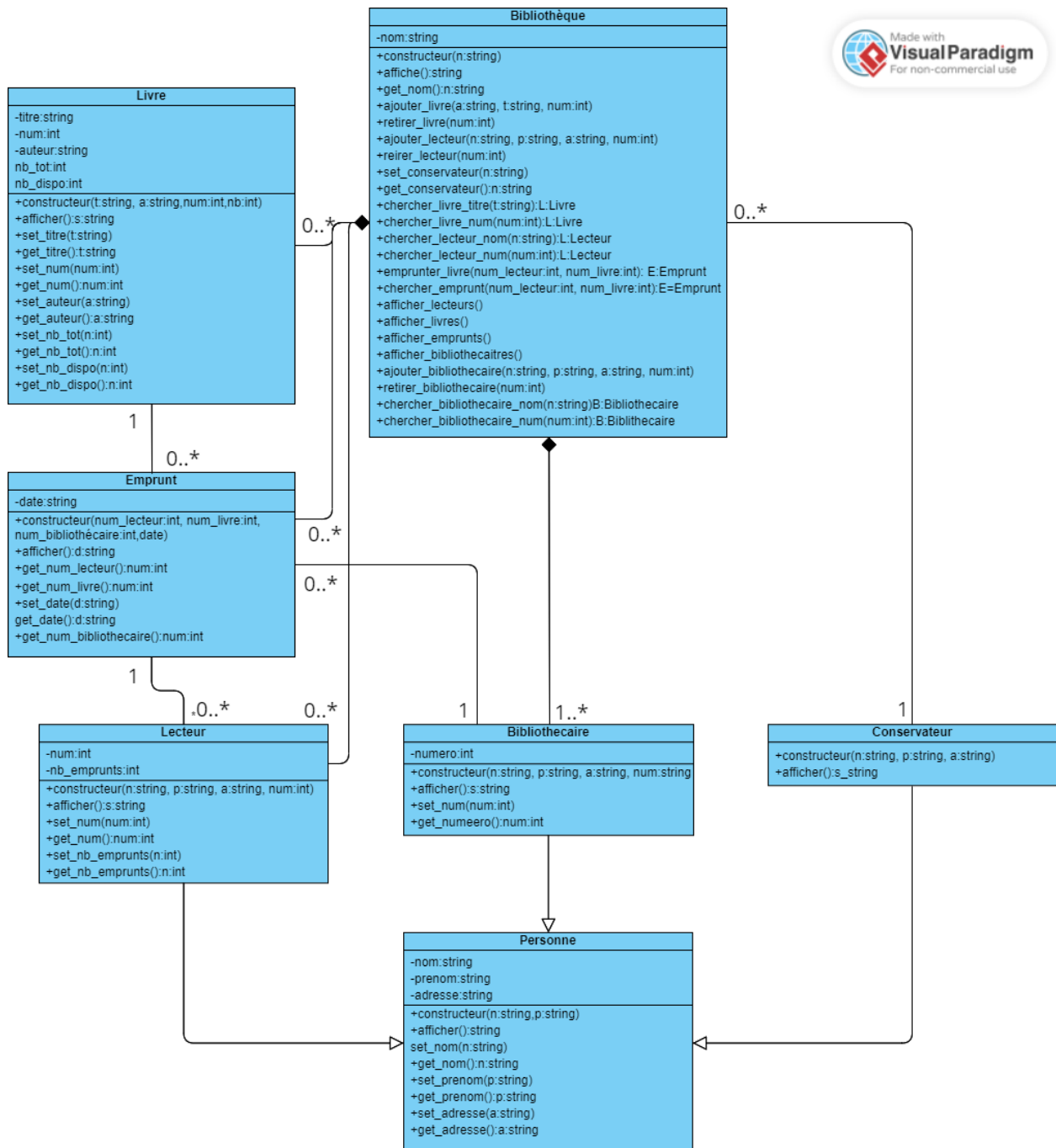


FIGURE 1 – Diagramme UML de la bibliothèque

2 Ajout des nouvelles classes

2.1 Programmation de la classe 'Bibliothecaire'

Avant tout, il est important de rappeler que les codes des classes Personne, Livre, Emprunt, Lecteur et Bibliotheque furent faits lors de la séance de TD précédente. Notre première tâche fut donc de créer la nouvelle classe 'Bibliothecaire'. Comme schématise le diagramme UML. Cette classe hérite de toute la classe personne et contient les fonctions `__init__`, `set_num`, `get_num` et `__str__` que nous retrouvons bien dans le diagramme des classes.

```
#Class Bibliothecaire

from personne import *

class Bibliothecaire(Personne):
    def __init__(self,nom,prenom,adresse,num):
        Personne.__init__(self,nom,prenom,adresse)
        self.__num=num

    def set_num(self,num):
        self.__num = num

    def get_num(self):
        return self.__num

    def __str__(self):
        return 'Bibliothecaire - Nom : {}, Prenom : {}, Adresse : {}, Numero : {}'.format(self.get_nom(),self.get_prenom(),self.get_adresse(),self.__num)
```

FIGURE 2 – Code de la classe 'Bibliothecaire'

2.2 Programmation de la classe 'Conservateur'

Dans un second temps, nous avons rajouté la classe 'Conservateur' qui, comme expliqué sur le diagramme des classes, dérive de la classe 'Personne' et présente les méthodes `__init__`, `get_bibliotheque` et `__str__`. Le code est donc le suivant :

```
#Class Conservateur

from personne import *
from bibliothecaire import *

class Conservateur(Personne) :

    def __init__(self,nom,prenom,adresse,bib):
        Personne.__init__(self,nom,prenom,adresse)
        self.__bibliotheque=[]
        self.__bibliotheque=bib

    def get_bibliotheque(self):
        return self.__bibliotheque

    def __str__(self):
        pr=Personne.get_prenom(self)
        nom=Personne.get_nom(self)
        ad=Personne.get_adresse(self)
        return f"Classe Conservateur - Nom : {nom}, Prenom : {pr}, Adresse : {ad}, Bibliotheque : {self.__bibliotheque}"
```

FIGURE 3 – Code de la classe 'Conservateur'

3 Programmation de la bibliothèque

3.1 Modification de la classe 'Emprunt'

Chaque emprunt doit pouvoir être relié à un l'identifiant du bibliothécaire l'ayant créé. Nous avons donc rajouté une fonction `get_num_bibliothecaire` qui renvoie le numéro du bibliothécaire qui a enregistré l'emprunt. La fonction `__init__` est elle aussi modifiée pour que chaque emprunt soit initialisé avec le numéro du bibliothécaire (comme la fonction `__str__` qui affichera maintenant le numéro du bibliothécaire). Les changements sont donc les suivants :

```
class Emprunt:
    def __init__(self, num_lecteur, num_livre, num_bibliothecaire, date):
        self.__num_lecteur = num_lecteur
        self.__num_livre = num_livre
        self.__date = date
        self.__num_bibliothecaire = num_bibliothecaire
```

FIGURE 4 – Modification de `__init__()`

```
def get_num_bibliothecaire(self):
    return self.__num_bibliothecaire
```

FIGURE 5 – Ajout de la fonction `get_num_bibliothecaire()`

3.2 Modification de la classe 'Bibliotheque'

Dans un premier temps, avons modifié la fonction `__init__` de cette classe. En effet, nous faisons le choix d'initier la bibliothèque directement avec le conservateur qui lui est associé. De plus, nous rajoutons parmi les listes déjà présentes la liste des bibliothécaires. Le code est le suivant :

```
11 class Bibliotheque:
12     def __init__(self, nom, c):
13         self.__nom = nom
14         self.__conservateur = c
15         self.__lecteurs = []
16         self.__livres = []
17         self.__emprunts = []
18         self.__bibliothecaires = []
19
```

FIGURE 6 – Modification de l'initialisation de la Bibliotheque

De plus, l'ajout des précédentes classes entraine le besoin d'ajouter les méthodes suivantes :

```

26 def get_conservateur(self):
27     return self.__conservateur

```

FIGURE 7 – Ajout de la fonction `get_conservateur()`

```

39 def ajouter_bibliothecaire(self,nom,prenom,adresse,num):
40     if self.chercher_bibliothecaire_num(num)==None:
41         self.__bibliotheques.append(Bibliothecaire(nom,prenom,adresse,num))
42     else:
43         print('Ajout impossible')
44         return None
45
46 def chercher_bibliothecaire_nom(self,nom):
47     for l in self.__bibliotheques:
48         if l.get_nom() == nom:
49             return l
50     return None
51
52 def supprimer_bibliothecaire(self,nom,prenom,adresse,num):
53     bibliothecaire = self.chercher_bibliothecaire_nom(nom) # On cherche le bibliothecaire
54     if bibliothecaire == None:
55         return False
56     self.__bibliotheques.remove(bibliothecaire) # On peut ici retirer le bibliothecaire de la liste
57     return True
58

```

FIGURE 8 – Ajout des fonctions `ajouter_bibliothecaire()`, `chercher_bibliothecaire_nom()`, `supprimer_bibliothecaire()`

4 Tests

Nous avons faits les test suivants pour verifier le bon fonctionnement de nos classes :

```

1  #Tests
2  from bibliotheque import *
3
4  #On ne testera pas les anciens programmes faits au TD2
5
6  print('-- Test sur Bibliothecaire --')
7  print('')
8
9  B=Bibliothecaire('Pierre', 'Louis', 'Ecully', 14)
10 print(B)
11
12 B.set_num(13)
13 print(B.get_num())
14
15
16 print('')
17 print('-- Test sur Conservateur --')
18 print('')
19
20 C=Conservateur('Olivier', 'Giroud', 'Bordeaux', 'Bibliotheque ECL')
21 print(C)
22 print(C.get_bibliotheque())
23
24 print('')
25 print('-- Test sur Emprunts --')
26 print('')
27
28 E=Emprunt(8, 88, 10, '15-03-2023')
29 print(E)
30
31 print(E.get_num_bibliothecaire())
32
33 print('')
34 print('-- Test sur la classe Bibliotheque--')
35 print('')
36
37 Biblio=Bibliotheque('Bibliothèque ECL',Conservateur('Moulin', 'Maud', 'Bordeaux', 'Bibliotheque ECL'))
38 Biblio.ajouter_bibliothecaire('Pierre', 'Louis', 'Ecully', 14)
39 Biblio.ajouter_bibliothecaire('Dupont', 'Fabien', 'Bordeaux', 8)
40
41 print(Biblio.affiche_bibliothecaires())
42
43 Biblio.ajouter_bibliothecaire('Pavard', 'Benjamin', 'Bordeaux', 8)
44
45 Biblio.supprimer_bibliothecaire('Pierre', 'Louis', 'Bordeaux', 8)
46 print(Biblio.affiche_bibliothecaires())
47
48 print(Biblio.get_conservateur())
49
50 print(Biblio.chercher_bibliothecaire_num(14))
51
52 Biblio.ajouter_livre('', '', 56, 2)
53 Biblio.ajouter_livre('auteur', 'titre', 94, 2)
54 Biblio.ajouter_lecteur('nom', 'prenom', 'adresse', 51)
55 Biblio.ajouter_lecteur('nom', 'prenom', 'adresse', 67)
56

```

FIGURE 9 – Programme de test