

INDEX

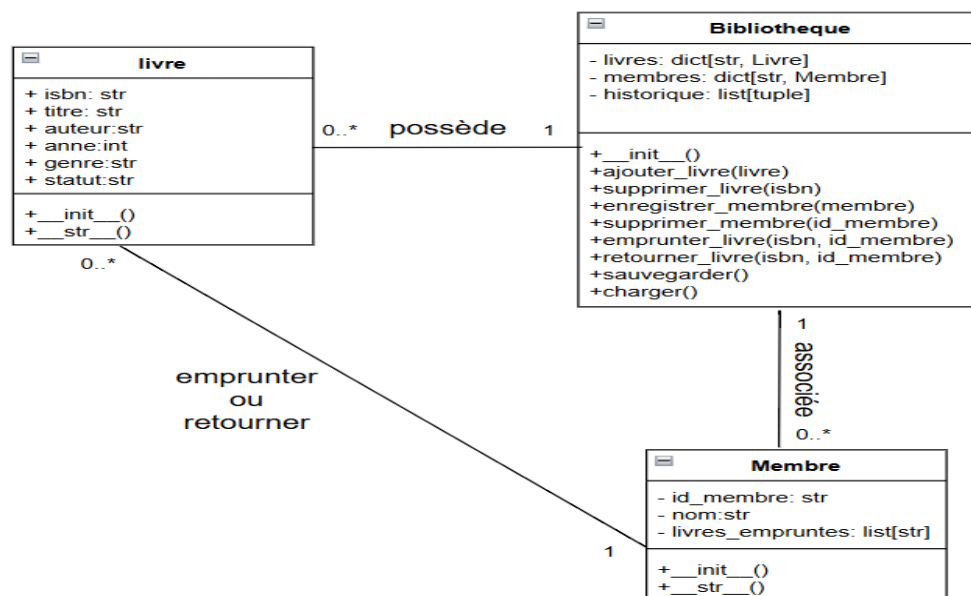
I.	Introduction	
II.	Diagramme UML	
	1 . Diagramme de classe	
III.	Explications des algorithmes clés	
	1 . Class Bibliothèque	
	2. Les exceptions.....	
	3. visualisation.....	
	4. Main.....	
IV.	Interface de l'application	
V.	Difficultés rencontrées et solutions	
VI.	Conclusion	

I .Introduction :

Ce projet est une implémentation d'un système de gestion de bibliothèque en Python avec une interface graphique réalisée à l'aide de Tkinter. Il permet de gérer les livres, les membres, les emprunts et les retours, tout en offrant des statistiques visuelles grâce à la bibliothèque Matplotlib. L'utilisateur peut facilement interagir avec l'interface pour ajouter ou supprimer des livres, enregistrer des membres et consulter l'état de la bibliothèque. Des graphiques illustrent la répartition des genres, les auteurs les plus empruntés, ainsi que l'activité des 30 derniers jours. L'application repose sur la programmation orientée objet, assure la sauvegarde des données via des fichiers, et propose une navigation fluide grâce à une interface conviviale enrichie d'icônes.

II. Diagramme UML :

1.diagramme de classe :



Le système repose sur trois entités : la bibliothèque, le livre et le membre. La bibliothèque a en stock les livres disponibles, les membres inscrits, ainsi que les actions de prêt, de retour et l'historique. Chaque livre, avec ses données (ISBN, titre, auteur, genre, année, statut), appartient à la bibliothèque et ne peut être prêté qu'à un membre en la jouissance exclusive. Chaque membre, identifié de manière unique, peut emprunter

plusieurs livres simultanément. La bibliothèque suit précisément les emprunts et retours.

III. Explications des algorithmes clés :

1 . class Bibliothèque:

La classe Bibliothèque est le cœur de l'application. Elle gère les livres, les membres et l'historique des emprunts et retours. Chaque livre possède des informations clés et un statut, chaque membre a un identifiant, un nom et

```
def emprunter_livre(self, isbn, id_membre): 1usage
    if id_membre not in self.membres:
        raise MembreInexistantError("Membre inexistant.")
    if isbn not in self.livres:
        raise LivreInexistantError("Livre inexistant.")
    if self.livres[isbn].statut != "disponible":
        raise LivreIndisponibleError("Livre déjà emprunté.")
    if len(self.membres[id_membre].livres_empruntes) >= 3:
        raise QuotaEmpruntDepasseError("Ce membre a atteint le quota maximal de 3 emprunts.")

    self.livres[isbn].statut = "emprunte"
    self.membres[id_membre].livres_empruntes.append(isbn)
    self._log_action(action="emprunt", isbn, id_membre)
    self.sauvegarder()
```

```
def retourner_livre(self, isbn, id_membre): 1usage
    if id_membre not in self.membres:
        raise MembreInexistantError("Membre inexistant.")
    if isbn not in self.livres:
        raise LivreInexistantError("Livre inexistant.")
    if isbn not in self.membres[id_membre].livres_empruntes:
        raise ValueError("Ce livre n'a pas été emprunté par ce membre.")

    self.membres[id_membre].livres_empruntes.remove(isbn)
    self.livres[isbn].statut = "disponible"
    self._log_action(action="retour", isbn, id_membre)
    self.sauvegarder()
```

ses emprunts. La classe permet d'ajouter ou supprimer livres et membres, gérer les prêts, et sauvegarder ou charger les données automatiquement.. Lors d'un **emprunt**, le programme vérifie que le membre existe, que le livre est disponible et que le membre n'a pas dépassé la limite. Pour un **retour**, il vérifie que le livre appartient bien au membre avant de le rendre disponible à nouveau. Toutes les modifications sont automatiquement enregistrées pour garder

les données à jour.

2.les exceptions :

On utilise des exceptions personnalisées pour gérer les erreurs de manière claire et structurée. **LivreIndisponibleError** est levée lorsqu'un livre déjà emprunté est demandé. **QuotaEmpruntDepasseError** signale qu'un membre a dépassé la limite d'emprunts autorisée. **MembreInexistantError**

apparaît lorsque l'identifiant d'un membre est introuvable, et **LivreInexistentError** est déclenchée si l'ISBN fourni ne correspond à aucun livre. Ces exceptions permettent d'afficher des messages explicites à l'utilisateur et améliorent l'expérience globale.

3.visualisation :

Les Trois fonctions de visualisation permettent d'analyser l'activité de la bibliothèque : un diagramme circulaire pour la répartition des livres par genre, un histogramme des auteurs les plus présents, et une courbe illustrant les emprunts sur les 30 derniers jours. Ces graphiques offrent un aperçu clair et rapide des données.

4.Main :

Ce code de gestion de bibliothèque, conçu avec *Tkinter*, permet de gérer les livres, les membres, les emprunts et les retours, tout en affichant des statistiques visuelles. L'interface comprend trois onglets (Livres, Membres, Statistiques) et un tableau central. La logique métier est regroupée dans la classe Bibliothèque, qui utilise la programmation orientée objet et des exceptions personnalisées pour traiter les erreurs.

IV. Interface de l'application :

Tous les Livres

ISBN	Titre	Auteur	Année	Genre	Statut	ID Membre	Nom Membre
9780439139595	Harry Potter et le Prisonnier d'Azkaban	J.K. Rowling	1999	Fantastique	emprunté	M001	Omar El Haddadi
9782070736315	Les Fleurs du Mal	Charles Baudelaire	1857	Poésie	emprunté	M004	Khadija Bouziane
9782370490096	Demain	Guillaume Musso	2013	Romanesque	emprunté	M002	Fatima Zahra Idrissi
9780140449266	Les Misérables	Victor Hugo	1862	Classique	emprunté	M006	Samia Choukri
9782363066329	Le Petit Prince	Antoine de Saint-Exupéry	1943	Conte, philosophie	emprunté	M003	Youssef Banielloum

✓ sauvegarder et quitter

Tous les Livres

ISBN	Titre	Auteur	Année	Genre	Statut	ID Membre	Nom Membre
9780439139595	Harry Potter et le Prisonnier d'Azkaban	J.K. Rowling	1999	Fantastique	emprunté	M001	Omar El Haddadi
9782070736315	Les Fleurs du Mal	Charles Baudelaire	1857	Poésie	emprunté	M004	Khadija Bouziane
9782370490096	Demain	Guillaume Musso	2013	Romanesque	emprunté	M002	Fatima Zahra Idrissi
9780140449266	Les Misérables	Victor Hugo	1862	Classique	emprunté	M006	Samia Choukri
9782363066329	Le Petit Prince	Antoine de Saint-Exupéry	1943	Conte, philosophie	emprunté	M003	Youssef Banielloum

✓ sauvegarder et quitter

Tous les Livres

ISBN	Titre	Auteur	Année	Genre	Statut	ID Membre	Nom Membre
9780439139595	Harry Potter et le Prisonnier d'Azkaban	J.K. Rowling	1999	Fantastique	emprunté	M001	Omar El Haddadi
9782070736315	Les Fleurs du Mal	Charles Baudelaire	1857	Poésie	emprunté	M004	Khadija Bouziane
9782370490096	Demain	Guillaume Musso	2013	Romanesque	emprunté	M002	Fatima Zahra Idrissi
9780140449266	Les Misérables	Victor Hugo	1862	Classique	emprunté	M006	Samia Choukri
9782363066329	Le Petit Prince	Antoine de Saint-Exupéry	1943	Conte, philosophie	emprunté	M003	Youssef Banielloum

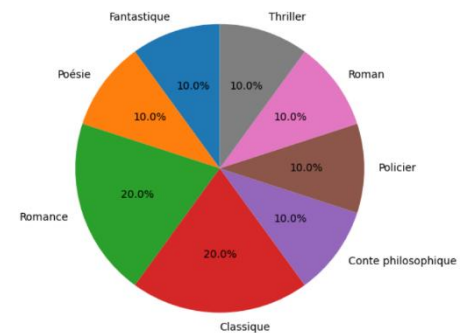
✓ sauvegarder et quitter

L'application se compose de trois interfaces principales. La première permet de gérer les livres (ajout, suppression, affichage des emprunts). La deuxième est dédiée à la gestion des membres et de leurs emprunts. La troisième interface

affiche des statistiques visuelles (genres, auteurs, activité) générées avec Matplotlib.

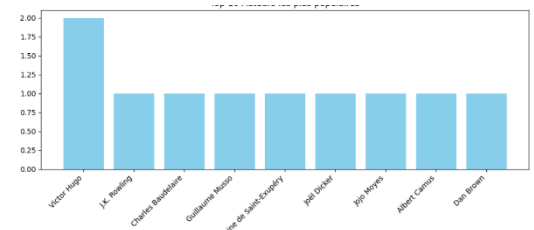
✓ **Diagramme circulaire (genres) :**

Affiche la répartition des livres par genre, permettant de visualiser les préférences en matière de catégories .



✓ **Histogramme (auteurs) :**

Représente les 10 auteurs les plus empruntés dans la bibliothèque, offrant une vue rapide des auteurs les plus populaires.



✓ **Courbe d'activité (emprunts) :**

Montre l'évolution quotidienne des emprunts sur les 30 derniers jours, permettant de suivre l'activité récente de la bibliothèque



V . Difficultés rencontrées et solutions :

- **Sauvegarde des données** : utilisation de fichiers texte mis à jour automatiquement pour conserver livres, membres et historiques.
- **Mise à jour de l'interface** : actualisation du tableau après chaque action pour refléter les modifications en temps réel.
- **Gestion des erreurs utilisateurs** : création d'exceptions personnalisées pour afficher des messages clairs et précis.

VI . conclusion :

Ce gestionnaire de bibliothèque propose une solution intuitive et complète pour gérer les livres, les membres, ainsi que les emprunts et retours. Grâce à une interface claire et des fonctionnalités bien conçues, il facilite le suivi des ressources tout en offrant des outils de visualisation pour mieux comprendre l'activité. Cette application suit pleinement les principes de la programmation orientée objet, la gestion des données et l'interface utilisateur, répondant ainsi aux besoins essentiels d'une bibliothèque moderne.