



جامعة محمد الأول بوجدة
UNIVERSITE MOHAMMED PREMIER OUJDA
ⵜⴰⵎⴰⵎⴰⵔⵜ ⵜⴰⵎⴰⵏⴰⵢⵜ ⵜⴰⵖⵔⴰⵏⵜ ⵜⴰⵏⴰⵢⵜ ⵜⴰⵖⵔⴰⵏⵜ

Systeme de Gestion de Bibliothèque

Douae Mahmoudi

INDEX

- **INTRODUCTION**

1. **Diagrammes UML**

- 1.1. Diagramme de classe.....

2. **Explications des algorithmes clés**

- 2.1. Classe Bibliotheque.....

- 2.1.1. Définition des classes Livre et Membre, initialisation de la classe Bibliotheque et gestion des fichiers.....

- 2.1.2. Gestion des livres, membres et emprunts avec persistance.....

- 2.2. Exceptions.....

- 2.3. Visualisations.....

- 2.4. Main.....

3. **Illustrations des visualisations**

- 3.1. Interface Livres.....

- 3.2. Interface Membres.....

- 3.3. Interface Statistiques.....

4. **Difficultés rencontrées et solutions**

- 4.1. Intégration des images dans Tkinter.....

- 4.2. Sauvegarde des données.....

- **CONCLUSION**

INTRODUCTION

Python est actuellement l'un des langages les plus populaires pour le développement de programmes puissants et faciles à maintenir. Associé à la bibliothèque Tkinter, il permet de créer des interfaces graphiques simples, interactives et efficaces. De plus, Python offre de nombreuses bibliothèques utiles pour la gestion des fichiers, la gestion des erreurs, et la génération de graphiques.

Dans une bibliothèque, la gestion manuelle des livres, des membres et des emprunts peut rapidement devenir complexe et source d'erreurs. C'est pourquoi ce projet vise à développer une application de gestion de bibliothèque qui automatise toutes ces tâches : ajout de livres, enregistrement de membres, gestion des emprunts et des retours, sauvegarde automatique des données, et création de statistiques visuelles.

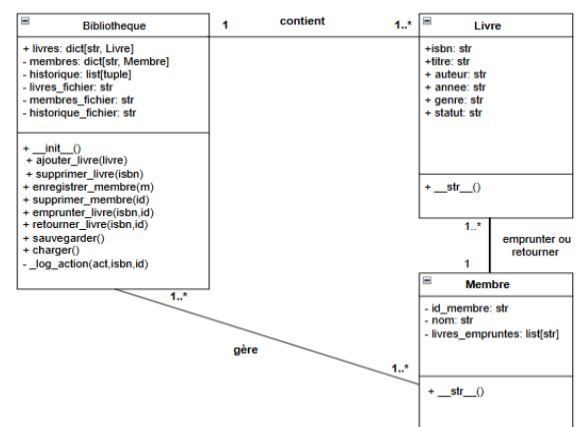
L'application est conçue selon une approche orientée objet, à partir des classes Livre, Membre et Bibliotheque. Elle utilise également des exceptions personnalisées pour gérer les erreurs possibles (par exemple : livre déjà emprunté, quota dépassé, membre inexistant, etc.). Toutes les données sont enregistrées dans des fichiers texte, CSV, ce qui permet de les conserver entre les différentes sessions d'utilisation.

Enfin, une interface graphique simple et intuitive a été développée avec Tkinter. Elle comprend des onglets organisés, des formulaires de saisie, des boutons d'action, ainsi que des messages d'alerte pour faciliter la prise en main de l'application.

1 .Diagrammes UML :

1.1 Diagramme de classe :

Cette illustration UML montre la structure de l'application avec les classes Bibliothèque, Livre et Membre, leurs attributs et méthodes. Elle met en évidence une composition forte entre la bibliothèque, ses livres et ses membres.



2 .Explications des algorithmes clés :

2.1 Class bibliothèque :

2.1.1 Définition des classes Livre et Membre et Initialisation de la classe Bibliotheque et gestion des fichiers :

Ces classes représentent les livres, les membres et la gestion globale de la bibliothèque avec stockage des données et historique des actions.

2.1. 2 Gestion des livres, membres et emprunts avec persistance :

Les méthodes permettent d'ajouter, supprimer et gérer les livres et membres avec des contrôles pour éviter les erreurs. Les emprunts et retours mettent à jour les statuts et l'historique des actions. Toutes les données sont sauvegardées et chargées depuis des fichiers pour assurer leur conservation entre les sessions.

```
def emprunter_livre(self, isbn, id_membre):
    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 = "emprunté"
    self.membres[id_membre].livres_empruntes.append(isbn)
    self._log_action(action="emprunt", isbn, id_membre)
    self.sauvegarder()
```

La fonction `emprunter_livre(self, isbn, id_membre)` permet à un membre d'emprunter un livre en vérifiant la disponibilité, l'existence et le quota d'emprunts. Elle met à jour le statut du livre, la liste des emprunts du membre, puis enregistre l'action et sauvegarde les données.

2.2 Exceptions:

Ce sont des exceptions personnalisées pour gérer les erreurs spécifiques : livre indisponible, quota d'emprunts dépassé, membre ou livre inexistant.

```
1 class LivreIndisponibleError(Exception):
2     pass
3
4 class QuotaEmpruntDepasseError(Exception):
5     pass
6
7 class MembreInexistantError(Exception):
8     pass
9
10 class LivreInexistantError(Exception):
11     pass
12
```

2.3 Visualisations:

Ces fonctions utilisent Matplotlib pour créer des graphiques : un camembert pour la répartition des genres, un histogramme des auteurs populaires, et une courbe montrant l'activité des emprunts sur 30 jours.

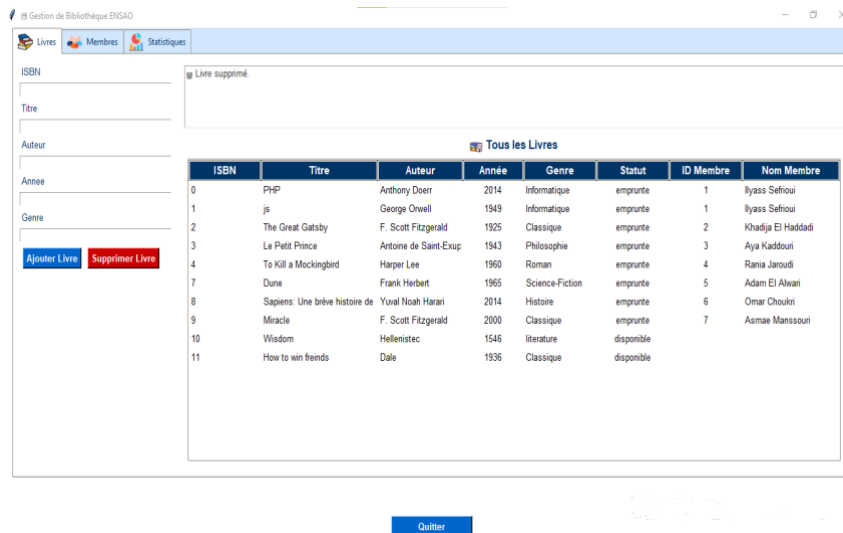
2.4 Main:

Ce code Python crée une interface graphique avec tkinter pour gérer une bibliothèque via trois onglets : gestion des livres, des membres, et affichage de statistiques avec matplotlib. Il charge des icônes personnalisées, met à jour dynamiquement les listes et tableaux, et sauvegarde les données à la fermeture.

3 . Illustrations des visualisations:

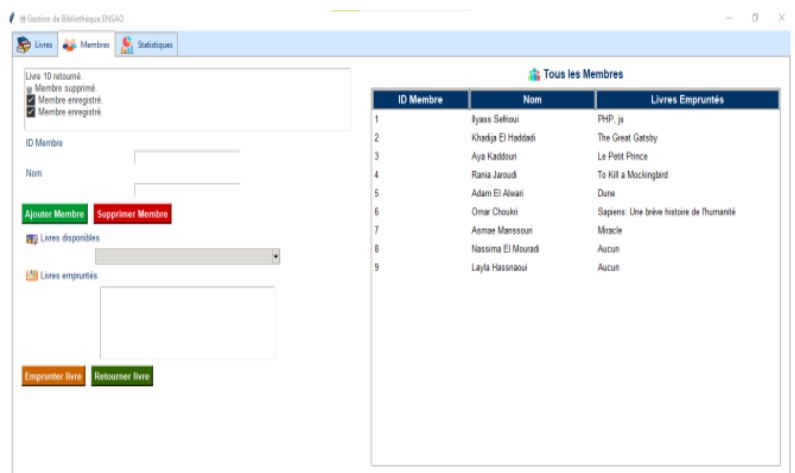
3.1 Interface Livres:

Cette interface sert à gérer les livres d'une bibliothèque. Elle affiche la liste des livres avec leurs détails (titre, auteur, genre, statut, etc.) et permet d'ajouter ou de supprimer un livre. Des messages s'affichent en haut pour indiquer les opérations effectuées, comme « Livre supprimé ». Un bouton « Quitter » est également disponible pour fermer l'application.



3.2 Interface Membres:

Cette interface permet de gérer les membres d'une bibliothèque : ajouter, supprimer, emprunter ou retourner un livre. Elle affiche les membres, leurs livres empruntés et des messages de confirmation des actions. Un bouton permet de quitter l'application.



3.3 Interface Statistiques:

Cette interface affiche les statistiques de la bibliothèque. En cliquant sur un bouton, un graphique correspondant s'affiche (comme montré dans les images suivantes) : par genre, top 10 auteurs ou activité des emprunts.

