Gestion de Bibliothèque en Scala “Documentation”

Description du Projet

> 💡 Ce projet consiste en une application de gestion de bibliothèque développée en Scala. L'application permet d'ajouter de nouveaux livres à la bibliothèque, d'emprunter des livres et de les rendre, et enregistrant les données dans le fichier **"bibliotheque.txt"**.

Classes Principales

1. `Livre`

- **Attributs** : `titre`, `auteur`, `anneeDePublication`, `estEmprunte` (détermine si le livre est actuellement emprunté ou non).
- **Méthodes** : `emprunter` (marque le livre comme emprunté), `rendre` (marque le livre comme non emprunté).

2. `Bibliothèque`

- **Attributs** : une liste de `Livre`.
- **Méthodes** :
  - `ajouterLivre` (ajoute un livre à la bibliothèque).
  - `emprunterLivre` (permet d'emprunter un livre en le marquant comme emprunté).
  - `rendreLivre` (permet de rendre un livre en le marquant comme non emprunté).
  - `rechercherParTitre` (recherche un livre par son titre).
  - `rechercherParAuteur` (recherche des livres par l'auteur).

3. `Main:`

- **L’utilisateur** :

- Peut `ajouter` un livre à la bibliothèque en fournissant les détails du livre (titre, auteur, année de publication).
- Peut `emprunter` et `rendre` des livres en fournissant le titre du livre ou le nom de l'auteur.
- Peut également `rechercher` des livres par titre ou par auteur.

Scénario : Gestion d'une bibliothèque

1. L'utilisateur démarre l'application.
2. L'application crée une instance de la classe `Bibliotheque` appelée `bibliotheque`.
3. L'application affiche le message de bienvenue : "Bienvenue dans la bibliothèque. Vous pouvez ajouter un livre ?"
4. L'application entre dans une boucle qui permet à l'utilisateur de choisir différentes opérations.
5. L'utilisateur a les options suivantes :
   - "1" : Rechercher un livre
   - "2" : Emprunter un livre
   - "3" : Rendre un livre
   - "4" : Ajouter un livre
   - "5" : Afficher tous les livres
   - "6" : Sauvegarder et quitter
6. Selon le choix de l'utilisateur, l'application effectue l'opération correspondante :
   - Si l'utilisateur choisit "1", l'application exécute `bibliotheque.rechercherLivre()`.
   - Si l'utilisateur choisit "2", l'application demande le titre du livre à emprunter et exécute `bibliotheque.emprunterLivre(titreChoisi)`.
   - Si l'utilisateur choisit "3", l'application demande le titre du livre à rendre et exécute `bibliotheque.rendreLivre(titreChoisi)`.
   - Si l'utilisateur choisit "4", l'application demande les détails du livre à ajouter et exécute `bibliotheque.ajouterLivre(bibliotheque.ajouterDetailLivre())`.

- Si l'utilisateur choisit "5", l'application exécute `bibliotheque.afficherTousLesLivres()`.
- Si l'utilisateur choisit "6", l'application sauvegarde les données dans le fichier "bibliotheque.txt" avec `bibliotheque.sauvegarderDansFichier("bibliotheque.txt")`, affiche un message de confirmation et quitte le programme avec `System.exit(0)`.

7. Si l'utilisateur entre un choix qui ne correspond à aucun des cas (par exemple, une chaîne non numérique), l'application affiche **"Choix invalide.".**
8. Le programme continue d'exécuter les étapes 4 à 7 jusqu'à ce que l'utilisateur choisisse l'option "6" pour sauvegarder et quitter.
9. Une fois que l'utilisateur choisit l'option "6", le programme se termine, enregistrant les données dans le fichier **"bibliotheque.txt"**.

Conclusion

Ce projet fournit une application fonctionnelle pour gérer une bibliothèque en **Scala**. Il répond aux exigences spécifiées en termes de fonctionnalités, de gestion des erreurs et de structuration du code.