

Introduction :

Ce rapport détaille une application Scala de gestion de bibliothèque. L'application est composée de plusieurs classes qui permettent de gérer les opérations courantes liées aux livres, notamment l'ajout, l'emprunt, le rendu, la recherche, la sauvegarde et le chargement de l'état de la bibliothèque.

Description du code :

Le projet est divisé en trois parties principales : les classes **Livre**, **Bibliotheque**, et un objet **Main**. Chaque classe joue un rôle spécifique dans le fonctionnement global de l'application.

Classe « Livre » :

La classe **Livre** représente un livre dans la bibliothèque. Chaque livre est caractérisé par un titre, un auteur, une année de publication, et son état d'emprunt. Elle contient les informations suivantes :

- **titre (String)** : Le titre du livre.
- **auteur (String)** : L'auteur du livre.
- **anneeDePublication (Int)** : L'année de publication du livre.
- **estEmprunte (Boolean)** : Indique si le livre est actuellement emprunté ou non.

La classe **Livre** propose les méthodes suivantes :

- **emprunter()** : Cette méthode permet d'emprunter le livre en mettant à jour son état. Si le livre est déjà emprunté, un message d'erreur est affiché.
- **rendre()** : Cette méthode permet de rendre le livre en mettant à jour son état. Si le livre n'est pas emprunté, un message d'erreur est affiché.
- **toString** : Cette méthode génère une représentation textuelle du livre, affichant son titre, auteur, année de publication et son état d'emprunt (oui ou non).

Classe « Bibliotheque » :

La classe **Bibliotheque** représente la bibliothèque elle-même. Elle contient une liste de livres et propose des méthodes pour gérer ces livres. Les principales méthodes sont les suivantes :

- **ajouterLivre(livre: Livre)** : Cette méthode permet d'ajouter un livre à la bibliothèque. Elle retourne un message d'erreur en cas d'échec.
- **emprunterLivre(titre: String)** : Cette méthode permet d'emprunter un livre de la bibliothèque en spécifiant son titre. Elle retourne un message d'erreur en cas d'échec.
- **rendreLivre(titre: String)** : Cette méthode permet de rendre un livre emprunté en spécifiant son titre. Elle retourne un message d'erreur en cas d'échec.
- **rechercherParTitre(titre: String)** : Cette méthode permet de rechercher des livres par titre. Elle retourne une liste de livres correspondant au titre ou `None` si aucun livre n'est trouvé.
- **rechercherParAuteur(auteur: String)** : Cette méthode permet de rechercher des livres par auteur. Elle retourne une liste de livres de l'auteur spécifié ou `None` si aucun livre n'est trouvé.
- **sauvegarderDansFichier()** : Cette méthode permet de sauvegarder l'état actuel de la bibliothèque dans un fichier CSV.
- **chargerDepuisFichier()** : Cette méthode permet de charger l'état de la bibliothèque depuis un fichier CSV.

Main (point d'entrée) :

Le **Main** est le point d'entrée de l'application de gestion de bibliothèque. Elle permet à l'utilisateur d'interagir avec la bibliothèque en choisissant diverses actions depuis un menu. Les actions possibles sont les suivantes :

```
*****
Application de Gestion de Bibliothèque
*****
1. Ajouter un livre
2. Emprunter un livre
3. Rendre un livre
4. Rechercher un livre par titre
5. Rechercher un livre par auteur
6. Sauvegarder et quitter
Votre choix (1-6) :
```

1. Ajouter un livre :

Dans ce premier choix de l'application, l'utilisateur a la possibilité d'ajouter un livre à la bibliothèque. Il est invité à saisir le titre et l'auteur du livre, ainsi que son année de publication.

```
Votre choix (1-6) : 1
***** Ajouter un livre *****
Titre : Le père Goriot
Auteur : Balzac
Année de publication : 1835
Livre ajouté à la bibliothèque.
```

2. Emprunter un livre :

L'action "Emprunter un livre" permet à l'utilisateur d'emprunter un livre de la bibliothèque en saisissant son titre. Si le livre est disponible, il peut être emprunté.

```
***** Emprunter un livre *****  
Titre du livre : Le père Goriot  
Livre avec le titre 'Le père Goriot' a été emprunté.
```

Si le livre n'est pas disponible, le message suivant s'affiche :

```
***** Emprunter un livre *****  
Titre du livre : Candide  
Erreur : Le livre avec le titre 'Candide' est déjà emprunté.
```

Si le livre n'existe pas, un message d'erreur s'affiche :

```
***** Emprunter un livre *****  
Titre du livre : aaa  
Erreur : Le livre avec le titre 'aaa' n'existe pas dans la bibliothèque.
```

3. Rendre un livre :

Cette action permet à l'utilisateur de retourner un livre emprunté à la bibliothèque. L'utilisateur doit fournir le titre du livre qu'il souhaite rendre.

```
***** Rendre un livre *****  
Titre du livre : Candide  
Livre avec le titre 'Candide' a été rendu.
```

Si le livre n'est pas emprunté, un message d'erreur s'affiche :

```
***** Rendre un livre *****  
Titre du livre : Germinal  
Erreur : Le livre avec le titre 'Germinal' n'est pas emprunté.
```

4. Rechercher un livre par titre :

L'utilisateur peut rechercher des livres dans la bibliothèque en spécifiant un titre. Tous les livres correspondant au titre saisi seront affichés.

```
***** Rechercher par titre *****  
Titre : Le petit prince  
Résultats :  
Titre: Le petit prince, Auteur: Antoine de Saint-Exupéry, Année de publication: 1943, Emprunté: oui
```

Si le titre n'existe pas, le message suivant s'affiche :

```
***** Rechercher par titre *****  
Titre : aaa  
Aucun livre trouvé dans la bibliothèque avec le titre spécifié.
```

5. Rechercher un livre par auteur :

De la même manière que la recherche par titre, l'utilisateur peut rechercher des livres par auteur. Les livres de cet auteur spécifique seront listés.

```
***** Rechercher par auteur *****  
Auteur : Emile Zola  
Résultats :  
Titre: Au bonheur des dames, Auteur: Emile Zola, Année de publication: 1883, Emprunté: non  
Titre: Germinal, Auteur: Emile Zola, Année de publication: 1885, Emprunté: non
```

Si l'auteur n'existe pas, le message suivant s'affiche :

```
***** Rechercher par auteur *****  
Auteur : bbb  
Aucun livre trouvé dans la bibliothèque de cet auteur.
```

6. Sauvegarder et quitter :

L'utilisateur peut choisir de sauvegarder l'état actuel de la bibliothèque dans un fichier csv avant de quitter l'application.

```
6. Sauvegarder et quitter  
Votre choix (1-6) : 6  
***** Sauvegarde en cours... *****  
Au revoir !
```

Conclusion :

L'application Scala de gestion de bibliothèque fournit une interface simple pour gérer une collection de livres. Elle permet d'ajouter, d'emprunter, de rendre, de rechercher et de sauvegarder les livres. Elle est conçue pour être conviviale et pratique pour les utilisateurs souhaitant gérer une bibliothèque de manière efficace. Les fonctionnalités de l'application peuvent être étendues à l'avenir pour inclure davantage de fonctionnalités, telles que la gestion des membres emprunteurs, des catégories de livres, des évaluations, etc. L'application actuelle sert de base solide pour de futures améliorations. Le code source fourni est correctement commenté et structuré, ce qui facilite sa compréhension et sa maintenance. L'application Scala de gestion de bibliothèque offre un exemple d'utilisation de la programmation orientée objet pour résoudre un problème de gestion de données. Elle montre comment Scala peut être utilisé pour créer des applications fonctionnelles et interactives.