Documentation du Projet
Système de Gestion de Bibliothèque

THIAKOU

Janvier 2026

Table des matières

Présentation du Projet

Ce projet consiste en un système de gestion de bibliothèque robuste développé en Java. Il met en application des concepts avancés de structures de données et d'algorithmes (SDA) pour manipuler efficacement une collection de livres.

Fonctionnalités Implémentées

- **Gestion CRUD :** Ajout, affichage, mise à jour et suppression (avec décalage de tableau).
- **Algorithmes de Recherche :** Recherche linéaire et recherche binaire ($O(\log n)$).
- **Algorithmes de Tri :** Tri à bulles (Titre) et QuickSort (Année).
- **Structures de Données Personnalisées :**
  - **Pile (Stack) :** Pour le journal d'activités (LIFO).
  - **Liste Chaînée :** Pour l'historique des emprunteurs.

Architecture Technique

Le projet suit une structure modulaire pour éviter les dépendances circulaires et faciliter la maintenance :

1. `Book.java` : Le modèle de données.
2. `LibraryManager.java` : Le moteur algorithmique (méthodes statiques).
3. `ActivityStack.java` & `BorrowingHistory.java` : Structures de données.
4. `Main.java` : Interface utilisateur console.

Instructions de Compilation

Pour résoudre les erreurs de type *"cannot find symbol"*, utilisez la procédure de compilation globale suivante :

Listing 1 – Commandes de compilation

```
# Se placer    la racine du code source
cd src/main/java/

# Compiler tous les fichiers du package en une fois
javac learn/java/*.java

# Executer l'application
java learn.java.Main
```

Analyse Algorithmique

5.1 La Suppression dans un Tableau

Contrairement à une liste dynamique, la suppression dans un tableau statique nécessite un décalage manuel des éléments pour maintenir l'intégrité de l'indexation.

5.2 Complexité

— **Recherche Binaire :** $O(\log n)$, extrêmement rapide mais nécessite un tableau préalablement trié.
— **QuickSort :** $O(n \log n)$ en moyenne, utilisé pour le tri par année de publication.

Conclusion

Ce projet démontre une compréhension approfondie de la gestion de la mémoire et de l'optimisation algorithmique en Java. L'utilisation de structures de données faites "à la main" (Pile et Liste Chaînée) prouve la maîtrise des concepts fondamentaux derrière les collections Java standards.