Projet de Gestion de Bibliothèque en C++

Matteo TOUTAIN, Timothée RICHARD

January 19, 2025

Introduction

Ce document décrit les classes utilisées pour la gestion d'un réseau de bibliothèques, leurs relations et leurs principales fonctionnalités.

1 Diagramme UML

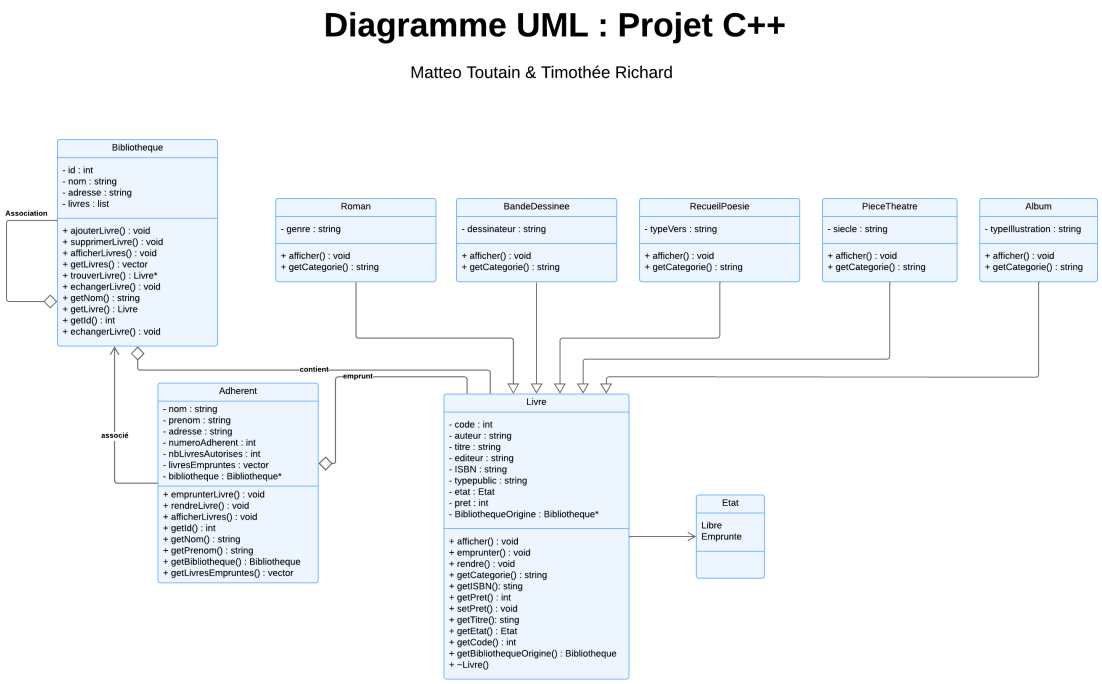

Figure 1: Diagramme UML des classes du projet

2 Classe `Livre`

La classe `Livre` est une classe de base représentant un livre générique. Elle inclut les attributs et méthodes suivants :

- **Attributs :**
  - `int code` : Identifiant unique du livre.

- `string auteur` : Auteur du livre.
- `string titre` : Titre du livre.
- `string editeur` : Éditeur du livre.
- `string ISBN` : Code ISBN du livre.
- `string typepublic` : Public cible du livre.
- `Bibliotheque* BibliothequeOrigine` : Pointeur vers la bibliothèque d'origine.
- `Etat etat` : État du livre (`Libre` ou `Emprunté`).
- `int pret` : Indique si le livre est prêté.

- **Méthodes :**
  - `void afficher()` : Affiche les détails du livre.
  - `void emprunter()` : Change l'état du livre à `Emprunté`.
  - `void rendre()` : Change l'état du livre à `Libre`.
  - `string getCategorie()` : Renvoie la catégorie du livre.

3 Sous-classes de `Livre`

3.1 `Album`

- **Attributs :**
  - `string typeIllustration` : Type d'illustration dans l'album.
- **Méthodes :**
  - `void afficher()` : Affiche les détails de l'album.
  - `string getCategorie()` : Retourne `Album`.

3.2 `BandeDessinee`

- **Attributs :**
  - `string dessinateur` : Nom du dessinateur.
- **Méthodes :**
  - `void afficher()` : Affiche les détails de la bande dessinée.
  - `string getCategorie()` : Retourne `BandeDessinee`.

3.3 `Roman`

- **Attributs :**
  - `string genre` : Genre du roman.
- **Méthodes :**
  - `void afficher()` : Affiche les détails du roman.
  - `string getCategorie()` : Retourne `Roman`.

3.4 `PieceTheatre`

- **Attributs :**
  - `string siecle` : Siècle de la pièce de théâtre.
- **Méthodes :**
  - `void afficher()` : Affiche les détails de la pièce de théâtre.
  - `string getCategorie()` : Retourne `PieceTheatre`.

3.5 `RecueilPoesie`

- **Attributs :**
  - `string typeVers` : Type de vers présents dans le recueil.
- **Méthodes :**
  - `void afficher()` : Affiche les détails du recueil de poésie.
  - `string getCategorie()` : Retourne `RecueilPoesie`.

4 Classe `Bibliotheque`

Cette classe représente une bibliothèque.

- **Attributs :**
  - `int id` : Identifiant unique de la bibliothèque.
  - `string nom` : Nom de la bibliothèque.
  - `string adresse` : Adresse de la bibliothèque.
  - `list<Livre*> livres` : Liste des livres.
- **Méthodes :**
  - `void ajouterLivre(Livre* livre)` : Ajoute un livre à la bibliothèque.
  - `void afficherLivres()` : Affiche les livres disponibles.
  - `void echangerLivre(Bibliotheque& autreBibliotheque, const string& ISBN)` : Échange un livre avec une autre bibliothèque.
  - `void supprimerLivre(int code)` : Supprime un livre à partir de son code.

5 Classe `Adherent`

Cette classe représente un adhérent.

- **Attributs :**
  - `string nom, prenom, adresse` : Informations personnelles.
  - `int numeroAdherent` : Identifiant de l'adhérent.
  - `int nbLivresAutorises` : Nombre maximal de livres autorisés.
  - `vector<Livre*> livresEmpruntes` : Livres empruntés.
- **Méthodes :**
  - `void emprunterLivre(Livre& livre)` : Permet à l'adhérent d'emprunter un livre.
  - `void rendreLivre(Livre& livre)` : Permet de rendre un livre.
  - `void afficherLivres()` : Affiche les livres empruntés par l'adhérent.

Fonctionnalités implémentées

- Afficher tous les livres ou par catégorie dans une bibliothèque. La bibliothèque peut afficher tous ses livres avec la méthode `afficherLivres()`. Elle peut également afficher les livres par catégorie avec un filtrage spécifique (par exemple, par type de livre).
- Un adhérent peut emprunter un livre en donnant son code. Cette fonctionnalité est gérée par la méthode `emprunterLivre()` de la classe `Adherent`. Un adhérent peut emprunter un livre en utilisant son code ISBN.
- Un adhérent ne doit pas pouvoir emprunter plus de livres qu'il n'en a le droit. Cela est contrôlé dans la méthode `emprunterLivre()`, où on vérifie si l'adhérent a atteint sa limite d'emprunts autorisés.
- Un adhérent ne doit pas pouvoir emprunter un livre non libre. L'état du livre est vérifié dans `emprunterLivre()`. Un livre doit être `Libre` pour pouvoir être emprunté.
- Une bibliothèque ne doit pas pouvoir prêter un livre non libre. Cela est vérifié à l'intérieur de la méthode `emprunterLivre()` de la bibliothèque, qui vérifie si le livre est `Libre`.
- Un adhérent peut rendre un livre. La méthode `rendreLivre()` de la classe `Adherent` permet à un adhérent de rendre un livre emprunté.
- Une bibliothèque peut demander un livre à une autre bibliothèque en donnant son ISBN. La méthode `echangerLivre()` de la classe `Bibliotheque` permet à une bibliothèque de demander un livre à une autre bibliothèque en donnant l'ISBN.
- Une bibliothèque peut acheter un nouveau livre. La méthode `ajouterLivre()` permet à une bibliothèque d'ajouter un nouveau livre.
- Une bibliothèque peut supprimer un livre, en donnant son code, soit parce qu'il est perdu, soit parce qu'il part au pilon. La méthode `supprimerLivre()` permet de supprimer un livre de la bibliothèque en fonction de son code.
- Une bibliothèque peut rendre les livres prêtés qui ne sont pas empruntés. Cela peut être implémenté par une méthode `rendreLivresPretes()` dans la classe `Bibliotheque` qui parcourt la liste des livres et rend ceux qui sont prêtés mais non empruntés.

Relations

- Une `Bibliotheque` contient plusieurs `Livre`s (relation de composition).
- Un `Livre` est associé à une `Bibliotheque` d'origine.
- Un `Adherent` peut emprunter plusieurs `Livre`s (relation d'agrégation).
- Les sous-classes de `Livre` héritent de la classe abstraite `Livre`.