

Exercice 1 – Modélisation UML

La municipalité souhaite une application de gestion pour sa bibliothèque municipale. Cette application doit permettre de :

- Gérer un catalogue de documents (livres et magazines),
- Enregistrer les membres,
- Gérer les emprunts de documents par les membres.

1. **Diagramme de classes** → Modélisez les classes suivantes :

- **Document** (classe abstraite)

- Attributs : id, titre, auteur, estEmprunte
- Méthodes :
 - afficherDetails() (abstraite)
 - estDisponible()
 - marquerCommeEmprunte()

- **Livre** hérite de **Document**

- Attribut : nombrePages
- Méthode : redéfinit afficherDetails()

- **Magazine** hérite de **Document**

- Attribut : numero
- Méthode : redéfinit afficherDetails()

- **Membre**

- Attributs : id, nom, historiqueEmprunts (liste de documents)
- Méthodes :
 - emprunter(Document d)
 - afficherDetails()

- **Bibliotheque**

- Attributs : nom, adresse, documents, membres
- Méthodes :
 - ajouterDocument(Document d)
 - ajouterMembre(Membre m)
 - afficherDetails()

- La classe **Bibliotheque** gère plusieurs documents
- La classe **Bibliotheque** gère plusieurs membres
- La classe **Membre** peut emprunter plusieurs documents : chaque membre conserve un historique de ses emprunts.
- Un **document** ne peut être emprunté que par un seul **membre** à la fois : l'attribut **estEmprunte** sert à vérifier sa disponibilité.

2. Diagramme de cas d'utilisation

Acteurs :

- **Bibliothécaire** : gère les documents et les membres.
- **Membre** : consulte et emprunte des documents.

Cas d'utilisation à représenter :

1. Ajouter un document
2. Ajouter un membre
3. Consulter le catalogue
4. Emprunter un document
5. Consulter l'historique des emprunts

3. Diagramme de séquence

Scénario : *Un membre emprunte un document via la bibliothèque.*

Objets :

- **:Utilisateur** (peut être le bibliothécaire ou l'interface)
 - **:Bibliotheque**
 - **:Membre**
 - **:Document**
1. **L'utilisateur** saisit l'identifiant du membre dans l'application.
 2. **AppBibliotheque** recherche le membre dans sa base de données.
 3. Si le membre est trouvé :
 - L'utilisateur sélectionne un document à emprunter.
 - **AppBibliotheque** vérifie si le document est disponible.
 - Si le document est disponible :
 - Le document est marqué comme emprunté.
 - Le document est ajouté à l'historique d'emprunts du membre.
 4. Si le membre n'est pas trouvé :
 - L'application informe que l'emprunt est impossible car le membre est inexistant ou non inscrit.

Exercice 2 – Implémentation Java

Développez une application en Java qui respecte la modélisation ci-dessus.

1. Implémentation des classes

Implémentez les classes suivantes en respectant :

- **Document** (classe abstraite)
- **Livre, Magazine**
- **Membre**
- **Bibliotheque**

2. Classe Main

Fonctionnalités à tester :

- Créer des livres et des magazines
- Ajouter des membres
- Enregistrer les documents dans la bibliothèque
- Simuler un ou plusieurs emprunts
- Afficher les détails de la bibliothèque et des membres