

ALOUANI Fadi

BTS SIO 1  
SLAM  
03/02/26

## **[Programmation] TD4 : le diagramme de classe**

### **Sommaire**

<a href="#"><u>Introduction</u></a> .....	1
<a href="#"><u>Introduction UML</u></a> .....	2
<a href="#"><u>Question :1</u></a> .....	2
<a href="#"><u>Question :2</u></a> .....	3
<a href="#"><u>Question :3</u></a> .....	4
<a href="#"><u>Question du TP</u></a> .....	5
<a href="#"><u>Question :1</u></a> .....	5
<a href="#"><u>Question :2</u></a> .....	5
<a href="#"><u>Question :3</u></a> .....	6
<a href="#"><u>Question :4</u></a> .....	6
<a href="#"><u>TD5</u></a> .....	7
<a href="#"><u>Conclusion</u></a> .....	7

### **Introduction**

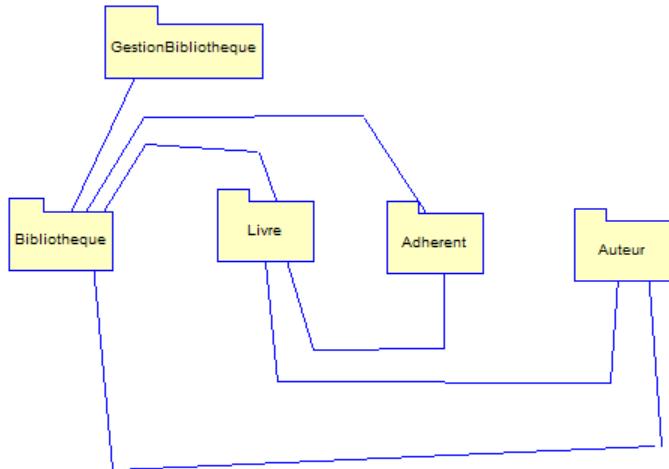
Ce travail pratique a pour objectif de mettre en application les concepts fondamentaux de la programmation orientée objet à travers la conception d'un diagramme de classes UML. Le cas étudié concerne la gestion d'une bibliothèque, permettant de modéliser les entités essentielles telles que les livres, les auteurs et les adhérents, ainsi que les relations qui les lient.

À l'aide de l'outil PowerAMC, ce TP vise à analyser les besoins fonctionnels, à structurer les données de manière cohérente et à générer automatiquement les classes Java correspondantes. Ce travail permet ainsi de faire le lien entre la modélisation conceptuelle et l'implémentation concrète en langage Java.

## Introduction UML

### Question :1

Faire le diagramme de package de votre projet



#### **Bibliothèque**

utilise :

- Livre
- Adhérent
- Auteur

Flèches :

- Bibliothèque → Livre
- Bibliothèque → Adhérent
- Bibliothèque → Auteur

#### **Livre**

utilise :

- Auteur

Flèche :

- Livre → Auteur

#### **Adhérent**

utilise :

- Livre

Flèche :

- Adhérent → Livre

#### **Auteur**

n'utilise personne

Donc aucune flèche qui part de Auteur

## Question :2

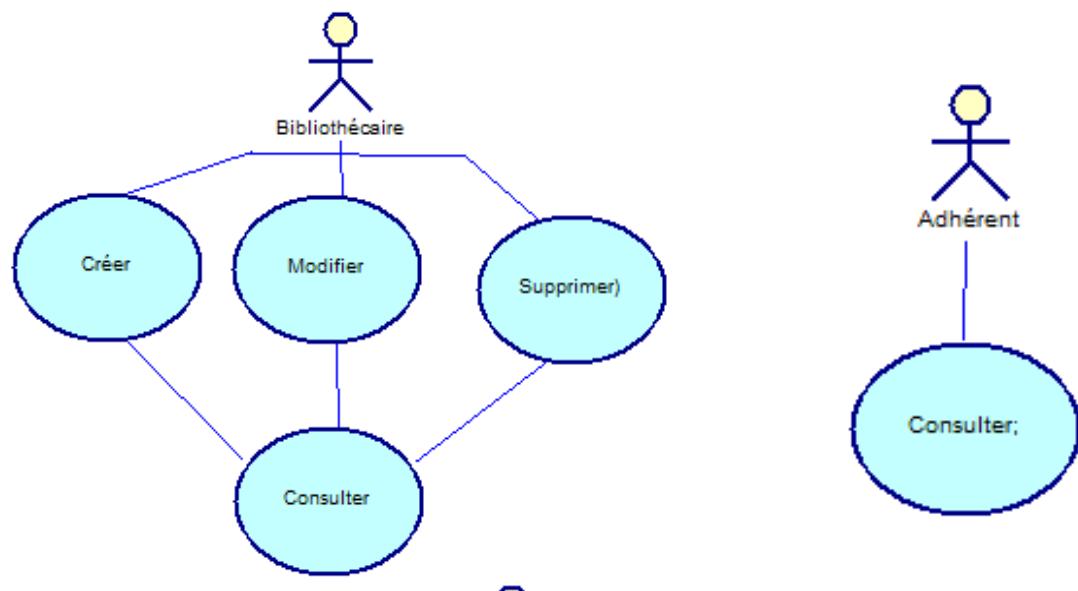
Faire le diagramme de cas d'utilisation pour la gestion des Compte rendu

**Acteurs :**

- Bibliothécaire
- Adhérent

**Use cases (cas d'utilisation) :**

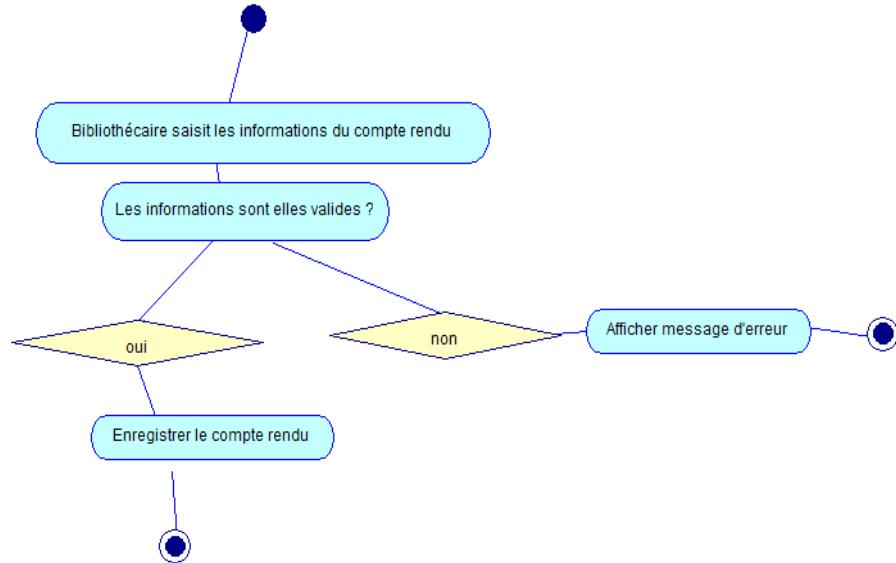
- Créer compte rendu
- Modifier compte rendu
- Supprimer compte rendu
- Consulter compte rendu



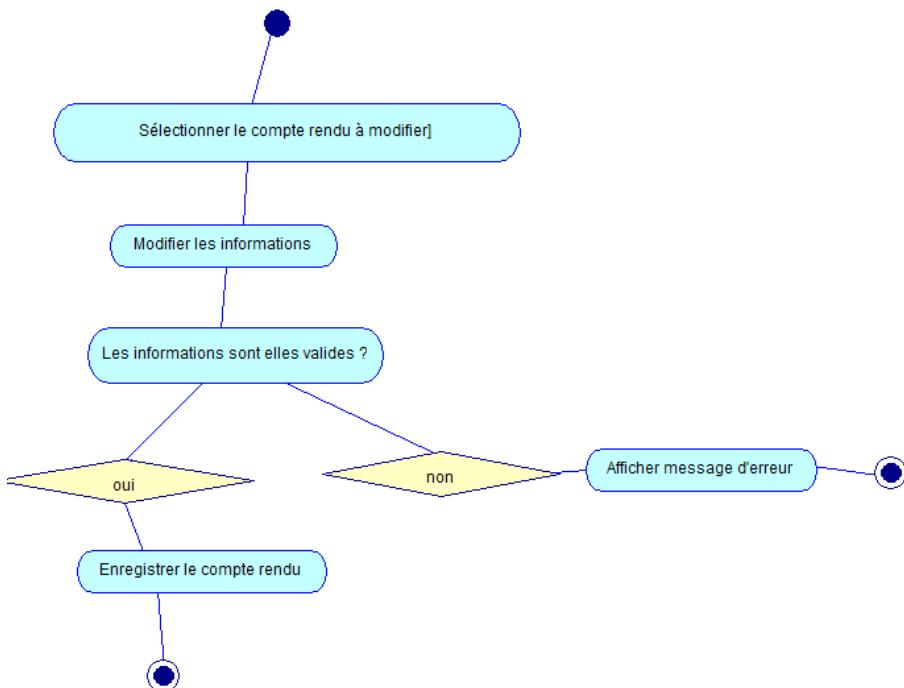
### **Question :3**

Faire les diagrammes d'activités :

#### **Insertion d'un compte rendu**



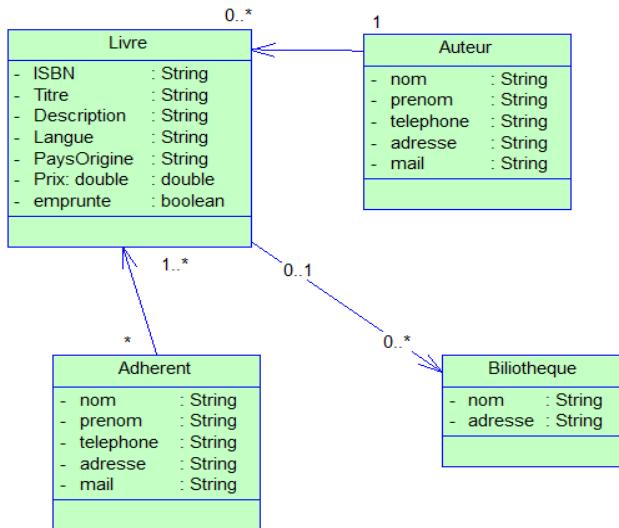
#### **Mise à jour d'un compte rendu**



## Question du TP

### Question :1

réaliser le diagramme de classe sur powerAMC



### Question :2

générer les classes

Restricted Mode is intended for safe code browsing. Trust this window to enable all features. [Manage](#) [Learn More](#)

J Adherent.java X J Auteur.java J Bibliotheque.java J Livre.java

C: > Users > eleve > Desktop > J Adherent.java > ...

```

1  ****
2  * Module: Adherent.java
3  * Author: eleve
4  * Purpose: Defines the Class Adherent
5  ****
6
7  import java.util.*;
8
9  /** @pdOid 0b0baf6c-be94-43f1-9261-c30d29e7f30a */
10 public class Adherent {
11     /** @pdOid 94cf05a9-8eb9-4692-ab30-180df237c7fc */
12     private String nom;
13     /** @pdOid f34ca4dd-3bd9-48ac-a74d-645f39392af5 */
14     private String prenom;
15     /** @pdOid 80068b5c-d8d2-4acb-819d-7be2ce25b60c */
16     private String telephone;
17     /** @pdOid fc66599e-02cc-4c7f-bdd1-68cb720c5518 */
18     private String adresse;
19     /** @pdOid c7da30c8-a91c-4bcf-86aa-d62581d58ca3 */
20     private String mail;
21
22     /** @pdRoleInfo migr=no name=Livre assc=association2 mult=1..*/
23     public Livre[] association2;
24
25 }
```

## Question :3

faire un programme principal permettant de créer ses objets (demander à votre professeur le fichier des données) :

```
J main.java > ↵ main > ↵ main(String[])
1  public class main {
2
3      Run | Debug
4      public static void main(String[] args) {
5
6          // ===== AUTEURS =====
7          Auteur a1 = new Auteur();
8          a1.setNom(nom: "Camus");
9          a1.setPrenom(prenom: "Albert");
10         a1.setDateNaissance(dateNaissance: "07/11/1913");
11         a1.setDescription(description: "Ecrivain, philosophe, romancier français");
12
13         Auteur a2 = new Auteur();
14         a2.setNom(nom: "Flaubert");
15         a2.setPrenom(prenom: "Gustave");
16         a2.setDateNaissance(dateNaissance: "12/12/1821");
17         a2.setDescription(description: "Ecrivain français");
18
19         Auteur a3 = new Auteur();
20         a3.setNom(nom: "Burd");
21         a3.setPrenom(prenom: "Barry");
22         a3.setDescription(description: "Professeur et auteur");
23
24         // ===== LIVRES =====
25         Livre l1 = new Livre();
26         l1.setISBN(isbn: "8573823224");
27         l1.setTitre(titre: "L'Étranger");
28         l1.setPrix(prix: 20.0);
29         l1.setDescription(description: "Roman de Albert Camus");
30         l1.setCategorie(categorie: "Roman");
31         l1.setAuteur(a1);
32
33         Livre l2 = new Livre();
34         l2.setISBN(isbn: "7495948500");
35         l2.setTitre(titre: "Madame Bovary");
36         l2.setPrix(prix: 19.5);
```

```
// ===== ADHERENTS =====
Adherent ad1 = new Adherent();
ad1.setNom(nom: "ABDALLA");
ad1.setPrenom(prenom: "Abdelrhman");
ad1.setTel(tel: "01.02.03.04.05");
ad1.setAdresse(adresse: "8 rue de Paris, 92 Courbevoie");
ad1.setMail(mail: "abdelrhman@gmail.com");

Adherent ad2 = new Adherent();
ad2.setNom(nom: "AGLAS");
ad2.setPrenom(prenom: "Lionel");
ad2.setTel(tel: "66.66.66.66");
ad2.setAdresse(adresse: "8 rue de Courbevoie, 75000 Paris");
ad2.setMail(mail: "lionel@gmail.com");

Adherent ad3 = new Adherent();
ad3.setNom(nom: "ALLAM");
ad3.setPrenom(prenom: "Ali");
ad3.setMail(mail: "Ali@gmail.com");

// ===== TESTS (question 4) =====
System.out.println(x: "== Liste des livres ==");
System.out.println(l1.getTitre() + " - " + l1.getAuteur().getNom());
System.out.println(l2.getTitre() + " - " + l2.getAuteur().getNom());
System.out.println(l3.getTitre() + " - " + l3.getAuteur().getNom());

System.out.println(x: "\n== Liste des adhérents ==");
System.out.println(ad1.getNom() + " " + ad1.getPrenom());
System.out.println(ad2.getNom() + " " + ad2.getPrenom());
System.out.println(ad3.getNom() + " " + ad3.getPrenom());
}
```

## Question :4

Test du programme

Faites emprunter le livre "Madame Bovary" à Lionel

Lister l'ensemble des adhérents de la bibliothèque

Lister les livres de la bibliothèque avec l'ensemble des informations (nom auteur, catégorie, état, nom de l'adhérent ayant emprunté le livre si il y a)

j'ai utiliser cette commander pour empreunter

```
/** @pdRoleInfo migr=no name=Biliotheque assc=associa
public Biliotheque[] association;

private String categorie;
private Auteur auteur;
private Adherent empruntePar;

public Livre() {
    this.emprunte = false;
    this.empruntePar = null;
}

public String getTisbn() {
```

on peut voir que cela fonctionner

```
--- Liste des adhérents de la bibliothèque ===
ABDALLA Abdelrhman - abdellhman@gmail.com
AGLAS Lionel - lionel@gmail.com
ALLAM Ali - Ali@gmail.com

--- Livres de la bibliothèque avec informations complètes ===
L'Étranger | Auteur: Camus | Catégorie: Roman | État: Disponible
Madame Bovary | Auteur: Flaubert | Catégorie: Roman | État: Emprunté par AGLAS Lionel
Java pour les nuls | Auteur: Burd | Catégorie: Education | État: Disponible
PS C:\Users\eleve\Desktop\tp_class> |
```

## **TD5**

- Modifier le diagramme de classe
- Générer les classes et observer le résultat

## **Conclusion**

À travers ce TP, nous avons pu concevoir un diagramme de classes représentant le fonctionnement d'une bibliothèque, en identifiant correctement les classes, leurs attributs et leurs relations. La génération automatique du code Java à partir du diagramme UML a permis de mieux comprendre l'intérêt des outils de modélisation dans le développement logiciel.

Ce travail met en évidence l'importance d'une bonne conception en amont pour faciliter l'implémentation, la maintenance et l'évolution d'une application. Il constitue ainsi une étape essentielle dans l'apprentissage de la programmation orientée objet et dans la maîtrise des méthodes de conception logicielle.