



---

# *Rapport TP POO*

Gestion d'une bibliothèque

En **JAVA**

---

Présenté par :

- LAIB Wissal
- LARABA Yamina Nesrine

Chargée du TP :

- F. MERROUCHE



## Présentation du projet :

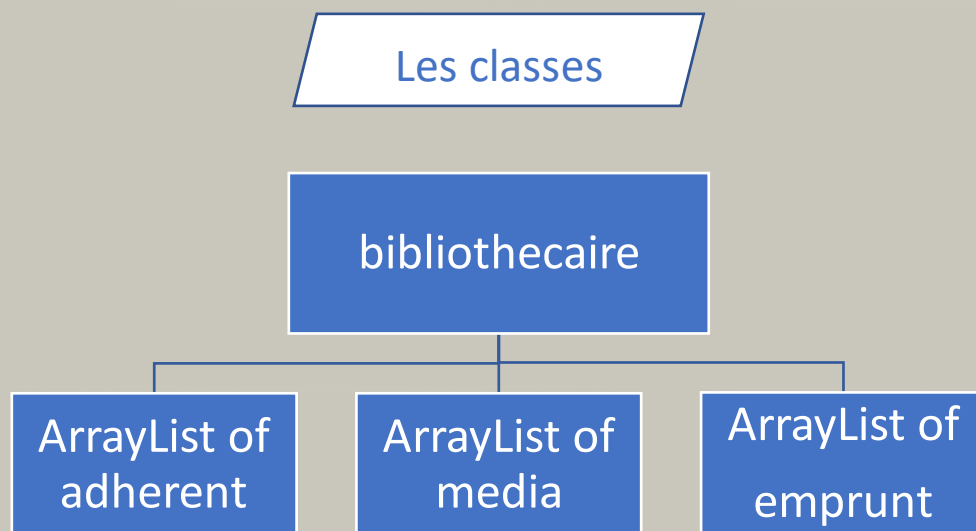
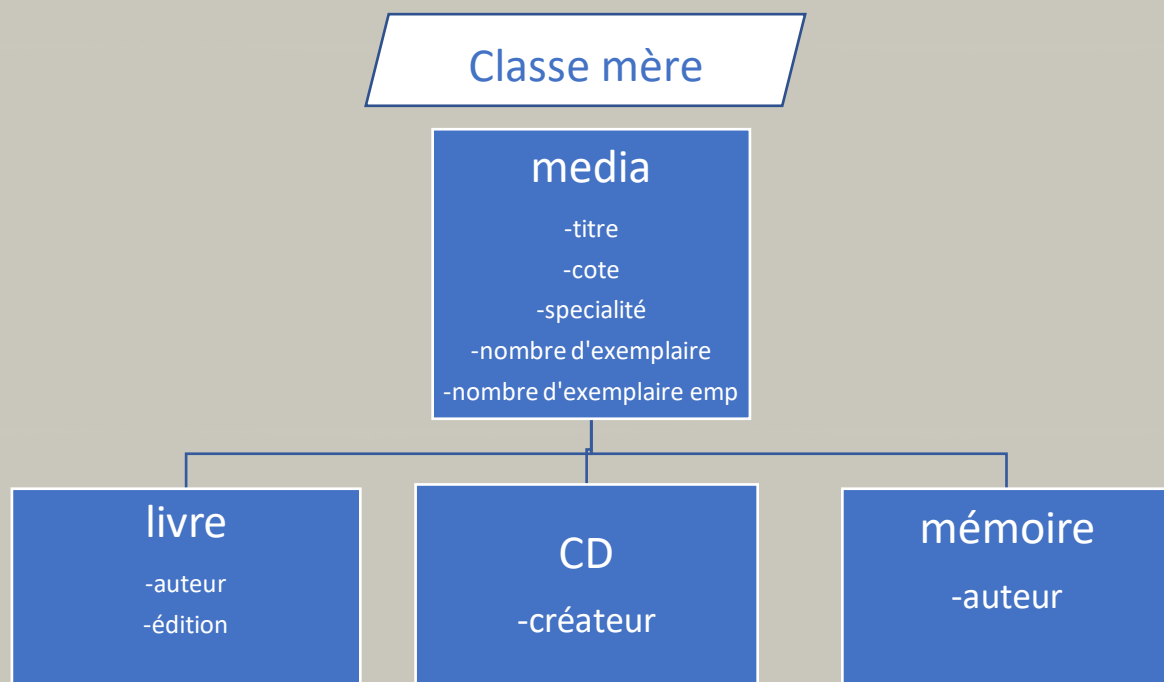
Comme son nom l'indique, le but de ce travail est de réaliser une multitude de tâches afin de gérer une bibliothèque.

Ce programme que nous allons vous présenter permet de :

- 1) Consulter la liste des media
- 2) Emprunter un media
- 3) Retourner un media
- 4) Modifier un media
- 5) Ajouter un media
- 6) Supprimer un media
- 7) Consulter la liste des adhérents
- 8) Modifier un adhérent
- 9) Ajouter un adhérent
- 10) Supprimer un adhérent
- 11) Consulter la liste des emprunts
- 12) Consulter la liste noire

Le schéma ci-dessous vous permet de mieux comprendre la gestion de cette bibliothèque.





## 1) Consulter la liste des media

Le but est très simple, pour afficher la liste des media

On a utilisé la méthode intitulée `afficher_media`, on parcourt ArrayList M avec une boucle et on teste le type de média avec `instanceof` ensuite on affiche les attributs

```
public void Afficher_Media() {  
    System.out.println("La liste des Media-s : ");
```

## 2) Emprunter un media :

Cela consiste à emprunter un media en utilisant `Add_Emprunt` qui

Fait un petit test pour vérifier l'existence du media à rajouter (en utilisant son titre et son type) dans ArrayList des media-s s'il existe déjà donc on décrémente son nombre d'exemplaire, sinon on affiche le message "`Ce media n'est pas disponible`"

```
public void Add_Emprunt(Emprunt E)
```



### 3) Retourner un media :

Le retour d'un media se fait par les deux méthodes `remove_emprunt` et `return_emprunt`.

On vérifie d'abord si un media existe dans la liste des emprunts, ce test se fait avec une boucle ou l'on compare le titre et le type, si le media qu'on veut retourner existe dans la liste, deux cas se présentent :

Soit l'étudiant a dépassé le délai de dix jours et dans ce cas on l'ajoute dans la liste noire, sinon on l'ajoute pas et dans les deux cas on incrémente le nombre d'exemplaires empruntés de la liste des media et le supprimer depuis la liste des emprunts

```
public void Remove_Emprunt(String Titre , int Type , int Matricule)
```

```
public void Retourner_Emprunt(String Titre , int Matricule , int Type)
```

### 4) Modifier un media :

On demande à l'utilisateur de donner le type et le titre de media à modifier, ensuite on lui demande de choisir le type de modification

(1= modifier la cote,2= modifier le nombre d'exemplaires) s'il choisit

de modifier la cote on appelle la méthode `modifier_cote_media`

sinon on appelle la méthode `modifier_nbr_exp_media`

```
public void Modifier_Media_Cote(int Type , String Titre , String NewCote)
```

```
public void Modifier_Media_Nbr_Exp(int Type , String Titre , int Newnbr)
```





## 5) Ajouter un media

On demande à l'utilisateur de choisir le type (1=livre ,2=mémoire , 3=CD )du media a rajouter et son titre, ensuite on vérifie l'existence du media a rajouter (en utilisant son titre et son type) dans ArrayList des media-s , si il existe déjà donc on incrémente son nombre d'exemplaire , sinon on le rajoute comme un nouveau media dans ArrayList media

```
while (i<this.M.size() && A==false) {  
    if(this.M.get(i) instanceof Livre && this.M.get(i).Titre.compareTo(M.Titre)==0) {  
        A=true;  
        this.M.get(i).Exp_Emp++;  
        System.out.println("Media ajoutée !");  
    }  
}
```

## 6) Supprimer un media :

Même chose pour la suppression, sauf qu'on décrémente le nombre d'exemplaire dans ce cas ou bien on le supprime carrément avec [remove\\_media](#)

```
public void Remove_Media(Media M)
```

## 7) Consulter la liste des adhérents :

Le but est très simple, pour afficher la liste des adhérents On a utilisé une procédure intitulée [afficher\\_adherent](#) , on parcourt ArrayList A et on affiche ses attributs

```
public void Afficher_Adherent() {  
    System.out.println("La liste des Adherents : ");  
    for(int i=0 ; i<this.A.size();i++) {  
        System.out.println("\nAdherent "+i+" : ");  
        System.out.println("Nom : "+this.A.get(i).Nom+" Prenom : "+this.A.get(i).Prenom+" Matricule : "+this.A.get(i).Matricule);  
    }  
}
```



## 8) Modifier un adhérent :

En utilisant [Modifier\\_Adherent](#) On demande à l'utilisateur de donner un matricule ensuite on vérifie si l'adhérent existe il entre son choix (1= nom, 2 =prénom) sinon on affiche le message "[Cet adhérent n'existe pas !](#)"

```
public void Modifier_Adherent(int Matricule , int Choix , String New) {
    if(this.Existe_Adherent(Matricule)==false) {
        System.out.println("Cet adherent n'existe pas !");
    }
    else {
        boolean E=false;
        int i=0;
        while(E==false && i<this.A.size()) {
            if(this.A.get(i).Matricule==Matricule) {
                if(Choix==1) {
                    this.A.get(i).Nom=New;
                }
                else {
                    this.A.get(i).Penom=New;
                }
                E=true;
            }
        }
    }
}
```

## 9) Ajouter un adhérent :

En effet un adhérent est caractérisé par plusieurs informations :

- Nom, prénom, matricule

Pour ajouter un adhérent, on demande à l'utilisateur d'entrer les caractéristiques de l'adhérent et vérifier d'abord s'il existe déjà avec [Existe\\_Media](#) , s'il existe on affiche « cet adhérent existe déjà » sinon on l'ajoute dans ArrayList adhérent du bibliothécaire avec [Add\\_Adherent](#)

```
public void Add_Adherent(Adherent A) {
    if(this.Existe_Adherent(A.Matricule)==false) {
        this.A.add(A);
        System.out.println("Adherent ajoutez !");
    }
    else {
        System.out.println("Cet adherent existe deja");
    }
}
```



## 10) Supprimer un adhérent :

Même principe pour la suppression on va utiliser la fonction [Remove\\_Adherent](#)

```
public void Remove_Adherent(Adherent A ) {
    if(this.Existe_Adherent(A.Matricule)==true) {
        this.A.remove(A);
    }
    else {
        System.out.println("Cet adherent n'existe pas !");
    }
}
```

## 11) Consulter la liste des emprunts

Le but est très simple, pour afficher la liste des emprunts  
On a utilisé une méthode intitulée [afficher\\_emprunt](#), on parcourt ArrayList E et on affiche ses attributs

```
public void Afficher_Emprunt() {
    String T=" ";
    System.out.println("La liste des emprunts : ");
    for(int i=0 ; i<this.E.size();i++) {
        System.out.println("Emprunt "+i+" :");
        if(this.E.get(i).Type==1) {
            T="Livre";
        }
        else {
            if(this.E.get(i).Type==2) {
                T="Memoire";
            }
            else {
                T="CD";
            }
        }
    }
}
```





## 12) Consulter la liste noire

Pour afficher la liste noire on a utilisé une procédure intitulée `afficher_ListeNoire`, on parcourt `arrayList L` et on affiche ses éléments

```
public void Afficher_ListeNoire() {  
    if(this.L.size()==0) {  
        System.out.println("La liste noire est vide");  
    }  
    else {  
        for(int i=0 ; i<this.L.size() ; i++) {  
            System.out.println("\nAdherent "+i+" : ");  
            System.out.println("Nom : "+this.A.get(i).Nom+" Prenom : "+this.A.get(i).Prenom+" Matricule : "+this.A.get(i).Matricule);  
        }  
    }  
}
```

