

Rapport de Projet : Application de Gestion de Bibliothèque

AZANGUE LEONEL DELMAT 3GI ENSPY

20 février 2025

0.1 Introduction

Ce projet a pour but de développer une application de gestion de bibliothèque en Java, permettant aux utilisateurs d'emprunter et de rendre des livres sous certaines conditions. Il intègre des concepts de **programmation orientée objet** tels que l'héritage, les interfaces, les classes abstraites et l'encapsulation.

0.2 Analyse des Besoins

L'application doit répondre aux fonctionnalités suivantes :

- Gérer l'inscription et la suppression des utilisateurs.
- Permettre aux utilisateurs d'emprunter et de rendre des livres avec une limite de ****20 jours****.
- Un utilisateur ne peut emprunter qu'un seul livre à la fois.
- Le coût d'emprunt dépend du livre et contribue aux gains de la bibliothèque.
- Un administrateur supervise les emprunts et la gestion des utilisateurs.

0.3 Conception du Système

L'application repose sur plusieurs classes principales :

- **Livre** : Définit les propriétés et méthodes pour la gestion des livres.
- **Utilisateur** : Représente un utilisateur qui peut emprunter un livre.
- **Administrateur** : Hérite de **Utilisateur** et gère la bibliothèque.
- **Bibliothèque** : Centralise la gestion des livres et des utilisateurs.
- **Main** : Exécute les tests en mode statique ou dynamique.

0.4 Architecture

VOIR ANNEXE

0.5 Description des Classes

0.5.1 Classe Livre

Cette classe représente un livre pouvant être emprunté. Elle contient :

- titre, auteur, prixEmprunt, estEmprunte.
- Des méthodes pour gérer l'emprunt et le retour.

Listing 1 – Classe Livre

```
1 package Package_Bibliotheque;
2
3 public class Livre {
4     private String titre;
5     private String auteur;
6     private double prixEmprunt;
7     private boolean estEmprunte;
8
9     public Livre(String titre, String auteur, double
10         prixEmprunt) {
11         this.titre = titre;
12         this.auteur = auteur;
13         this.prixEmprunt = prixEmprunt;
14         this.estEmprunte = false;
15     }
16
17     public boolean estDisponible() { return !estEmprunte; }
18     public void emprunter() { this.estEmprunte = true; }
19     public void rendre() { this.estEmprunte = false; }
20     public double getPrixEmprunt() { return prixEmprunt; }
21 }
```

0.5.2 Classe Utilisateur

Cette classe représente un utilisateur pouvant emprunter un livre.

Listing 2 – Classe Utilisateur

```
1 package Package_Bibliotheque;
2 import java.time.LocalDate;
3
4 public class Utilisateur {
5     private String nom;
6     private Livre livreEmprunte;
7     private double totalDepense;
8     private LocalDate dateEmprunt;
9
10    public Utilisateur(String nom) {
11        this.nom = nom;
12        this.livreEmprunte = null;
13        this.totalDepense = 0;
14    }
15
16    public void emprunterLivre(Livre livre, Bibliotheque
17        biblio) {
18        if (livreEmprunte != null) {
19            System.out.println(this.nom + " doit rendre " +
20                this.livreEmprunte.getTitre());
21            return;
22        }
23        if (livre.estDisponible()) {
24            livre.emprunter();
25            this.livreEmprunte = livre;
26            this.totalDepense += livre.getPrixEmprunt();
27            this.dateEmprunt = LocalDate.now();
28            biblio.ajouterGain(livre.getPrixEmprunt());
29        }
30    }
31 }
```

0.5.3 Classe Administrateur

Cette classe représente un administrateur de la bibliothèque.

Listing 3 – Classe Utilisateur

```
1 package Package_Bibliotheque;
2
3 public class Administrateur extends Utilisateur {
4     /* Gère l'inscription des utilisateurs et l'
5        attribution des livres */
6 }
```

```

5      /*La classe Administrateur herite de celle
6          Utilisateur */
7
8      private Bibliotheque biblio; //Bibliotheque qu'il
          administre
9
10     public Administrateur(String nom,Bibliotheque biblio)
        {
11         super(nom);
12         this.biblio = biblio;
13     }
14     /*Gestion des utilisateurs par l'administrateur */
15     public void ajouterUtilisateur(Utilisateur
        utilisateur) {
16         this.biblio.inscrireUtilisateur(utilisateur);
17         System.out.println("'" + utilisateur.getNom() + '
            "' + " a t inscrit avec succ s");
18     }
19
20     public void supprimerUtilisateur(Utilisateur
        utilisateur) {
21         this.biblio.supprimerUtilisateur(utilisateur);
22         System.out.println("'" + utilisateur.getNom() + '
            "' + " a t supprim avec succ s");
23     }
24     public void afficherUtilisateurs(){
25         System.out.println("====Liste des
            Utilisateurs====");
26         int count = 0;
27         for (Utilisateur user : this.biblio.getUser()){
28             System.out.println("- "+user.getNom());
29             count ++;
30         }
31         if (count == 0){
32             System.out.println("      Aucun UTILISATEUR n'a
                t enregistr ");
33         }
34     }
35
36     /*Gestion des livres */
37     public void ajouterLivre(Livre livre){
38         this.biblio.ajouterLivre(livre);
39         System.out.println("Le livre "+livre.getTitre() +
            " a t ajout avec succ s");
40     }

```

```

41
42 public void afficherLivres(){
43     /*Methode permettant d'afficher la liste des
44         livres */
45     System.out.println("=====Liste des Livres
46         Emprunt s=====");
47     int count = 0;
48
49     for (Livre livre : this.biblio.getBook()){
50         System.out.println(count+1 +": "+livre);
51         count ++;
52     }
53     if (count == 0){
54         System.out.println("Aucun livre n'a t
55             enregistr ");
56     }
57 }
58
59 public void afficherLivresEmpruntesby(Utilisateur
60     user){
61     /*Methode permettant d'afficher le livre
62         emprunt par user */
63     System.out.println("=====Liste des Livres
64         Emprunt s par "+user.getNom()+"====="
65         ");
66     if (user.getLivreEmprunte() == null){
67         System.out.println(user.getNom() +" n'a aucun
68             livre emprunt ");
69         return;
70     }
71
72     System.out.println(user.getNom() + " a emprunt
73         le livre "+user.getLivreEmprunte().getTitre()
74         +"de "+user.getLivreEmprunte().getAuteur() );
75 }
76
77 public void afficherLivresEmpruntes(){
78     System.out.println("=====Liste des Livres
79         Emprunt s=====");
80     int count = 0;
81     /*Methode permettant d'afficher les livres
82         emprunt s */
83     for (Utilisateur user : this.biblio.getUser()){
84
85         if (user.getLivreEmprunte() != null){
86             System.out.println("- "+user.

```

```

75         getLivreEmprunte());
76         count ++;
77     }
78     if (count == 0){
79         System.out.println("Aucun livre n'a t
80             emprunt ");
81     }
82
83
84     public void voirGains() {
85         /*Permet de voir le gain total des empruntes */
86         System.out.println("Gain total de la
87             biblioth que : " + this.biblio.getGainTotal()
88             + " FCFA");
89     }
90 }

```

0.6 Tests et Résultats

Pour chaque class , une classe Test est crée pour tester les différentes méthodes.

0.6.1 Explication de la classe Main

La classe Main.java est le point d'entrée du programme et propose deux modes :

- **Mode statique** : Exécute un scénario prédéfini.
- **Mode dynamique** : Permet à l'utilisateur d'interagir via un menu.

Exécution du Mode Statique

1. Inscription de trois utilisateurs (Alice, Bob, Oscar).
2. Ajout de quatre livres.
3. Suppression de l'utilisateur Alice.
4. Bob emprunte puis rend des livres.
5. Oscar emprunte un livre.
6. Affichage des gains de la bibliothèque.

Listing 4 – Exécution du Mode Statique

```
1 Hello, Club GI ENSPY!
2 Bienvenue dans notre biblioth que
3
4 =====Inscription des Utilisateurs=====
5 Alice a t inscrite.
6 Bob a t inscrite.
7 Oscar a t inscrite.
8
9 =====Enregistrement des Livres=====
10 Livre 'Java pour les nuls' enregistr .
11 Livre 'Python avanc ' enregistr .
12
13 =====Bob emprunte un livre=====
14 Bob a emprunt 'Java pour les nuls'.
15
16 =====Bob tente d'emprunter un autre livre=====
17 Bob doit rendre son livre avant d'en emprunter un autre.
18
19 =====Bob rend son livre=====
20 Bob a rendu 'Java pour les nuls'.
21
22 =====Liste des Livres Emprunt s=====
23 1. Oscar : 'Python avanc '
24
25 =====Gains=====
26 Gain total : 1200 FCFA
```

Exécution du Mode dynamique

VOIR ANNEXE

0.7 Conclusion et Perspectives

Ce projet a permis de mettre en application des principes avancés de programmation orientée objet en Java. Les prochaines améliorations pourraient inclure :

- Ajout d'une interface graphique.
- Gestion des pénalités pour retard.
- Utilisation d'une base de données pour stocker les informations.

0.8 Annexe

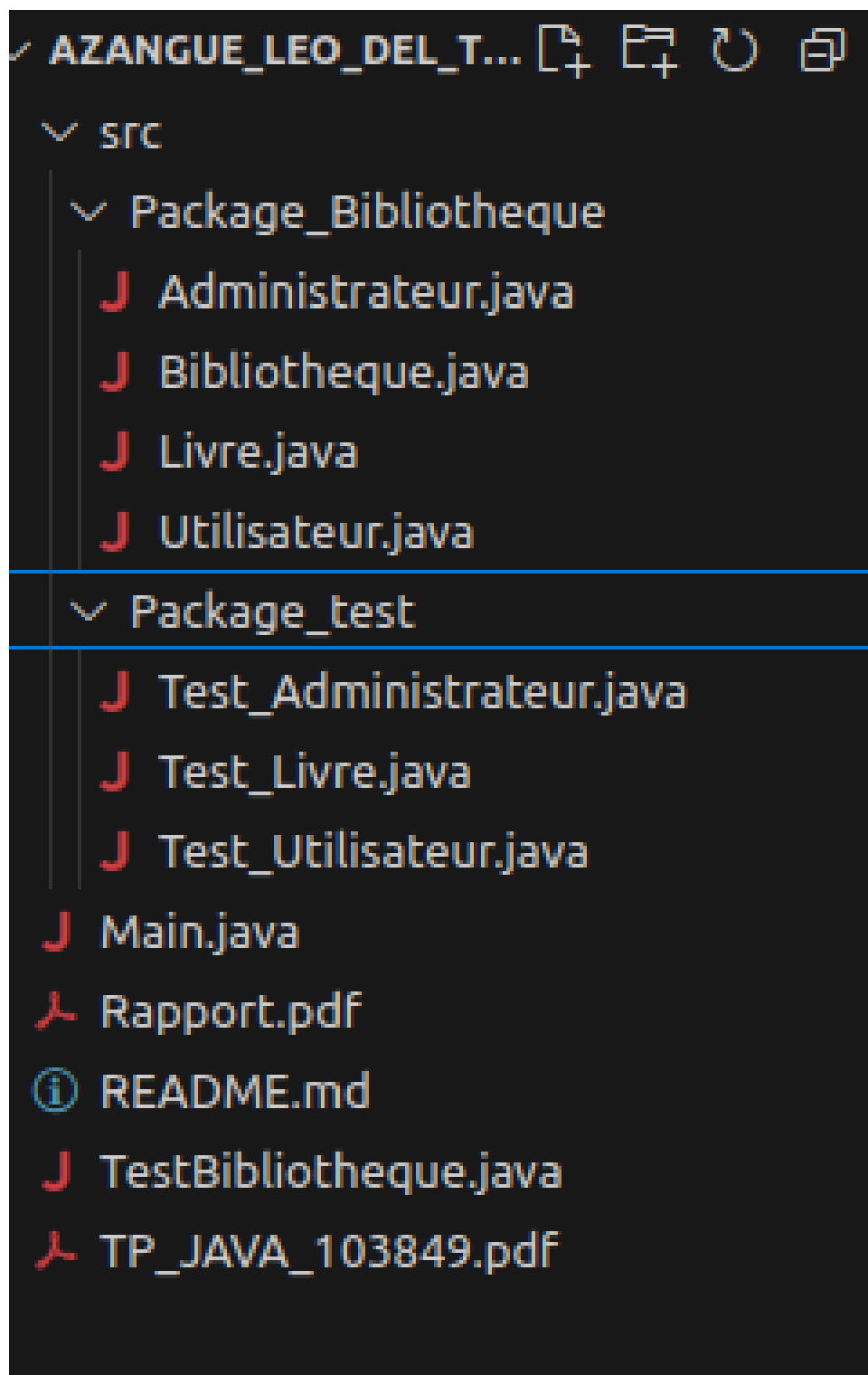


FIGURE 1 – extrait Mode dynamique

```
Pages: ep / home / default / config / code / user / notspaces / storage /
Hello, Club GI ENSPY!
Bienvenue dans notre bibliotheque

Voici les differentes fonctionnalités de notre bibliotheque
1.Mode statique
2.Mode dynamique

Entrez votre choix: 2
Vous avez choisi le mode dynamique

=====MENU=====
[1].Création de l'administrateur
[2].Creation d'utilisateurs
[3].Enregistrement des livres
[4].Emprunté un livre
[5].Rendre un livre
[6].Supprimer un utilisateur
[7].Liste utilisateurs
[8].Liste livres
[9].Liste livres emprunté
[10].Livre emprunté par un utilisateur
[11].Voir gain emprunt

Entrez votre option: █
```

FIGURE 2 – extrait Mode dynamique