

```
import java.util.ArrayList;
import java.util.HashMap;
import java.util.Scanner;

class Livre {
    //nous avons ici les proprietes du livre
    private String titre;
    private String auteur;
    private int anneePublication;
    private String ISBN;
    //nous avons ici Constructeur pour initialiser les propriétés
    public Livre(String titre, String auteur, int anneePublication, String
ISBN) {
        this.titre = titre;
        this.auteur = auteur;
        this.anneePublication = anneePublication;
        this.ISBN = ISBN;
    }
    // nous avons ici Méthodes getters et setters pour accéder et modifier les
propriétés on a aussi la Méthode toString pour afficher les informations du
livre
    public String getTitre() {
        return titre;
    }

    public void setTitre(String titre) {
        this.titre = titre;
    }

    public String getAuteur() {
        return auteur;
    }

    public void setAuteur(String auteur) {
        this.auteur = auteur;
    }

    public int getAnneePublication() {
        return anneePublication;
    }

    public void setAnneePublication(int anneePublication) {
        this.anneePublication = anneePublication;
    }

    public String getISBN() {
        return ISBN;
    }
}
```

```

    public void setISBN(String ISBN) {
        this.ISBN = ISBN;
    }

    @Override
    public String toString() {
        return "Livre{" +
            "titre='" + titre + '\'' +
            ", auteur='" + auteur + '\'' +
            ", anneePublication=" + anneePublication +
            ", ISBN='" + ISBN + '\'' +
            '}';
    }
}

class Utilisateur {
    // nous avons ici les Propriétés d'un utilisateur
    private String nom;
    private int numeroIdentification;
    private ArrayList<Livre> livresEmpruntes;

    // ici c'est pour Limite le nombre d'emprunts à 2
    private static final int MAX_EMPRUNTS = 2;
    // nous avons ici le Constructeur pour initialiser les propriétés de
    // l'utilisateur
    public Utilisateur(String nom, int numeroIdentification) {
        this.nom = nom;
        this.numeroIdentification = numeroIdentification;
        this.livresEmpruntes = new ArrayList<>();
    }

    public String getNom() {
        return nom;
    }

    // nous avons implementer ici la méthode pour emprunter un livre
    public boolean emprunterLivre(Livre livre) {
        if (livresEmpruntes.size() < MAX_EMPRUNTS) {
            livresEmpruntes.add(livre);
            System.out.println("Livre emprunté avec succès : " +
livre.getTitre());
            return true;
        } else {
            System.out.println("Limite d'emprunts atteinte pour cet
utilisateur.");
            return false;
        }
    }

    // nous avons implementer ici la méthode pour retourner un livre
    public void retournerLivre(Livre livre) {

```

```

        livresEmpruntes.remove(livre);
    }
// nous avons implementer ici la méthode pour afficher le livre emprunter
// par l'utilisateur
    public void afficherLivresEmpruntes() {
        if (livresEmpruntes.isEmpty()) {
            System.out.println("Aucun livre emprunté par cet utilisateur.");
        } else {
            System.out.println("Livres empruntés par " + nom + " :");
            for (Livre livre : livresEmpruntes) {
                System.out.println(livre);
            }
        }
    }
}

class Bibliotheque {
    // nous avons ici les Propriétés de la bibliothèque
    private ArrayList<Livre> listeLivres;
    private HashMap<Utilisateur, ArrayList<Livre>> empruntsUtilisateurs;
    //le Constructeur pour initialiser les propriétés de notre bibliotheque
    public Bibliotheque() {
        this.listeLivres = new ArrayList<>();
        this.empruntsUtilisateurs = new HashMap<>();
    }
// nous avons implementer ici la méthode pour ajouter un livre
    public void ajouterLivre(Livre livre) {
        listeLivres.add(livre);
    }
// nous avons implementer ici la méthode pour supprimer un livre
    public void supprimerLivre(Livre livre) {
        listeLivres.remove(livre);
    }
// nous avons implementer ici la méthode pour supprimer un livre aussi mais
// par le ISBN du livre
    public void supprimerLivreParISBN(String ISBN) {
        Livre livreASupprimer = null;
        for (Livre livre : listeLivres) {
            if (livre.getISBN().equals(ISBN)) {
                livreASupprimer = livre;
                break;
            }
        }
        if (livreASupprimer != null) {
            listeLivres.remove(livreASupprimer);
            System.out.println("Livre supprimé avec succès : " +
livreASupprimer);
        } else {
            System.out.println("Aucun livre trouvé avec l'ISBN : " + ISBN);
        }
    }
}

```

```

    }
}
// nous avons implementer ici la la possibiliter de rechercher un livre par
// son titre, auteur, ISBN
    public Livre rechercherLivreParTitre(String titre) {
        for (Livre livre : listeLivres) {
            if (livre.getTitre().equals(titre)) {
                return livre;
            }
        }
        return null;
    }

    public Livre rechercherLivreParAuteur(String auteur) {
        for (Livre livre : listeLivres) {
            if (livre.getAuteur().equals(auteur)) {
                return livre;
            }
        }
        return null;
    }

    public Livre rechercherLivreParISBN(String ISBN) {
        for (Livre livre : listeLivres) {
            if (livre.getISBN().equals(ISBN)) {
                return livre;
            }
        }
        return null;
    }
// nous avons implementer ici la méthode pour enregistrer un emprunt
    public void enregistrerEmprunt(Utilisateur utilisateur, Livre livre) {
        if (empruntsUtilisateurs.containsKey(utilisateur)) {
            empruntsUtilisateurs.get(utilisateur).add(livre);
        } else {
            ArrayList<Livre> livresEmpruntees = new ArrayList<>();
            livresEmpruntees.add(livre);
            empruntsUtilisateurs.put(utilisateur, livresEmpruntees);
        }
    }

    public void enregistrerRetour(Utilisateur utilisateur, Livre livre) {
        if (empruntsUtilisateurs.containsKey(utilisateur)) {
            empruntsUtilisateurs.get(utilisateur).remove(livre);
        }
    }
// Logique d'éligibilité à personnaliser selon les besoins
    public boolean verifierEligibilite(Utilisateur utilisateur) {
        return true;
    }

```

```

    }
    // Ajouter la logique pour ajouter un utilisateur à la bibliothèque
    public void ajouterUtilisateur(Utilisateur utilisateur) {

        if (!empruntsUtilisateurs.containsKey(utilisateur)) {
            empruntsUtilisateurs.put(utilisateur, new ArrayList<>());
            System.out.println("Utilisateur ajouté avec succès : " +
utilisateur.getNom());
        } else {
            System.out.println("Cet utilisateur existe déjà dans la
bibliothèque.");
        }
    }
}

public class Biblio {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        Bibliotheque bibliotheque = new Bibliotheque();
        boolean quitter = false;

        while (!quitter) {
            System.out.println("*           Menu Principal           *");
            System.out.println("1. Ajouter un livre");
            System.out.println("2. Rechercher un livre par titre");
            System.out.println("3. Rechercher un livre par auteur");
            System.out.println("4. Rechercher un livre par ISBN");
            System.out.println("5. Enregistrer un emprunt");
            System.out.println("6. Enregistrer un retour");
            System.out.println("7. Vérifier l'éligibilité d'un utilisateur");
            System.out.println("8. Afficher les livres empruntés par un
utilisateur");
            System.out.println("9. Supprimer un livre par ISBN");
            System.out.println("10. Ajouter un utilisateur");
            System.out.println("11. Quitter");

            System.out.print("Votre choix : ");
            int choix = scanner.nextInt();
            scanner.nextLine(); // Consommer la nouvelle ligne

            switch (choix) {
                case 1:
                    // Ajouter un livre
                    System.out.print("Titre : ");
                    String titre = scanner.nextLine();
                    System.out.print("Auteur : ");
                    String auteur = scanner.nextLine();
                    System.out.print("Année de publication : ");
                    int anneePublication = scanner.nextInt();

```

```

        scanner.nextLine(); // Consommer la nouvelle ligne
        System.out.print("ISBN : ");
        String ISBN = scanner.nextLine();

        Livre nouveauLivre = new Livre(titre, auteur,
anneePublication, ISBN);
        bibliotheque.ajouterLivre(nouveauLivre);
        System.out.println("Livre ajouté avec succès : " +
nouveauLivre);

        break;

    case 2:
        // Recherche par titre
        System.out.print("Entrez le titre du livre à rechercher :
");

        String titreRecherche = scanner.nextLine();
        Livre livreParTitre =
bibliotheque.rechercherLivreParTitre(titreRecherche);
        if (livreParTitre != null) {
            System.out.println("Livre trouvé : " + livreParTitre);
        } else {
            System.out.println("Aucun livre trouvé avec le titre :
" + titreRecherche);
        }
        break;

    case 3:
        // Recherche par auteur
        System.out.print("Entrez l'auteur du livre à rechercher :
");

        String auteurRecherche = scanner.nextLine();
        Livre livreParAuteur =
bibliotheque.rechercherLivreParAuteur(auteurRecherche);
        if (livreParAuteur != null) {
            System.out.println("Livre trouvé : " +
livreParAuteur);
        } else {
            System.out.println("Aucun livre trouvé avec l'auteur :
" + auteurRecherche);
        }
        break;

    case 4:
        // Recherche par ISBN
        System.out.print("Entrez l'ISBN du livre à rechercher :
");

        String ISBNRecherche = scanner.nextLine();
        Livre livreParISBN =
bibliotheque.rechercherLivreParISBN(ISBNRecherche);

```

```

        if (livreParISBN != null) {
            System.out.println("Livre trouvé : " + livreParISBN);
        } else {
            System.out.println("Aucun livre trouvé avec l'ISBN : "
+ ISBNRecherche);
        }
        break;

    case 5:

        // Enregistrer un emprunt
        System.out.print("Nom de l'utilisateur : ");
        String nomUtilisateur = scanner.nextLine();
        System.out.print("Numéro d'identification de l'utilisateur
: ");

        int numeroIdentification = scanner.nextInt();
        scanner.nextLine(); // Consommer la nouvelle ligne
        Utilisateur utilisateur = new Utilisateur(nomUtilisateur,
numeroIdentification);

        System.out.print("ISBN du livre à emprunter : ");
        String ISBNEmprunt = scanner.nextLine();
        Livre livreEmprunt =
bibliotheque.rechercherLivreParISBN(ISBNEmprunt);

        if (livreEmprunt != null) {
            // Vérifier si l'utilisateur peut emprunter un nouveau
livre

            if (utilisateur.emprunterLivre(livreEmprunt)) {
                bibliotheque.enregistrerEmprunt(utilisateur,
livreEmprunt);

                System.out.println("Emprunt enregistré avec
succès.");
            } else {
                System.out.println("Limite d'emprunts atteinte
pour cet utilisateur.");
            }
        } else {
            System.out.println("Aucun livre trouvé avec l'ISBN : "
+ ISBNEmprunt);
        }
        break;

    case 6:

        // Enregistrer un retour
        System.out.print("Nom de l'utilisateur : ");
        String nomUtilisateurRetour = scanner.nextLine();

```

```

        System.out.print("Numéro d'identification de l'utilisateur
: ");

        int numeroIdentificationRetour = scanner.nextInt();
        scanner.nextLine(); // Consommer la nouvelle ligne
        Utilisateur utilisateurRetour = new
Utilisateur(nomUtilisateurRetour, numeroIdentificationRetour);

        System.out.print("ISBN du livre à retourner : ");
        String ISBNRetour = scanner.nextLine();
        Livre livreRetour =
bibliotheque.rechercherLivreParISBN(ISBNRetour);
        if (livreRetour != null) {
            bibliotheque.enregistrerRetour(utilisateurRetour,
livreRetour);

            System.out.println("Retour enregistré avec succès.");
        } else {
            System.out.println("Aucun livre trouvé avec l'ISBN : "
+ ISBNRetour);
        }
        break;

    case 7:
        // Vérifier l'éligibilité d'un utilisateur
        System.out.print("Nom de l'utilisateur : ");
        String nomUtilisateurEligibilite = scanner.nextLine();
        System.out.print("Numéro d'identification de l'utilisateur
: ");

        int numeroIdentificationEligibilite = scanner.nextInt();
        scanner.nextLine(); // Consommer la nouvelle ligne
        Utilisateur utilisateurEligibilite = new
Utilisateur(nomUtilisateurEligibilite, numeroIdentificationEligibilite);

        boolean estEligible =
bibliotheque.verifierEligibilite(utilisateurEligibilite);
        if (estEligible) {
            System.out.println("L'utilisateur " +
nomUtilisateurEligibilite + " est éligible pour emprunter un livre.");
        } else {
            System.out.println("L'utilisateur " +
nomUtilisateurEligibilite + " n'est pas éligible pour emprunter un livre.");
        }
        break;

    case 8:
        // Afficher les livres empruntés par un utilisateur
        System.out.print("Nom de l'utilisateur : ");
        String nomUtilisateurEmprunt = scanner.nextLine();
        System.out.print("Numéro d'identification de l'utilisateur
: ");

```



```

        int numeroIdentificationEmprunt = scanner.nextInt();
        scanner.nextLine(); // Consommer la nouvelle ligne
        Utilisateur utilisateurEmprunt = new
Utilisateur(nomUtilisateurEmprunt, numeroIdentificationEmprunt);

        System.out.println("Livres empruntés par l'utilisateur
:");

        utilisateurEmprunt.afficherLivresEmpruntes();
        break;

    case 9:
        // Supprimer un livre de la bibliothèque par ISBN
        System.out.print("Entrez l'ISBN du livre à supprimer : ");
        String ISBNASupprimer = scanner.nextLine();
        bibliotheque.supprimerLivreParISBN(ISBNASupprimer);
        break;

    case 10:
        // Ajouter un utilisateur
        System.out.print("Nom de l'utilisateur à ajouter : ");
        String nomNouvelUtilisateur = scanner.nextLine();
        System.out.print("Numéro d'identification de l'utilisateur
à ajouter : ");
        int numeroIdentificationNouvelUtilisateur =
scanner.nextInt();
        scanner.nextLine(); // Consommer la nouvelle ligne
        Utilisateur nouvelUtilisateur = new
Utilisateur(nomNouvelUtilisateur, numeroIdentificationNouvelUtilisateur);
        bibliotheque.ajouterUtilisateur(nouvelUtilisateur);
        System.out.println("Utilisateur ajouté avec succès : " +
nouvelUtilisateur);
        break;

    case 11:
        // Quitter
        quitter = true;
        break;

    default:
        System.out.println("Choix invalide. Veuillez choisir une
option valide.");
        break;
    }
}

// Fermeture du scanner
scanner.close();
}
}

```

```
PROBLEMS 46 OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
PS C:\Users\hp\Desktop\sokhna code> javac Biblio.java
PS C:\Users\hp\Desktop\sokhna code> java Biblio
*      Menu Principal      *
1. Ajouter un livre
2. Rechercher un livre par titre
3. Rechercher un livre par auteur
4. Rechercher un livre par ISBN
5. Enregistrer un emprunt
6. Enregistrer un retour
7. Vérifier l'éligibilité d'un utilisateur
8. Afficher les livres empruntés par un utilisateur
9. Supprimer un livre par ISBN
10. Ajouter un utilisateur
11. Quitter
Votre choix : 1
Titre : mariama
Auteur : mariamaba
Année de publication : 2024
ISBN : 123
Livres ajoutés avec succès : Livre{titre='mariama', auteur='mariamaba', anneePublication=2024, ISBN='123'}
*      Menu Principal      *
1. Ajouter un livre
2. Rechercher un livre par titre
3. Rechercher un livre par auteur
4. Rechercher un livre par ISBN
5. Enregistrer un emprunt
6. Enregistrer un retour
7. Vérifier l'éligibilité d'un utilisateur
8. Afficher les livres empruntés par un utilisateur
9. Supprimer un livre par ISBN
10. Ajouter un utilisateur
11. Quitter
```

```
Nom de l'utilisateur : maimouna
Numéro d'identification de l'utilisateur : 123
Livres empruntés par l'utilisateur :
Aucun livre emprunté par cet utilisateur.
*      Menu Principal      *
1. Ajouter un livre
2. Rechercher un livre par titre
3. Rechercher un livre par auteur
4. Rechercher un livre par ISBN
5. Enregistrer un emprunt
6. Enregistrer un retour
7. Vérifier l'éligibilité d'un utilisateur
8. Afficher les livres empruntés par un utilisateur
9. Supprimer un livre par ISBN
10. Ajouter un utilisateur
11. Quitter
Votre choix : 10
Nom de l'utilisateur à ajouter : fatou
Numéro d'identification de l'utilisateur à ajouter : 1234
Utilisateur ajouté avec succès : fatou
Utilisateur ajouté avec succès : Utilisateur@76fb509a
*      Menu Principal      *
1. Ajouter un livre
2. Rechercher un livre par titre
3. Rechercher un livre par auteur
4. Rechercher un livre par ISBN
5. Enregistrer un emprunt
6. Enregistrer un retour
7. Vérifier l'éligibilité d'un utilisateur
8. Afficher les livres empruntés par un utilisateur
9. Supprimer un livre par ISBN
10. Ajouter un utilisateur
```