# Gestion d’une bibliothèque

Ce projet porte à modéliser une bibliothèque sur le langage de programmation Java. Le programme est en mesure de gérer les actions relatives à l’activité d’une bibliothèque. Certains utilisateurs sont définis administrateurs afin de pouvoir effectuer des tâches qui leur sont propres telles que l’ajout et la suppression d’un livre ou encore l’ajout d’utilisateurs admin.

Le projet Library est composé des classes : Book, User, Library, Menu et de la classe Main. Dans ce rapport, les classes seront expliquées indépendamment les unes des autres. Chacune de leurs méthodes regroupées par thématique seront détaillées afin d’en comprendre leur fonctionnement.

Inventaire des classes

[Gestion d’une bibliothèque 1](#_Toc181222808)

[Book 1](#_Toc181222809)

[Attributs 1](#_Toc181222810)

[Méthodes 2](#_Toc181222811)

[User 2](#_Toc181222812)

[Attributs 2](#_Toc181222813)

[Méthodes 2](#_Toc181222814)

[Library 3](#_Toc181222815)

[Attributs 3](#_Toc181222816)

[Méthodes 3](#_Toc181222817)

[Menu 4](#_Toc181222818)

[Attributs 4](#_Toc181222819)

[Méthodes 4](#_Toc181222820)

[Main 5](#_Toc181222821)

[Pistes de progression 5](#_Toc181222822)

## Book

### Attributs

On objet Book a les paramètres suivants :

* *ISBN* : numéro unique permettant de distinguer les ouvrages
* *title* : titre
* *description* : décrit rapidement l’ouvrage
* *author* : auteur du livre
* *price* : prix de vente
* *nbCopies* : nombre d’exemplaires
* *nbAvailable* : nombre de d’exemplaires encore disponibles dans la bibliothèque

Tous les attributs d’un livre sont définis privés principalement pour pouvoir contraindre leur déclaration selon certains critères. Leur déclaration est impérativement faite par les Setter. Le prix et les nombres d’exemplaires doivent être des entiers positifs. Le titre et l’auteur doivent impérativement être renseignés.

La classe comporte l’attribut isbnCounter. Il permet d’incrémenter d’une unité le compteur lors de la création d’un livre. Cet attribut est propre à la classe, et non aux instances de celle-ci.

### Méthodes

La classe Book comporte seulement le constructeur, les Getters, les Setters et les méthodes de la classe Objet surchargées.

## User

### Attributs

La classe User possède les attributs suivants :

* isAdmin : détermine si l’utilisateur est admin ou non
* identifier : numéro unique permettant d’identifier l’utilisateur
* borrowedBooks : liste des livres empruntés par l’utilisateur

La principale raison pour laquelle les attributs d’un utilisateur sont définis privés est la protection des données. Certaines données peuvent être à caractère personnel.

User comporte l’attribut de classe idCounter qui permet de distinguer les utilisateurs entre eux. A chaque création d’une instance de la classe User, ce compteur est incrémenté d’une unité.

Un utilisateur est créé à l’aide du constructeur sans qu’aucun paramètre ne soit spécifié. Son identifiant est unique et généré automatiquement lorsque l’utilisateur est créé. Sa liste de livres empruntés est par défaut vide.

### Méthodes

#### Déterminer ce que l’utilisateur possède

La méthode canRentMoreBooks permet de déterminer si l’utilisateur a déjà loué trois livres, et donc s’il est en mesure d’en emprunter davantage.

hasAlreadyRented prend un paramètre un numéro ISBN et détermine si le livre est déjà loué par l’utilisateur.

#### Actions utilisateur

La méthode borrowBook permet à l’utilisateur d’emprunter un livre. Le numéro ISBN du livre corrspondant est alors ajouté à la liste borrowedBooks.

La méthode returnBook permet de retourner un livre emprunté à la bibliothèque. Son numéro ISBN est alors supprimé de la liste borrowedBooks.

## Library

### Attributs

Un objet Library possède les attributs suivants :

* books : liste de l’ensemble des livres de la bibliothèque. Le numéro ISBN d’un livre correspond à sa position dans la liste books.
* users : liste de l’ensemble des utilisateurs de la bibliothèque. L’identifiant d’un utilisateur correspond à sa position dans la liste users.

Ces deux attributs sont définis privés afin d’appliquer le principe d’encapsulation et assurer une bonne maintenabilité.

La classe Library possède l’attribut instance qui permet de définir le Singleton.

### Méthodes

#### Constructeur

La méthode getInstance permet de créer la bibliothèque ; une seule et unique bibliothèque peut être créée car la méthode vérifie la valeur de l’attribut de classe instance.

#### Connexion de l’utilisateur

La méthode login permet à l’utilisateur souhaitant se connecter à l’interface et accéder au menu de se connecter.

#### Vérification du contenu de listes

La méthode isAvailable détermine si un livre spécifié par son numéro ISBN est disponible ou non.

La méthode contains permet de vérifier que la bibliothèque dispose d’un livre spécifié par son ISBN.

La méthode isRegistered permet de déterminer si un utilisateur spécifié par son identifiant est déjà inscrit dans la bibliothèque.

#### Actions utilisateur

La méthode getAvailableBooks permet d’afficher en console la liste des livres disponibles dans la bibliothèque.

La méthode borrowBook permet d’enregistrer l’emprunt d’un livre spécifié par un numéro ISBN par l’utilisateur connecté spécifié par son identifiant. La méthode vérifie tout d’abord si l’utilisateur est bien inscrit à la bibliothèque. Elle vérifie ensuite que le livre soit toujours disponible. Enfin, elle vérifie s’il dispose de moins de trois livres empruntés et s’il n’a pas déjà emprunté ce livre afin de déterminer s’il est en mesure de l’emprunter. Ces conditions étant réunies, l’emprunt peut être réalisé.

La méthode returnBook permet d’enregistrer le retour d’un livre. Elle prend en paramètre l’ISBN d’un livre et le numéro d’identifiant de l’utilisateur qui s’est connecté.

#### Actions admin

La méthode getAllBooks permet d’afficher en console la liste des livres de la bibliothèque, y compris ceux dont aucun exemplaire n’est encore disponible.

La méthode addCopy permet d’ajouter un exemplaire d’un ouvrage déjà contenu dans la bibliothèque.

La méthode addBook permet aux gestionnaires de la bibliothèque d’ajouter des livres à la bibliothèque. Si la bibliothèque dispose déjà de l’ouvrage, le nombre d’exemplaires de celui-ci est incrémenté d’une unité à l’aide de la méthode addCopy. Si l’ouvrage n’est en revanche pas encore dans la bibliothèque, un nouvel élément est ajouté dans la liste books de l’instance de Library.

La méthode removeBook permet de supprimer un exemplaire de la bibliothèque. La méthode vérifie préalablement si l’ouvrage contient bien un exemplaire ou plus. Si tel est le cas, son nombre d’exemplaires et son nombre d’exemplaires disponibles sont décrémentés d’une unité.

La méthode addUser détermine tout d’abord si l’utilisateur passé en paramètre est déjà enregistré ou non. S’il ne l’est pas il est ajouté dans la liste users de Library. Cette méthode sera utilisée lors du chargement des utilisateurs contenus dans le fichier JSON répertoriant l’ensemble des utilisateurs mais aussi lorsqu’un administrateur veut ajouter un utilisateur.

#### Obtention d’un livre ou d’un utilisateur

La méthode getBookDetails prend un numéro ISBN et permet de renvoyer l’objet Book s’il est contenu dans la liste books.

La méthode printBook utilise la méthode getBookDetails et convertit alors l’objet Book en chaîne de caractères.

La méthode getUserById permet de renvoyer l’objet User dont le numéro d’indentifiant est spécifié en paramètre.

## Menu

### Attributs

La classe Menu possède les attributs suivants :

* library : Singleton de Library
* userID : numéro d’identifiant de l’utilisateur qui s’est connecté
* scanner : lecteur permettant de récupérer les saisies de l’utilisateur en console

### Méthodes

#### Affichage du menu

La méthode displayMenuOptions permet de lister la liste des choix possibles par l’utilisateur.

La méthode showMenu permet de d’afficher les choix possibles en appelant la méthode displayMenuOptions. Elle capture ensuite le choix saisi en console par ’utilisateur. Lorsque l’interface de console se ferme, la méthode indique à l’utilisateur que tel est le cas et la session initiée par l’utilisateur prend fin.

#### Vérification des commandes saisies par l’utilisateur

La méthode getValidatedChoice permet de s’assurer que l’utilisateur a entré un nombre entier.

La méthode handleUserChoice d’effectuer l’action choisie par l’utilisateur parmi celles listées dans le menu.

La méthode promptforISBN permet de demander à l’utilisateur le numéro ISBN concerné par l’action qu’il veut effectuer. Le message qui s’affiche pour cette saisie s’adapte selon l’action choisie par celui-ci.

#### Actions à effectuer

Les actions qui peuvent être effectuées par l’utilisateur sont les suivantes :

* handleBookDetails : obtenir les informations d’un livre
* handleBookBorrow : emprunter un livre
* handleBookReturn : retourner un livre emprunté
* handleBookAdd : ajouter un livre à la bibliothèque
* handleBookCreation : créer une instance de Book si le l’ouvrage n’est pas encore contenu dans la bibliothèque
* handleBookRemove : supprimer un exemplaire de la bibliothèque
* handleAdminAddition : ajouter un administrateur à la liste des utilisateurs

Les trois premières méthodes sont réalisables par tout utilisateur, tandis que les quatre suivantes sont réservées aux administrateurs.

Chaque méthode qui concerne un livre en particulier prend en paramètre son numéro ISBN.

#### Sauvegarde des données lors de modifications

Les méthodes saveBooks et saveUsers permettent de sauvergarder respectivement les livres et les utilisateurs dans leurs fichiers JSON respectifs. Elles font appel à la méthode saveToFile qui prend en paramètre l’emplacement du fichier ainsi que son nom, le format du type de données.

## Main

Dans la méthode principale, une instance de Library est d’abord créée. Ensuite les livres et les utilisateurs de la bibliothèque sont chargés dans leurs fichiers JSON. L’utilisateur est ensuite invité à se connecter. Sa session commence alors

Les méthodes loadBooks et loadUsers permettent de charger les données contenues dans leur fichier JSON respectif. Les instances sont respectivement ajoutées dans les liste books et users.

## Pistes de progression

* La gestion des utilisateurs peut être améliorée, notamment avec l’ajout d’informations telles que le nom d’utilisateur et un mot de passe permettant de se connecter à son compte d’utilisateur.
* La liste des livres empruntés par l’utilisateur aurait pu être gérée avec des objets Book.
* Les identifiants peuvent mettre en cause la suppression des utilisateurs puisque leur position dans la liste users est définie par leur numéro d’identifiant.
* Il aurait été souhaitable de réaliser davantage de tests unitaires afin de s’assurer de la fiabilité du programme.
* Un travail de clarification/simplification du code est également envisageable.