

# Compte rendu TP1 :

## Introduction

Le binôme étant constitué de débutants en programmation C++ orientée objet, pour commencer, un travail pratique de niveau débutant a été choisi. Ce premier travail pratique concerne donc la gestion d'une bibliothèque. La bibliothèque contient plusieurs livres qui peuvent être empruntés par des lecteurs. Evidemment, un livre déjà emprunté ne pourra pas être emprunté de nouveau s'il n'a pas déjà été rendu. Mais alors comment s'y prendre pour la réalisation de ce travail pratique ? Nous répondrons à cette question en parlant d'une part de l'organisation du travail et d'autre part du développement du code.

## I. Organisation du travail

Le binôme travail en même temps sur le même travail pratique. Pour une meilleure compilation du code, l'outil utilisé ici sera GitHub. Ainsi, chaque membre du groupe peut envoyer du code et le recevoir tout en réglant les conflits qu'il peut y avoir entre les différents bouts de codes générés. Pour ce travail pratique, les différentes questions à traiter ont été réparties entre les membres du groupe. Le code source du projet est disponible dans le dépôt git ayant été utilisé lors de la réalisation de celui-ci.

Ainsi, chacun sait ce qu'il doit faire et comment le partager avec l'autre. Dans la suite nous entrons donc plus en détail dans le code de programmation.

## II. Développement du code

Pour la réalisation de cette bibliothèque, les classes qui ont été créées sont les suivantes : Auteur, Date, Livre, Lecteur, Emprunt et Bibliothèque.

### 1. Classe Date :

La classe date a été reprise de celle étudiée en cours magistral et adaptée pour prendre en compte les années aussi et non plus uniquement les jours et les mois.

### 2. Classe Livre :

La classe Livre est constituée des variables suivantes :

- Un titre qui est considéré comme une chaîne de caractères ;
- Un auteur qui a pour type la classe auteur qu'on verra dans la suite ;
- Une langue qui est considérée comme une chaîne de caractères ;
- Une genre Une langue qui est considérée comme une chaîne de caractères ;
- Une date de publication qui a pour type la classe Date vu précédemment ;
- Un ISBN qui est considérée comme une chaîne de caractères ;
- Une liste des identifiants des précédents emprunteurs qui est prise ici comme un **vector** ;
- Un statut qui sera considéré comme une valeur booléenne. Le statut ici va permettre de savoir si le livre a été emprunté ou non dans la bibliothèque.

Pour pouvoir manipuler facilement cette classe, on lui ajoute :

- Des getters pour chacune de ses variables ;
- Deux setters à savoir un pour ajouter un identifiant d'un nouvel emprunteur à sa liste des identifiants et un autre pour changer la valeur de la variable booléenne lorsque le livre est disponible ou pas ;
- Des surcharges pour les opérateurs == et != pour pouvoir comparer des livres de la bibliothèque ;
- Une surcharge de l'opérateur << pour pouvoir afficher un livre sans difficultés et contrôler ses informations.

### 3. Classe Auteur :

La classe Auteur est constituée des variables suivantes :

- Un identifiant numérique qui est considéré comme une chaîne de caractères ;
- Un nom et un prénom qui sont considérés comme des chaînes de caractères ;
- Une date de naissance qui est du type de la classe Date.

Pour pouvoir manipuler facilement cette classe, on lui ajoute :

- Des getters pour chacune de ses variables ;
- Des surcharges pour les opérateurs == et != pour pouvoir comparer des auteurs de la bibliothèque ;
- Une surcharge de l'opérateur << pour pouvoir afficher un auteur sans difficultés et contrôler ses informations.

### 4. Classe Lecteur :

La classe Lecteur est constitué des variables suivantes :

- Un identifiant unique qui est considérée comme une chaîne de caractères ;
- Un nom et un prénom qui sont considérée comme des chaînes de caractères ;
- Une liste des ISBN des livres empruntés qui est prise ici comme un **vector**.

Pour pouvoir manipuler facilement cette classe, on lui ajoute :

- Des getters pour chacune de ses variables ;
- Un setter pour ajouter un ISBN d'un livre emprunté ;
- Des surcharges pour les opérateurs ==, !=, <, >, <= et >= pour pouvoir comparer des Lecteurs de la bibliothèque ;
- Une surcharge de l'opérateur << pour pouvoir afficher un auteur sans difficultés et contrôler ses informations.

### 5. Classe Emprunt :

La classe Emprunt est constituée des variables suivantes :

- Une date d'emprunt qui est du type de la classe Date ;
- Un ISBN d'un livre ;
- Un identifiant d'un lecteur.

Pour pouvoir manipuler facilement cette classe, on lui ajoute :

- Des getters pour chacune de ses variables ;
- Des setters pour modifier la date, l'identifiant ou l'ISBN ;
- Des surcharges pour les opérateurs == et != pour pouvoir comparer des Lecteurs de la bibliothèque ;
- Une surcharge de l'opérateur << pour pouvoir afficher un emprunt sans difficultés et contrôler ses informations.

## 6. Classe Bibliothèque :

La classe Bibliothèque est constituée des variables suivantes :

- Une liste des identifiants des auteurs qui est prise ici comme un **vector** ;
- Une liste des identifiants des livre qui est prise ici comme un **vector** ;
- Une liste des identifiants des lecteurs qui est prise ici comme un **vector** ;
- Une liste des emprunts qui est prise ici comme un **vector**.

Pour pouvoir manipuler facilement cette classe, on lui ajoute :

- Des getters pour chacune de ses variables ;
- Des setters pour ajouter des livres, auteurs, lecteurs et emprunts a la bibliothèque ;
- Des surcharges pour les opérateurs == et != pour pouvoir comparer des bibliothèques différentes ;
- Une surcharge de l'opérateur << pour pouvoir afficher toute la bibliothèque sans difficultés et contrôler ses informations.

Pour simuler une bibliothèque, on crée plusieurs auteurs, livres, emprunts et lecteurs dans le fichier main. Une bibliothèque est aussi déclarée. Pour gérer la bibliothèque on doit lui ajouter différentes fonctions et méthodes qui sont :

- ✓ Une méthode **emprunter** pour permettre à un lecteur d'emprunter un livre disponible ;
- ✓ Une méthode **restituer** pour permettre à un lecteur de restituer un livre qu'il a lui-même emprunté ;
- ✓ Une méthode **livre\_auteur** pour afficher tous les livres écrits par un auteur ;
- ✓ Une méthode **livre\_emprunt** pour afficher tous les livres empruntés et le pourcentage de livres empruntés dans la bibliothèque ;
- ✓ Une méthode **lecteur\_emprunt** pour afficher tous les livres empruntés par un lecteur ;
- ✓ Une méthode **arrange** pour ranger la liste des lecteurs en fonction du nombre de livres empruntés.

## Conclusion :

Arrivés au terme de ce rapport, il en ressort que nous pouvons dire que ce projet nous aura permis d'appréhender les notions vues en cours de C++ de manière pratique. Ce projet nous aura aussi permis de pouvoir utiliser le langage C++ afin de régler un problème notamment la gestion d'une bibliothèque ce qui nous a donc fait gagner en expertise de programmation vu que c'était notre premier vrai projet donc dorénavant nous savons comment réagir à un tel problème. Des problèmes de compilation que nous ne comprenions pas avant par exemple ne sont plus

vraiment de grands problèmes pour nous. Nous pouvons donc dire qu'il aura été un grand plus pour la suite de notre formation d'ingénieur.