

# Compte-rendu de TP

## TP 1

**ROTA-GRAZIOSI Othilie**

**WENDL Tobias**

**Groupe TD2-TP3**

### **Introduction :**

Le but de ce TP est de gérer une bibliothèque. Cette bibliothèque comporte des livres pouvant être empruntés par des lecteurs. Un livre ne peut être emprunté que par un seul lecteur à la fois et doit être rendu à la bibliothèque avant de pouvoir être emprunté de nouveau.

Nous avons choisi ce TP car il s'agit d'un TP de niveau débutant, permettant de mettre en pratique tous les principes de la programmation orienté objet vu en cours.

Ayant suivi le cursus GEIPI tous les deux, nous avons certes déjà suivi des cours de C++ et de python mais il nous semblait judicieux de reprendre les bases en traitant un TP de niveau débutant.

Il est à noter que nous avons traité ce TP après avoir traité le TP numéro 2. Les deux TP étant similaires nous avons pu réutiliser une partie du travail que nous avons effectué précédemment afin de nous concentrer sur l'écriture d'un code propre et la correction de bugs. Ceci nous a également permis d'affiner notre compréhension du fonctionnement du code dans sa globalité.

### **Répartition du travail :**

La création des classes (de manières basique et simple) a été faite en amont hors séance de TP. Ceci comprend la mise en place des constructeurs, des getters, des setters et des attributs. Les fonctionnalités avancées de chaque classe ont été faite par l'autre binôme en séance de TP.

Cela comprend l'ajout des pointeurs dans la classe Borrow notamment. Pour la suite du TP, nous avons travaillé en parallèle en corrigeant mutuellement nos bugs et en testant les fonctionnalités codées par l'autre binôme.

Nous avons utilisé le même fonctionnement que lors du TP 2.

### **Choix techniques retenues et organisation des classes :**

Concernant l'organisation des classes, nous avons suivi l'énoncé. Nous avons simplement ajouté la possibilité à classe Date de prendre en compte les années.

Concernant les surcharges de l'opérateur <<, nous avons choisi d'en faire une fonction friend sur les conseils de notre enseignant. Nous avons utilisé cette technique dans les 2 TPs.

Nous avons choisi d'utiliser des pointeurs plutôt que des références à certains moments car nous avons besoin de passer des objets d'autres classes en attribut de classe. Ceci n'est pas possible avec de simples références.

### Difficultés rencontrées :

Nous n'avons rencontré qu'un seul réel problème lors de ce TP (que nous n'avons pas réussi à résoudre malgré l'aide de notre enseignant). Lorsqu'un client est « pré-crée » dans le main et qu'il effectue une réservation ; lors du rendu du livre, le programme s'arrête brusquement.

En effet nous obtenons l'erreur suivante :

```
terminate called after throwing an instance of 'std::bad_alloc'
what():  std::bad_alloc
signal: aborted (core dumped)
```

Avec l'aide de notre enseignant, nous avons tenté d'utiliser la fonctionnalité de débogage pour trouver la source du problème, en vain. Nous pensons que cela est dû à une mauvaise utilisation des pointeurs dans notre programme. Malheureusement, nous n'avons pas eu le temps de trouver l'erreur afin de la corriger.

### Outils utilisés :

Nous avons utilisé GitHub pour faire du versionning. Afin de travailler en parallèle, nous avons utilisé l'outil repl.it

( Vous pourrez directement exécuter le code avec le lien suivant :

<https://replit.com/join/nashbeaenw-geryaw> )

Repl.it est un outil en ligne nous permettant de travailler simultanément sur le même projet sans avoir à utiliser GitHub en permanence. De plus, l'outil Git est directement intégré et facile d'utilisation.

Vous pourrez accéder au projet complet sur GitHub via le lien suivant :

<https://github.com/G3R74W/TP1.git>

Nous nous sommes également servi de forum comme StackOverflow pour trouver comment trier un vecteur de lecteur en fonction de leur nombre de livres empruntés :

```
std::sort(readers_list.begin(), readers_list.end(), [](Reader &a,
Reader &b) {
    return a.get_borrowed_books() < b.get_borrowed_books();
});
```

Nous avons dû trier un vecteur d'objet dans la fonction reader\_ranking, permettant de trier un vecteur d'objet.

### Conclusion :

Traiter un TP de niveau débutant nous a permis de consolider nos bases en programmation orientée objet et de mettre en place de bonnes pratiques de programmation. Avoir eu à faire face à une erreur majeure nous a permis d'utiliser l'outil de débogage. Cet outil est essentiel peu importe le

langage de programmation. Nous avons donc acquis un savoir-faire essentiel de la programmation orientée objet.