

Compte rendu C++ par KUATE Nelson & LATTARD Jean

TP1 : Bibliothèque

Ce TP porte sur la gestion d'une bibliothèque dans la cadre de l'approfondissement des notions de programmation C++ vu durant le semestre. Cette bibliothèque comporte des livres qui peuvent être empruntés par des lecteurs en respectant la règle suivante : « Un Livre peut être emprunté par un seul Lecteur à un instant donné et doit donc être rendu à la bibliothèque par ce Lecteur avant d'être à nouveau emprunté par le même lecteur ou par un autre lecteur ».

L'organisation adoptée pour mener à bien ce travail est celle proposée par l'enseignant dans le sujet de TP à savoir que chaque membre du binôme travaille sur une question de façon alterné. Ceci donna donc :

- KUATE (Question1, question3, question5, question7)
- LATTARD (Question2, question4, question6, question8)

Etant tout deux des débutants en C++, ce TP a été choisi tout d'abord pour son niveau de difficulté ensuite pour son niveau de compréhension. En effet, de par notre profession, le fonctionnement d'une bibliothèque ne nous est pas étranger.

I. Outils utilisés :

Après la "Description du problème", on voit tout de suite qu'il faut créer des classes on aura donc les classes : Livre, Lecteur, Emprunt et Date. On nous demande par la suite de créer une classe Auteur et cela s'ajoute naturellement une classe Bibliothèque suivit d'une classe test.

Les outils utilisés pour effectuer la programmation sont multiples :

- ✓ Editeurs de texte en l'occurrence Sublime test pour le binôme et quelques fois VS Code
- ✓ Compilateur à savoir le terminal WSL pour les deux membres du binôme qui permet la compilation, l'exécution mais aussi de se connecter à git.
- ✓ Un outil de versionning notamment GIT choisit ici par l'enseignant pour notre apprentissage.

II. Création des classes :

- Création de la classe date :

La création de la classe date fut simple puisque il n'a suffi que de récupérer l'exemple présenté durant le cours par l'enseignant et d'effectuer minime modification.

- Création de la classe Auteur :

Comme mentionnée dans le support de TP, la classe Auteur a été créé avec ses attributs que sont le nom et prénom de l'Auteur en string, un id en string pour une meilleure manipulation et une date de naissance utilisant ainsi de l'héritage. A cela s'ajoute les getters de ces différents attributs et la surcharge de l'opérateur afin de permettre l'affichage correct d'un auteur.

- Création de la classe Livre :

Ici, après déclaration des attributs et des getters, la particularité est la présence d'un vecteur (le nombre de manuels dans une bibliothèque n'étant pas fixe) comportant des livres ainsi que la surcharge de l'opérateur <<. A cela s'ajoute une variable booléen qui pointe à 1 ou à 0 selon la disponibilité ou non du livre.

- Création de la classe Lecteur :

La particularité de cette classe est la présence de la liste de livres que le Lecteur a eu à emprunter pour une meilleure traçabilité mais aussi à fin de de lui faire des propositions de livres à consulter selon les préférences de ce dernier. Ceci semble poser les bases de la récolte des données. Afin de réaliser ceci, nous avons essayé sans succès de maintenir une certaine sécurité associé dans le tableau associés à chaque Lecteur non pas les noms des manuels consultés mais les isbn de ceux-ci. Ayant beaucoup de difficultés (notamment le type des fonctions car cette solution est survenu en fin de projet) avec cette solution bien qu'étant la plus efficace, nous avons optes pour l'affichage de la liste de Livres empreintes en elle-même.

- Création de la classe Emprunt :

Un emprunt étant par définition accompagné d'information telle que le manuel emprunté, le lecteur et la date de l'emprunt, cette classe possède des attributs étant eux-mêmes des objets. Afin de maintenir un semblant d'anonymat et installé une première couche de sécurité, nous avons cette fois réussi à ne faire apparaitre que l'isbn du livre emprunté et l'identifiant du lecteur ayant emprunté le manuel. Ceci se faisant dès la création de l'objet dans le constructeur.

Etant donné que nous avons besoin de plusieurs classes héritant chacune des autres, nous avons utilisé la commande « Pragma Once » permettant à chaque inclusion de classe de se faire une seule fois.

- Création de la classe Bibliothèque :

La classe bibliothèque regroupe toutes les classes précédemment réalisées ; elle est un assemblage composée de Vecteur des autres classes. Elle possède une méthode permettant d'indiquer la disponibilité d'une classe.

- Création de la classe d'implémentation :

Celle a permis de nous rassurer de la validité des objets créés au fur et à mesure de l'avancement des travaux. Ainsi donc nous avons créés plusieurs Lecteurs, livres, dates, auteurs et même des emprunts afin d'avoir un exemple concret de l'avancé de nos travaux.

III. Problèmes rencontrés et apports :

La Principale difficulté a été l'implication de M. Jean LATTARD, qui n'a pas été assez actif et n'a pas assez communiqué sur ces difficultés. En effet, celui-ci ne sachant que faire la déclaration d'une classe, et de l'ayant signalé que très tard, mener à bien le projet jusqu'à son terme n'a pas été chose facile.

A cela s'ajoute, la prise en main et l'utilisation de git/github nécessaire au versionning du code. La surcharge des différents opérateurs et l'emploi de ceux-ci a été une tâche assez complexe car n'étant pas habitué aux subtilités du langage Orienté objets et du C++.

La difficulté lors de la création des classes est de bien savoir quel est le **type** de chaque attribut de la classe. Les strings sont utilisés pour du texte comme les noms des auteurs ou des lecteurs tandis que les int sont utilisés pour des entiers comme les ISBN ou l'identifiant numérique. Mais pour ce qui concerne une date comme la date de publication d'un livre ou la date d'emprunt, le type de l'attribut est une date liée à une classe date.

Concernant les apports, ceux-ci ont été multiples :

- ✓ L'apprentissage du versionning git qui est un outil dont la maîtrise est primordiale pour la gestion d'un projet de groupe et pour tout programmeur chevronné.
- ✓ L'emploi de la surcharge nous a permis de mieux percevoir et saisir la notion mais aussi des nuances d'un langage orienté objet telles que le C++.
- ✓ L'apprentissage de la gestion d'un projet et des réalités/difficultés d'un projet et du nécessaire pour mener à bien un projet.