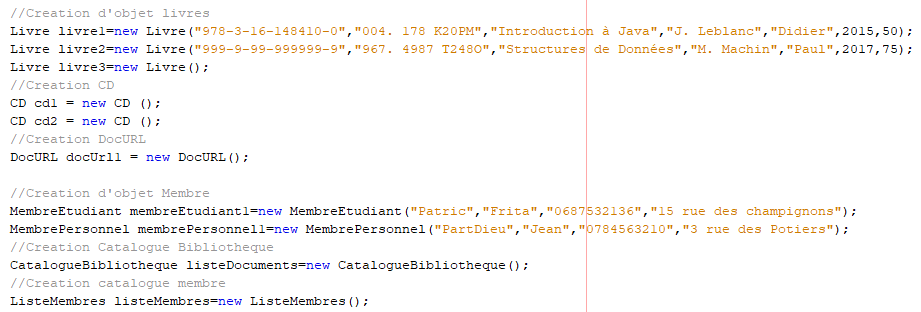
PS :

-Voici la création des objets des clases que j’utiliserais pendant le tp (démonstration)

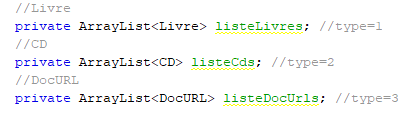


La bibliothèque contiendra 3 livres 2 cd et 1 document url.

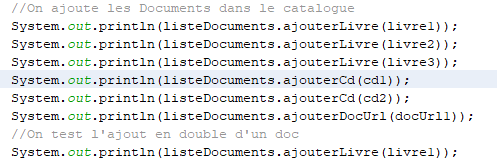
* True correspondra à = Opération Réussie
* False correspondra à = Opération non Réussie

1. Démonstration du catalogue pouvant stocker différents types de DocBibliotheque

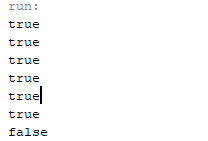
La classe Catalogue bibliothèque contient 3 listes de chaque type et permet donc de stocker différents types de documents



On test donc l’ajout des 6 documents et j’essaye d’ajouter un même document (pour vérifier qu’on ne peut pas avoir de doublons)

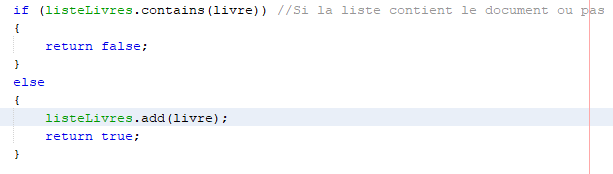


Vérifions les actions avec la console

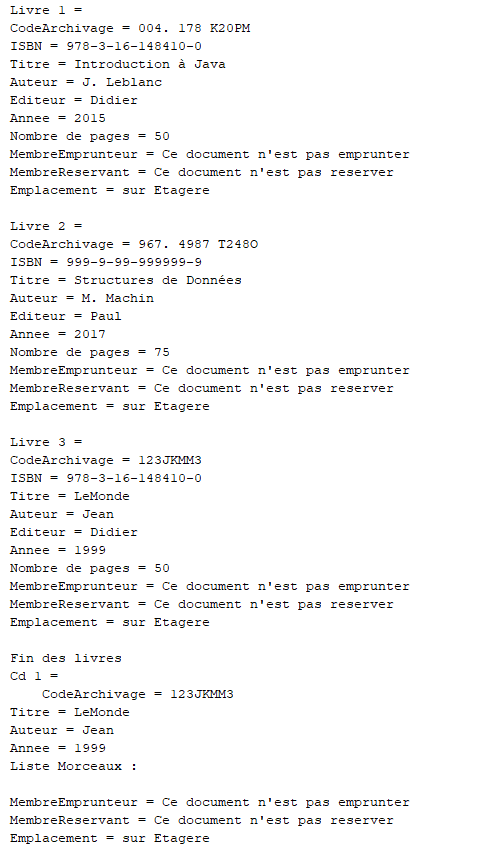


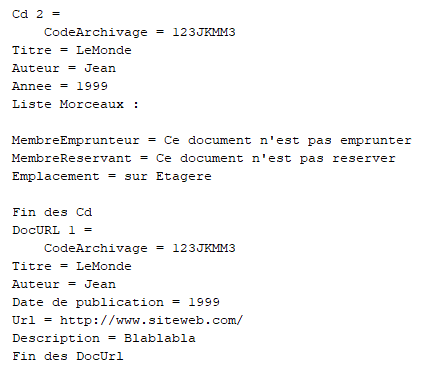
On remarque que toutes les actions ont fonctionner sauf la dernière car dans la fonction ajouter, il regarde s’il n’est pas déjà dans le catalogue :

Exemple code pour le livre :

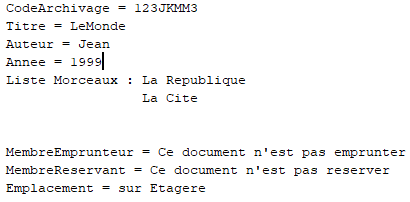


Suite à ces opérations, on peut les retrouver dans le catalogue en affichant celui-ci :





On remarque que les  « listes des Morceaux » des Cds sont vides : car je n’ai pas ajouté de morceaux à eux. Rajoutons des morceaux au Cd1 puisque la classe CD contient aussi une liste :



Et Voila :

1. Démonstration qu’un document Url ne peut être emprunté



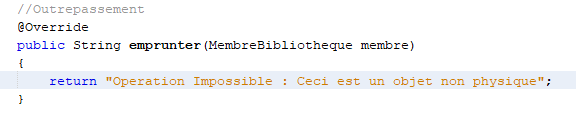


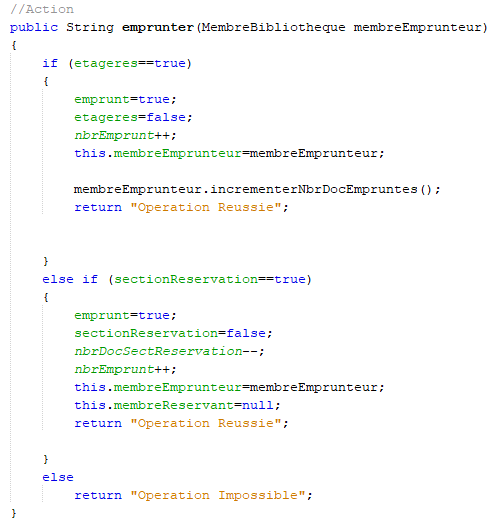
La console me renvoi « Opération impossible » car un document url n’est pas un objet physique.

Comme on ne peut pas réserver un document QUI n’est pas emprunté, toutes les actions (reserver, retourner…) sont impossible.

Cette action se réalise automatiquement grâce à l’overriding dans la classe docUrl :

Il va donc réaliser cette fonction :



A la place de l’autre (qui est situé dans la classe mère DocBibliotheque)  


1. Démonstration qu’un membre ne peut emprunter plus de 4 documents

Essayons d’emprunter des documents avec un membre Etudiant ( donc limité à 4 emprunts)

On envoit donc 3 paramètre : listeDocuments.emprunteDoc(ID,membre,TYPE)

Id= position de l’objet dans la liste (comme à 0)

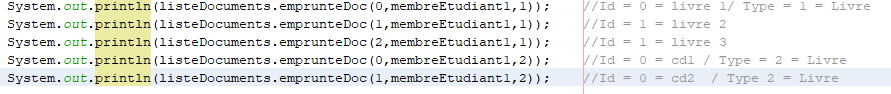
Membre = Objet membre (membreEtudiant1 ou membreEtudiant2)

TYPE = Nous demandons à l’utilisateur quel « type » de  document veut-il emprunter  
 (soit Livre, cd, docUrl)

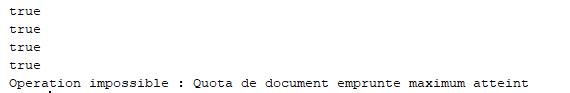
=1 =2 =3

Si le type docUrl est choisi, le programme renvoit une erreur (Operation impossible objet non physique… (conf 2) )

Empruntons alors 3 livres et 1 cd et essayons d’emprunter un 5ème document (le cd2)

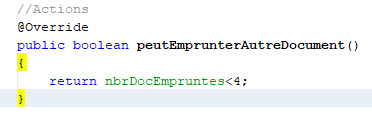


Réponse de la console :



Le quota est dépassé donc l’étudiant ne peut pas emprunter un 5 ème document tant qu’il n’a pas retourner ses anciens documents.

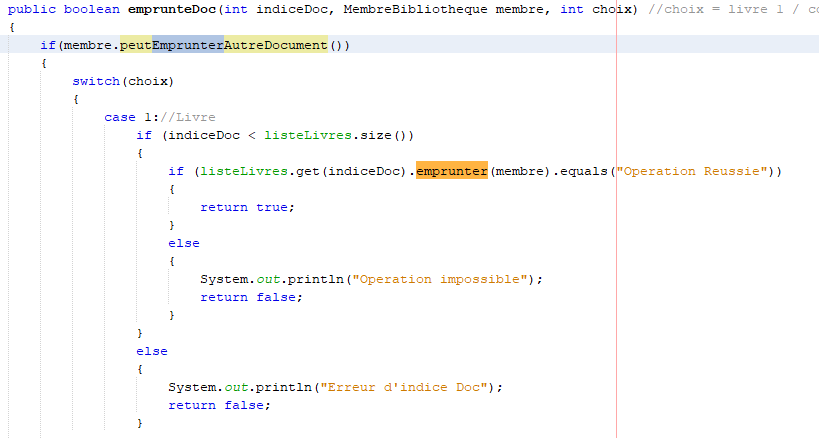
Cela fonctionne aussi grâce à de l’overriding :

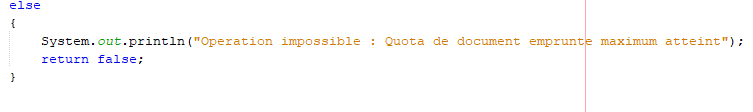


Il retourne true si nbrDocEmpruntes<4

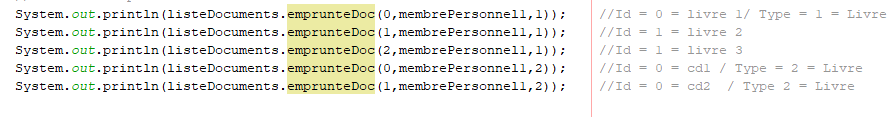
Sinon, il retourne false.

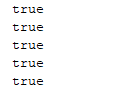
Voici la fonction qui regarde le nbrDocEmpruntes avant de faire l’emprunt





Faisons aussi le test avec un membrePersonnel (qui peut emprunter au maximum 8 documents





On en conclut que l’opération à réussie car la classe membrePersonnel contient un override différent :

