



Cégep **André-Laurendeau**

Hiver 2025

Cégep André-Laurendeau

Travail pratique #2

420-445 Programmation dans un environnement transactionnel

François Lacoursière

Hiver 2025

Consignes du travail

- Le travail est à réaliser individuellement (Sauf pour établir les stories et acteurs).
- Ce travail compte pour 10% de la note finale.
- La date de remise est le mardi 11 mars 2025 à 09:00 du matin.
- Tout retard dans la remise entraînera une pénalité de 10% par jour de retard jusqu'à concurrence de 5 jours. Après cette date, la note zéro sera attribuée au travail.

Ce qu'il faut faire

Dans ce travail pratique vous allez mettre en pratique les notions suivantes :

- Programmation java en utilisant le framework Hibernate et une base de données H2. Vous devrez implanter les fonctionnalités suivantes tout en suivant le modèle des couches présentés en classe.
 - Enregistrement d'un nouveau client (emprunteur)
 - Ajout de livres, CD et DVD à la bibliothèques (vous devez gérer les multiples copies d'un même livre, CD et DVD, aussi appelés 'exemplaires'). S'il n'y a pas d'exemplaire de disponible, l'emprunteur ne peut donc pas emprunter
 - Recherche de livre selon les critères suivant : Titre, auteur et année. Pour le titre, on doit effectuer une recherche qui permette qu'elle ne soit pas exacte. Par exemple, si le titre d'un livre est : « Les parapluies sont disparus », on devra pouvoir obtenir le document avec les critères de recherche contenant simplement le mot « parapluies » ou le mot « disparus ». Il faut également pouvoir chercher les titres et artistes des CD et DVD
 - L'emprunt d'un livre, CD ou DVD en tenant compte s'il reste au moins un exemplaire de disponible.
 - Le client doit pouvoir obtenir la liste de ses emprunts ainsi que les

dates auxquelles il doit retourner chacun des documents empruntés. Vous devez me démontrer ça avec un `System.out.println()` dans un script `Main`. Assurez-vous que l'on puisse voir le client adéquatement avec ses emprunts, les noms des documents ainsi que les dates d'emprunt et de retour. Comme vous ne faites toujours pas de code 'frontend', vous devrez exécuter vos scénarios dans des scripts dans une classe 'Main'. Un `toString()` approprié permettra de bien voir ces informations.

- Vous NE DEVEZ PAS prendre le `fetch eager` dans les entités mais plutôt utiliser des queries lorsque vous devez avoir les collections enfants initialisées. Utilisez plutôt les 'Queries JPQL' tel que vu en classe.
- Le code doit être remis dans un repo github classroom avec des commits ayant la bonne granularité. (-5 première infraction, -2 pour les autres jusqu'au maximum de points pour les commits)
- La bd à utiliser est H2 avec un URL en mémoire exemple `:jdbc:h2:mem:tp2nometudiant` (-10 pour non respect)
Remplacez 'nometudiant' par votre propre nom.

Livrable

Chaque étudiant devra livrer dans GitHub classroom les éléments suivants :

- Le code java dans le repo github. Je veux avoir des commits sensibles, c'est-à-dire ayant la bonne granularité avec des descriptions adéquates. Quand une fonctionnalité est terminée, aussi petite soit-elle, je veux un commit. Je ne veux pas de commit qui chevauchent 2 ajouts de fonctionnalité (Ex : Emprunt d'un document et recherche de document dans le même commit) Il devrait y avoir AU MOINS une quinzaine de commits.

Besoin client (rappel)

La librairie de la ville de Javatown a besoin de moderniser sa bibliothèque. Un système manuel est présentement en place et ne fait plus l'affaire, plusieurs livres sont perdus, les amendes des clients se perdent et plusieurs autres petits problèmes hantent les opérations de cette bibliothèque.

Votre tâche tout au cours de la session sera de produire et de livrer une nouvelle solution pour informatiser les opérations de la bibliothèque.

Votre première tâche sera d'identifier les acteurs du système ainsi que de produire les 'stories' décrivant les tâches et responsabilités de chacun. Dans le TP2, on connaît déjà les acteurs : Préposé et Emprunteur

La bibliothèque prête plusieurs types de documents soit évidemment des livres mais également des cd et aussi des dvd. Les clients, qui sont des résidents de Javatown, peuvent emprunter gratuitement ces documents pour une période de 3 semaines pour les livres, 2 semaines pour les CD et 1 semaine pour les DVD. Si un emprunteur enfreint les délais, une amende de 25 cents par jour lui est facturé. Celui-ci ne pourra emprunter de nouveaux documents s'il a des frais accumulés à son compte. Un préposé de la bibliothèque s'occupe d'entrer les documents dans le système. Pour les livres, il y consigne notamment le titre, l'auteur, l'éditeur, l'année de publication, le nombre de pages. Pour les cd et dvd, on y consigne plutôt le titre, l'artiste, le nombre de chansons et la durée totale. Les utilisateurs peuvent rechercher des documents à l'aide de la nouvelle plate-forme. Ils peuvent

notamment chercher pour les documents d'un auteur, d'un éditeur,
...

Les gestionnaires de la bibliothèque peuvent interroger le système pour savoir le nombre de documents emprunter par mois et également le nombre d'amendes produites.

Ils peuvent également savoir à quelle date un document doit revenir à la bibliothèque.

Grille de correction

Les objectifs pédagogiques visés par ce travail :

- Programmation adéquate par couche, tel que vu en classe
- Utilisation des concepts clés de la programmation orientée objet.
- Programmation de la couche persistance à l'aide de JPA.
- Définition adéquate de l'API à la couche service

Grille de correction

Grille de correction						
Éléments visés	Excellent	Très bien	Bien	Moyen	Insuffisant	Nul
Qualité du document. (Phrases syntaxiques; l'orthographe lexicale et grammaticale)	5	4	3	2	1	0
Programmation adéquate des couches. (Service, Persistence et Modèle)	5	4	3	2	1	0
Enregistrer un nouveau client	15	14-12	11-8	7-5	4-1	0
Enregistrer un nouveau livre	15	14-12	11-8	7-5	4-1	0
Recherche de livre (Titre, Auteur, Année) Recherche de cd (Titre, artiste)	15	14-12	11-8	7-5	4-1	0
Emprunt d'un livre en prenant compte du nombre d'exemplaires	20	19-16	15-12	11-6	5-1	0
Liste des emprunts	15	14-12	11-8	7-5	4-1	0
Bonne granularité des commits et bonne utilisation des concepts OO, bonne configuration	10	9-8	7-6	5-4	3-1	0

UML à utiliser

