

Projet COO-POO-Java

• Modalités et rendus :

Le projet est par binôme. Les binômes sont constitués librement au sein d'un même groupe de TD. Il y a deux types de rendus :

1. Un rendu UML : il s'agit de rendre un diagramme de classe complet de l'application. Vous pouvez construire votre diagramme de classe avec n'importe quel logiciel. Ce rendu doit être adressé à votre professeur de TD.
2. Un rendu POO-Java : il s'agit de rendre les fichiers sources et l'exécutable de votre projet. Ce rendu doit être adressé à votre professeur de TD de POO-JAVA.

Les soutenances du projet POO-JAVA auront lieu pendant la semaine du 8 Juin 2020.

• But du projet

Le but de ce projet est de développer une application pour une vidéothèque afin de faciliter la gestion des locations de ses produits. A cette fin, on utilisera les concepts de la programmation objets. Dans un premier temps on doit élaborer un diagramme de classe qui correspond à la gestion de la vidéothèque. Ensuite, on doit effectuer son implémentation en programmation objets java. Le résultat final du projet est de présenter une interface graphique qui permet d'établir une facture correspondant à la location des produits de la vidéothèque par un client donné.

• Sujet :

Une vidéothèque propose ses produits à la location. Le prix à la location dépend du tarif par jour de chaque produit. Chaque produit est défini par un identifiant, un titre et un tarif journalier. Parmi, les produits proposés, il y a les documents et les supports numériques. Les documents sont de deux types : les livres et les dictionnaires. les livres ont un auteur et les dictionnaires ont une langue. Dans la catégorie des livres, il y a trois types de produits : les romans, les manuels scolaires et les BD. Dans la catégorie des supports numériques, il y a les CD qui ont une année de sortie et les DVD qui ont un réalisateur. Les clients de la vidéothèque sont définis par un identifiant, un nom et un prénom. Une commande a un identifiant, une date de creation et une réduction. Une commande d'un client est constituée de plusieurs emprunt. Chaque ligne de commande correspond à l'emprunt d'un produit. Un emprunt a une date de début et de fin de location. La date de début de l'emprunt correspond à la date de création de la commande. En ce qui concerne les clients de la vidéothèque, il y a deux types de clients : les clients fidèles et les clients occasionnels. Les premiers bénéficient d'une réduction de 10% sur toute la commande, alors que les seconds n'ont aucune réduction. On peut ajouter ou supprimer une commande d'un client, accéder à son nom et à sa réduction.

1. Il est demandé d'établir un diagramme de classe complet de l'application. Il s'agit d'écrire entièrement les classes qui interviennent dans la vidéothèque en évitant toute redondance de code. Il faut préciser les classes, les classes abstraites, les relations d'héritage, de composition, d'agrégation, les cardinalités, les attributs et les méthodes. Notez que la vidéothèque pourrait, par la suite, étendre sa gamme de produits (avec des livres audio ou des e-books par exemple). La solution que vous retenez doit permettre d'intégrer aisément ces nouveaux produits sans modifier les classes déjà écrites.
2. Il est demandé d'implémenter une interface graphique pour la gestion de location des produits de la vidéothèque. Cette application permet entre autres, d'effectuer des opérations sur la commande d'un client et doit tenir compte de la disponibilité d'un produit dans le stock :
 - (a) Ajouter un client, ajouter une commande, ajouter un emprunt à la commande.
 - (b) Modifier une commande en ajoutant ou en supprimant un emprunt.
 - (c) Calculer le montant total d'une commande d'un client avec réduction ou sans réduction.
 - (d) Ajouter un produit au stock de la vidéothèque.
 - (e) Récupérer la liste des produits disponibles (ou indisponibles) dans le stock.
 - (f) Récupérer la quantité d'un produit du stock. etc

• Extensions possibles :

- On peut établir une facture d'un client sous forme d'un fichier (.txt, csv ou pdf par exemple). Cette facture comporte les références de la commande et du client, le nom du client, ainsi que la liste de produits loués.
- On peut utiliser les tests unitaires (JUnits) et produire la documentation à l'aide de javadoc pour chacune des classes.
- On peut aussi produire un projet qui respecte la modèle MVC (Modèle, Vue , Controleur). Le module Modèle définit les données utilisées par l'application, le module Vue définit la manière dont les informations vont être affichées sur l'écran. Ici, il s'agit de l'interface graphique. Le controleur traite les requêtes lors de l'interaction de l'utilisateur avec la Vue.