Projet 2

Date de remise : mardi 24 novembre à 23h55.

Objectifs d'apprentissage

- Analyser un problème
- Créer une interface graphique conviviale
- Choisir des structures de données adéquates
- Implanter une application robuste
- Pratiquer le modèle MVC
- Utiliser le composant TableView
- Utiliser le composant TabPane
- Sérialiser des objets
- Utiliser les singletons

Remarques

- Votre interface graphique doit comporter au moins une barre de menu, une image et doit être facile à utiliser.
- Compléter le diagramme de classes réalisé dans le cours d'analyse.
- Utiliser le modèle MVC pour implanter vos classes
- Les fichiers de données seront disponibles sur LÉA.

Type d'évaluation

Sommative (10% de la note du cours)

À remettre

Le dossier contenant votre **projet complet** doit être déposé via LEA

Barème

• Fonctionnement (fiabilité, validité, conformité aux spécifications)	85 %
• Style (Programmation objet : utilisation de classes pour modéliser et instancier les différents composants de l'application, encapsulation, réutilisation, commentaires, respect des normes de programmation, clarté et disposition du code)	15%

Consignes particulières

- Un projet clairement INCOMPLET ou un projet qui ne compile pas entraîne automatiquement la note 0. Il est donc préférable de remettre un projet en retard qu'un projet incomplet. Par INCOMPLET, j'entends quelque chose qui ne fonctionne pas du tout ou qui n'est qu'une ébauche de programme. Vous n'aurez pas de points juste pour fournir des lignes de code. Ça doit ressembler à un projet complet qui fait à peu près ce qui est demandé.
- Une pénalité de 10% s'applique pour chaque jour de retard (max. 40%). Après 4 jours de retard, la note est 0.
- Commentez judicieusement votre code et toutes vos variables locales.
- La validité est évaluée surtout en utilisant le programme. Vous n'obtenez donc aucun point pour le code non fonctionnel. De plus, si un bogue ou un code non fonctionnel m'empêche d'utiliser d'autres parties du programme, vous ne pourrez pas obtenir de points pour ces parties.

Énoncé du problème : Médiathèque

Le travail consiste à coder un système très simplifié de gestion de documents pouvant se retrouver sous différents supports, en utilisant les concepts de la programmation orientée objet.

Notre médiathèque gère un catalogue des documents divers : livres, périodiques et DVD.

Le préposé peut inscrire un prêt, un retour, procéder au paiement d'une amende.
Le préposé peut aussi : acheter un nouveau document et l'ajouter au catalogue, supprimer un document du catalogue.
Le préposé peut aussi ajouter, supprimer et modifier un adhérent. Noter que seuls l'adresse et le numéro de téléphone peuvent être modifiés.

Les adhérents et le préposé peuvent consulter le catalogue. Pour chaque catégorie de documents (livres, périodiques et DVD), la recherche doit être possible par auteur ou par mots clés. Les résultats peuvent être affichés en ordre croissant ou décroissant selon les noms des auteurs ou les dates de publication. La liste doit aussi informer sur la disponibilité des documents affichés.

Tous les documents sont identifiés par un numéro de document, un titre, le nombre de prêts ainsi que la date de publication. En plus de ces informations de base, les livres sont aussi caractérisés par un nom d'auteur, les périodiques possèdent un numéro de volume et un numéro de périodique, les DVD comportent un nom de réalisateur et le nombre de disques.

Un adhérent est identifié par un numéro d'inscription, un numéro de téléphone, un nom, un prénom et une adresse. Il peut consulter son dossier à tout moment en fournissant son numéro de téléphone ou son nom et prénom.

Tous les adhérents ont droit d'emprunter au maximum deux DVD, trois livres et un périodique. La durée d'un prêt est deux semaines pour les livres, 3 jours pour les périodiques et une semaine pour les DVD. Une amende de 0,50$ par jour est imposée pour tout retard lors de retour de documents. Un adhérant ayant une amende impayée ne peut emprunter un nouveau document.

Important :

- Complétez les fichiers textes à l'aide de mots clés de votre choix.
- Votre application doit contenir au moins **deux ensembles de classes** : un pour les classes de données et un pour les classes de l'interface graphique.
- Vous devez utiliser le composant ***TableView*** et ***TabPane*** pour l'affichage des documents (voir démonstration).
- La première fois que l'application est exécutée :
    - Lire les données à partir des fichiers textes (livres.txt, DVD.txt, periodiques.txt,..) pour initialiser les structures de données en mémoire.

    - Quand on quitte l'application, sauvegarder vos données sérialisées dans les fichiers livres.ser, DVD.ser, periodiques.ser,…Notez que vous pouvez sérialiser directement la liste de documents. On peut sauvegarder directement un Arraylist d'objets sérialisables.

- Pour toutes les autres exécutions, les données en mémoire seront initialisées à partir des fichiers .ser
- Toutes les données entrées par l'utilisateur doivent être validées.