

TD5B : développement d'une bibliothèque (MVC)

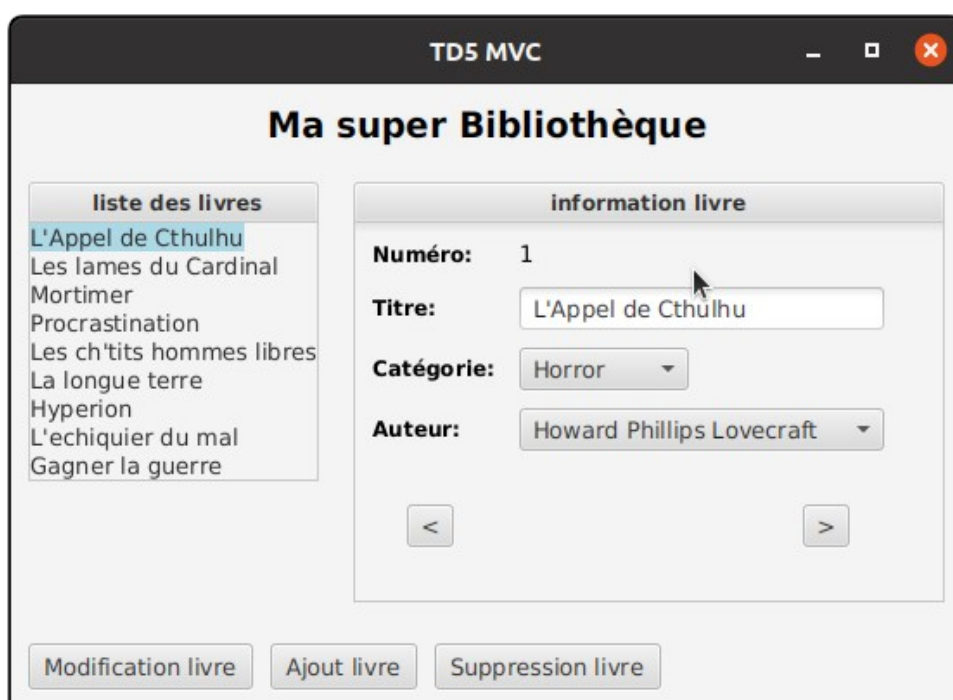
Nous allons mettre en place une architecture MVC (Modèle-Vue-Contrôleur) dans le cadre du développement d'une IHM pour la gestion simplifiée d'une bibliothèque. Rappelez vous du principe. L'utilisateur interagit via la vue avec l'application. Lorsque des événements sont déclenchés, ce sont les contrôleurs qui font le lien dans les 2 sens entre le modèle et la vue et ils peuvent aussi réaliser des traitements. La vue et le modèle ne se connaissent pas.

1) Etude des classes liées au modèle

vous commencerez par étudier la classe qui se nomme **Bibliotheque** (développée par Arnaud Lanoix) qui permet de définir le modèle de l'application. Vous étudierez aussi les classes **Auteur** et **Livre** qui sont dans le répertoire **librairie** du projet et qui sont des classes "outil" qui seront utilisées indifféremment par toutes les couches de l'architecture MVC. **Déterminer le diagramme de classe relatif à ces 3 classes.**

2) Etude des classes liées à la vue

Etudiez la vue principale de l'application pour comprendre comment elle est implémentée => classe **MainVue** et classe **TitledPaneLivre**. **Réaliser le graphe de scène.** Pour l'instant, vous n'avez que le code de la partie vue où les données du modèle ne sont pas encore utilisées. Ce sera votre prochain travail.



3) Affichage de la vue précédente

La classe **Main** qui vous est fournie (on peut considérer cette classe comme le contrôleur principal), permettra de lancer votre application *javaFX*.

Vous allez, dans la méthode **start(...)**, instancier une **MainVue** et un modèle **Bibliothèque**, insérer des données dans la bibliothèque (préremplir) et mettre à jour le panneau droit et le panneau gauche de la vue. Pour l'instant, nous ne nous préoccupons pas des autres contrôleurs. Vous devez avoir le même rendu que dans la capture d'écran précédente.

4) Mise en place du contrôleur *ControleurDetailLivre*

Il permet lors d'un clic souris sur un élément de la liste de gauche, d'afficher dans le panneau droit le détail du livre sélectionné. Lors d'un clic sur un élément, celui qui était déjà sélectionné se désélectionne (méthode dans **MainVue**)

- Les éléments dans la grille de gauche sont des **Label**. Vous pouvez récupérer l'élément cliqué via la propriété *source* de l'événement

- vous pouvez récupérer le numéro de ligne de l'élément cliqué dans la grille via la méthode de classe de **GridPane**

=> `getRowIndex(Node)`

Vous avez accès dans les contrôleurs aux classes du modèle et de la vue.

N'oubliez pas de mettre aussi à jour le panneau droit.

Une fois le contrôleur développé, Il faut maintenant que vous abonniez les éléments de votre liste gauche à ce contrôleur. Ceci se passe dans la classe **Main**. Une méthode existe dans **MainVue** pour le réaliser.



5) Mise en place des contrôleurs de type *ControleurLivrePrecedent* et *ControleurLivreSuivant*

a) Lors du clic sur le bouton "<", dans le panneau de droite s'affichent les détails du livre précédent dans la bibliothèque. Lors du clic sur le bouton ">" dans le panneau de droite s'affichent les détails du livre suivant dans la bibliothèque. Attention de ne pas déclencher d'exceptions lorsque la fin et le début de liste sont atteints.

N'oubliez pas d'abonner les deux boutons précédents à leur contrôleur respectif (méthode dans **MainVue**). Ceci est à réaliser dans le contrôleur principal.

b) Lorsque la fin de liste est atteinte, c'est à dire qu'il n'y a plus de livre suivant, le bouton ">" doit être désactivé.

Réaliser la même chose pour le bouton "<" quand le début de liste est atteint et qu'il n'y a plus par conséquent de livre précédent.

c) Le panneau gauche et le panneau droit doivent être synchronisés. Lorsque les livres défilent dans le panneau de droite à l'aide des deux boutons, le même élément doit être sélectionné dans la grille de gauche (couleur bleu clair). Si un élément est sélectionné à gauche avec la souris, si c'est le premier de la liste alors le bouton "<" est désactivé. Si c'est le dernier de la liste alors le bouton ">" est désactivé.