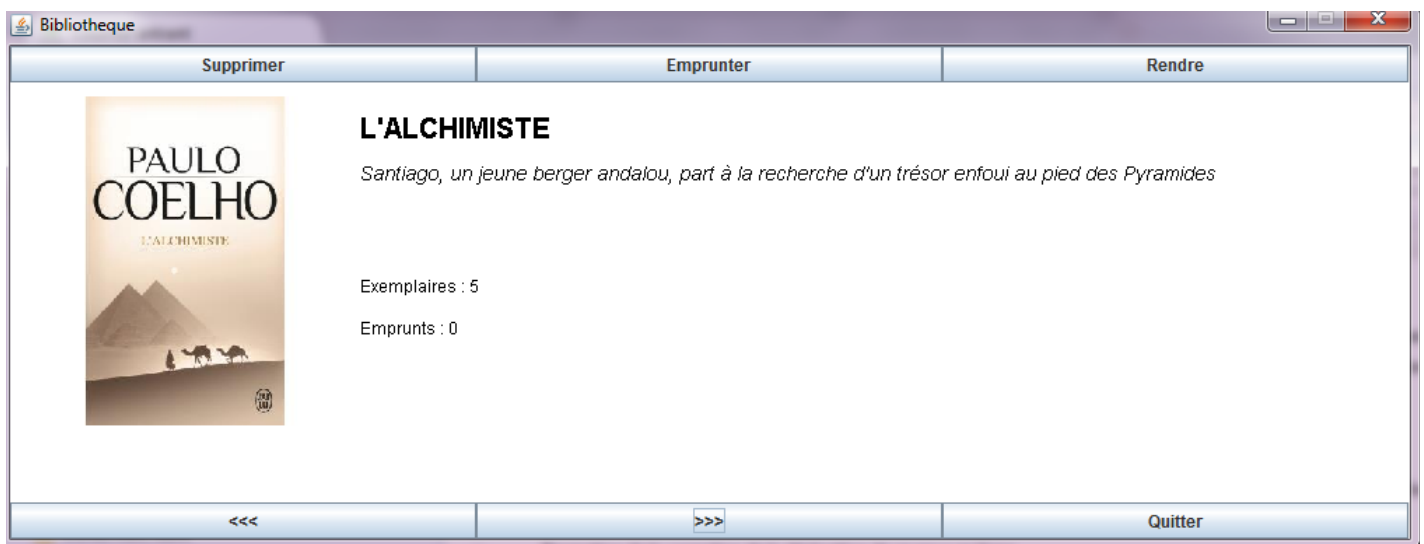


Objectif : Classe, ArrayList, intégration dans une fenêtre. Dissocier le mécanisme de la structure de données et l'affichage graphique.

Conception d'une bibliothèque

🕒 2h30'

Nous voulons gérer une collection de livres et donc établir une bibliothèque qui nous permettra de connaître le livre mais aussi les emprunts dont il peut faire l'objet.



Le nombre d'exemplaires absolue en stock n'est pas amené à changer.

Le nombre d'emprunts changera au fur et à mesure que l'on appuiera sur les boutons Emprunter ou Rendre.

Si l'on appuie sur supprimer, le livre courant affiché est supprimé et remplacé par le suivant. Attention à cette règle si l'on supprime le dernier livre. Attention si la bibliothèque est vide.

Les boutons <<< et >>> font défiler les livres vers la gauche ou droite avec une rotation en bout.

Exécutez en double cliquant sur le .jar dans le dossier DemoBibliotheque ou voir la vidéo de démo.

- Téléchargez le fichier du projet TP2Bibliotheque sur Moodle
- Dézippez le fichier dans le dossier de votre choix sur votre W :
- Ouvrez la classe UnLivre, la classe est complète et nécessite que vous compreniez entièrement son code.
- Exécutez le fichier TestLivre.java

On vous demande de faire la classe Bibliotheque puis d'intégrer la création d'une bibliothèque dans une fenêtre graphique comme pour le diaporama.

Création de Bibliothèque et tests de la classe

- Ouvrez la documentation de la classe *Bibliotheque* en double cliquant sur le fichier *Bibliotheque.html* dans le dossier *JavaDoc Bibliotheque*. Vous pourrez voir le constructeur, les méthodes que vous devez développer.
- Faites les tests unitaires de *Bibliotheque* dans le fichier *TestBibliotheque.java*. Les résultats sont ici affichés sous forme de texte. L'idée étant de valider la mécanique pour passer d'un livre à un autre, de récupérer un livre

Intégration dans une fenêtre graphique

La fenêtre graphique vous donne un conteneur dans lequel vous pouvez placer ce que vous souhaitez. Ici nous allons y placer une bibliothèque.

Vous devez vous inspirer de l'exemple du Diaporama.

Regardez la documentation de la classe *Graphics* sur Google afin d'utiliser ses méthodes dans *paint()*.

1- Déclarez en attribut de la classe une *Bibliotheque*, un livre et une *Image* (pour les couvertures de livre).

2- Constructeur : à la création de la fenêtre, on doit charger la bibliothèque et récupérer le livre courant à afficher.

3- Affichez les éléments du livre dans *paint()* en utilisant la variable *g* de type *Graphics*. Dessinez la couverture puis les textes...

4- Associez aux boutons le traitement qu'il convient. Par exemple, lorsqu'on appuis sur >>> on doit récupérer le nouveau livre et charger une nouvelle couverture. *Repaint()* se chargera de tout redessiner...

Améliorations :

- On souhaite voir en permanence à l'écran la somme de tous les emprunts de tous les livres.
- On souhaite voir en permanence le livre le plus emprunté. Vous devrez détecter le max mais aussi le cas particulier où tous les livres, on le même nombre d'emprunts. Auquel cas, aucun affichage sera fait.

Application autonome Java

L'idée est de pouvoir exécuter par un double clic votre application.

1- Vérifiez que pour aller produire un fichier *.jar* : *Clic droit sur le nom du projet / Propriétés*

2- Allez dans le menu *Exécuter/ Clean and Build*

3- Allez voir sur votre disque dur, dans l'arborescence de votre projet, dans le dossier *Dist*.

4- Faites un dossier sur votre disque dur *MonAppli*.

Placez y votre fichier *.jar* et tout ce que votre application à besoin comme données : images, fichiers txt.

5- Double cliquez sur le fichier *.jar* votre application doit s'exécuter sans avoir besoin de NetBeans.

