OBJECTIFS DU PROJET

- Modéliser à l'aide d'un diagramme de classes UML une application manipulant plusieurs objets
- Manipulant plusieurs objets polymorphes
- Développer une interface graphique
- Intégrer et manipuler plusieurs sources de données (fichier texte/JSON/XML/Binaire)
- Développer une interface graphique manipulant deux fenêtres qui interagissent
- Signaler et prendre en charge des exceptions
-

ÉNONCÉ

L'objectif de ce travail est le développement d'une application Java de gestion de divers documents d'une bibliothèque.

CONTRAINTES

- Le travail doit se faire en binôme.
- Le travail doit être structuré en respectant le principe de la POO et doit se faire en 2 parties comme suit :

  Partie 1 :
  - Conception et représentation du diagramme des classes (UML) de l'application à l'aide du **logiciel de votre choix**
  - Codage en Java des classes et Test de la validation des classes

  Partie 2 :
  - Développement **des interfaces utilisateur (IUG**) de l'application pour la gestion interactive de la bibliothèque

PARTIE 1 : (55PTS)

Conception de diagramme des classes (**UML**) de l'application (**Classes** UML, **relations** entre les classes, **cardinalité**) et **Codage** en java

1. Modélisez et développez vos classes sachant que pour la gestion d'une bibliothèque :

Un document est caractérisé par un code unique et un titre

- ☞ *Le code d'un document est automatiquement généré*
- ☞ *Tout document doit être comparable* par catégorie **et** par code

- Un document de la bibliothèque peut être soit un journal, soit un volume. Un volume peut être soit un livre, soit une BD (bande dessinée)

- Un volume est un document qui a en plus un nom d'auteur.

- Un livre est un volume qui a en plus une année d'édition, un genre (Roman, Fiction, Technique, etc), un nombre total de copies du livre et un nombre de copies disponibles.
  Par exemple : si le nombre total de copies d'un livre est 4. Il est possible qu'il n'y en ait que 1 copie disponible. Les autres copies ayant été empruntées.

- Une bande dessinée est un volume qui a en plus un numéro d'édition

- Une bibliothèque est caractérisée par un nom et une collection de documents. Elle possède une capacité qui peut contenir un nombre **maximum de 500** documents

  - ☞ *Utiliser un tableau dynamique(**ArrayList**) pour l'implantation d'une collection de documents.*

  Les fonctionnalités de base d'une bibliothèque sont :

  - L'ajout d'un nouveau document à la collection de documents.
    - ☞ Au cas où il n'y a plus de place pour le document (si on dépasse la capacité de la bibliothèque), on doit lever une exception
  - La suppression d'un document de la collection de documents.
    - ☞ *Le code du document doit exister dans la bibliothèque avant de le supprimer*
  - L'obtention d'un document à partir de sa position *dans la collection de documents*
  - La recherche d'un document par code
  - La recherche d'un document par titre
  - Le tri des documents par catégorie puis par code

- Seuls les livres peuvent être empruntés.
  - ☞ *Un ajustement du nombre de copies disponibles doit être effectué selon les prêts et les retours*
  - ☞ *Le prêt d'un livre donné revient à vérifier s'il a une copie disponible.*
  - ☞ *Le retour d'un livre revient à rendre disponible une copie de ce livre*
  - ☞ *Au cas où l'on essaye d'emprunter un document qui n'est pas un livre,* une exception doit être levée(signalée)
- Obtenir la liste de tous les documents triés afin de les afficher
- Obtenir la liste triée de tous les livres afin de les afficher.

2. Pour chacune des classes de l'application

- ☞ Fournir un ou des **constructeurs** (obligatoirement un constructeur sans paramètre)
- ☞ Fournir des méthodes **getter/setter** selon le besoin
- ☞ Redéfinir une méthode **toString()**
- ☞ **Toutes** les classes doivent être **Serializable**
- ☞ Signaler des Exceptions selon le besoin

CONTRAINTES :

- Les classes seront créées dans un même package **(classes)** et doivent inclure des commentaires **javadoc**(/** ...*/)
- Utiliser des **énumérations** quand c'est nécessaire
- Utiliser un **ArrayList** et une boucle **foreach** pour parcourir une liste
- Les fichiers ou images seront dans leur propre dossier ou package
- Il est recommandé d'utiliser des classes utilitaires quand c'est nécessaire. Ces classes seront créées dans un même package(**util**)

NORMES DE PROGRAMMATION

- Votre projet doit aussi respecter les normes concernant les règles de nommage, l'indentation et les commentaires.
- Utiliser les commentaires javadoc (/** …*/), expliquez – au-dessus de chaque méthode – ce qu'elle fait, les paramètres dont elle a besoin (s'il y a lieu) et le résultat qu'elle retourne (s'il y a lieu).

3. Développez une *classe Test* qui permet de valider : (voir résultats à l'exécution)

- ✓ **Instancier** une bibliothèque dont le nom est « **Ma Bibliothèque Ahuntsic »**
- ✓ **Ajouter** des documents (Journal, Livre, BD)
- ✓ **Afficher l**a liste triée des documents de la bibliothèque
- ✓ **Rechercher** un livre connaissant son titre puis **l'afficher**
- ✓ **Emprunter** ce livre puis l'**affiche**r si trouvé.
- ✓ **Emprunter** un document (autre qu'un livre). Afficher un message si l'emprunt n'est pas permis
- ✓ **Afficher** la liste triée des livres
- ✓ **Sérialiser** le contenu de la bibliothèque dans un **fichier binaire(.dat)** et un fichier **.json**
    - ☞ *Le fichier binaire généré pourrait être utilisé dans la partie 02 du projet*

Exemple de résultats à l'exécution :

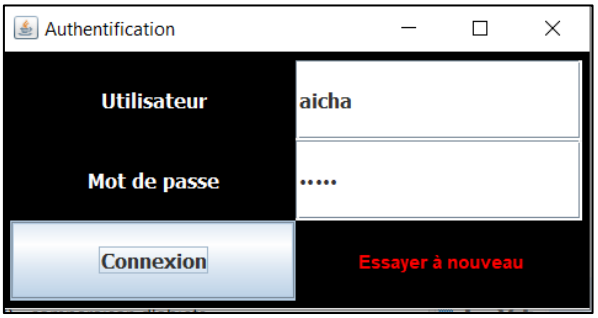
```
Liste triée des documents :
BD : JC-8F600-5F, Pif   Edition: 109
Journal : 21-C65B6-25, Le monde, 2021-03-27
Journal : 22-44B63-27, Devoir, 2022-02-13
Livre : AC-4AFB6-63, L'étranger  1959  Roman 4/4
Livre : VH-99D96-47, Les Misérables 1942  Roman 3/3

Livre trouvé par titre :Livre : AC-4AFB6-63, L'étranger  1959  Roman 4/4
Pret:  Livre : AC-4AFB6-63, L'étranger 1959  Roman 3/4
Pret: Journal: Emprunt non permis

Liste triée des livres :
Livre : AC-4AFB6-63, L'étranger  1959  Roman 3/4
Livre : VH-99D96-47, Les Misérables 1942  Roman 3/3

Bibliothèque sérialisée avec succès
```

PARTIE 2 : (45 POINTS)

Développez l'interface utilisateur (IUG) de l'application qui comportera deux fenêtres:

- L'une pour l'authentification d'un utilisateur
- Et l'autre pour la gestion de la bibliothèque une fois l'authentification réussie

1. La **fenêtre d'authentification** permet d'authentifier un utilisateur (représenté par un ***userId*** et un ***password***). Les informations de l'authentification sont extraites d'une base de données, dans notre cas un fichier texte).



☞ Un composant **JPasswordField** est utilisé comme un **JTextField**.
String pwd = String.valueOf(txtPwd.**getPassword**())

✓ **La fenêtre de gestion** de la bibliothèque est affichée seulement si l'authentification est réussie. Ses principales fonctionnalités permettent de :

- **Charger** le fichier **binaire**, créé dans la partie 1, à l'ouverture de la fenêtre
  ☞ *Ajouter les codes des documents dans la liste déroulante des codes*
- **Supprimer** un document sélectionné à partir de son code
  ☞ *Lors de la suppression d'un document, son code doit être supprimé de la liste déroulante contenant les codes*
- **Lister** les documents triés par type de document puis par code : dans une zone de texte (*JTextArea*) dont le contenu peut être parcouru grâce à des barres de défilement (*JScrollBar*)
- **Ajouter** un nouveau document :
  ☞ *Ajouter son code dans la liste déroulante des codes*
- **Lister** seulement les livres : dans une zone de texte(JTextArea)

- ***Rechercher*** *un document par son* ***code*** *:*
  - ☞ *À la sélection d'un code dans la liste déroulante, les informations du document seront affichées les zones de texte appropriées, le bouton radio de son type sera sélectionné : Ajouter une méthode* ***afficherDoc()*** *qui permettra de vérifier les informations selon le type du document*
- **Rechercher** un document par **titre** :
  - ☞ *Son code sera sélectionné dans la liste déroulante ou bien afficher un message s'il n'existe pas*
- **Emprunter** ou **Retourner** un document :
  - ☞ *Aviser l'utilisateur par un message spécifiant si l'opération est permise ou pas. Dans le cas où elle est permise, si elle s'est effectuée avec succès ou pas.*
- **Effacer** le contenu des zones de texte : *Ajoutez une méthode* ***effacerTout()*** *qui permettra d'effacer le contenu des zones de texte*
- **Quitter** l'application **:**
  - ☞ *Sérialiser le contenu de la bibliothèque dans un fichier* ***json*** *ET dans un fichier* ***binaire***

CONTRAINTES :

- L'interface graphique sera créée dans un nouveau package **(gui)**
- Si vous utilisez des images dans l'interface graphique, regroupez les images dans un package **images**
- Le diagramme de toutes les classes de l'application sera placé dans son propre package (**uml**)

ÉCHÉANCIER ET BARÈME DE CORRECTION :

L'interface graphique d'authentification :	5 points
L'interface graphique de gestion :	10 points
Fonctionnement, exactitude des résultats :	26 points
Originalité de l'IUG, lisibilité du code :	2 points
Runnable:	2 points

ÉCHÉANCIER ET BARÈME DE CORRECTION :((100 Points) 27 MAI 2022

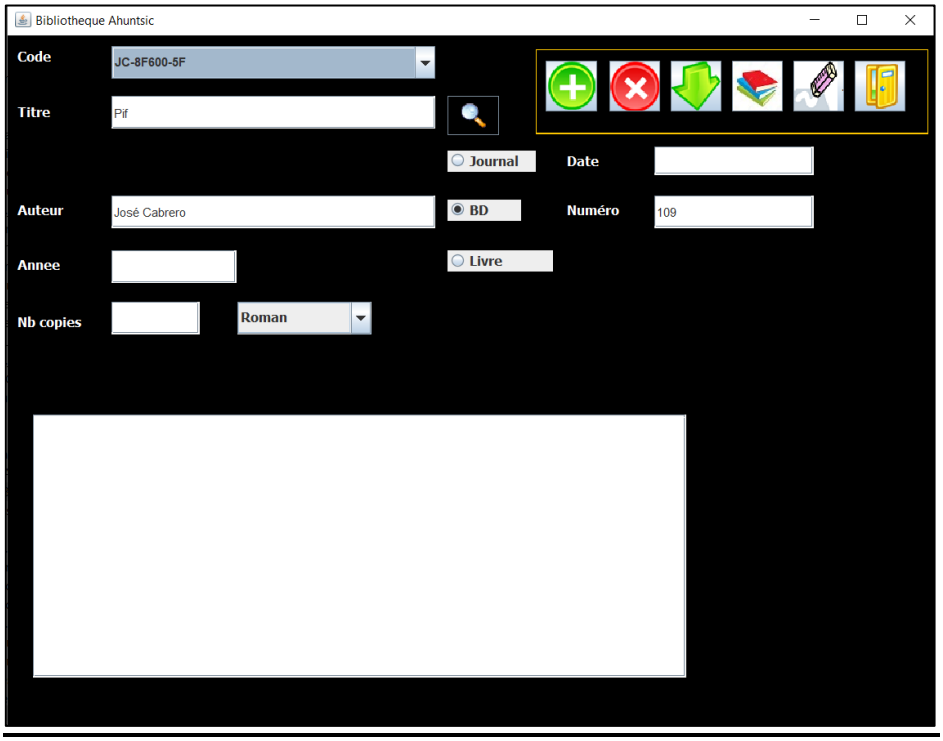
Le diagramme UML des classes :	5 points
Le codage des classes :	25 points
La classe Test :	10 points
Respect des contraintes :	10 points
Validations :	5 points
Interface graphique	45 points

REMISE

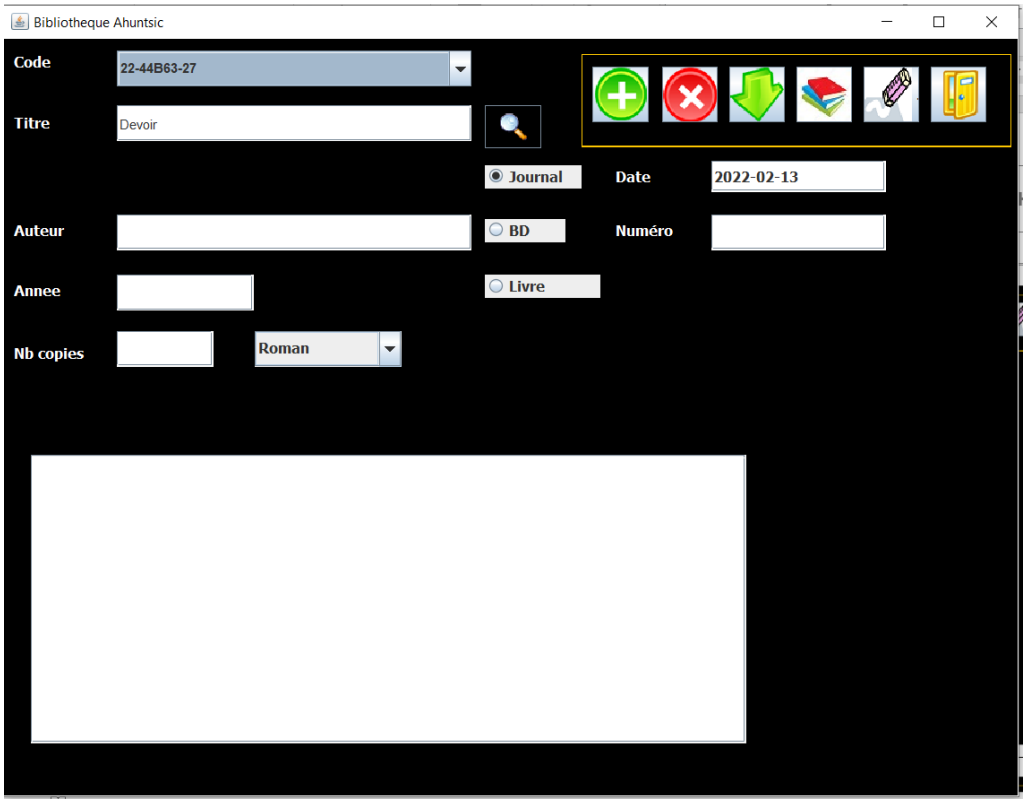
- ✓ Créez un fichier **241Projet3NomPrenom.zip** contenant tout le contenu de votre projet (les packages **uml**, **classes** et **gui**, les **images** et les **fichiers**) ainsi que le **Runnable** (**.jar**)
- ✓ Déposez le fichier compressé dans le dossier Remise sur **LEA** de votre professeur

EXEMPLE D'EXÉCUTION :

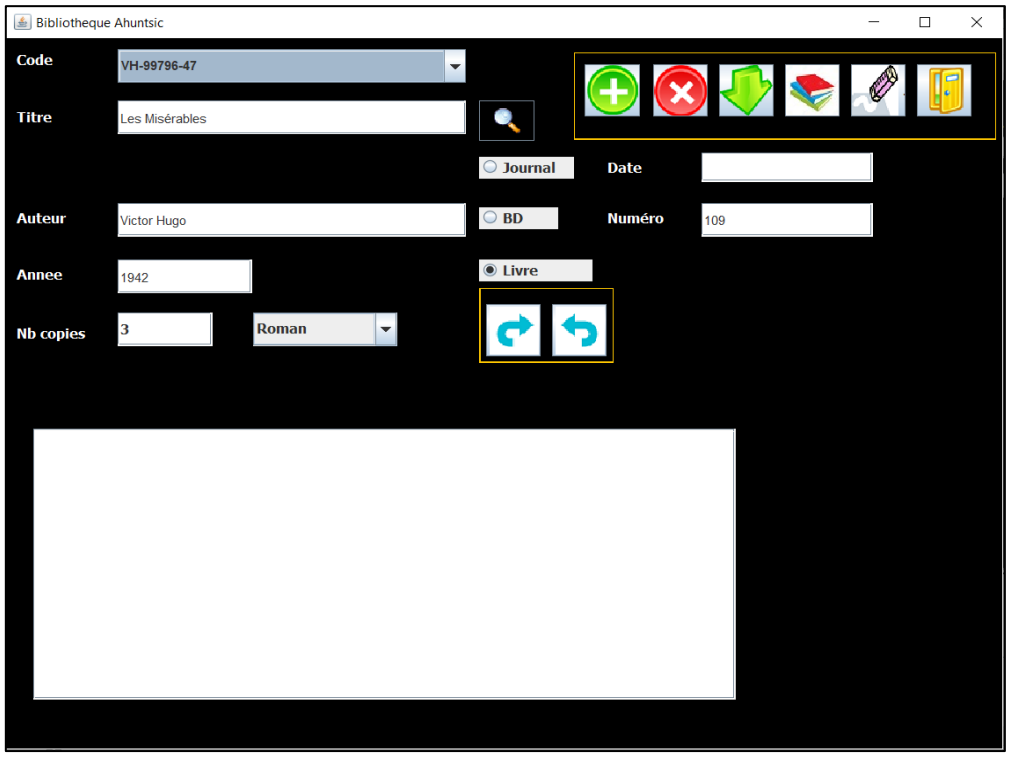
1.Après chargement de la fenêtre de gestion



2.Après sélection du code d'un journal

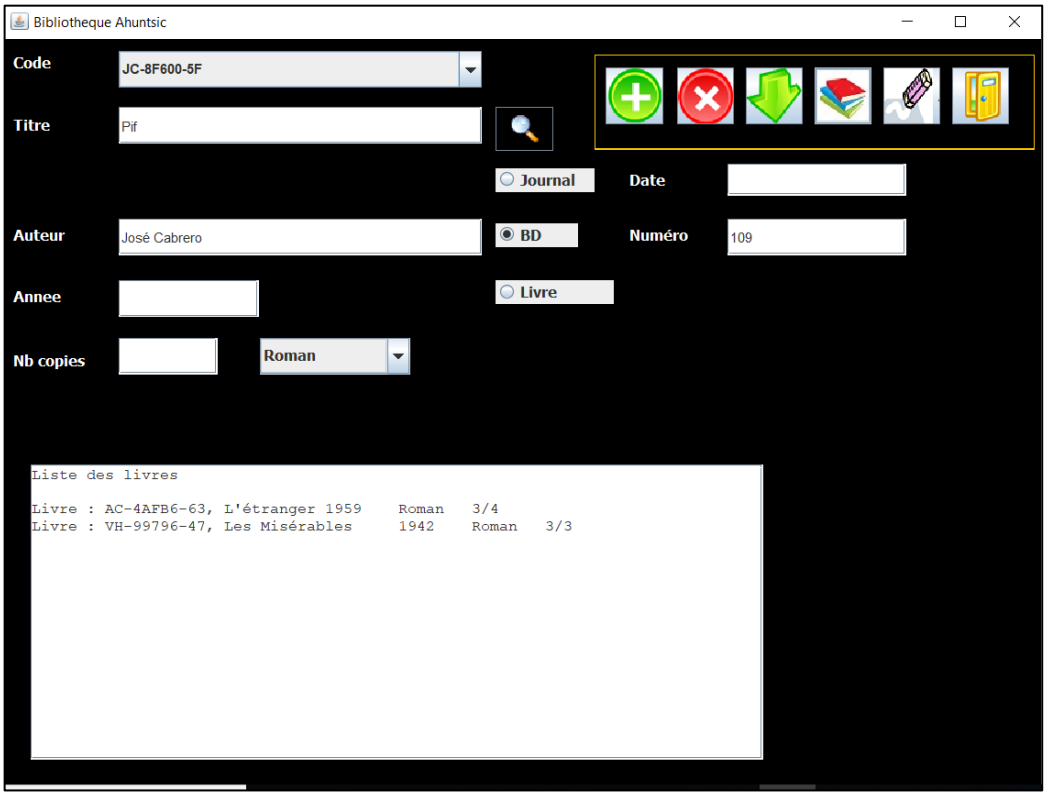


3.Après sélection du code d'un journal



4.Après l'affichage des documents triés de la bibliothèque

5.Après l'affichage des livres de la bibliothèque


Bibliotheque Ahuntsic
Code
JC-8F600-5F
Titre
Pif
Journal
Date
Auteur
José Cabrero
BD
Numéro
109
Annee
Livre
Nb copies
Roman
Liste des livres
Livre : AC-4AFB6-63, L'étranger 1959 Roman 3/4
Livre : VH-99796-47, Les Misérables 1942 Roman 3/3