

---

# Sujet de TP N°2 - Gestion d'une bibliothèque en Kotlin (Android)

## Modalité de remise :

- **Le projet doit être déposé sur un repository GIT.**
- **La compilation du projet se fera directement dans Android Studio.**

## Préambule

Ce travail pratique est noté et contribue à la note finale du module. Il vise à évaluer vos compétences tant sur le plan fonctionnel que technique.

## Critères de notation :

1. **Partie fonctionnelle** : Le rendu doit répondre aux besoins énoncés dans le sujet du TP.
2. **Partie technique** :
  - Respect des bonnes pratiques abordées en cours.
  - Qualité du code (clarté, structure, nomenclature).
  - Gestion des erreurs et absence de bugs bloquants.

## VUE D'ENSEMBLE

Dans ce TP, vous allez développer une application de gestion de bibliothèque en utilisant Kotlin et Android. Ce projet vous permettra de mettre en pratique les concepts de la programmation orientée objet tout en vous familiarisant avec le développement Android.

## OBJECTIFS

Créer une application de gestion de bibliothèque qui permet de gérer les livres, les auteurs et les utilisateurs, en suivant les étapes ci-dessous. Les fonctionnalités suivantes devront être mises en œuvre :

- **Gestion des livres** : L'utilisateur doit pouvoir ajouter, supprimer et consulter la liste des livres disponibles dans la bibliothèque.
- **Emprunt de livres** : Un utilisateur peut emprunter un livre papier disponible. Lorsqu'un livre est emprunté, son état doit passer à "emprunté", et aucun autre utilisateur ne pourra l'emprunter tant qu'il n'est pas restitué.

- 
- **Téléchargement de livres numériques** : Un utilisateur peut télécharger un livre numérique. Le nombre de téléchargements doit être mis à jour en conséquence.
  - **Recherche de livres** : L'utilisateur doit pouvoir rechercher des livres par différents critères, tels que le titre, l'auteur ou l'année de publication.
  - **Affichage des opérations** : Les résultats des actions effectuées (ajout, emprunt, recherche, etc.) doivent être clairement affichés dans la console pour vérification.

## Étapes du TP

### 1. Création de la structure de base

- Créez un nouveau projet Android en **Kotlin**, choisissez le template **Empty Views Activity**.
- Nommez votre application : `tp_1_name1_name2`.
- Choisir comme langage **Kotlin** et comme version minimum SDK l'API 26 (**Android 8.0**).
- Organisez vos packages pour refléter la structure de votre application (par exemple `model`, `view`, `controller`).

### 2. Modèle de données

#### 1. Créez une classe abstraite **Book** dans le package **model**.

- Propriétés de base à inclure dans **Book** :
  - `title: String`
  - `author: String`
  - `publicationYear: Int`

#### 2. Créez deux classes **PaperBook** et **DigitalBook** qui héritent de **Book**.

- Ajoutez des propriétés spécifiques pour chaque sous-classe :
  - **PaperBook** : `numberOfPages`
  - **DigitalBook** : `downloadNumbers`

#### 3. Ajout de l'état (**status**).

- Créez une enum **BookStatus** dans le package **model** qui représente l'état du livre. Elle aura deux valeurs :
  - **AVAILABLE** (indique que le livre est disponible)
  - **BORROWED** (indique que le livre est emprunté)
- Ajoutez une propriété **status** à la classe **PaperBook** en utilisant cette enum.

---

#### 4. Ajout des catégories (**category**).

- Créez une enum **Category** dans le package `model` pour représenter les différentes catégories de livres. Quelques exemples de valeurs :
  - `FANTASTIC`, `FANTASY`, `FICTION`...
- Ajoutez une propriété **category** à la classe `Book` en utilisant cette enum.

#### 5. Créez une classe **Utilisateur** dans le package `model` qui représente un utilisateur de la bibliothèque.

- Propriétés à inclure dans `Utilisateur` :
  - `name` (nom de l'utilisateur)
  - `borrowedPaperBooks` (liste des livres papier empruntés par l'utilisateur)
  - `downloadedDigitalBooks` (liste des livres numériques téléchargés par l'utilisateur)

#### 6. Créez une classe **Bibliothèque** dans le package `model`.

- Propriétés à inclure dans `Bibliothèque` :
  - `availableBooks` (liste contenant les instances de `Book`)
- Ajoutez des méthodes pour gérer les livres :
  - `addBook` (Ajoute un livre à la liste)
  - `removeBook` (Supprime un livre de la liste)

### 3. Gérer les emprunts et les achats de livres

- Ajoutez des fonctionnalités pour gérer les emprunts de livres papier et les téléchargements de livres numériques.
- Assurez-vous que la logique de gestion des emprunts respecte les conditions d'emprunt (ex. : un seul exemplaire pour les livres papier).
- Implémentez des méthodes pour mettre à jour les états des livres en fonction des actions des utilisateurs.

---

## 4. Chargement des données

- Créez une méthode permettant de charger des données de livres à partir d'une chaîne de caractères.
- Cette chaîne de caractères est disponible dans fichier texte (`books.txt`).
- La méthode doit être capable de parser les données et de créer les instances appropriées de livres.

## 5. Recherche de livres

- Définissez une interface pour la recherche de livres.
- Implémentez des méthodes permettant de rechercher des livres par différents critères, tels que le titre, l'auteur et l'année de publication.

## 6. Affichage

- Pour cette première version du TP, concentrez-vous sur l'affichage des opérations dans la console.
- Aucune interface graphique n'est requise pour cette partie, mais les résultats des opérations doivent être clairement visibles pour le débogage et la vérification des fonctionnalités.