

# Exercice Avancé en Java sur la POO

---

Cet exercice met l'accent sur les concepts avancés de la Programmation Orientée Objet (POO) en Java.

## Concepts abordés

- **Encapsulation, Héritage, Polymorphisme, Abstraction** : Principes fondamentaux de la POO.

## Scénario

Vous devez développer un simple **système de gestion de bibliothèque** permettant de suivre différents types d'éléments (par exemple, livres, magazines).

## Spécifications

### Classes et *Records*

1. **LibraryItem** : Classe de base abstraite pour tous les éléments de la bibliothèque.
  - Champs : `id` (int), `title` (String), `publicationYear` (int).
  - Méthode abstraite : `String getDetails()`.
2. **Book (sous-classe)** :
  - Champs supplémentaires : `author` (String), `genre` (String).
3. **Magazine (sous-classe)** :
  - Champ supplémentaire : `issueNumber` (int).
4. **BorrowRecord** :
  - Champs : `itemId` (int), `borrowerName` (String), `borrowDate` (String).

### Fonctionnalités principales

- Ajouter des éléments à une collection.
- Lister tous les éléments de la bibliothèque.
- Emprunter un élément et créer un *BorrowRecord*.
- Voir les livres disponibles et ceux empruntés.

### Exemple de sortie

```
Ajout des éléments à la bibliothèque...
Liste des éléments disponibles :
- Livre : [Titre : Effective Java, Auteur : Joshua Bloch, Genre : Programmation,
Année : 2018]
- Magazine : [Titre : National Geographic, Numéro : 42, Année : 2023]

Emprunt d'un élément...
```

```
BorrowRecord[itemId=1, borrowerName=John Doe, borrowDate=2024-12-01]
```

Liste des éléments empruntés :

- Livre : [Titre : Effective Java, Auteur : Joshua Bloch, Genre : Programmation, Année : 2018]

Liste des éléments disponibles :

- Magazine : [Titre : National Geographic, Numéro : 42, Année : 2023]