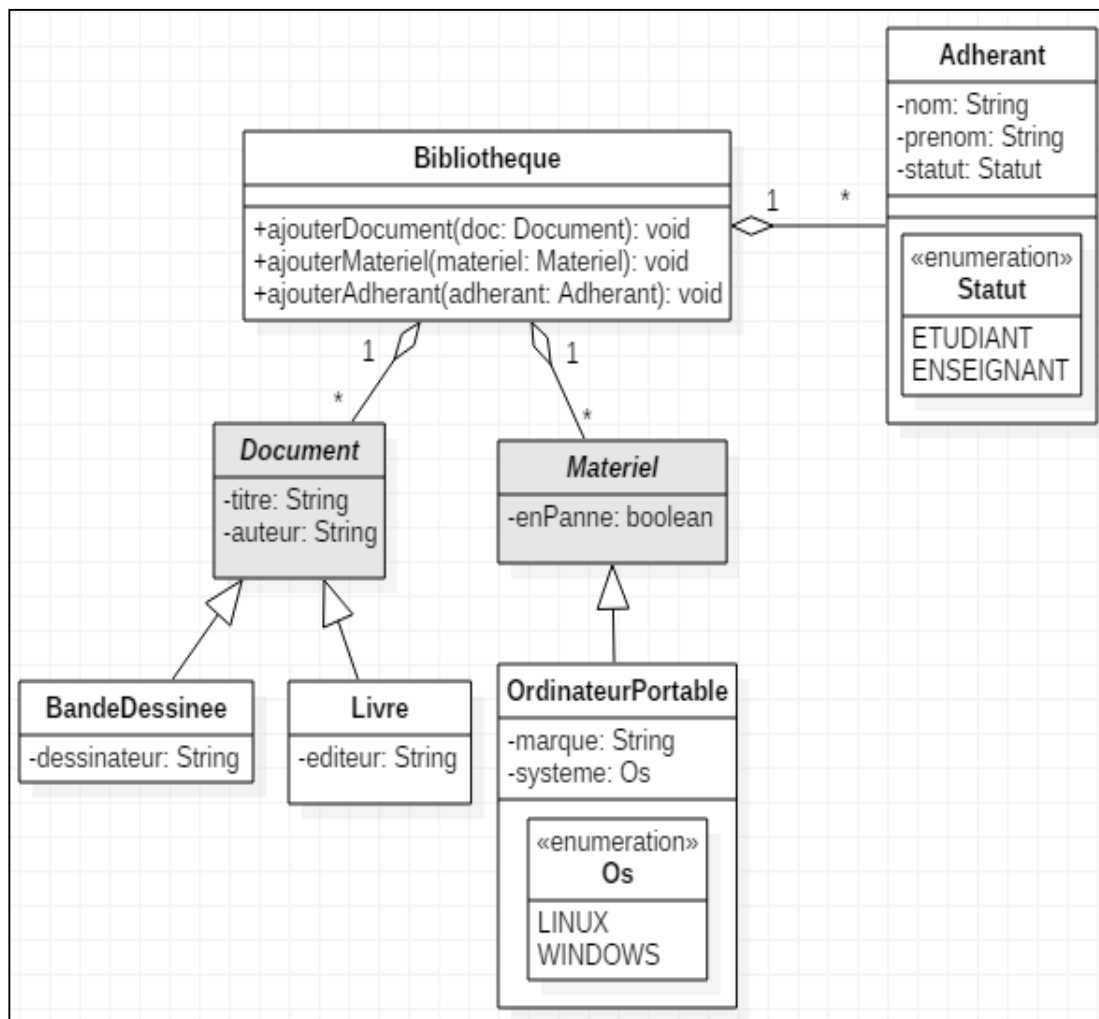


# Université de Nanterre

## Projet de Gestion de Bibliothèque

### Objectif du Projet :

Le présent projet a pour but de concevoir une application de gestion pour une bibliothèque qui met à disposition des livres et des ordinateurs à l'usage des enseignants et des étudiants. Pour ce faire, il est impératif de s'appuyer sur **le diagramme de classes UML** fourni afin d'implémenter toutes les classes nécessaires à l'application de gestion.



Les classes `Document` et `Materiel` sont considérées comme abstraites dans cette architecture.

## Partie 1 : Héritage et Polymorphisme

### Implémentation des Classes :

1. Établir des relations d'héritage entre les classes selon le diagramme UML.
2. Transformer les relations d'agrégation (symbolisées par un losange vide) en compilant les documents, les matériels et les adhérents dans trois listes dynamiques respectives : Document, Matériel, et Adhérent.
3. Introduire des opérations d'ajout dans la bibliothèque conformément aux signatures définies dans le diagramme UML.
4. Créer une classe de test nommée TestBiblio, pour y ajouter une instance de la bibliothèque avec 2 livres, 1 bande dessinée, et 2 ordinateurs portables.

## Partie 2 : Interface

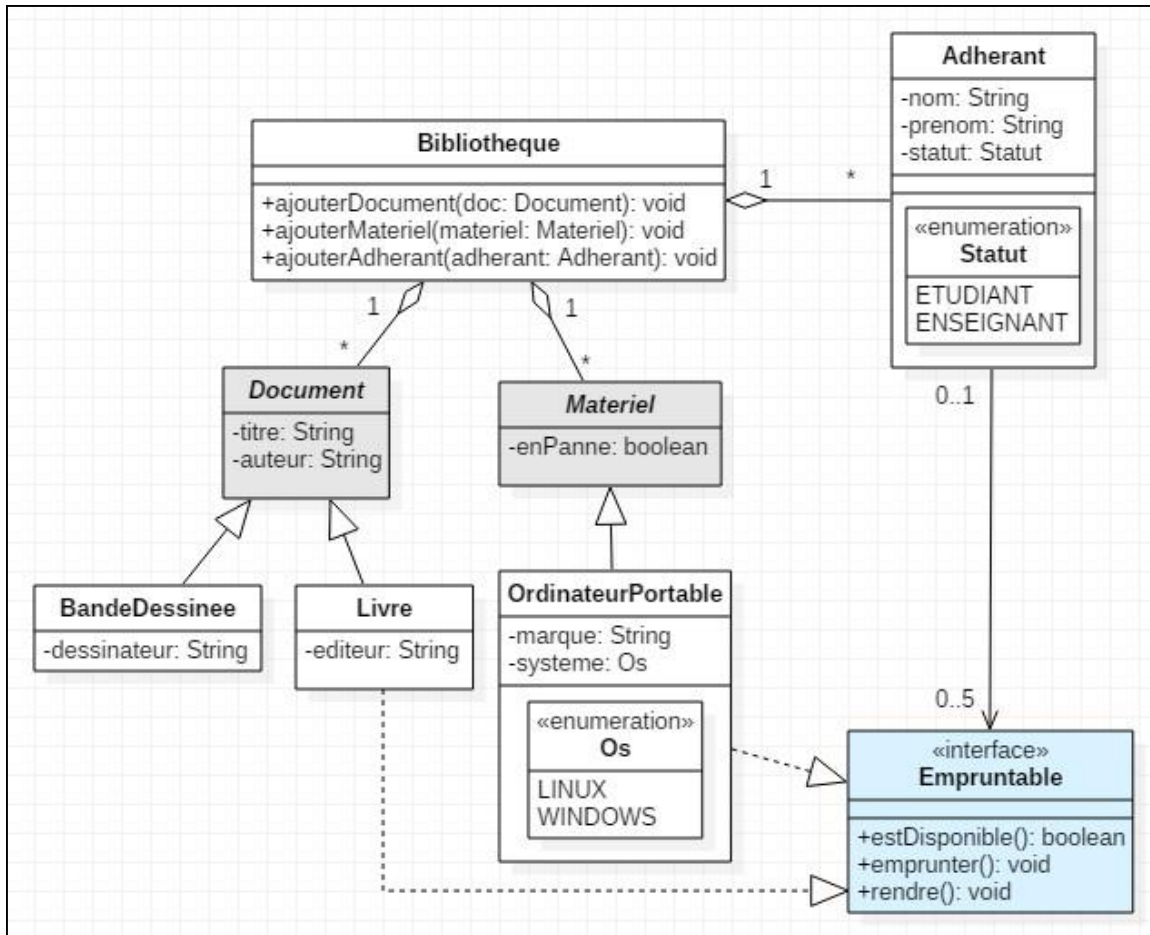
### Introduction de l'Interface Empruntable :

Dans la suite de l'application de gestion de la bibliothèque, nous visons à assurer que **seuls les livres et les ordinateurs portables peuvent être empruntés par les adhérents.**

Afin de mettre en place ceci, nous proposons l'interface Empruntable définie comme suit :

```
/**
 * Interface modélisant un objet qui
 * peut être emprunté et rendu.
 */
public interface Empruntable {
    public boolean estDisponible();
    public void emprunter();
    public void rendre();
}
```

Nous proposons, ainsi, la nouvelle conception du diagramme de classes UML suivante :



1. Ajouter l'interface Emprunable au même package que celui défini précédemment (Utilisation de NetBeans : clic droit sur le nom du package > New > Java Interface)
2. Assurer que les classes Livre et OrdinateurPortable implémentent cette interface Important : Il est nécessaire d'ajouter un état de disponibilité pour les objets empruntables afin d'éviter les emprunts simultanés par plusieurs utilisateurs.
3. Écrire le code pour la gestion des emprunts, en incluant des méthodes pour emprunter et rendre dans la classe des adhérents, en prenant soin de vérifier les conditions d'emprunt spécifiées.
  - a. Pensez à bien vérifier qu'un objet n'est pas emprunté deux fois (vérifier que cela marche en faisant un test dans une classe nommé TestBiblio) et qu'un adhérent n'emprunte pas plus de cinq objets (voir cardinalité dans diagramme UML).
  - b. Ajouter également les messages d'erreurs adéquats.
4. Modifier les opérations d'emprunts pour vérifier que l'on emprunte aux plus cinq livres **mais un seul portable au maximum.**

## **Partie 3 : Bonus**

### **Interface Graphique et Base de Données :**

1. Concevoir une interface graphique pour faciliter la manipulation des ressources de la bibliothèque.
2. Connecter votre application à une base de données pour assurer la persistance des données.