

## Les associations UML et les collections Java

---

### Exercice 1 *Bibliothèque*

---

Une bibliothèque commence l'informatisation de son catalogue. Le catalogue regroupe des informations sur les ouvrages dont elle dispose. Un ouvrage est représenté par sa notice bibliographique.

Une notice bibliographique a un identifiant unique qui est son ISBN. Une notice est caractérisée par son titre, éventuellement son sous-titre et ses contributeurs. Un contributeur est une personne qui a participé à la réalisation de l'ouvrage. Une même personne peut être contributrice dans plusieurs ouvrages, par exemple comme rédacteur, illustrateur, traducteur, rédacteur de la préface, etc. Par exemple, Daniel Pennac est auteur de "La fée Carabine" et rédacteur de la préface de "Une langue venue d'ailleurs" de Akira Mizubayashi. Une notice a un public cible qui est soit enfant, soit junior, soit adulte.

Un abonné possède un numéro d'abonné qui lui est communiqué lors de l'inscription. Il existe des abonnés particuliers qui sont les mineurs (abonné de moins de 18 ans).

La bibliothèque peut disposer de plusieurs exemplaires physiques pour une notice bibliographique donnée. Un exemplaire peut être emprunté puis rendu par un abonné. Un exemplaire a un état parmi : neuf, bon état, mauvais état.

On désire également qu'une notice ait un attribut ou une méthode `estDisponible?` précisant si un des exemplaires de la notice est disponible à l'emprunt. Un exemplaire est indisponible quand il est emprunté ou quand le bibliothécaire l'a mis en réparation (l'exemplaire redevient disponible après réparation).

Un abonné peut emprunter jusqu'à 5 livres à la fois. Les abonnés mineurs ne peuvent emprunter que des ouvrages pour enfants ou juniors.

**Question 1.** Proposez un diagramme UML où vous ferez apparaître les classes, les attributs et les associations qui vous semblent judicieux. On s'intéressera particulièrement aux classes d'associations introduites pour l'emprunt et la contribution. Proposez un diagramme d'instance significatif.

**Question 2.** Proposez une traduction en Java.

**Question 3.** Modifiez votre modélisation et votre implémentation pour avoir, depuis la bibliothèque, un accès aux notices bibliographiques indexé par l'ISBN.

---

### Exercice 2 *Relations maritales*

---

On s'intéresse une fois à la modélisation et l'implémentation d'un petit logiciel gérant les contrats de vie commune pouvant être établis entre deux personnes : le mariage, et le PACS (Pacte civil de solidarité). Dans la loi française, ces deux contrats de vie commune peuvent être contractés par deux personnes, quel que soit leur sexe. Un contrat de vie commune a une date de début et une date de fin (pour simplifier ici, on considérera la date comme un entier et dans l'implémentation, quand un contrat de vie commune est encore en cours, la date de fin sera positionnée à -1). On souhaite conserver pour chaque personne connue du logiciel à la fois son contrat de vie commune courant (s'il existe un tel contrat) et les éventuels anciens contrats de vie commune. Un PACS est contracté dans un tribunal dont on veut stocker le nom, un mariage est contracté dans une mairie, dont on veut conserver le nom, et également le nom de la personne ayant célébré le mariage. De plus, un mariage est associé à un type de contrat de mariage (Communauté, séparation, autre).

**Question 4.** On s'intéresse tout d'abord au mariage seulement, avant l'existence du PACS et du temps où le mariage était réservé aux couples mixtes. Proposez une modélisation des couples par mariage, avec tour à tour une association réflexive, une classe d'association puis une classe à part entière. Discutez.

On se place maintenant dans le cadre plus général décrit plus haut (celui des lois françaises actuelles).

**Question 5.** Dans un diagramme de classes placez une classe `Personne`, une classe `ContratVieCommune`, et de quoi représenter également les mariages et les PACS. De plus prévoyez une classe pour représenter les organismes gérant ces aspects de l'état civil, qui seront dans notre contexte soit des mairies, soit des tribunaux.

**Question 6.** Complétez le diagramme afin de représenter :

- les dates de début et fin des contrats de vie commune (mariage ou PACS),
- le fait qu'un mariage ou un PACS lie 2 personnes, et qu'on conserve à la fois le contrat courant d'une personne, et ses contrats résiliés,
- le fait qu'un mariage est associé à un type de contrat de mariage, qu'on garde mémoire pour un mariage de la mairie de célébration, et du nom de la personne ayant célébré le mariage,

— le tribunal dans lequel est contracté un PACS.

**Question 7.** Ajoutez au diagramme l'ensemble des constructeurs pertinents.

**Question 8.** Donnez un diagramme d'objets représentant 2 personnes de nom A et B, ayant contracté un PACS en 2012 à la mairie de Montpellier, et n'ayant jamais été ni mariées ni pacsées auparavant.

**Question 9.** Écrivez en Java l'ensemble du code nécessaire à l'implémentation d'une méthode permettant de **résilier** un contrat de vie commune. Quand on résilie un tel contrat, on s'assure que le contrat à résilier est bien le contrat courant des 2 personnes impliquées dans le contrat (si ce n'est pas le cas, on affiche un message d'erreur). On fixe la date de fin de contrat à la date courante. L'année courante peut être obtenue avec l'expression `LocalDate.now().getYear()`.

**Question 10.** Écrivez en Java l'ensemble du code nécessaire à l'implémentation d'une méthode permettant de **contracter** un contrat de vie commune. Cette méthode positionnera les deux personnes impliquées dans le contrat, et fera en sorte que l'instance courante devienne le contrat courant de ces deux personnes. Dans le cas d'un mariage, il faudra au préalable vérifier qu'aucune des deux personnes impliquées n'est mariée (si l'une ou les deux sont pacsées, les PACS contractés doivent automatiquement être résiliés). Dans le cas d'un PACS, il faudra au préalable s'assurer qu'aucune des deux personnes impliquées n'a de contrat de vie commune en cours.

**Question 11.** Écrivez en Java une méthode qui permet de détecter les erreurs de dates dans les anciens contrats, c'est-à-dire qui vérifie qu'aucun ancien contrat n'en chevauche un autre. On pourra pour cela trier les contrats par date de début croissante.