

Java DUT 1 Feuille TP3
Université Paris-Est Marne-la-Vallée

Exercice 1.—

On définit une classe `Book` pour représenter des livres. Chaque livre possède un titre (`title`) et un auteur (`author`) qui seront des chaînes de caractères, c'est-à-dire des objets `String`.

- a) Écrire un constructeur pour la classe `Book`. Les champs seront `private`.
- b) Écrire une classe `BookTest` avec une méthode `main` pour tester le constructeur et les méthodes de `Book`. Testez le constructeur.
- c) On considère que les champs d'un `Book` ne seront jamais changés une fois que celui-ci a été créé. Quels sont les accesseurs que l'on doit mettre ? Écrivez les. Que doit on ajouter aussi ?
- d) Écrire un autre constructeur qui prend juste un `title` et pas d'`author`. On initialisera le champ `author` avec "no author" dans ce cas. Comment le compilateur fait-il pour savoir quel constructeur appeler ?
- e) Comment faire maintenant pour que le second constructeur appelle le premier ? On utilisera l'appel `this()` placé en tête du constructeur de la façon suivante.

```
public Book(String title) {  
    this(title, "no author");  
}
```

Exercice 2.—

On définit une classe `BookStore` pour stocker des livres. La classe comprend un champ `catalog` de type `ArrayList<Book>` qui est une liste d'objets `Book`. Un livre pourra figurer plusieurs fois dans le catalogue. On verra plus tard comment utiliser d'autres classes pour s'assurer qu'un même livre figure une seule fois ou pour tenir compte du nombre d'exemplaires.

- a) Écrire le début de la classe `BookStore` avec un constructeur qui crée un catalogue vide et un "getter".
- b) Écrire une méthode `add` dans la classe `BookStore`. qui permet d'ajouter un livre dans le catalogue.
- c) Écrire une classe `Test` avec une méthode `main`. Créez trois livres, un catalogue et mettez les livres dans le catalogue. Afficher le catalogue. Quelles méthodes doit-on ajouter pour que l'affichage fonctionne bien ?

d) Qu'affiche le code ci-dessous dans `Test` ?

```
public static void main(String[] args){
    Book b1 = new Book("Lolita", "Nabokov");
    Book b2 = new Book(null, null);
    BookStore bookStore = new BookStore();
    bookStore.add(b1);
    bookStore.add(b2);
    System.out.println(bookStore);
}
```

Où se situe le problème ?

- e) Corrigez le problème en empêchant la construction d'un `Book` avec des champs `null`. On utilisera la méthode `java.util.Objects.requireNonNull`.
- f) Écrire une méthode `contains(Book b)` dans la classe `BookStore` qui renvoie `true` si le livre en argument figure dans le catalogue et `false` sinon. Tester la méthode `contains` dans `Test` avec le code suivant.

```
public static void main(String[] args){
    Book b1 = new Book("Lolita", "Nabokov");
    Book b2 = new Book("Les fruits du Congo", "Vialatte");
    Book b3 = new Book("Lolita", "Nabokov");
    BookStore bookStore = new BookStore();
    bookStore.add(b1);
    bookStore.add(b2);
    System.out.println(bookStore.contains(b3));
}
```

Que remarque-t-on ? Pourquoi ?

- g) Modifier la classe `Book` pour que `contains` fonctionne correctement.