

Conception et programmation orientée objet

Modèle, constructeurs et visibilité

1 Livre

On souhaite écrire un programme pour une bibliothèque. Cette dernière nous informe que, pour identifier un livre, elle utilise son auteur, son titre et son nombre de pages.

1. De quels attributs avez-vous besoin pour représenter une telle classe ? Pour chacun, précisez leur type.
2. Par défaut, supposez qu'un livre est composé de 5 pages. Comment faire cela en pratique ?

Supposons qu'un tel livre soit représenté par la classe `Book`. On souhaite que, par défaut, un `Book` soit instancié avec un auteur « `UNKNOWN` », un nombre de pages égal à 5 et un titre « `EMPTY` ». En plus de cela, on veut pouvoir instancier un livre en spécifiant explicitement un auteur et un titre. En outre, le nombre de pages des livres que l'on souhaite manipuler ne doit pas être supérieur à 100 pages.

3. En POO, comment instancie-t-on une classe ?
4. Proposez une implantation permettant de répondre aux besoins d'instanciation énoncés.
5. Proposez une implantation de la classe `Book`. En particulier, il doit être possible de récupérer le titre, l'auteur et le nombre de pages d'un livre, mais une fois un livre créé, il ne peut pas être modifié.

En `Java`, il est possible de définir une méthode `main` au sein d'une classe, qui a la particularité de servir de point d'entrée pour l'exécution de la classe. Dit autrement, on peut intégrer un programme principal au sein d'une classe et exécuter cette classe comme s'il s'agissait d'un programme, dont le point de départ est la méthode `main`. Par exemple, pour définir cette méthode dans la classe `Book`, on écrit :

```

public class Book {
    // ...
    public static void main(String[] args) {
        // Le code de notre programme
    }
}

```

En supposant que la classe `Book` soit entièrement définie, écrivez un programme principal dans `Book` dans lequel :

5. vous créez un livre par défaut ;
6. vous créez un livre à partir d'un titre et d'un auteur choisis.

2 Personne

Toujours pour la bibliothèque, elle nous informe que, lorsqu'un livre est emprunté, la personne est identifiée par un nombre unique, ainsi que par son nom et son prénom.

1. De quels attributs avez-vous besoin pour représenter une classe `Person` ? Pour chacun, précisez leur type.
2. Comment peut-on garantir que l'identifiant attribué à chaque personne soit unique ?
3. Peut-on instancier une personne à partir de son identifiant, de son nom et de son prénom ? Justifiez.

La bibliothèque nous informe qu'une personne peut réserver au plus un seul livre. Elle nous indique également que la durée de réservation est gérée par un autre logiciel. Il n'est donc pas nécessaire pour nous de la prendre en compte. De plus, elle ne maintient pas d'historique des personnes ayant emprunté un livre. Ainsi, toute personne existante dans le logiciel a nécessairement emprunté un livre.

4. Comment peut-on modifier notre classe `Book` pour ajouter la possibilité qu'un livre soit emprunté ?
5. Comment peut-on représenter le fait qu'une personne ait emprunté un livre en particulier ?
6. De quels constructeurs a-t-on besoin dans la classe `Person` ?
7. Est-il nécessaire de pouvoir modifier le livre emprunté par une personne ? Justifiez.
8. Proposez une implantation de la classe `Person`.

3 Des livres et des personnes

L'objectif de cet exercice est de comprendre la notion de visibilité des variables et des méthodes, et plus spécifiquement les visibilités `public` et `private`. Pour cela, on souhaite écrire un programme principal (c'est-à-dire une méthode `main`) dans la classe `Personne`. Dans ce programme :

1. instanciez deux livres `book1` et `book2` ;
2. instanciez deux personnes `person1` et `person2`.

Désormais, on suppose que la classe `Book` dispose des éléments suivants :

```

public class Book {
    private int reference;
    public int prix;
    // ...
    public void afficherLivre() {
        // Affiche un livre
    }
    private void lireLivre() {

```

```
    // Lit un livre
  }
}
```

4. Peut-on appeler la méthode `afficherLivre` sur *book1* et *book2* ? Justifiez.
5. Peut-on appeler la méthode `lireLivre` sur *book1* et *book2* ? Justifiez.
6. Peut-on affecter un `prix` négatif à un livre ? Si oui, que peut-on changer pour garantir qu'un prix soit modifiable ET qu'il soit toujours positif ?
7. Peut-on directement modifier la `reference` d'un livre ? Justifiez.

Pour la suite de cet exercice, on souhaite désormais écrire un programme principal dans la classe

`Book`. Dans ce programme :

8. Instanciez deux livres `book1` et `book2` ;
9. Instanciez deux personnes `person1` et `person2` ;
10. Peut-on appeler la méthode `afficherLivre` sur `book1` et `book2` ? Justifiez.
11. Peut-on appeler la méthode `lireLivre` sur *book1* et *book2* ? Justifiez.
12. Peut-on modifier directement la `reference` d'un livre ? Justifiez.