



[Update readme.md](#)

[Jonas Solhaug Kaad](#) authored 2 months ago



Forked from an inaccessible project.

**readme.md** 3.07 KiB

## Practice Assignment 3

### Tasks - Collections Framework & JavaFX

Assume, you are a Software Engineering/Software Technology student. In the Advanced Object-Oriented Programming course, you are asked to model and implement a Book, which is characterised by different properties. You are also asked to create a GUI, which is used to insert/retrieve book information for later use.

Supplied Code:

- Book.java
- BookType.java

`BookType.java` is an Enum containing 4 different book types:

- Audiobook, eBook, Hardcover & Paperback

### Task 1 - Creating the Book Class

1. In the `Book.java` class declare the following variables:

- `String name`
- `String author`
- `BookType bookType`

*(Remember to apply encapsulation)*

2. Create a constructor to initialize these variables.

3. Override the `toString()` method such that the output looks as following:

The Great Gatsby is written by Francis Scott Fitzgerald and is a HARDCOVER

To test the `toString()` method you can uncomment the code in the `main()` method of `Book`.

### Task 2 - Creating a HashMap

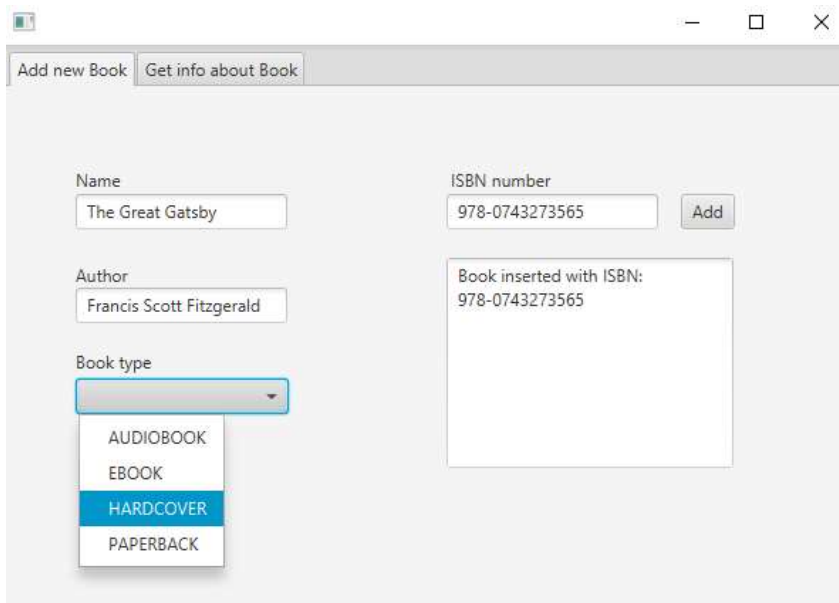
1. In the `PrimaryController.java` declare a `HashMap` to hold a reference between a Book and its ISBN.

1. *(ISBN stands for "International Standard Book Number". It is a unique identifier that every book has. E.g. **978-0743273565**)*

2. Initialize the `HashMap` in the `initialize()` method.

*Hint: (Use `HashMap<String, Book>`)*

## Task 3 - JavaFX

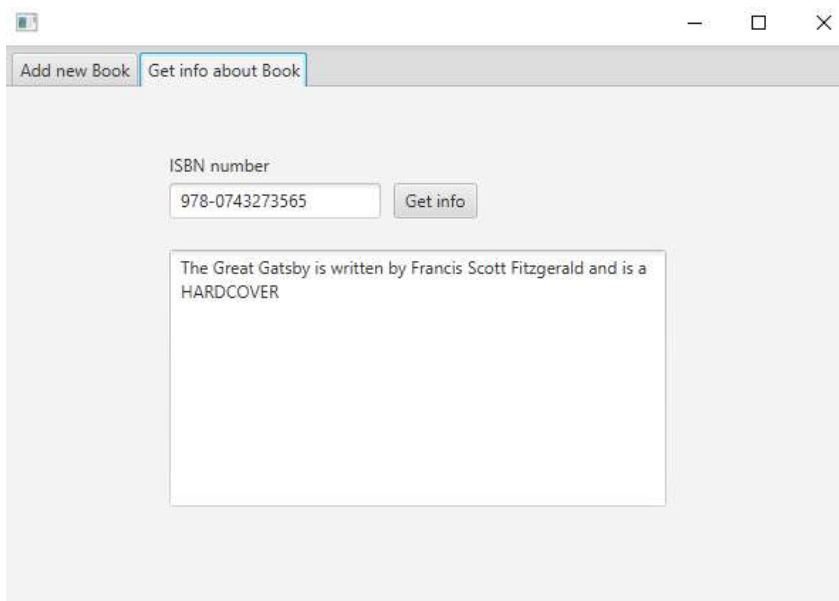


Open the `primary.fxml` file located in resources.

In the first tab create the following elements:

- 4 Labels , one for each of the inputs (i.e. name, author, book type & ISBN)
- 3 `TextField` , one for the name, one for the author and one for the ISBN
- A `ChoiceBox` for picking the book type
- A `TextArea` to display the newly added ISBN
- A `Button` for calling the method to add a new Book to the `HashMap`

*Hint: When declaring the `Choicebox` , use the type `BookType` ( `ChoiceBox<BookType>` )*



In the second tab create the following elements:

- A `Label` and `TextField` for the ISBN that should be looked up
- A `TextArea` to display the information about the Book
- A `Button` for calling the method to get a Book from the `HashMap`

## Task 4 - Implementing the functionality

Open the `PrimaryController.java` file.

1. In the `initialize()` method fill the `ChoiceBox` with the elements from `BookType` .

*Hint: Use `.getItems().addAll(BookType.values())` on the declared `ChoiceBox` variable to add all elements from `BookType`*

2. Implement a method/ActionHandler on the `Add` -button such that:

- A new `Book` is created from the `Name` , `Author` and `BookType` fields.
- The new `Book` is added to the `HashMap` using the `ISBN` as a key.
- The `ISBN` is printed out in the `TextArea` below.

*(Remember to check if all fields are valid, before adding a new Book to the HashMap )*

3. Implement a method/ActionHandler on the `Get info` -button that:

- Looks up the `ISBN` in the `HashMap` and retrieves the `Book`.
- If the `ISBN` is found, it prints out the information about the `Book` in the `TextArea` below.

```
1 package vop;
2
3 public class Book {
4     String bookname;
5     String author;
6     BookType booktype;
7
8     public Book(String bookname, String author,
9 BookType booktype){
10         this.bookname = bookname;
11         this.author = author;
12         this.booktype = booktype;
13     }
14
15
16     @Override
17     public String toString() {
18         return "The Name of the Book :" + bookname +
19         "\n"+"Author of the book :" + author + "\n"+ "Which
20 Booktype is it :" + booktype;
21     }
22
23     public static void main(String[] args) {
24         // For testing the Book Class
25
26         Book b = new Book("The Great Gatsby", "
27 Francis Scott Fitzgerald", BookType.HARDCOVER);
28         System.out.println(b.toString());
29     }
30 }
```

Add new Book

Get info about Book

Name

ISBN-Number

ADD

Author

Booktype

Ovenpå, kan det ses at jeg har lavet de forskellige komponenter for JavaFX i SceneBuilder. I kan ignorere størrelsen på scenen, det vigtigste er bare at indsætte de rigtige knapper, figurer og labeler med tekst og id.

```
1 package vop;
2
3 import javafx.event.ActionEvent;
4 import javafx.fxml.FXML;
5 import javafx.scene.control.Button;
6 import javafx.scene.control.ChoiceBox;
7 import javafx.scene.control.MenuButton;
8 import javafx.scene.control.TextField;
9 import javafx.scene.input.MouseEvent;
10
11 import java.util.HashMap;
12
13 public class PrimaryController {
14
15     @FXML
16     private Button add;
17
18     @FXML
19     private TextField author;
20
21     @FXML
22     private ChoiceBox<BookType> choiceBox;
23
24     @FXML
25     private TextField insert;
26
27     @FXML
28     private TextField isbn;
29
30     @FXML
31     private TextField name;
32
33     @FXML
34     private Button newadd;
35
36     @FXML
37     private TextField text;
38
39
40
41
```

```
42     @FXML
43     public void initialize(){
44         HashMap<String, BookType> hashmap = new
HashMap<>();
45         choiceBox.getItems().addAll(BookType.values
());
46     }
47
48     @FXML
49     public void addButtonClicked(ActionEvent event){
50         String name = name.getText();
51         String author = author.getText();
52         BokType bookType = choiceBox.getValue();
53
54         if(name.isEmpty() || author.isEmpty() ||
bookType == null){
55             text.setText("Please insert all the
values");
56             return;
57         }
58         Book newBook = new Book(name, author,
bookType);
59         hashmap.put(newBook.getISBN, newBook);
60
61         text.setText("The new book with the following
ISBN-number has been added" + newBook.getISBN);
62     }
63 }
```