Library Project: Shelf.java

In the project, you will develop four classes called Book, Reader, **Shelf**, and Library. These classes will be used to store, manipulate, and analyze information about the objects stored in a library.

This details the creation of the Shelf.java class. The UML is listed below.

All the fields have getters and setters. It is also important to note that the class has a hashCode() method and an equals() method.

Field Details

Constant Fields

SHELF_NUMBER_
SUBJECT_

These fields will hold an integer that will be used to index the values shown below. The values will be read from a String array that results from using the split() method on the comma separated Strings shown below.

Library00.csv (shelf values listed in bold)

1,sci-fi
2,education
3,Adventure

HashMap<Book, Integer> books

This holds an instance of each book, and the count of that book that is currently on the shelf.
This means if the shelf normally holds a copy of a book, but they are all checked out, the book will be in the list but will have a count of 0.

Method Details

Getters and Setters

Getters and setters are standard getter and setter methods.

equals() and hashCode()

Equals and hashCode() both use the shelfNumber and subject fields. The HashMap is NOT used in either equals() or hashCode()

toString()

Returns a string that looks like:
2 : education

Where **2** is the shelfNumber field and **education** is the subject.

getBookCount(Book book)

This returns the count of the book parameter stored on the shelf. If the book is not stored on the shelf it should return a -1

addBook(Book book)

Adds the parameter 'book' to the list of books stored on the shelf
If the book already exists on the shelf, the count should be incremented, and a **SUCCESS** code should be returned.
If the book does NOT exist on the shelf, and the subject matches, add the book to the shelf, set the count of the book to 1, and return a **SUCCESS** Code.
If the book does NOT exist on the shelf, and the subject DOES NOT match, return a **SHELF_SUBJECT_MISMATCH_ERROR** Code.

If the book is successfully added, print a message that displays the toString for the Book, concatenated with " added to shelf " concatenated with the toString of the current Shelf.

removeBook(Book book)

If the parameter book is not present in the books hashMap, return a **BOOK_NOT_IN_INVENTORY_ERROR** Code and print a message like
`Hitchhikers Guide To the Galaxy is not on shelf sci-fi`

If the parameter book IS present in the books hashMap but has a count of 0, return a **BOOK_NOT_IN_INVENTORY_ERROR** Code and print a message like
`No copies of Hitchhikers Guide To the Galaxy remain on shelf sci-fi`

If the parameter book is in the books hashMap and the count is greater than 0 decrement the count of books in the hashMap, return a **SUCCESS** Code and print a message like
`Hitchhikers Guide To the Galaxy successfully removed from shelf sci-fi`

listBooks()

Returns a String listing all of the books on the shelf. The listing of books should match the following (Each of the following is a separate shelf):

```
2 books on shelf: 2 : education
Headfirst Java by Grady Booch ISBN:1337 2


1 book on shelf: 3 : Adventure
Count of Monte Cristo by Alexandre Dumas ISBN:5297 1

3 books on shelf: 1 : sci-fi
Hitchhikers Guide To the Galaxy by Douglas Adams ISBN:42-w-87 2
Dune by Frank Herbert ISBN:34-w-34 1
```

Testing Shelf

Constructors, getters, setters, equals

These should be tested in the same way similar methods were tested in the previous assignments

addBook(Book book)

- Create a Book that would be stored on the shelf (i.e. the subjects match)
    - Add it to the shelf and check the return Code. It should equal SUCCESS.
    - Check the count (this can be checked using the getters from above)
- Add another copy of the book, check the code, check the count.
- Create a Book that WOULD NOT be stored on the shelf (i.e. the subjects do NOT match)
    - Add the non-matching Book to the shelf and check the return code and the count.

removeBook(Book book)

- Create a Book that DOES NOT exist on the Shelf
    - Remove the Book and check the code
- Use addBook() to add a Book that does belong, check the count, check the Code.
    - Remove the book, check the count has decremented and the correct Code has been returned
- Remove ALL copies of a Book and check the return code and the count.

getBookCount(Book book)

- Create a Book, add a random number of copies (Use the Random object)
- Check the count matches.
- Remove 1 and check the count
- Remove the rest and check the count
- Check the count of a Book that is NOT on the shelf

listBooks()

Like a toString this is tricky to test.
The only real way to test this is to add a book (e.g. Count of Monte Cristo) and test that 'list book' returns a String that is equal to the one we would expect.
Do your best.

UML Diagram of Reader.java