Book Inventory Program

It’s hard work keeping track of all of those books on your bookshelf in your office, right? Right. Now that we know some basic java, we can begin to create a database management system that can keep track of it all for you!

All jokes aside, here are the **basic** things I’d like to see the program be able to do:

1. Via’ a command line interface, the ability for a user to create a new ‘database’ of books
2. Allow a user to add a book to a certain ‘database’
3. Allow a user to remove a book from a certain ‘database’
4. Print the contents of a ‘database’
5. Switch databases (aka: the ability to handle multiple ‘databases’)
6. Perform a basic text search for a book in a ‘database’

Now, you might have wondered ‘will, why are all of the databases in quotes?’ Well, that’s because you’re not going to be using a full blown database (a la: MySQL, Postgres, etc.) because that’s more complexity than this program needs. Anytime I mention a database, I simply mean some sort of text file storage, so that you’ll get a little experience reading/writing to and from a file.

What are some ‘fun’ expansions you can do if you just rip through this project?

1. Add extra categorization to the inventory system
2. Hook it up to an **actual** database!
3. Make a real GUI for it
4. Make the inventory system not specific to books, make it inventory any category of item!
5. Make it pick a book at random for some nice daily reading
6. Keep a record of who added/removed a book and when
7. Implement a basic user login / user access system.

If you need more than that for a 1-week program… damn! But let me know, there are some more fun expansions we can do with it if you want to take it even further (the above would make it borderline useful for real world situations).

Here are the basic classes I would expect which would be necessary for the basic functionality:

1. A ‘Main’ class (a la: Main.java) that will handle all of the user input/the primary loop that
2. A “DatabaseManager’ class that handles all of the interaction with your file database (aka: write a new book to the database, get records, etc.) the methods I would expect to see would have signatures akin to the following:
   1. Public Boolean writeNewBookToDB(Book book, String dbID){}
   2. Public Boolean getBookFromDB(String bookID, String dbID){}
   3. Public List<Book> get all books from DB(Strign dbID){}
   4. Public Boolean removeBookFromDB(String bookID, String dbID){}
   5. Public boolean searchForBookInDB(String searchString, String dbID){}
3. A ‘Book’ class which would essentially just be a POJO (Plain Old Java Object). It would have fields such as:
   1. ISBN
   2. Name
   3. Category
   4. Edition
   5. Author
   6. ???

So, as you can see, nothing too elaborate for this program. Those are hardly all the methods you may or may not need, and I imagine the Book POJO can be more/less detailed depending on what you’re trying to save information wise. I really encourage you to be as creative/exploratory as you can with this project. The future projects are going to jump in complexity FAST.

Some useful resources to look at that will help you with this program:

* <https://docs.oracle.com/javase/tutorial/essential/io/cl.html> << teaches you about how to get input from users on the command line
* <https://docs.oracle.com/javase/tutorial/essential/io/fileio.html> << teaches you how to interact with files in Java (this is java 8 specific, the latest version, but trust me, it’s waaaaaaaaaaaaaaaaaaaaaaaaaaaaaaay better than it used to be in previous java versions)
* <https://docs.oracle.com/javase/tutorial/java/generics/index.html> << this teaches about java ‘generics’, a hugely important concept in any non-trivial program.
* <https://docs.oracle.com/javase/tutorial/java/package/> << teaches you about ‘packages’ and what they are/how to use them/how to organize your code, etc.
* Also, like I mentioned before, if you haven’t worked on the Java CodeAcademy, I highly HIGHLY recommend it.