======================================================================

Assignment Overview

======================================================================

In this assignment, you will implement a database for a media library. This

library stores three types of media items:

- books

- music albums

- movies

You will start with the following six Java source files:

- MyLibrary.java -- make no changes to this code

- Library.java

- Item.java -- which has the following subclasses:

- Book.java

- MusicAlbum.java

- Movie.java

Library.java, Item.java, Book.java, MusicAlbum.java, and Movie.java are in a

Java package named library. In the project folder, thay are contained in a

subfolder with the same name as the package.

Library.java contains the methods that are used by MyLibrary.java. As

provided, Library.java contains a stub implementation for each method.

You should not add new .java files to this project, and you should not remove

any of the provided .java files from this project.

======================================================================

Assignment Implementation Details

======================================================================

Your task for this assignment is to write code for the method stubs in

Library.java to implement the database functionality. In addition, you will

need to add code to Item.java and its subclasses to support the Library class

as it stores and retrieves data about the various items. Your code should

exhibit high cohesion -- the code in each class should do things that match

what the physical item corresponding to that class would do in the real

world.

Once you have properly implemented the database functionality, your code's

output should be identical to the contents of one of the provided text output

file (expected.txt). This will require all of your output lists (including

keywords, band members, movie casts) to be sorted alphabetically. Lists of

Items should be sorted by title.

Make sure to do all of your printing from the appropriate place, as defined

by object-oriented design. This would include your items' toString() methods.

As always, you should use the Java Collections Framework where appropriate.

Also, for this assignment you should code for high performance, meaning fast

access. This means that you should choose components of the Collections

Framework that support the highest possible speed consistent with the tasks

your code needs to perform. To put this another way, do not do linear searches

when a faster method is available in the Collections Framework. For example,

the set of items for a particular keyword should not be collected by a linear

search through all of the items contained in the library. The requirement to

look things up quickly will mean that you will need to have multiple data

structures to store information.

======================================================================

JAR Files

======================================================================

Once you're ready to submit your work, you will need to create a jar

file containing your code. Make sure that the jar file contains your java

files.

======================================================================

Testing Your Code

======================================================================

When your program is working correctly it will generate the exact same output

as in the expected.txt file.

======================================================================

Assignment Specific Grading Items

======================================================================

The following items are specific grading items you should pay attention to

(see the general course information about how programs are graded for more

information).

- The program must avoid using linear searches for any of the lookups. Notice

that there are lookups by author for books, lookups by keyword for all

items, etc. None of those should do a linear search.

- The program must put the correct data in the correct classes. For example,

don't put a data member for movie cast members in the book class. Also,

don't put data items in the base class (Item) that don't apply to all

subclasses.

- The program must use inheritance as it is provided in the program starting

files. That is, don't just put all of the implementation in the Item class

and ignore the others.

- The program must not use a "media type" field of any sort anywhere to

detemine what type of item you have.