Preparing People to Lead Extraordinary Lives

LOYOLA
UNIVERSITY
CHICAGO

DEPARTMENT OF
Computer Science

Comp 271 - Assignment
Book and BookList

As in the course syllabus, late assignments will not be accepted unless you have a completed the process for using a late pass before the due date.

1. Submit

The zipped project file in Sakai assignment following the correct file naming convention.

2. Format

Java program using proper programming style and with appropriate comments in the code.

3. Content

Summary: Write a complete Java project using correct Object Oriented Programming techniques. The purpose of this assignment is to evaluate your knowledge of and ability to write a basic OO program.

Most Important: Use good OOP techniques! Implement proper classes with good encapsulation. Use public and private appropriately. Use standard naming and style conventions. Make a high quality, well structured, readable program.

Write the following program as a Java project with three .java files. The three classes are Book, BookList, and TestBooks (the latter one includes main). Use these names for your classes.

Book Class: represents a book and can be online, digital, or hardcopy. Books have a title, author, number of pages, and a publication year (or year of last revision if online). Required items (beyond the basics to set up a properly structured class):

1. A constructor that takes input of title, author, publication year, number of pages, as two Strings and two ints

2. A getPages method that returns the number of pages in the book at this time.

3. A toString() method that returns a String suitable for printing out the key information about the book. You decide the format but it should all be one line when the string is printed. Be sure to include the number of pages in the book in the returned string.

4. An updateBook method that works with the existing book object to change its publication year and number of pages. Get the new values for publication year and pages as parameters.

BookList Class: used to keep track of some number of books. A book can be on zero, one, or more lists at any time. Each BookList object has a name of the list and knows or calculates the number of books in the list. A book list can look up a Book object and see if it is on the list. A book list can look for any Book on the list by a given author. Required items:

1. A constructor that takes input of a list name and a maximum number of books to be on the list as a String and an int. You can ignore the maximum number of items if you don't need it in your program but still allow it to be specified in the constructor call. You can decide how you wish to store the books in the list.

2. A getListName accessor that returns the name of the current list.

3. A addBook method that takes as input a Book object and adds it to the list. You do not need to worry about books that are possibly duplicated in the list. Each one counts as a book on the list. Think of it as keeping track of how many times I have read this book for example.

4. A numberOfBooks method that returns as an int the actual number of books currently in the list (not the maximum number possible).

5. A isOnList method that returns a boolean to be true if the Book object parameter is currently on the list, false otherwise.

6. A findByAuthor method that returns a Book object that is in the list with the same author given as String input parameter to the method. If there is more than one book by that author on the list, anyone Book can be returned. If the author does not have a book on the list return null. Be sure you can match author names without regard to upper and lower case.

7. A toString method that returns a single String with information on all the books now on the list, one book per line if that String is printed.

8. A totalPages method that returns an int which is the sum of all the pages in the books on the list at this time.

A TestBooks class the includes main and uses the other two classes to test your work. Requirements:

1. After you set up your main program, copy into it the mainStart file attached. You CAN NOT CHANGE anything in this code, add it as is to your main. It will create several objects for your testing including two BookLists. Include the comments at the start and end of this code; put all your code outside of the mainStart text.

2. Print out the current contents of both lists with a Title or subject line at the start and a blank line between the two lists.

3. Add two books to each list with suitable code in main. Make up two new books on your own and add each one to both lists.

4. Update the two books by Lewis Carroll giving them new dates and numbers of pages.

5. Check to see if the book object "Alice in Wonderland" is in either list; if so, say you found it in that list and give the number of pages in the book, if not, say it is not in the list.

6. Check to see if any book by Edgser Dijkstra is in the wishList list and print out the results.

7. Print out a title line with the name of each list, the number of books in the list, and then list all the books. Do this for both lists of books.

8. When you are finished add a comment at the top of main, give an estimate of the number of hours you worked on the entire project. List clearly (you can use a format such as "Book – 1,2,4")all those numbered items in these lists that you believe you implemented correctly and fully (there are 4 items for Boo, and 8 each for BookList and TestBooks.)

Test your program fully and carefully. Make sure the output from the program looks good and is easy to follow.

For items you are not sure how to do, keep a list of questions. Post them in Sakai Forum or bring them to class to discuss.

4. Grading

100 points total. 35 points for each of Book and BookList class. 15 points for main. No points if will not compile. -5 points for each missing class feature not fully defined or not well structured. -3 to -5 for logic errors and missing requirements depending on severity. Up to -10 for poor programming style.

Assignment BookandBookListEvaluation
Version 1 (java edition) Jan 2017

//end assignment