CSC 143 PROGRAMMING ASSIGNMENT 4

For 25 points

Due on or before: see our Canvas class site

No due date extensions

Book Cataloging System

Write a program to manage a Book Catalog. The catalog would contain **a collection of books**, with following fields of information: **book code** (unique for each book; must be valid – see ISBN validation rules in this assignment for more info), **author's last name, author's first name, book title**, **year of publication** and **price**. The book catalog should use a file for storing books.

The information in the file uses tabs to separate each of the field of a book, with one book per line. Refer to sample file: **booklist.txt** for more details.

Usage Requirements:

1. At the start, the program should load the catalog from file (Books.txt from current directory), if it doesn't exist, create a new catalog.
2. The program should allow the user to **add a new book** to the catalog. The program should ask for the required information and then add the book to the catalog, only if same book code doesn't already exist in the catalog and if the book is a valid. Book is considered valid, if the book code is valid, name and title are valid (name can't start with numbers, name/title can't be empty, price is not negative or zero, year cannot be greater than 2013)
3. The program should allow user to **find a book** in the catalog, given one of the following: author **first name** or **last name** or the **book code**. If the book is found, all the information about the book should be displayed. If the book is not found, an appropriate message should be displayed.
4. The program should allow user to **delete an existing book** from the catalog.
5. The driver (or client) program should have a **menu driven interface** (console) to allow users to select one of the options to perform the above tasks and repeat them till the program is exited. The program should display appropriate messages to the user for each operation, successful or not.
6. When exiting, the program should ask to **save Yes/No** the book catalo**g to file**, taking in to account any of the above changes, for future reuse.

Design/Implementation Requirements:

1. **Book Class** – represents one book. In addition to accessor (get) and mutator (set) methods for instance variables, this class should not allow instantiation of book objects unless the initialization values are valid including book code. This class also should override the *toString* method.
2. **BookCatalog Class** – represents a collection of books. The data structure used for storing books should be a **linked list of books**. You are required to implement this linked list structure yourself and NOT use any Java API collection, Arrays and Collections types. This class should override *toString* method and provide methods for all book catalog operations as specified by the usage requirements.
3. **BookCatalogClient Class** – This class contains the *main* method. The above classes should be I/O independent to be utilized by console, applet, file based applications. Therefore, for this program, the file save and load operations are implemented by the client class.

Book code validation rules:

An ISBN (International Standard Book Number) identifies a unique publication. An ISBN is ten digits. The first nine digits must be decimal digits (0...9). The tenth digit can be either a decimal digit or the letter X. Three single dashes may be between any of the characters. (i.e., an ISBN number may either have no dashes or exactly three dashes). Also, an ISBN must not begin or end with a dash, and sequential dashes are not allowed.

Some valid example ISBNs:
0-201-88337-6
0-13-117334-0
0821211315 (no dashes is ok)
1-57231-866-X

The last character of an ISBN number is a checksum. The checksum is determined from the first 9 digits; it is computed by taking modulo 11 (the remainder after dividing by 11) of the sum of each digit multiplied by its position in the ISBN. The letter X corresponds to a value of 10.

Here are two ISBNs and the calculations that show how the check sum is determined:

0-201-88337-6:(0*1 + 2*2 + 0*3 + 1*4 + 8*5 + 8*6 + 3*7 + 3*8 + 7*9)mod 11=6
1-57231-866-X:(1*1 + 5*2 + 7*3 + 2*4 + 3*5 + 1*6 + 8*7 + 6*8 + 6*9)mod 11=10(X)
For more info, check out: www.isbn.org.

Some **invalid** ISBNs:
0-201-8A337-6 (bad digit)
0-201-88337-63 (too many digits)
0-201-88-337-6 (too many dashes)
0-201883376 (not enough dashes)
0-201-88337-3 (wrong check sum)
-013-117334-0 (beginning or ending dash)
157231--866-X (sequential dashes)
013-1134-0 (too few digits)

Comment your code including: (1) Header with Course Name, Program Name, File Name, Date, Programmer Name and Program Description, and (2) comments throughout the source code.

Submit the following by using Canvas Online assignment submission:
1. Source-code files for Book, BookCatalog and Driver (*.java),
2. Comprehensive sample runs to demonstrate the key features of your program,
3. Summary of your work – problems/issues with this assignment, any additional things/features that you've thought of etc. Feel free to combine items: 2 & 3 in a single text document,
4. Place all of the above files in a folder, zip the folder and attach when submitting.

Please retain a copy of everything that you submit for your records.