Jason Graves'

Library Management System

Software Development I
CEN-3024C-24667

Main Menu

This program will have a menu where the user will input a number to do the stated option. It is designed to be used by librarians to keep track of a library's books.

1. Add Book(s) From Text
2. Enter Book in manually
3. Remove Book by id number
4. Save book library
5. Load book library
6. List all books in library

The menu will use a Do… While loop to continuously cycle through itself until the user breaks the loop and ends the program. Book will be a custom object with the following attributes, Title, Author, idNumber. These objects will be stored into an array while running, with an option to export which will use a text file with a custom extension stored in the root of where this program lives.

Add Book(s) From Text

Option 1 will be accomplished by reading a text file which should be formatted as shown in the assignment overview. If the text doesn't match this format the program will display an error message and kick the user back to the main menu.

Enter Book in Manually

Option 2 will display a prompt for each book attribute, have the user input each piece of information, and will add the book to the text file with the library's information.

Remove Book by id Number

Option 3 will first display the entire book library using the same code from option 6, and will then prompt the user for which number book they would like to delete from the library. The id number will just be the number the book is in the library, for example the first book would have an id number of "1"

Save Book Library

Option 4 will export the book library currently in the program to a text file with a custom extension.

Load Book library

Option 5 will import a book library from a previously exported file. This will essentially use the same code as option 1, only the file it's reading from will have a different extension.

Print All Books

Option 6 will print every book in the library. It will accomplish this by simply printing the toString for every object in the array the books are stored.

Development Steps

-First I'll make the menu, which should be a simple Do… While loop. Using a Case Switch statement, the menu will guide the program into the proper methods. Each option will be a different method, with the menu running off the main.

-Next I'll create a Book object class with the three attributes, title, author, and id. The title and author will be strings and the id will be an integer. The class will be standard with private attributes and public getters and setters for each of them. There will also be a toString, which will be configured to output a book's information. In order to test this the menu will have an option to simply print out a book object, which will be removed later. This test book will be pre initiated from inside the program, no user input required.

-I will then implement an arraylist comprised of my now confirmed to be working Book objects. It will be initialized when the program starts.

-In order to use my arraylist I will then write a method which stores books from user defined attributes. This will become the code for option 2. Using the Book's setters it will take in the book's title and author from separate prompts from the user. The id will be set to 0 as a default while the object is initialized.

-Then a separate method will then run through the array assigning every book an id number based on its index. For example if it's the 3rd book it's id number will be 3. The id number is the index of the arraylist plus 1. This will be a separate method not built into the above bullet point because I plan to re-run this every time a change is made to the arraylist, for example removing a book.

-To test if the arraylist is working properly, I will then write the code to display all books in a separate method. Using a For loop it will cycle through running the toString for every book. If the list prints out properly then it means that the arraylist is working.

-After this removing a book by id Number should be easy to implement. I will use the .remove command to remove the index of the book to be removed. The index will be the id minus 1. So if a user wants to remove the 3rd book this method will need to remove index 2. After the desired book is removed from the arraylist, this method will then assign new id numbers for each book in the arraylist.

-Next will be to implement the reading from a text file method. I expect to do a lot of googling here, because I don’t remember how this works off the top of my head, however I’ve done similar things in the past and am confident that I’ll be able to figure out how it’s done this time. This method will parse out a book’s id number, title, and author based on the format as outlined in the assignment and add it (or them) to the arraylist, I will then run the method to re-assign id numbers. This method ideally will work with .txt files as well as the custom extension files this program will export. I expect this method to be the hardest obstacle.

-Finally will be the text exporter. This should be as simple as running a loop through the arraylist and writing the output to a text file. I plan to use a custom extension for this file, because that should be easy to do.

Deployment

The project will be uploaded to my GitHub and will be free to use and view by anybody. Complete with a Readme on how to use.