Computer Science 205

Project #2 : *The Great Books Program*

50 Points

Due Date : Friday, October 13th

Objective

The purpose of this assignment is to extend students' familiarity with developing classes and working with arrays of objects, strings, and the selection sort and binary search algorithms.

Assignment Summary

For this second program, you will be setting up a library catalog search program which will allow the user to determine whether a particular book is found in a library's computerized database holdings.

If a book is found, you will provide the complete online book record. If a book is not found, you will indicate that the book was not found by the program. All searches will be book title searches. That is, you will only be setting up a search based on the title field of a record.

Before beginning this project, you should copy over some files from our class directory by typing :

```
cp /pub/digh/CSC205/Prog2/* .
```

To get a sample run, type `Sample`.

Data Structure

The data structure you will be using for this program will be an array of records where each record in the array represents one book and each of the components within the record are the book fields. Feel free to use an `ArrayList` as opposed to a traditional array. If you use a traditional array, you may assume the input size will never exceed 50.

Each book record should have the following five fields set up in this order:

1. Title – a string object
2. Author's Name (Last Name, First Name) – a string object
3. Copyright
4. Price
5. Genre (fiction, nonfiction, drama, poetry) – an enumerated type

It would probably be best to develop a class, say `LibraryBook`, that contains each of these as private attributes and includes public getter methods to access each of them.

Opening the Input File

You will need a method to allow the user to input all book records currently stored in the library's database file. This database file is an external file with a `.dat` extension whose name you will prompt for at run-time.

You should provide the user with a listing of the data files in the current directory they can choose.

Inputting your Book Records

Within your input method, you will also want to input all the fields of your book records. You may assume the format shown in your input file will always be the same.

You will need to use a while not end of file loop rather than a for loop to set up each of your book records since you will not know how many slots there are to fill. After inputting your library data, you will need to call a method to sort it (see below on coding this method).

The total number of records that have now been input and sorted should then be printed to the screen with a prompt such as the following :

```
A total of 4 books have been input and sorted by title.
```

Before leaving this method, you can let the user continue the program at their own pace by using a please hit return to continue.

Sorting

The input data should be sorted using the selection sort algorithm.

You will need to slightly modify the headings of your selection sort code we went over in class so that it includes the number of books as an input parameter. The reason for this is that not all of the slots in our 50-slot array will be used. You will also be sending in an array of records rather than an array of integers.

You will also need to tweak a few other things. Your temporary variable used for swapping will need to be of the book record type. Also, you're not comparing entire array slots, you are comparing title fields within those array slots.

Menu for Program

Following the input and sort, the user should be able to continue searching for book records until they are ready to quit. Call a method that prints a menu such as the following:

```
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
            THE GREAT BOOKS SEARCH PROGRAM
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
1)  Display all book records
2)  Search for a book by Title
3)  Exit Search Program
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
Please Enter Your Choice >
```

This method should return the user's menu selection to the main program which will utilize the user's choice in a `switch` statement. The selection variable should be an integer and will only take on the values of 1 through 3. The user should receive an error message if they enter any integer value other than these three.

Searching

You will have a method which prompts the user for the title they are searching for, and then calls a search method to see if this title is in your array of book records. A binary search is ideal here since we have a sorted array and it will be quicker than a linear search.

If the book title is found, the corresponding book record should be displayed by calling a method to print the record at the found location. Otherwise, they should be informed that the book was not found.

Displaying and Printing Records

You should have a method `displayRecords` that is called to display all your book records. This method should use a `for` loop to call a `printRecord` method that is used to print out a book record by its record number. For example,

```
for (int i = 0;  i < numBooks;  i++)
            printRecord (Books, i);
```

You may also want to add some code to this `for` loop to let the user hit return to continue, and then clear the screen before displaying the next record. You may also want to let them enter a code for the menu, and exit the method via a `return;` if they so desire.

The `printRecord` method will also be called when a book is found by your book search. All fields of the book record including its record number (position within the array) should be displayed with appropriate labels to the user such as seen below.

```
Record #1 :
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
             Title :              Animal Farm
     Author's Name :            Orwell,George
         Copyright :                     1978
             Price :                    10.00
             Genre :                  Fiction
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

Analysis & Design

Before beginning your program, you are required to type up a problem description, the program specification, and your program algorithm.

The problem description should describe the problem at hand, and give a sample input along with a discussion of the output provided by each of the different menu choices. Briefly explain the different sorting and searching algorithms used by the program, and give an example of how they work using an array of five integers.

The program specification should describe exactly what type of data is being input and output as well as the location of the input and output files. The algorithm should describe your plan of action for solving this problem in outline format. All methods you plan on writing should be discussed. Avoid all code references in your algorithm.

Create this file name it `README`. Store it with your source code in your project directory.

Extra Credit and Revision Policy

You will receive at least five extra credit points added to your score out of 50 if you hand in this assignment one class day before the due date. You may also receive extra credit points for adding significant additional features beyond the minimum requirements.

All assignments handed in on or before the due date that you do not receive full credit for in either implementation or documentation are eligible for revision.

Sample Run and Typescript

You should have some test data for your program that you copied over from our class directory. This file is named `library.dat` and contains a total of 15 book records. `vim` the file and look at the data to see what different book titles are being input and sorted.

I encourage you to create your own small input file first with just a couple of books in it to test your program before running it on a large input file.

In your typescript, include a listing of your program using the `cat` command as well as a compilation and sample run of your program. In your sample run, after inputting and sorting your book records, I would like for you to search for at least the following five titles:

- *Odyssey*
- *Animal Farm*
- *Last of the Mohicans*
- *Whisper of the River*
- *Macbeth*

The last title, *Macbeth*, should not be found.