

## CS 116

### Lab Assignment # 4: Re-Implementing the Library Database Using Vector/ArrayList

- **Points: 3**
- **Submission**
  - Deadline: Wednesday 03/09 11:59 PM
  - Submit on Blackboard under assignment “Lab4”. Please make sure that you click the “Submit” button and not just “Save”.
- **Late Submission Policy**
  - You can do a late submission until Friday 03/11 11:59PM with a 5% penalty on the total points for this assignment.
  - After that solutions will be posted and no submission will be accepted
- **Early Submission**
  - You can also get 5% extra point on your score on the assignment for early submission if you submit by Tuesday 03/08 11:59PM.
- **Getting help**
  - From instructor during office hours in SB228 or by email.
  - By seeing one of the TAs during the listed TA office hours in room SB108. (check the course website <http://cs.iit.edu/~jkorah/cs116/>)
  - By visiting the ARC (Academic Resource Center).
- **Academic Dishonesty Policy**
  - Working with a partner: You can work with a partner as assigned by the instructor. Otherwise this should be considered individual work.
    - Even if you are working with a partner, you and your partner are required to make individual submissions.
  - **Please note: In case two submissions are declared identical (and if you are not supposed to work together) the excuse: we worked together, does not hold and both submission will be treated according to ethics rules.**

#### Objectives:

The list below indicates the Java concepts needed for this exercise.

1. Calling static and non-static methods from a static method.
2. Reading data from a File.
3. Using while loops.

4. Develop the logical algorithm that will accomplish the required task.
5. Using methods of the String class
6. Using Vector/ArrayList
7. Packaging a class
8. Using enumerations.
9. Binary Search
10. Sorting algorithm

#### **PROGRAMMING TASK : Reimplementing the library database using Vector/ArrayList class**

- Please read all steps carefully first, and then start coding.
- You can use the solutions to practice exercises and any other help including lecture presentations and your text book.
- The current directory where your source code files are located should be a folder named <LastName>-<FirstName>-Lab4.

**In a nutshell, this programming task is about re-implementing the library database, using Java Vector or ArrayList class to store (and also sort and search) the BookRecord objects, with all the functionality specified in the lab assignments 2 & 3.**

**NOTE: You solutions for Lab 4 should have ALL the functionality that was specified in Lab 2 & 3.**

**NOTE: You can reuse whatever code you need from your solutions for the lab assignments 2 & 3**

**You can also use the solutions for Lab 2& 3 posted on the course website**

Programming Task specification:

1. **Implement the service class** BookRecord.java is the service class of the java application. As mentioned before, the service class SHOULD contain, at the minimum, class attributes variables that represent the following information:

- a. A unique record id for the book.
- b. A static variable to provide unique record ids for the books (as you have done in previous assignments).
- c. Title of the book
- d. A list of authors. You should have an array to store this as there may be multiple authors.
- e. Genre of the book: Each book record has only one genre value. This will be an enumeration datatype called

BookGenre which has the following values {GENRE\_HISTORY, GENRE\_SCIENCE, GENRE\_ENGINEERING, GENRE\_LITERATURE} . You will implement the enumeration class in a separate java source file.

f. A string variable to store the “Tag” information for the books. A tag is a UNIQUE 8 digit alpha numeric code (combination of letters and numbers) for each book. This information is provided in the input text file.

g. An int variable to store the length (number of pages) in the book. This information is provided in the input text file.

h. equals(): You will also implement an equals() class method that will compare the instance variables of the two objects . Specifically, you will compare the values of the title, list of authors, genre of two objects, tag and page length. DO NOT compare the record id.

i. toString(): You will print out details of the object specifically the title, authors (authors separated by , or space), genre, the tag and page length information.

j. Accessor and mutator methods: implement appropriate accessor and mutator methods for the class attributes.

You should implement the service class in a package called library.service.classes.

2. Implement the Enumeration class: Implement an enumeration class called BookGenre which has the following values {GENRE\_HISTORY, GENRE\_SCIENCE, GENRE\_ENGINEERING, GENRE\_LITERATURE}. The enumeration class is implemented in a java file called BookGenre.java in package library.service.classes.

3. Implement the client class: The client class should have the following attributes

a. An Vector or ArrayList of BookRecord objects: You will use a Vector or ArrayList (upto you) to store the details of the books as a collection of BookRecord objects. Since you are using Vector/ArrayList, you don’t have to worry about resizing.

b. main method: The main class will take the following user parameter:

- Name of the text file containing the book records.

The main function will also support the following key functionalities:

- Reading the book records from the textfile [0.5 pt]: The file text “books.txt”, provided along with the assignment has lines with the following format:

title:genre:author-1,author-2,....author-m:tag:no-of-pages

Read each line, create BookRecord object for each book and insert in the Vector/ArrayList.

- Duplicate records [0.5 pt]: Note that the book records in your Vector/ArrayList should be unique. However the text file may contain duplicate records. Therefore you will need to check for duplicated records before you insert them in the Vector/ArrayList. Every time you find a duplicate, print out the record. Two records are said to be identical if their title, authors and genre are the same. Assume that the authors for a particular title will always be provided in the input text file in the same order.
- Implement a class method called sortString() with the following signature:[0.75pt]

```
public Vector< BookRecord > sortPages(Vector< BookRecord > myArray)
or
public ArrayList< BookRecord > sortPages(arrayList< BookRecord > myArray)
```

The method takes the Vector/ArrayList of BookRecord object, you wish to sort, as the and also the number of objects in the array. You will implement the selection sort algorithm in this method to sort the BookRecord object Vector/ArrayList. Once the books.txt file has been read and the Vector/ArrayList of BookRecord objects has been created, you will call the sortString() method. This method will use selection sorting algorithm to re-arrange the BookRecord objects in the Vector/ArrayList in a lexicographic increasing order with respect to their tag values. Note: DO NOT convert your Vector/ArrayList into array and then apply selection sort.

- Implement a second class method called sortPages() with the following signature[0.5pt]

```
public Vector<BookRecord> sortPages(Vector<BookRecord> myArray)
```

or

```
public ArrayList<BookRecord> sortPages(ArrayList<BookRecord> myArray)
```

This method will take in the Vector/ArrayList of BookRecord objects and sort the objects in an increasing order with respect to their page length values. You will use a selection sort algorithm to accomplish this. The purpose for implementing this method is explained below. Again, DO NOT convert your Vector/ArrayList into an array and then apply selection sort.

- Implement a third class method called searchTag() with the following signature [0.75pt]

```
public void searchTag(String tag)
```

This method will take in a tag value as a parameter. It should search the BookRecord objects in the Vector/ArrayList, using Binary Search, for a particular tag value and display the record (call the toString() method of the record). If such a record does not exist in the array, the method should print “No match found” and return. **You should implement binary search to find the object in the Vector/ArrayList. If you implement the sequential search algorithm, you will NOT get partial points!** Again, DO NOT convert your Vector/ArrayList into array to use Binary Search.

- Interactive search: After reading the records from the text file, you should prompt the user with the following options:

**Select an option:**

**Type "S" to list books of a genre**

**Type "P" to print out all the book records**

**Type "T" to search for a record with a specific tag**

**Type "Q" to Quit**

- Options “S”: When the user selects this option, the set of genres are listed and the user is prompted to type in a particular genre. The books belonging to the selected genre are listed. You will display the books of the selected genre in the increasing order of their page length. This means that books with shortest length are

listed first, followed by the second sorted book and so on. You will use the sortPages() method to implement this feature.

- If the user types in the option “P”, you will list all the book records in the Vector/ArrayList. Since you sorted the book records in an increasing (lexicographic) order, the book records should also be displayed in that order.
- Option “T”: This options allows the user to search for a book record that has a particular tag value. Remember the tag value is unique. When a user selects this option, your code should prompt the user to enter the tag value that he/she is looking for. Once the user enters the search term, your code will call the searchTag() method which will then display the results.
- If the user types in the option “Q”, you will quit the program. Remember you will continue to display the menu options to the user and execute the selections, until he/she selects “Q” to exit the program.

#### Submission instructions

- In your submission you must include
  - a. The source code files and the compiled files for the program.
- Zip all files and name the zip file using your last name followed by your first name followed by the name of the assignment  
i.e. Doe\_Jane\_Lab4.zip
- Upload the file on assignment folder: Lab4 on Blackboard.

#### **Sample Output:**

```
>java library.client.classes.library books.txt
```

Found a duplicate

=====

Tag:MNBV3456

Title:English landscaping and literature, 1660-1840

Genre: GENRE\_LITERATURE

Authors: E. Malins

No. of Pages: 980

=====

Found a duplicate

=====

Tag:NJKG7456

Title:Nikola Tesla

Genre: GENRE\_HISTORY

Authors: Sean Patrick

No. of Pages: 987

=====

Found a duplicate

=====

Tag:HJKG2342

Title:Microfabricated microneedles, a novel approach to transdermal drug deliver

y

Genre: GENRE\_ENGINEERING  
Authors: S. Henry D. V. McAllister M. G. Allen  
No. of Pages: 378  
=====

Select an option:  
Type "S" to list books of a genre  
Type "P" to print out all the book records  
Type "T" to search for a record with a specific tag  
Type "Q" to Quit

S  
Type a genre. The genres are:

GENRE\_HISTORY  
GENRE\_SCIENCE  
GENRE\_ENGINEERING  
GENRE\_LITERATURE  
GENRE\_SCIENCE  
=====

Record No:10015  
Tag:KJHG8909  
Title:Gene Ontology  
Genre: GENRE\_SCIENCE  
Authors: M. Ashburner C. A. Ball J. A. Blake D. Botstein H. Butler  
No. of Pages: 90  
=====

=====

Record No:10010  
Tag:HGH8909  
Title:Free radicals in biology and medicine  
Genre: GENRE\_SCIENCE  
Authors: B. Halliwell J. M. C. Gutteridge  
No. of Pages: 234  
=====

=====

Record No:10008  
Tag:BGHF8976  
Title:The comparative method in evolutionary biology  
Genre: GENRE\_SCIENCE  
Authors: P. H. Harvey M. D. Pagel  
No. of Pages: 234  
=====

=====

Record No:10011

Tag:LKJH2345  
Title:Electron transfers in chemistry and biology  
Genre: GENRE\_SCIENCE  
Authors: R. A. Marcus N. Sutin  
No. of Pages: 890  
=====

Select an option:

Type "S" to list books of a genre  
Type "P" to print out all the book records  
Type "T" to search for a record with a specific tag  
Type "Q" to Quit

T

Input the tag of the book you want to search for:  
LKJH2345

Found a match  
=====

Record No:10011  
Tag:LKJH2345  
Title:Electron transfers in chemistry and biology  
Genre: GENRE\_SCIENCE  
Authors: R. A. Marcus N. Sutin  
No. of Pages: 890  
=====

Select an option:

Type "S" to list books of a genre  
Type "P" to print out all the book records  
Type "T" to search for a record with a specific tag  
Type "Q" to Quit

T

Input the tag of the book you want to search for:  
LKJH2344

No match found

Select an option:

Type "S" to list books of a genre  
Type "P" to print out all the book records  
Type "T" to search for a record with a specific tag  
Type "Q" to Quit

P

=====

Record No:10000  
Tag:ABEW2345  
Title:Thomas Jefferson and the Tripoli Pirates  
Genre: GENRE\_HISTORY  
Authors: Brian Kilmeade Don Yaeger

No. of Pages: 234

=====

=====

Record No:10008

Tag:BGHF8976

Title:The comparative method in evolutionary biology

Genre: GENRE\_SCIENCE

Authors: P. H. Harvey M. D. Pagel

No. of Pages: 234

=====

=====

Record No:10007

Tag:FDST9878

Title:Climate and atmospheric history of the past 420,000 years

Genre: GENRE\_HISTORY

Authors: J. R. Petit J. Jouzel D. Raynaud N. I. Barkov J. M. Barnola

No. of Pages: 675

=====

=====

Record No:10010

Tag:HGH8909

Title:Free radicals in biology and medicine

Genre: GENRE\_SCIENCE

Authors: B. Halliwell J. M. C. Gutteridge

No. of Pages: 234

=====

=====

Record No:10001

Tag:HGNH4567

Title:Component-oriented programming

Genre: GENRE\_ENGINEERING

Authors: C. Szyperski J. Bosch W. Weck

No. of Pages: 456

=====

=====

Record No:10005

Tag:HJGF7645

Title:A history and theory of informed consent

Genre: GENRE\_HISTORY

Authors: R. R. Faden T. L. Beauchamp N. M. King

No. of Pages: 654



=====

=====

Record No:10002

Tag:HJKG2342

Title:Microfabricated microneedles, a novel approach to transdermal drug delivery

Genre: GENRE\_ENGINEERING

Authors: S. Henry D. V. McAllister M. G. Allen

No. of Pages: 378

=====

=====

Record No:10006

Tag:JHGF9089

Title:The Feminist Companion to Literature in English Women Writers From the Middle Ages to the Present

Genre: GENRE\_LITERATURE

Authors: V. Blain P. Clements I. Grundy

No. of Pages: 767

=====

=====

Record No:10013

Tag:JHKG2343

Title:Device electronics for integrated circuits

Genre: GENRE\_ENGINEERING

Authors: R. S. Muller T. I. Kamins M. Chan P. K. Ko

No. of Pages: 654

=====

=====

Record No:10015

Tag:KJHG8909

Title:Gene Ontology

Genre: GENRE\_SCIENCE

Authors: M. Ashburner C. A. Ball J. A. Blake D. Botstein H. Butler

No. of Pages: 90

=====

=====

Record No:10011

Tag:LKJH2345

Title:Electron transfers in chemistry and biology

Genre: GENRE\_SCIENCE

Authors: R. A. Marcus N. Sutin

No. of Pages: 890

=====

=====

Record No:10004

Tag:MNBV3456

Title:English landscaping and literature, 1660-1840

Genre: GENRE\_LITERATURE

Authors: E. Malins

No. of Pages: 980

=====

=====

Record No:10003

Tag:NJKG7456

Title:Nikola Tesla

Genre: GENRE\_HISTORY

Authors: Sean Patrick

No. of Pages: 987

=====

=====

Record No:10009

Tag:UYHG1223

Title:Human-computer interaction

Genre: GENRE\_ENGINEERING

Authors: J. Preece Y. Rogers H. Sharp D. Benyon S. Holland

No. of Pages: 889

=====

=====

Record No:10014

Tag:YUTY9098

Title:An outline of English literature

Genre: GENRE\_LITERATURE

Authors: G. C. Thornley G. Roberts

No. of Pages: 89

=====

Select an option:

Type "S" to list books of a genre

Type "P" to print out all the book records

Type "T" to search for a record with a specific tag

Type "Q" to Quit

Q

Quitting program