CS 116

Lab Assignment # 2: Implementing a Library Database

- **Points: 2**
- **Submission**
  - **Deadline: Monday 02/15 11:59 PM**
  - **Submit on Blackboard under assignment "Lab2". Please make sure that you click the "Submit" button and not just "Save".**
- **Late Submission Policy**
  - **You can do a late submission until Wednesday 02/17 11:59PM with a 5% penalty**
  - **After that solutions will be posted and no submission will be accepted**
- **Early Submission**
  - **You can also get 5% extra point for early submission if you submit by Sunday 02/14 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. It is considered a review of CS115 concepts with the addition of the new concepts of Enumeration, packaging, array of objects, and array expansion.

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 split method of String class or StringTokenizer.
6. Using Wrapper classes to parse Strings.
7. Saving objects in an array.
8. Packaging a class
9. Using enumerations.
10. Using methods of the String class

THEORY: ENUMERATIONS:[1]

Keep in mind that when we create an enumeration class the data of the enumeration is not String types.
The data are objects of the enumeration class we created.
The enumeration can be in a file on its own and be compiled accordingly including the packaging aspect if there is a package.
The data is accessed by using the name of the enumeration class i.e Suppose we have an enum class called MyEnum and one of the data is DATA1:
MyEnum me=MyEnum.DATA1;
Here the variable me holds the value DATA1.

If we wanted to convert a String to an enum value then we can do it as follows:

Suppose we have the String value "DATA1" which is stored in String type variable called mydata.
We can convert it to an enum type using the enum MyEnum as an example, as follows:
MyEnum me=MyEnum.valueOf(mydata);
The variable me now holds the enum version of the String value DATA1.

THEORY: MAKING ARRAYS SCALABLE

Arrays datatypes are static with a fixed size that needs to be provided during instantiation. However such arrays can be made scalable to store any number of data types by including appropriate array expansion functionality in your code. To expand an array while maintaining its original values, you can follow these steps

1. Instantiate an array with the new size and a temporary name
2. Copy the original elements to the new array
3. Point the original array reference to the new array
4. Assign a *null* value to the temporary array reference

PROGRAMMING TASK : Building a database for a library

- **Please read all steps carefully first, and then start coding.**

[1] Enumeration tutorial provided by George Koutsogiannakis

- **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 Lab2.**

Your programming task is to design and implement a database for a library. The database will contain details of books, such as title, author list and genre, in the library. The service class will store the record of a book. The client class will read the records of a set of books from a text file, and store them as an array of service class objects. The client class will also have functionality to interact with the user, and search the books records as specified below.

Programming Task specification:

1. **Implement the service class** [0.6 pts] 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 a 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. 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 and genre of two objects. DO NOT compare the record id.

g. toString(): You will print out details of the object specifically the title, authors (authors separated by , or space) and genre.

h. 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[0.2 pts]: 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: should have the following attributes

a. An array of BookRecord objects: The client class will store the details of the books as an array of BookRecord objects. **The initial size of the array should be 5.** However your implementation of the array should be scalable, which means that the array implementation should be able to handle any number of books. As you read in more book records from the text file, and when you have completely filled the array, you will need to automatically expand the array to accommodate additional book records. Read the theory section EXPANDING ARRAYS, in the earlier part of the assignment.

b. An integer attribute to represent the number of records in the array. Since the array may not be full, you cannot use the length attribute of the array to figure out the number of book records.

c. main method: The main class will take the following two user parameters:

- Name of the text file containing the book records.
- An integer value called the expansion factor. As you reach the capacity of your object array, you will add additional space equivalent to the expansion factor. For example, if you expansion factor is 5, then every time the array becomes filled, you implementation should increase the size of the array by 5.

The main function will also support two key functionality:

- Reading the book records from the textfile [0.4 pts]: As you read in the book records from the text file, you will need to expand your array. You should implement code that will check if the array has been filled, and if needed, call methods that will expand the array. This additional space is controlled by the expansion factor, an argument provided by the user. Every time you need to expand the array, you will print out the current size and the expanded size.
- Duplicate records [0.4]: Note that the book records in your array 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 array. 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.
- Interactive search [0.4]: 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 "Q" to Quit

  - If the user types in the option S, then you should prompt the user for the specific genre. Something like this –

    Type a genre. The genres are:
    GENRE_HISTORY
    GENRE_SCIENCE
    GENRE_ENGINEERING
    GENRE_LITERATURE

    You can expect the user will correctly type in the genre he/she is interested in. Once the user enters her selection, you will list all the books records of that particular genre.
  - If the user types in the option P, you will list all the book records in the array.
  - If the user types in the option Q, you will quit the program.

Here is a skeleton code for the main program that you can use:

```
public static void main(String []args){
//arg[0]: text file //arg[1]: resize factor
    /* some code */
    try {
        // read the text file
        //read the tokens in each line and create an object of type book record
        //insert into the client class array if no duplicate exists
        //after insertion check if the array is completely filled and needs to be expanded
    } catch(IOException ioe){
        System.out.println("The file can not be read");
```

```
        }

    //User interactive part

        while(true){
            System.out.println("Select an option:");
            System.out.println("Type \"S\" to list books of a genre");
            System.out.println("Type \"P\" to print out all the book recors");
            System.out.println("Type \"Q\" to Quit");

            //get input from the user

            switch (option1) {
                case "S":System.out.println("Type a genre. The genres are:");
                                for (BookGenre d : BookGenre.values()) {
                                        System.out.println(d);
                                }
                                option2=scan.nextLine(); //assume the use will type
in a valid genre
                                //print out records of the selected genre
                                break;

                case "P":   //list out all the records
                                break;

                case "Q":   System.out.println("Quitting program");
                                System.exit(0);

                default:  System.out.println("Wrong option! Try again");
                                break;

            }

        }

}
```

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 first name followed by your last name followed by the name of the assignment

    i.e. Jane_Doe_Lab2.zip

- Upload the file on assignment folder: Lab2 on Blackboard.

Sample Output:

```
>java library.client.classes.library books.txt 2
Resized the array from 5 to 7
Resized the array from 7 to 9
Resized the array from 9 to 11
Resized the array from 11 to 13
Found a duplicate
==================================
Record No:10012
Title:English landscaping and literature, 1660-1840
Genre: GENRE_LITERATURE
Authors:  E. Malins
==================================

Resized the array from 13 to 15
Resized the array from 15 to 17
Found a duplicate
==================================
Record No:10016
Title:Nikola Tesla
Genre: GENRE_HISTORY
Authors:  Sean Patrick
==================================

Found a duplicate
==================================
Record No:10017
Title:Microfabricated microneedles, a novel approach to transdermal drug deliver
y
Genre: GENRE_ENGINEERING
Authors:  S. Henry D. V. McAllister M. G. Allen
==================================

Select an option:
Type "S" to list books of a genre
Type "P" to print out all the book recors
Type "Q" to Quit
S
Type a genre. The genres are:
GENRE_HISTORY
GENRE_SCIENCE
GENRE_ENGINEERING
GENRE_LITERATURE
GENRE_SCIENCE
==================================
Record No:10008
```

Title:The comparative method in evolutionary biology
Genre: GENRE_SCIENCE
Authors: P. H. Harvey M. D. Pagel
==========

==========
Record No:10010
Title:Free radicals in biology and medicine
Genre: GENRE_SCIENCE
Authors: B. Halliwell J. M. C. Gutteridge
==========

==========
Record No:10011
Title:Electron transfers in chemistry and biology
Genre: GENRE_SCIENCE
Authors: R. A. Marcus N. Sutin
==========

==========
Record No:10015
Title:Gene Ontology
Genre: GENRE_SCIENCE
Authors: M. Ashburner C. A. Ball J. A. Blake D. Botstein H. Butler
==========

Select an option:
Type "S" to list books of a genre
Type "P" to print out all the book recors
Type "Q" to Quit
P
==========
Record No:10000
Title:Thomas Jefferson and the Tripoli Pirates
Genre: GENRE_HISTORY
Authors: Brian Kilmeade Don Yaeger
==========

==========
Record No:10001
Title:Component-oriented programming
Genre: GENRE_ENGINEERING
Authors: C. Szyperski J. Bosch W. Weck
==========

==========

Record No:10002
Title:Microfabricated microneedles, a novel approach to transdermal drug deliver
y
Genre: GENRE_ENGINEERING
Authors: S. Henry D. V. McAllister M. G. Allen
==========

==========
Record No:10003
Title:Nikola Tesla
Genre: GENRE_HISTORY
Authors: Sean Patrick
==========

==========
Record No:10004
Title:English landscaping and literature, 1660-1840
Genre: GENRE_LITERATURE
Authors: E. Malins
==========

==========
Record No:10005
Title:A history and theory of informed consent
Genre: GENRE_HISTORY
Authors: R. R. Faden T. L. Beauchamp N. M. King
==========

==========
Record No:10006
Title:The Feminist Companion to Literature in English Women Writers From the Mid
dle Ages to the Present
Genre: GENRE_LITERATURE
Authors: V. Blain P. Clements I. Grundy
==========

==========
Record No:10007
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
==========

==========
Record No:10008
Title:The comparative method in evolutionary biology

Genre: GENRE_SCIENCE
Authors: P. H. Harvey M. D. Pagel
==========

==========
Record No:10009
Title:Human-computer interaction
Genre: GENRE_ENGINEERING
Authors: J. Preece Y. Rogers H. Sharp D. Benyon S. Holland
==========

==========
Record No:10010
Title:Free radicals in biology and medicine
Genre: GENRE_SCIENCE
Authors: B. Halliwell J. M. C. Gutteridge
==========

==========
Record No:10011
Title:Electron transfers in chemistry and biology
Genre: GENRE_SCIENCE
Authors: R. A. Marcus N. Sutin
==========

==========
Record No:10013
Title:Device electronics for integrated circuits
Genre: GENRE_ENGINEERING
Authors: R. S. Muller T. I. Kamins M. Chan P. K. Ko
==========

==========
Record No:10014
Title:An outline of English literature
Genre: GENRE_LITERATURE
Authors: G. C. Thornley G. Roberts
==========

==========
Record No:10015
Title:Gene Ontology
Genre: GENRE_SCIENCE
Authors: M. Ashburner C. A. Ball J. A. Blake D. Botstein H. Butler
==========

```
        Select an option:
        Type "S" to list books of a genre
        Type "P" to print out all the book recors
        Type "Q" to Quit
        Q
Quitting program
```