ECE 242: Data Structures and Algorithms Fall 2015, Project 1

Due Date

11:50pm on September 23, 2015

Description

This project is intended to familiarize you with basic Java programming as well as basic Object Oriented features. The goal of the project is to design and implement a basic media library system in which administrators can add or remove different items in a list. These items are defined either as books or movies.

The media library is organized as follows. Each entry in the library consists of a **Item Title**, **Item Reference**, **Item Price**. A sample media library, which the Title, Media-type, Reference and Price of each item is given in Table 1. In addition, the book media will include **Item Author**, while the movie media will include **Item Director**, **Item Main Actor**. These additional information are provided in Table 2.

Number	Title	Reference	Media	Price
1	2001: A Space Odyssey	TU2RL012	Movie	11.99
2	Data Structures and Algorithms	UI7P3J6Y	Book	32.49
3	Stars Wars: Episode IV	R2D2C3P0	Movie	13.83
4	The Alchemist	TR3FL0EW	Book	3.99
5	The Good, The Bad and The Ugly	PO5T7Y89	Movie	9.99
6	A Brief History of Time	GV5N32M9	Book	10.17
7	Gone with the Wind	2FG6B3N9	Movie	9.99
8	Gone with the Wind	6Y9OPL87	Book	7.99
9	Thus Spoke Zarathustra	F2O9PIE9	Book	6.97
10	North by Northwest	1DB6HK3L	Movie	8.99

Table 1: A sample media library.

Number	Author- Book	Director- Movie	Main Actor- Movie
1	N/A	Stanley Kubrick	Keir Dullea
2	Robert Lafore	N/A	N/A
3	N/A	Georges Lucas	Mark Hamill
4	Paulo Coelho	N/A	N/A
5	N/A	Sergio Leone	Clint Eastwood
6	Stephen Hawking	N/A	N/A
7	N/A	Victor Fleming	Vivien Leigh
8	Margarett Mitchell	N/A	N/A
9	Friedrich Nietzsche	N/A	N/A
10	N/A	Alfred Hitchcock	Cary Grant

Table 2: Additional information for the sample media library.

When a administrator runs the library application, he/she is shown the library with the following restricted fields: **Item Title**, **Item Media-type**, **Item Price**. The administrator can then use the following options: (i) request remaining information about a given item, (ii) add one item to the library (iii) delete one item of the library, or (iv) define a maximum price above which the items of the library are hidden. Specifically, the following functionality must be implemented by the library application:

1. The user is presented with following choices: **1- See List of Items**, **2- See Item Description**, **3- Add one Item**, **4- Remove one Item**, **5- Set Maximum Price 0- Exit**.

2. The media library is initialized using a limited capacity (we will assume that it cannot exceed 20 items for our example). We will also assume that the maximum price for a given item in the library is initialized at $100.

3. If the option "1" is selected, the library is displayed as presented in Table 1.

4. If the option "2" is selected, the user is asked to choose a given item. Depending on the nature of the media, the essential additional information are displayed (e.g. the author name if the selected item is a book, or if it is a movie, the director and the main actor names).

5. With the option "3", the user can add one item to the library; the item will be added at the end of the library.

6. With the option 4, the user can remove an item of the library; remaining items are shifted up in the listing (see sample execution below).

7. If the option "5" is selected then the user can define a maximum price for the items shown, items that are more expensive are not shown (but they are not removed from the list). Therefore, some items of the library will be hidden to the user (i.e. option "1" will display only the items whose prices are below the selected maximum price).

Here is a sample execution of the media application to illustrate the basic functionality of the library:

```
>java MediaLibraryApplication bestmedia.txt

Welcome to Best Media
==================

Menu
====
1-See List of Items
2-See Item Description
3-Add one Item
4-Remove one Item
5-Set Maximum Price
0-Exit

Command: 1

Media/Title/Price
-----------------
1 Movie  '2001: A Space Odyssey' $11.99
2 Book  'Data Structure and Algorithms' $32.49
3 Movie  'Stars Wars: Episode IV' $13.83
4 Book  'The Alchemist' $3.99
5 Movie  'The Good, The Bad and The Ugly' $9.99
6 Book  'A Brief History of Time' $10.17
7 Movie  'Gone with the Wind' $9.99
8 Book  'Gone with the Wind' $7.99
9 Book  'Thus Spoke Zarathustra' $6.97
10 Movie  'North by Northwest' $8.99

Menu
====
1-See List of Items
2-See Item Description
3-Add one Item
4-Remove one Item
5-Set Maximum Price
0-Exit
```

```
Command: 2
Enter item:
3
 Title : Stars Wars: Episode IV (Ref : R2D2C3P0, Price : 13.83);
 Movie Director : Georges Lucas; Main Actor : Mark Hamill

Menu
====
1-See List of Items
2-See Item Description
3-Add one Item
4-Remove one Item
5-Set Maximum Price
0-Exit

Command: 2
Enter item:
4
 Title : The Alchemist (Ref : TR3FL0EW, Price : 3.99);
 Author : Paulo Coelho

Menu
====
1-See List of Items
2-See Item Description
3-Add one Item
4-Remove one Item
5-Set Maximum Price
0-Exit

Command: 2
Enter item:
11
This item is not in the Library!

Menu
====
1-See List of Items
2-See Item Description
3-Add one Item
4-Remove one Item
5-Set Maximum Price
0-Exit

Command: 4
Which item do you want to remove ?
4

Menu
====
1-See List of Items
2-See Item Description
3-Add one Item
4-Remove one Item
5-Set Maximum Price
0-Exit

Command: 1
```

```
Media/Title/Price
-----------------
1 Movie  '2001: A Space Odyssey' $11.99
2 Book  'Data Structure and Algorithms' $32.49
3 Movie  'Stars Wars: Episode IV' $13.83
4 Movie  'The Good, The Bad and The Ugly' $9.99
5 Book  'A Brief History of Time' $10.17
6 Movie  'Gone with the Wind' $9.99
7 Book  'Gone with the Wind' $7.99
8 Book  'Thus Spoke Zarathustra' $6.97
9 Movie  'North by Northwest' $8.99

Menu
====
1-See List of Items
2-See Item Description
3-Add one Item
4-Remove one Item
5-Set Maximum Price
0-Exit

Command: 3

Menu
====
1-Book
2-Movie
1
Enter Book Title:
The Alchemist
Enter Book Reference:
TL3FL0EW
Enter Book Price:
3.99
Enter Author Name:
Paulo Coelho

Menu
====
1-See List of Items
2-See Item Description
3-Add one Item
4-Remove one Item
5-Set Maximum Price
0-Exit

Command: 1

Media/Title/Price
-----------------
1 Movie  '2001: A Space Odyssey' $11.99
2 Book  'Data Structure and Algorithms' $32.49
3 Movie  'Stars Wars: Episode IV' $13.83
4 Movie  'The Good, The Bad and The Ugly' $9.99
5 Book  'A Brief History of Time' $10.17
6 Movie  'Gone with the Wind' $9.99
7 Book  'Gone with the Wind' $7.99
8 Book  'Thus Spoke Zarathustra' $6.97
9 Movie  'North by Northwest' $8.99
```

```
10 Book  'The Alchemist' $3.99

Menu
====
1-See List of Items
2-See Item Description
3-Add one Item
4-Remove one Item
5-Set Maximum Price
0-Exit

Command: 5
Which Maximum Price ? (currently 100.0)
10

Menu
====
1-See List of Items
2-See Item Description
3-Add one Item
4-Remove one Item
5-Set Maximum Price
0-Exit

Command: 1

Media/Title/Price
-----------------
4 Movie  'The Good, The Bad and The Ugly' $9.99
6 Movie  'Gone with the Wind' $9.99
7 Book  'Gone with the Wind' $7.99
8 Book  'Thus Spoke Zarathustra' $6.97
9 Movie  'North by Northwest' $8.99
10 Book  'The Alchemist' $3.99

Menu
====
1-See List of Items
2-See Item Description
3-Add one Item
4-Remove one Item
5-Set Maximum Price
0-Exit

Command: 5
Which Maximum Price ? (currently 10.0)
100

Menu
====
1-See List of Items
2-See Item Description
3-Add one Item
4-Remove one Item
5-Set Maximum Price
0-Exit

Command: 1
```

```
Media/Title/Price
-----------------
1 Movie  '2001: A Space Odyssey' $11.99
2 Book  'Data Structure and Algorithms' $32.49
3 Movie  'Stars Wars: Episode IV' $13.83
4 Movie  'The Good, The Bad and The Ugly' $9.99
5 Book  'A Brief History of Time' $10.17
6 Movie  'Gone with the Wind' $9.99
7 Book  'Gone with the Wind' $7.99
8 Book  'Thus Spoke Zarathustra' $6.97
9 Movie  'North by Northwest' $8.99
10 Book  'The Alchemist' $3.99

Menu
====
1-See List of Items
2-See Item Description
3-Add one Item
4-Remove one Item
5-Set Maximum Price
0-Exit

Command: 0
Goodbye!
```

Requirements and Hints

1. You may use either the `EasyIn.java` class for input/output or the Scanner class; recall that EasyIn.java must be in the same directory as the .java file for this project.

2. The media collection is stored in the file "bestmedia.txt". This file will be included as a command line argument. All the data must be read from the file and your code must be able to process any other similar generated files. The `ReadEasy.java` class provides a couple of examples on how to read from a file.

3. You will create a parent class **MediaItem** and two child classes **Book** and **Movie**. The option "2- See Item Description" which is presented to the user will be calling a method **WhoAmI** which will be redefined in both **Book** and **Movie**. As a requirement for this project, the method **WhoAmI** is defined as follows in the class **MediaItem**: public String

   ```
   public String WhoAmI() {
       String result;
       result = new String(" Title : " + title + " (Ref : " + reference +
               ", Price : " + price + ");\n");
       return result;
     }
   ```

4. You will also create a class **MediaCollection** which can be used to initialize the library. You will need to define several methods here, at least one per menu option. It may also be useful to isolate functionality that is used repeatedly into a method. You may want to use the following names: the method `InitializeCollection`, `AddItem`, `RemoveItem`, `PrintCollection` (in association with option "1- See list of Items"), etc. . . The library should be stored using an array, where each element of the array is a reference to a `MediaItem`.

5. Finally, you will create a class which contains the main method that you will call **MediaLibraryApplication**. The class MediaCollection can then be instantiated and used in this class. The user interface will also be in this class.

Skeleton of the code

File: MediaItem.java

```
public class MediaItem{

    // protected/private member variables
    // no public member variables!

    // constructors here (you may use more than one constructor)

    // protected methods here

    protected void SetTitle....

    protected void SetRef....

    protected void SetPrice....


   // public methods here

    public String WhoAmI() {
        String result;
        result = new String(" Title : " + title + " (Ref : " +
                            reference + ", Price : " + price + ");\n");
        return result;
    }

    public double GetPrice()....

    public String GetRef()....

    public String GetTitle()....

    // other methods here

}
```

File: Book.java

```
class Book extends MediaItem {

    private String author;

    // constructors here (you may use more than one constructor - a couple should be needed)

    // define WhoAmI here

    }
}
```

File: Movie.java

Follow similar idea than for the class **Book.java**

File: MediaCollection.java

```
class MediaCollection {
    private String name;
    private MediaItem[] item;

    // protected/private member variables
    // no public member variables!

    // constructor here
    public MediaCollection(String name, int maxItems, double maxItemPrice) {
        // complete
    }

    // Method
    public String GetName() {
return(name);
    }

    public void InitializeCollection()....

    public void PrintCollection().....

    public String DetailedItem...//Contains the method WhoAmI

    public void AddItem()....

    public void RemoveItem()...

    public void MaximumPrice()...


}
```

File: MediaLibraryApplication.java

```
public class MediaLibraryApplication{

    public static void main(String args[]){
        // declare instance of EasyIn class, if you are using it
        EasyIn easy = new EasyIn();
        MediaCollection store1=new MediaCollection("Best Media",20,100.0);
        store1.InitializeCollection(args[0]);
        System.out.println("Welcome to "+store1.GetName());
        System.out.println("==================");


        // To complete
        // more user interface code here, you may want to define
        // supplementary methods



        System.out.println("Goodbye!");

}
```

Submissions

You will work in groups of two (working alone is accepted but not recommended). **Each group should submit one copy of the solution. Please do not forget to include the name of both partners!**

Include the **name**, **student ID**, and **email address** of both partners. This information should be included in a README file. All files must be zipped into a single file and uploaded on Moodle. Include all files needed to run your project, including the `EasyIn.java` file, if it was used. Include (in the README file) special features or assumptions made in your program we should know about.

Grading Policy

This project will be graded out of 100 points:

1. Your program should compile successfully. (30 points)
2. Your program should implement basic functionality and run correctly. (50 points)
3. Overall programming style: source code should have proper identification, and comments. (20 points)

Extra Credit

You can earn 10 additional points by extending the functionality of your library. Implement an option "6" that sorts the library by increasing prices, book listed first (in increasing price order) follows by movies (in increasing price order). You could use the bubble sort algorithm.