CENG 211

PROGRAMMING FUNDAMENTALS

HOMEWORK-2

Due Date: 13 November 2016, 23:55

***** A bookshelf has books so that each book has the following features:

- title,
- ISBN number,

  → Each ISBN number is composed of 13 numbers by grouping them with dashes. (i.e. 978-975-17-3298-9)
- genre,

  → SCIENCE, DRAMA, ADVENTURE, HORROR, HISTORY, COMICS
- publishing year,
- publishing company (publisher),
- price,
- author(s).

  → A book might have more than one author.

***** Additionally, each author has the following features:

- name,
- birth year,
- birth place,
- book(s).

  → An author might have more than one book.

***** You should define a class so that it should be responsible for reading a file from the given path. **(DataAccessLayer.java)**

***** You should define a class so that it should be responsible for keeping list of all authors that have at least one book in the bookshelf. **(AuthorList.java)**

***** You should define a class so that it should be responsible for user menu operations.**(Book-ShelfMenu.java)**

***** You should define a Bookshelf application class that should do the following operations:

→ **Firstly**, your application should read the file named as "authors.dat" and "books.dat". Then you should create corresponding book objects and add them to the bookshelf. You should also create corresponding author objects and add them to the author list.

→ **Then,** your application should show a menu on the console that has following options:

1. **Adding a book:** For this option, your application should ask all features of the book to user.

   → If author information is not recorded before, application should ask features of the author and add him/her to the author list.

   → If the author of the book already exists in the author list then you should get the author information from the list and add the book to the author's book list.

2. **Removing a book:** For this option, your application should ask the ISBN number of book to user and remove accordingly. If this book is the last book of the author, then author information should be removed from the author list.

3. **Searching books by genre:** For this option, your application should ask genre to user and list books which have the given genre.

4. **Searching books by publishing year:** For this option, your application should ask publishing year to user and list books accordingly.

5. **Searching books by author name:** For this option, your application should ask name of author to user and list books that are written by given author.

6. **Searching books by price:** For this option, your application should ask lower and upper price limit to user and list books whose price is in the given range.

7. **Calculating total price of books in the bookshelf:** For this option, your application should display total price of the books in the bookshelf.

8. **Calculating total price of books by author:** For this option, your application should ask author's name to user and calculate total price of all books of the given author.

IMPORTANT NOTES:

- You should define at least the following Java classes: Book.java, Author.java, BookShelf.java, AuthorList.java, DataAccessLayer.java, BookShelfMenu.java, and BookShelfApp.java
- You should use **Scanner** instead of BufferedReader **for file input operations**.
- You should define a **genre** as an **ENUM** class.
- You should check validity of **ISBN** with **regular expression**. You should also check validity of other inputs.
- In the main method of BookShelfApp.java, you should define an object of AuthorList, BookShelf, DataAccessLayer and BookShelfMenu classes. Then, you should call corresponding method of DataAccessLayer object and read the given files in order to create book and author objects. After that, you should call **init(BookShelf bookShelf, AuthorList authorList)** method of BookShelfMenu object.

SUBMISSION RULES:

- You should create your Java project as **ID1_ID2_HW2** and export as **ID1_ID2_HW2.zip**
- You should upload your zip file **ID1_ID2_HW2.zip** to the CMS.
- One of the group members is sufficient to upload homework to the CMS.
- You should add an author comment to the top of each class that you implement.