CSCD 211
Lab 2

PROGRAM SPECIFICATIONS

For this lab you will create a bunch of classes to solve the problem. You must have the appropriate @Override and enforce the appropriate preconditions. All parameters must be final.

Genre

- Enumerated type that contains Fiction – value-10, NonFiction – value-20, Romance – value-5, SciFi – value 30, Education – value 15 – the ordinal values are fiction – 0, nonfiction – 1, romance – 2, scifi – 3, education - 4
- Private explicit value constructor
- toString that returns fiction, non-fiction, romance or science fiction
- getValue

Publisher

- String publisher name
- String city
- No DVC
- Explicit Value Constructor (takes a string for the name and one for the city)
- getMethods – as you deem necessary
- setMethods – as you deam necessary
- compareTo – sort by name – must use parameterized type
- toString
    - format is: Name, City
      (Example) Pearson, New York City

Author

- String first name
- String last name
- No DVC
- Explicit Value Constructor (takes a string for the first and the last)
- getMethods – as you deem necessary
- setMethods – as you deam necessary
- compareTo – sort by last name if the last names are the same then first name – must use parameterized type
- toString
    - format is: last, first
      (Example) Steiner, Stuart

Book

- String title
- String ISBN
- int pages
- An array of Authors
- Genre reference
- Publisher reference

A book contains as it public methods:

- EVC takes all parameters (title, ISBN, pages, genre as an enum, a publisher reference, an array of strings each element is a full author name)
  ```
  public Book(final String title, final String isbn, final int
  pages, final Genre type, final Publisher pubs, final String []
  authors)
  ```
  - you must enforce all preconditions
  - you must ensure that a new object is created for the publisher

- EVC takes all parameters (title, ISBN, pages, genre as a string, string pub name, string city, Author array)
  - you must enforce all preconditions
  - you must ensure that a new object is created for the authors

- Get Methods
  - getTitle
  - getISBN
  - getPages
  - getPublisher
  - getGenre
  - getAuthors
  - getFirstAuthor

- Set Methods –must enforce all preconditions as shown under EVC
  - setTitle
  - setISBN
  - setPages
  - setPublisher – passed a string
  - setGenre – passed a string

- toString
  - format is: Title, authors, ISBN: ISBN number, Publisher, Pages: pages
    (Example) Operating Systems Concepts 9th edition, Siblerschatz, Abraham, Baer Galvin, Peter, Gagne, Greg, ISBN: 978-1118063330, John Wiley & Sons, Hoboken, Pages: 972

- equals – Must enforce preconditions as discussed in class for equals
  - checks title and ISBN

- hashCode
  - the title

- compareTo – must enforce preconditions and use the Generic <Book>
  - First by publisher, if pubs are the same then title, if titles are the same then by ISBN

CSCD211Lab2.java
I have provided a comprehensive main. You may not change this file.

CSCD211Lab2Methods.java
Public methods that you must write

- `fillArray – must call the first EVC above for Book`
- `menu`
- `printBooks`
- `readFileName`
- `addBook`
- `createBook – must call the second EVC above for Book`
- `getGenre`

The menu choices are

1. Print the books to the screen
2. Print the books to a file
3. Sort the book using compareTo
4. Sort the books by first author as a Comparator
5. Add a book
6. Quit

SPECIFICATIONS

- You must verify all values are in range menu, number of authors, etc.
- You must use my package for the FileUtil

FILE SPECIFICATIONS

# of Books in the file
Title
ISBN
Pages
Genre
Pub Name
Pub City
Number of Authors
One Author per line

Genre strings are guaranteed to be one of the following in any case

- fiction
- non-fiction
- romance
- education
- sci-fi

TO TURN IN

There are multiple turn ins for this lab.

1. Your stubbed out methods ensuring all code compiles will be submitted via git commit to your repository on GitHub. This is required by the date and time specified in canvas).

2. A zip file, in Canvas, by the required due date containing:
   - all java files
   - all input file(s) used to test your program program
   - all output file(s)
   - a test run named cscd211lab2out.txt – testing all aspects
   - We should be able to download the zip and compile your code, and then run your code.
   - Name your zip, your last name first letter of your first name lab2.zip (Example: steinerslab2.zip)