Week 5 Software Homework

FTC Team Mech-A-Mind #202300307

Summary

We are going to be using Classes, Constructors, and Inheritance to create a library of books! We will include different types of books in our library, and we will touch on Arrays in order to organize them. If you are stuck or having trouble at any time, please feel free to reach out and we will be happy to help.

Getting started

Please make sure that you have forked and cloned the repository from GitHub. You can view instructions on how to do that in the starter code's README, which you can find towards the bottom of the page here.

As for the code, you are already provided with all of the files you will need in order to complete the assignments. Let's start by looking at the **Book** class.

Book class

The **Book** class is mostly filled in for you, and should look like a review from Week 5's lesson.

- Please fill in the blank constructor and the constructor that provides input parameters.
- Write the method that compares the price of two books. This method should return a boolean, and use an if statement to compare the price of "this" book and the book that is passed in as a parameter to the method.

Textbook class

The **Textbook** class is different from the **Book** class in that it has "editions" that can expand upon each other. The **Textbook** class **extends** the **Book** class, which means it contains all of the properties of the **Book** class, but has some of its own as well. Read more about inheritance in java. In your own words, explain the "extends" keyword on line 4, explain the "super" keyword on line 12, and explain "@Override" on line 29.

- First, we want to finish the constructors for the class. Please initialize the edition variable in both constructors, and use super() to initialize title, author, and price in the second constructor.

- Next, write getter and setter methods for edition at the end of the **Textbook.java** file. If you are confused about getter and setter methods, please refer back to the **Book** class.
- Write the Override for the **toString** method. We essentially want to copy the toString method in the Book class, but include additional information that a Textbook would have compared to a Book.
- Lastly, we want to write the **canSubstituteFor** method. The conditions for the method are as follows:
    - We take **Textbook t** as a parameter and we are returning a **boolean**
    - A textbook can be a substitute if it has the same title and it is a newer edition than the parameter textbook.
    - Example:

      Textbook bio2019 and bio2015 have the same title, but have a different title than math2023. bio2019's edition is **3** while bio2015's edition is **2**.

      bio2019.canSubstituteFor(bio2015) should return **true**
      bio2015.canSubstituteFor(bio2019) should return **false**
      bio2019.canSubstituteFor(math2023) should return **false**

FictionBook class

Write the FictionBook class on your own. A FictionBook class should include
- All the same parameters of a Book class (FictionBook will extend the Book class)
- A genre variable (String)
- A gradeLevel variable (Integer)
- An empty constructor and a constructor with parameters, both utilize super()
- An @Override for toString to contain the additional information that a FictionBook contains
- Getter and setter methods for genre and gradeLevel
- A boolean method that checks if two books are the same genre
- Three boolean methods: one that checks if two books' grade levels are equal, one that checks if the parameter book's grade level is lower, and one that checks if the parameter book's grade level is higher

Bonus: Create your own class

As a bonus, create another class that extends the Book class. This book contains your own parameters and methods that you come up with. Ideas include a ChildrenBook class, a ColoringBook class, and an Encyclopedia class.

Library class

In the library class, we are using an object called ArrayList<Book> to store our books. An ArrayList is an Array but with an undefined length. Normal arrays have a length that you must define, and we want our library of books to be able to grow and shrink as we need it. So, we use an ArrayList instead of a regular Array. You won't need to worry too much about ArrayLists. We won't be coming across them much this season. However, if you are interested, feel free to play around with ArrayList and look up more information or documentation on it!

- Write the method **printBooks()**. This method should call toString() for all books that are in the library.
- Finish the method **sumOfPrices()**. Write the method, which calculates the sum of prices of all the books and returns an integer.

For these tasks, you may want to use a for loop to iterate over the **length** of the ArrayList.

Main class

Now, the main class. Let's wrap the program all together. Here are the steps you have to complete:

1. **Initialize a new library**. This is as simple as creating a new library object. The library constructor takes no parameters.
2. **Create books**. Create a few books, preferably at least one of each type of book you made. Initialize them with all the proper variables - do not leave their values to be the default values initialized by the blank constructor.
3. **Add all the books to the library**. One by one, add all of the books to the library object using library.addBook(Book). As a bonus - try and optimize this so that you can add the books to the library faster.
4. **Print all the books in the library**. Call the method that prints all of the books in the library.
5. **Print the sum of the prices of all books in the library**. You may want to use a print method since the method you will be calling returns an integer.

Run your program

Run the program! If you get any errors, try and debug them yourself and get the program working. If you finish earlier in the week or you felt that this assignment wasn't challenging enough, feel free to try to expand the project - how could we make things more efficient?

Submit assignment

View the instructions on how to submit the assignment at the end of the page here.