3802ICT

Programming Languages

Practice-based Assignment

Assignment 2

Trimester 2, 2021

Background

“The purpose of Data-Oriented programming (DOP) is to reduce the complexity of software systems, by promoting the treatment of data as a first-class citizen”. To treat data as a first-class citizen, we need to ensure the application of the 3 principles, given as follows:

1. Code is **separated** from data
2. Data is **immutable**
3. Data access is **flexible**

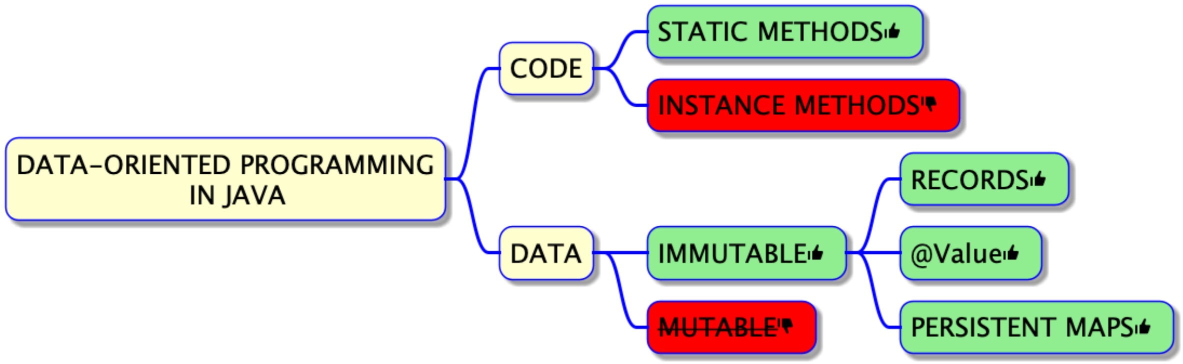

Fig 1. Data-oriented programming principles

In this assignment, you will build a library management system (LMS) with the following requirements (RQ#):

RQ1. Two kinds of users: library members and librarians
RQ2. Users log in to the system via email and password
RQ3. Members can borrow books
RQ4. Members and librarians can search books by title or by author
RQ5. Librarians can block and unblock members (e.g., when members return a book late)
RQ6. Librarians can list the books currently on loan to a member
RQ7. There could be several copies of a book

A classic Java design for such a system could be made of the following classes:

- **Library**: The central part for which the system is designed
- **Book**: A book
- **BookItem**: A book can have multiple copies, each copy is considered as a book item
- **BookLending**: When a book is lent, a book lending object is created
- **Member**: A member of the library
- **Librarian**: A librarian
- **User**: A base class for Librarian and Member
- **Catalog**: Contains a list of books
- **Author**: A book author

A class diagram to achieve DOP in the above LMS is given below: each class of the system is split in two classes.

1. A code class with **static methods** only.
2. A **data class with members** only.

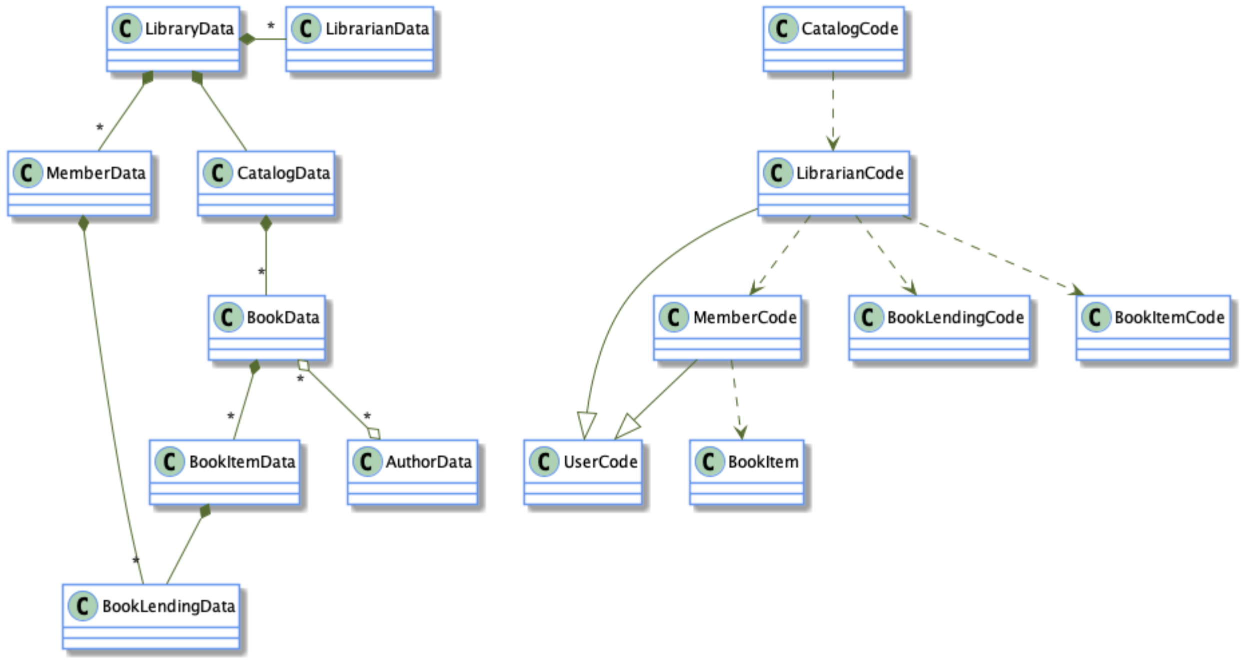

Fig 2. Class diagram of the library management system: Data classes are on the left and Code (software) classes are on the right

Tasks

You need to accomplish the following tasks:

1. Create data and code classes for an LMS as per the illustration given in Fig 2.
    a. Demonstrate the 3 principles of DOP in both data and code classes.
    b. Fulfill the requirements RQ1-RQ7 of the LMS.
2. Develop a **MyLibrary** class with a main method to demonstrate that all requirements of the LMS have been accomplished.

Constraints

1. You are allowed to use Java built-in packages.
2. You need to demonstrate the use of @Value from Lombok to demonstrate data immutability in your data classes.
3. You can demonstrate the use of var, lambda expression, Java functional interfaces, interfaces and generic classes. [**Hint**: You don’t need to demonstrate all of them - demonstrate the use of these advanced Java features in your code whenever possible]

4. A general requirement is to write readable and well-commented code.

Marking Criteria

Task	Mark Detail	Task Total
Data and Code Classes	0.8 mark for each Data Class	8x0.8+1.2x7=14.8
	1.2 marks for each Code Class	
MyLibrary Class	main method Demonstration of RQ1-RQ7	1+7=8
Demonstration of advanced Java features, readable and well-commented code		2.2
Total		**25**

References

1. Y. Sharvit, "Data-Oriented programming in Java", http://tutorials.jenkov.com/java/record.html
2. Why Java's Records Are Better* Than Lombok's @Data and Kotlin's Data Classes: https://nipafx.dev/java-record-semantics/
3. Y. Sharvit, How to access data dynamically in Java without losing type safety? https://blog.klipse.tech/java/2021/03/28/dynamic-data-access-in-java.html

Good luck ☺