SFA Java – Homework Assignment 1

Library Management Application

1. Abstract

The idea behind **Homework Assignment No: 1** is to incrementally develop an application, that will go by the name: Library Management Application (LMA), in the period of a few weeks. Each lecture new functionalities should be added, as part of a separate task, which is considered as one increment of the application.
Demo is advised to be presented for each increment/task. And the code for each task should be pushed to your personal repository on Github.

2. Overview of the business logic of the application

Library Management Application (LMA) will be a console application, which, as the name suggests, will be an application for internal library management. It should allow librarians to store information on an electronic device rather than on paper. Librarians should have the possibility to register clients, create book orders and gather information regarding books, clients, and orders in an easier and faster manner.

The main goal is to give the librarian the ability to create an order for a book and assign it to a client. The client and book are to be chosen from a list of pre-existing ones.
Librarians should be able to retrieve a list of all placed orders along with information for each.
Books can also be added to the catalogue (pre-existing file of books with their additional information).

3. Guidelines for developing the application:

Architecture: Three-layer

1. Presentation Layer: All of the application's functionality should be accessed using the Command Line Interface (Console/CLI)
2. Business Logic Layer: All data processing classes
3. Data Access Layer: Read/Write operations to the files should happen in separate classes

Presentation Layer: Console application

Since the application is imagined to be developed as console application, that means that you should present some 'menu' on the console which will allow the user to make a certain choice, based on their intentions and then proceed from there. Imagine it as an ATM 😊 You have options to choose from, displayed on an interface, and you can do different things based on your choice, e.g., check your balance, withdraw cash etc.

In the context of the Library Management application that means the librarian should have a menu presented, from which he/she will choose certain option like: create an order, get a list of all available authors or clients etc.

So, when running your app, a certain type of menu should be presented to the console in the following format:

Example:

Welcome to Library Management Application!
Please choose a menu, to proceed further:

1. Order Management
2. Book Management
3. Client Creation
4. Author Creation

If you choose a menu option, it should look something like this:

1. Order Management
   a. Order information
   b. Create order
      .
      .

Working with file system:

-In order to access, modify and add information you will have to work with files (which in later stages will be changed with database); from which you should read the data about books, clients, authors, and orders, and then also add new records for the already mentioned objects.

For example, if you choose as librarian to see all the books available, you should read a list of all books saved in Books.txt file and present them on the console.

4. Tasks:

Task 1. Develop Library Management Application functionality
Task 2. Add a build tool
Task 3. Add custom exceptions

Task 1: Develop Library Management Application functionality

Required functionalities:

Order Management:

The librarian should be able to create an order by choosing a client from a predefined list. They should then pick a book either from all available books, or choose a book filtered by specific author. In case the latter is chosen, an author is picked from a predefined list. The order is then saved to file.

Each order should be saved with the following information:

- Client
- Book
- Issue Date
- Due Date (All books should be returned a month after they were borrowed from the library)

i.e., Petar Shaw_The Island of missing trees_20-03-2023_20-03-2023

Book Management:

The librarian should be able to create a book by entering its name, date of publishing and selecting an author from predefined list.

e.g.: The Island of missing trees_12-12-2018_Elif Shafak

Order Information:

The librarian should be able to get the following lists of orders:

- All orders
- All orders by client
- All orders issued on/before/after specific date
- All orders due by/before/after specific date

Extra credit functionalities

- Add functionality to add new clients.
- Add functionality to add new authors.

- Validate that each client listed for an order also exists in the predefined list of available clients.

Task 2: Add a build tool

Task 3: Add custom exceptions

Add-ons - optional

Catalogue Management

Books now have limited availability. The current number of available books should be tracked and if it's 0, no more orders should be allowed to be created.

Order Management

- Add the functionality to have multiple books per order.
- Add the functionality to extend order due by date. Its length, as well as whether it is days, weeks or months, is taken from user input.

Add JDBC

Replace files with a database as a data-storage solution. Integrate the Java project with the DB using JDBC.

5. Assignment requirements:

In all the assignments, writing quality code that builds without warnings or errors, and then testing the resulting application and iterating until it functions properly is the goal.

Homework assignments will *not* be graded, but we might check projects at random to measure progress. The conditions we don't want to see are:

- Project does not build.
- Project builds with warnings
- One or more items in the Required functionalities section were not satisfied.
- A fundamental concept was not understood.
- Code Quality -Your solution is difficult (or impossible) for someone reading the code to understand due to:
    - Code is visually sloppy and hard to read (e.g., indentation is not consistent, etc.).
    - No meaningful variable, method and class name following Java code style guide. Over/under used methods, classes, variables, data structures or code comments.
- Assignment is not submitted as per Assignment Submission section below.

6. Assignment submission:

Homework should be submitted using Git.
It should be pushed according to the structure: "{AcademyFolder}/LibraryManagement"