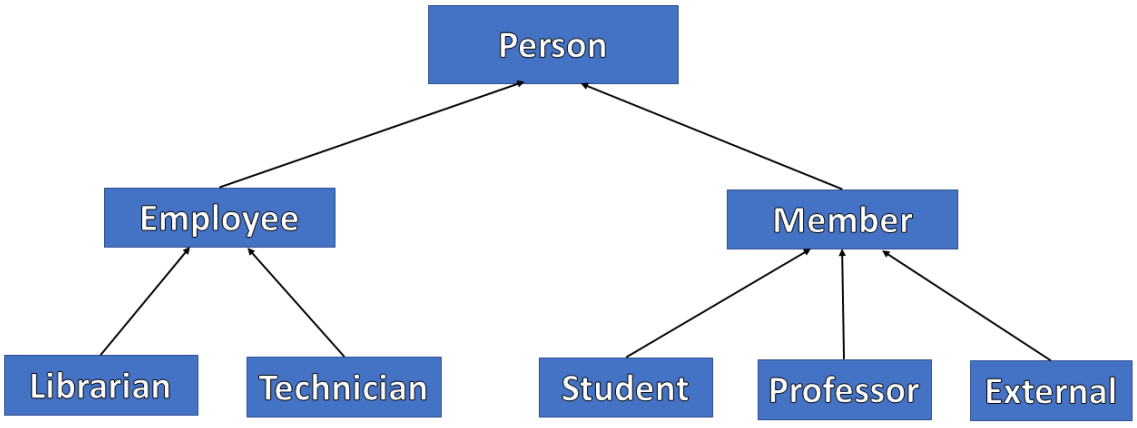
CS3330 Java Project- Part 1

Simple University Library System

Consider the following case study:

University of Java has a manual library system. The library has collections of books, newspapers, DVDs, and journals, which are shelved in different sections of the library. The sections include Arts, Sciences, Newspapers, and Law. All the collections have a unique identification number which is alphanumeric and is 6 characters long. Besides the usual information, the books and DVDs have ISBN numbers while the others have ISSN numbers. The library interacts with different people. These people can be employees and members. The employees are of two categories librarian and technician. The librarians are responsible for managing the collections and helping customers with the resources. Librarians also manage the memberships. The technicians tend to sort the returned books and re-shelve them in the correct location. Three types of people can borrow from the library- professors, students and external people. A Professor may supervise multiple students while, a student can have only one professor as an advisor. The external members are usually people who want to become members of the library but are not affiliated with the university. All members have a unique identification number. For simplicity, we do not consider Professors and Students as employees under the library system. Members can borrow items from the collection- not more than five materials all together. They are allowed to keep each material for a maximum of two weeks. An email is sent to the members on the 12th day as a reminder that the material deadline is reaching. On the 14th day another email is sent informing the members that the material is immediately due that day. From the next day onwards, if the material is not returned, a fine of $1 is added into the members account for each day the material is not returned. If the material is not returned within the next two weeks, another email is sent. Also, a letter is printed out and physically mailed to the members address. The item will be declared "lost" if it is missing for at least a month and the member will be charged the full price of the item in addition to the accumulated fines. The member may renew the item anytime before the due date for an additional two weeks.

Given the above scenario, you are asked to automate the system using object oriented programming (OOP) and Java. You may use any built-in Java libraries. The following is a class diagram for the People associated in the system:



1. Implement the above class diagram in Java. You may use whichever variables and methods you see fit judging from the case description, class diagram and general knowledge. As a software engineer, you are also allowed to ask the client (in this case: **Me**) appropriate questions for further information. You may add *additional* classes to this UML. A sample UML for the People class is given as follows which you may find useful (This is just a sample; you may add or remove the variables and methods as appropriate; overload as you wish):

People
name: String # address: String # dob: Date # email: String # ssn: SSN
+ toString(): String + setName(String) + setAddress(String) + setDate(Date) + setEmail(String) + setSSN(SSN) + getName(): String + getAddress(): String + getDate(): Date + getEmail(): String + getSSN(): SSN + getSSN(): String + getPeople(): People

2. Using knowledge from the case write-up and the client, design the UML class diagram for the LibraryCollection class. Show all inheritances and any additional classes you feel needs to be added or implemented. If a class is not inherited, keep it separate from the hierarchy diagrams.

3. Your project implementation should reflect sufficient knowledge acquired from this course. Which means, you need to implement Exception handling, validation checks, use abstraction as appropriate, etc. from our coursework.

4. Try to setup a system that communicates with the objects of the different classes. Think of certain events that can occur in real life, and then setup the event. Like for example, NewMembershipEvent, NewEmployeeEvent, NewCollectionEvent, BorrowsEvent, etc. It will be useful for us to visually see how objects communicate in each event, and then implement the event.

5. Try to use version control like GitHub to maintain your progress.

6. IMPORTANT: This is first part of the project. The concept will be carried forward to the second part of the project. If you feel that you are implementing too much within the stated deadline, you may skip certain portions and then implement them in the final part of the project.