LAB 5

Java programming

Today (and next week), you will construct a simple program that simulates the operation of a library. You need to design the class hierarchy, that represents the library borrowing system. There are three kinds of materials you can borrow from the library: regular books, journals, and films.

Regular books may be borrowed for two weeks, films can be borrowed for 2 days. Journals can be taken out for three days, if you are a student, and for one week if you are a faculty member. Students can have on a loan maximally one movie at the time, three books, and three journals. There is no limit for the faculty members.

There is an overdue fee, 0.5 $ per day for a book, 2 $ per day for a journal, and 5 $ per day for a movie.

All items have a unique library ID assigned. Books have `Title`, `Author`, `Genre` and `Publisher` entries. Journals have Title, eISSN, Publisher, Latest issue and Journal URL entries. Movies have Title, Genre, Director, Year, Runtime (Minutes) and Rating entries. Note, that some fileds could be incomplete.

The library users are 80 students and 20 faculty members. Out of them, 67 people always return the book on time, 33 can be delayed.

Simulate the operation of the library over one year (365 days). You may assume, that at the beginning, all items are in the library. For each user, for each day, there is the probability α_{item} (e.g., $\alpha_{book} = 0.05$, $\alpha_{journal} = 0.08$, $\alpha_{film} = 0.05$) that the user borrows the item, and the probalility β (e.g., $\beta = 0.02$), that the user returns the borrowed item. The ”on-time” users always return the item on the last day of the loan period, if it has not been already returned. The item is picked randomly, from those that are in the library (are not on loan).

The day when the item is taken out is not counted, however the day when it is returned is counted as the loan day.

Define the abstract class `LibraryItem` (which is extended by Book, Journal and Film classes), and lift those fields that can be lifted to this class.

Design the method `daysOverdue` that returns the number of days this item is overdue. If the number is negative, the item can still be out for that many days.

Design the method `isOverdue` that produces a boolean value, that informs us whether the item is overdue on the given day.

Design the method `computeFine`, that computes the fine for this item, if the item is returned with the delay.

All the instances, representing the library items should be held in the Library class.

Design a `borrowItem` method of the `Library` class that allows you to borrow any item on any day. This method will be useful for creating unit tests for `daysOverdue`, `isOverdue`, `computeFine` methods. Create such tests and run them for specific examples using JUnit.