Assignment 3

Due Date : (12.05.2023 - 23:59)
Programming Languages: Java
Subject : Object Oriented Design, Inheritance, Polymorphism

1 Introduction

In this project, you are expected to implement a simple library management system. You need to design a system that handles daily operations in the library. The main objective of the program is to analyze the incoming requests, which are provided in the form of an input list, and to take the necessary actions. The members of the library are students and academics. They can borrow books and return them before the specified deadline, otherwise they must pay a fee to borrow another book. It is also possible for a member to extend the deadline only once for each book. In addition, certain books cannot be taken out of the library, but can only be read within the library, since they are rarely found and only academics can get to them. Of course, members must be registered in the system before such operations, so that they can use the library's services.

Besides of the typical daily operations, the book collection can be enlarged by adding a new book.

2 Experiment

This section contains possible actions actions in the library. Actions and details are given below.

```
1 => addBook
2 => addMember
3 => borrowBook
4 => returnBook
5 => extendBook
6 => readInLibrary
7 => getTheHistory
```

Adding a new book:

While adding a new book to the library collection, only the type of the book will be given and the book id will be determined by Library. The id of a book is at most a six digit number and an id should be unique to a book. In other words, no two books can have same id, even if one of them is a printed book and one of them is a handwritten book. Successive numbers should be assigned as id values of the books according to the order of adding them to the library collection starting indexing from 1, such that the first book in the library must have the id equals to 1 and the second book has id 2 and so on. The format of adding a new book command is as following. P means the type of the book is Printed. H means the type of the book is Handwritten.

```
addBook<TAB><typeofbook>
addBook    P
addBook    H
```

Adding a new member:

While adding a member, only the type of the member will be given and the member id will be automatically determined by the library. The id of a member is at most a six digit number and no two members can be assigned with the same id. The procedure of adding a member is the same as with the procedure of adding a new book. The format of adding a new member command is as following. S means the type of the member is Student. A means the type of the member is Academic.

```
addMember<TAB><typeofmember>
addMember    S
addMember    A
```

Borrowing a book: After the necessary checks are made for the borrowing process, the borrowing process is completed after the necessary procedures are carried out for the relevant book. The format of the command of the borrowing a book is as following:

```
borrowBook<TAB><Id_Of_Book><Id_Of_Library_Member><TAB><date>
borrowBook    1    1    2023-04-07
```

If the book specified in the system cannot be borrowed, a warning message will be printed in the output file as follows:

```
You cannot borrow this book!
```

Returning a book: After the necessary checks for the validity of the return of a book, the member returns the book to the library. However, if the deadline has passed, the member must pay the penalty in order to borrow another book. The date format will be given as YYYY-MM-DD. The format of the command of the returning a book is as following:

```
returnBook<TAB><Id_Of_Book><TAB><Id_Of_Library_Member><TAB><date>
returnBook    2    1    2023-04-07
```

Fee should be paid = (the returning date) - (the deadline of the book)

If the deadline has passed, a warning message will be printed in the output file as follows:

```
You must pay a penalty!
```

Extending a book: Before the deadline, the member can extend the deadline further in this command. A member can extend the deadline only once for each book.The necessary check should be carried out for a possible extension. After the extension:

deadline = the current deadline + time limit for the LibraryMember.

The format of the command of the extending deadline is as following. Following currentDate refers to the date the library member requested the extension.

```
extendBook<TAB><Id_Of_Book><TAB><Id_Of_Library_Member><TAB><currentDate>
extendBook    1    1    2023-04-07
```

If a member wants to extend a book more than once, a warning message will be printed in the output file as follows:

```
You cannot extend the deadline!
```

Read In library: If a member desires to read the book instead of borrowing it, the necessary instructions should be executed by this method. Contrary to extendBook command, there is no deadline for returning the book in reading in the library. You will not be given a scenario where books read in the library are not returned. The format of the command of the read in library is as following:

```
readInLibrary<TAB><Id_Of_Book><TAB><Id_Of_Library_Member><TAB><currentDate>
readInLibrary    2    1    2023-04-07
```

If a member of the student type wants to read a handwritten book in library, a warning message will be printed in the output file as follows:

```
Students can not read handwritten books!
```

If a member wants to read a book borrowed by someone else in the library, an error message like the one below should be displayed.

```
You can not read this book!
```

Getting the history: This command is used for getting the history of library such as number of academic members registered in the library, number of student members, number of books borrowed, number of books read in the library. The format of the command of the getting the history of library is as following:

```
getTheHistory
```

The output of this command is shown in Figure 2. In the output of the getTheHistory command, for books borrowed and read in the library, print those that have not yet been returned.

2.1 Some Remarks

There are some rules about this automation system:

- Each Library Member has a book limit defines the quantity of the books that could simultaneously be borrowed. This book limit depends on the type of the Library Member (Student/Academic).
- Library Member have a constant time limit to return the borrowed book for each book which is defined through time Limit. This time limit also depends on the type of the Library Member. If the Book is not returned in time, then the Library Member must have to pay a fee.
- An academician can have 4 books at most simultaneously.
- The time limit to return a borrowed book should be defined as two week for academicians.

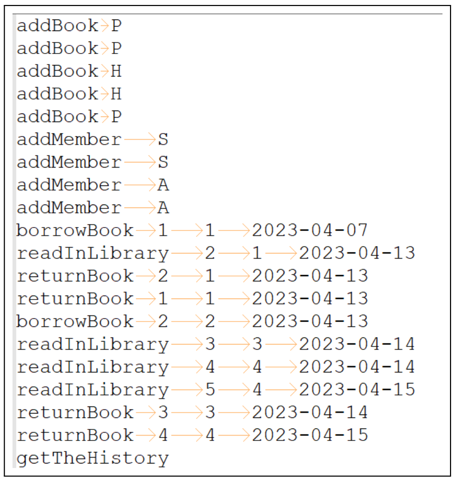
- Only the academicians have an access to the Handwritten books.
- A student can borrow 2 books at most simultaneously.
- The time limit to return a previously borrowed book is one week for students.
- While academicians are defined as “A” , the students should be represented through “S”.
- The students do not have permission to read a Handwritten book.
- While some of the books are borrowable by the members and can be taken out of the library, the Handwritten books are not permitted to be borrowed by any member. On the other hand, any member can borrow a Printed book, but it is also possible to read it without borrowing.
- For the extension process, if the due date has not yet been missed and the due date of the borrowed book has not been extended before, the time allowed to the Library Member is doubled.
- If there is an exception not mentioned in the pdf, you can specify it in an appropriate warning message.

2.2 Constraints

1. The methods’ and attributes’ names should be satisfied the most common naming conventions in Java.
2. **You should model entities of the system with classes.**
3. Your design will be graded. You are expected to propose a suitable design for the problem. Please keep in mind that providing the necessary accessibility and visibility is important: you should not implement everything as public, even though the necessary functionality is implemented. The usage of appropriate access modifiers and other Java keywords (super, final, static etc.) play an important role in this project since there will be a partial credit, specifically for the software design.
4. All the input files and output file will be taken as command line arguments.

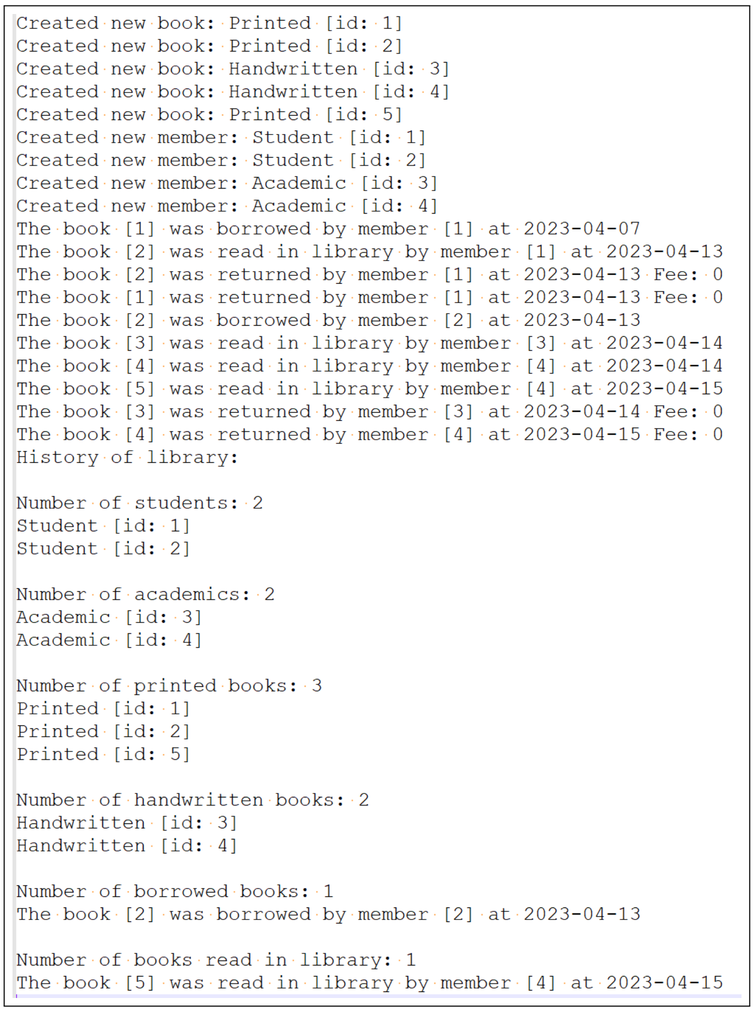
3 Input and Output

The example input and output files is given as follows:



```
addBook	P
addBook	P
addBook	H
addBook	H
addBook	P
addMember	S
addMember	S
addMember	A
addMember	A
borrowBook	1	1	2023-04-07
readInLibrary	2	1	2023-04-13
returnBook	2	1	2023-04-13
returnBook	1	1	2023-04-13
borrowBook	2	2	2023-04-13
readInLibrary	3	3	2023-04-14
readInLibrary	4	4	2023-04-14
readInLibrary	5	4	2023-04-15
returnBook	3	3	2023-04-14
returnBook	4	4	2023-04-15
getTheHistory
```

Figure 1: Sample input file



```
Created new book: Printed [id: 1]
Created new book: Printed [id: 2]
Created new book: Handwritten [id: 3]
Created new book: Handwritten [id: 4]
Created new book: Printed [id: 5]
Created new member: Student [id: 1]
Created new member: Student [id: 2]
Created new member: Academic [id: 3]
Created new member: Academic [id: 4]
The book [1] was borrowed by member [1] at 2023-04-07
The book [2] was read in library by member [1] at 2023-04-13
The book [2] was returned by member [1] at 2023-04-13 Fee: 0
The book [1] was returned by member [1] at 2023-04-13 Fee: 0
The book [2] was borrowed by member [2] at 2023-04-13
The book [3] was read in library by member [3] at 2023-04-14
The book [4] was read in library by member [4] at 2023-04-14
The book [5] was read in library by member [4] at 2023-04-15
The book [3] was returned by member [3] at 2023-04-14 Fee: 0
The book [4] was returned by member [4] at 2023-04-15 Fee: 0
History of library:

Number of students: 2
Student [id: 1]
Student [id: 2]

Number of academics: 2
Academic [id: 3]
Academic [id: 4]

Number of printed books: 3
Printed [id: 1]
Printed [id: 2]
Printed [id: 5]

Number of handwritten books: 2
Handwritten [id: 3]
Handwritten [id: 4]

Number of borrowed books: 1
The book [2] was borrowed by member [2] at 2023-04-13

Number of books read in library: 1
The book [5] was read in library by member [4] at 2023-04-15
```

Figure 2: Sample output file

3.1 Submit Format

- File hierarchy must be zipped before submitted (Not .rar, only .zip files are supported by the system)

- $\langle studentid \rangle .zip(example : 1234567.zip)$

    - [src]

        - Main.java

        - *.java

4 Grading and Evaluation

- Your work will be graded over a maximum of 100 points.
- Your total score will be partial according to the grading policy stated below.

Taking input and output files from command line	6p
Each command (addMember,addBook,etc.) *12p	84p
Code design, clean and readable code, algorithmic perspective, comments	10p

- **Your code will be tested on dev platform. And one output file will be expected as output file. If your program cannot compile on dev server, you cannot get any points. Please be sure that your program can be compile on dev server.**
  - Upload your files to your server account (dev.cs.hacettepe.edu.tr)
  - Compile your code (javac *.java)
  - Run your program (java Main input.txt output.txt)
  - Control your output (output.txt).

5 Notes

- You have to design your application in complete object oriented design approach. You are expected to define a java class for everything that is suitable for a class.
- The assignment must be original, individual work. Downloaded or modified source codes will be considered as cheating. Also the students who share their works will be punished in the same way.
- We will be using the Measure of Software Similarity (MOSS) to identify cases of possible plagiarism. Our department takes the act of plagiarism very seriously. Those caught plagiarizing (both originators and copiers) will be sanctioned.
- Please do not miss that the name of the input and output files will be fixed. Use the file names as indicated in the leaflet. File names with different names will not be evaluated.

- You can ask your questions through course's piazza group and you are supposed to be aware of everything discussed in the piazza group. General discussion of the problem is allowed, but DO NOT SHARE answers, algorithms, source codes and reports .
- Ignore the cases which are not stated in this assignment and do not ask questions on Piazza for such extreme cases.
- Don't forget to write comments of your codes when necessary.
- The names of classes', attributes' and methods' should obey to Java naming convention.
- Do not submit your project without first compiling it on dev machine.
- Save all work until the assignment is graded.
- Do not miss the deadline. Submission will be end at 12/05/2023 23:59, the system will be open until 23:59:59. The problem about submission after 23:59 will not be considered.