Project 2 Management of a library

Objectives :

This last project is about designing a program that simulates a library : members, books, registrations, loans, returns of books, etc. You must provide data types and functions that answer the below specification requirements, and design menus that rely on these functions. When several choices are possible, we advise you to propose (and implement) these different solutions, justifying their choice and comparing them, from the point of view of the implementation and the facilities/limitations they impose on the user. All written programs must be readable, commented, and modular in order to evolve easily. Developed programs must be easy and fun to use for a novice user (other than you).

The graphical interface is a plus but is not mondatory, a console display is enough. You are also asked to write a structured and concise report in which you describe your algorithmic design, the difficulties encountered and the envisaged improvements of your application.

Requirement specification :

You must manage two databases, *db-members* and *db-books*, representing library members and books respectively. The *db-member* database contains as many members records as members registered in the library. Each member record is composed of the following elements :

- First and last names of the member.
- The mailing address and e-mail address of the member.
- Function of the member
- List of borrowed book codes (up to 3 borrowings) and return date for each of them (15 days maximum).

The base-book database *db-books* contains as many books as books that can be borrowed from the library. Each book includes :

- The title and author of the book.
- A code XXX-YYY with 7 characters, where XXX is the theme of the book (NOV for novel, CAR for cartoon, etc ...), and YYY is the number of the book into the XXX theme. For example, the book referenced NOV-255 is the 255th novel within the theme NOV in the database.
- Total number of copies.
- The number of available copies.

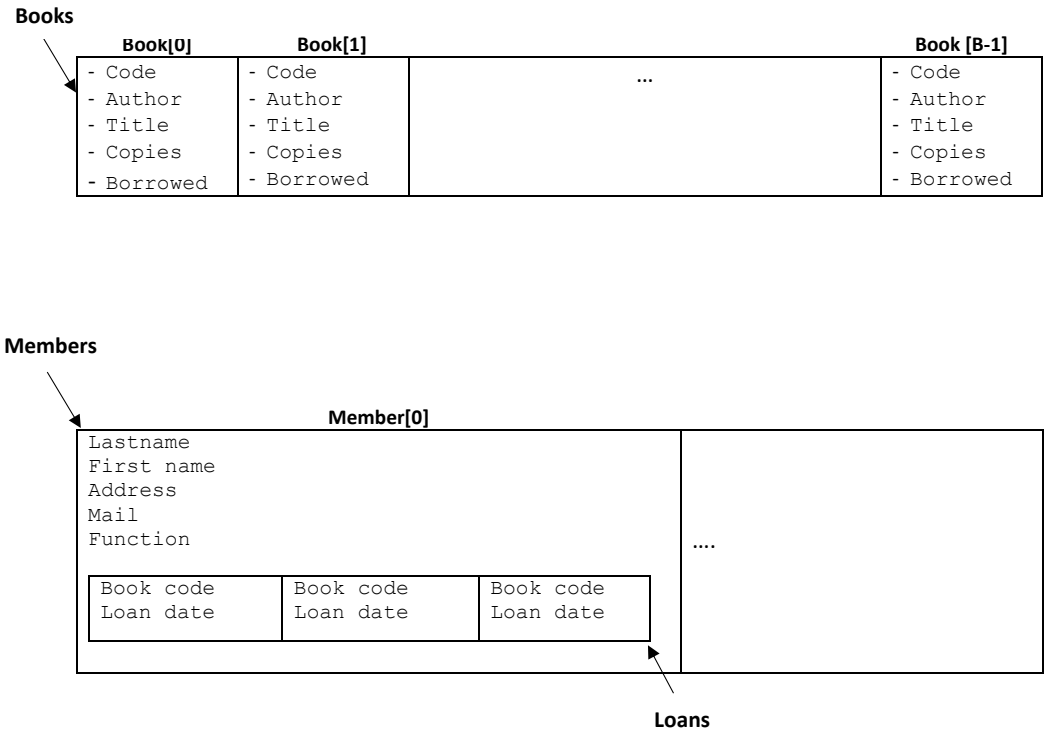

FIGURE 1 – The library data structures de données de la librairie

Storage of data

The data of the library (members, books, loans, ...) are saved in files of your choice (type, structuration of the data, ...). For sake of efficiency, once your application starts, the data (list of members, list of books, list of loans, ...) is read from these files and saved in appropriate data structures. It will therefore be necessary to provide functions for loading this data into dynamic arrays (depending on the number of records in the databases) from the files.

All the modifications are made in memory (on these arrays) and loaded into the files before closing the program.

Figure illustrates the storage of the members and books of the library in arrays.

Other data structures may be necessary for the implementation of your program (lists of loans for example). So do not hesitate to define whatever you consider useful for the design of your application and to facilitate the implementation of the functionalities requested below.

Menu :

You must offer two menus to the librarian.

1. A read only access service menu allowing you to :
   (a) View the list of members ordered alphabetically.
   (b) Display the list of books according to one of the following orders : code, title, author, ... A preliminary sorting of the list of books (according to the chosen criterion) should be considered. You must therefore provide a sorting function by criterion.
   (c) Search for information about a book (again offer several search criteria). To speed up the search, the sorting functions established in the previous functionality will be useful.
   (d) Display the list of borrowed books whose return date has been exceeded, and, for each, the information about the borrower member.
2. A menu that interfaces with the librarian. This menu provides :
   (a) Adding a new member. You must first check that the member does not already exist in the list. To do this, browsing a sorted list will allow this verification and will also facilitate the insertion of the new member at the right position in order to avoid re-sorting the list. You can use the member's first and last names to ensure that the same member does not appear twice in the database.
   (b) Add a new book. As with the previous functionality, you must first check that the book does not already exist in the list. To do this, browsing a sorted list will allow this verification and will also facilitate the insertion of the new book at the right position in order to avoid re-sorting the list. You can rely on the title of the book to guarantee that the same book does not appear twice in the database (uniqueness of the title). Note that the number of the book in the theme must be generated automatically as the last number in the theme + 1.
   (c) Deleting an adherent (only if he/she has borrowed book currently) or a book (only if it is not currently borrowed) from the data-base.
   (d) The record of a new loan if possible (implies the modification of a member and a book).
   (e) Taking into account the restitution of a book (implies the modification of a member and a book). If the return date is not respected, a message is displayed on the screen. One can imagine a sanction imposed on the member if several cases of non-compliance with the return date are recorded. This penalty can be the prohibition of borrowing during a period, or the decreasing of the number of simultaneous loans for this member. The implementation of this functionality can induce changes in the used data structures.

This specification is indicative and minimalist, it should allow you to design a basic version of the application. You can enrich the specifications as you like.

Some ideas of enrichment :

- Search for the books written by a given author,
- Look for members who hold (at this moment) a given book,
- Find the number (total or over a given period) of loans for a given theme (or a given book),
- Find the number of books in the library that belong to a given theme
- Fine the theme/book/author that has the largest number (global or A given period) of loans.
- ...

Planning

— Subject available on April 31, 2020
— Follow-up session during week of May 11, 2020
— Defense of the project, and upload of reports and source files on May 24, 2020

Project Defense Each project team has 20 minutes to defend its work, as follows :

— During the first 10 minutes, you will briefly introduce the project, describe the organization of the team, expose your achievements with respect to the requirements, and the problems you faced. Both team members must speak in turn. Note that you must prepare and rehearse your speech.
— For the next 10 minutes, you will answer the questions of the examiner. Each team member must be able to answer any question, on any part of the project.

Please note that, depending on the quality of their performance, project members may get different marks.