University of Bahrain
College of Information Technology
Department of Computer Science
First Semester, 2021-2022
ITCS214 / ITCS215 / ITCS216 (Data Structures)

Project

It is required to develop an application called **Library System** to maintain information about books and members in a library.

(1) Write a class called **Book** having the following data members:
title (String), author1 (String), author2(String), publisher (String), yearPublication (int), isbn (String), accessionNum (long), issuedTo (**LibMember**).
Include following public methods in the class:
Default constructor (without any parameters), constructor with 6 parameters (except for issuedTo – initialize issuedTo to null), set and get methods for all attributes, equals and toString.
NOTE: The ISBN is a unique 13 digit number assigned by the publisher. It is like barcode. As there can be multiple copies of a book in a library, the libraries assign a unique number, called accession number, to every book in the library to keep track of the books in the library. Assume that accession number starts from 1001.

(2) Write a class called **LibMember** having the following data members:
firstName (String), lastName (String), gender (char), cprNum (long), teleNum (String), booksIssued (array of type **Book**, size = 10) and numBooksIssued (int).
Include following public methods in the class:
Default constructor (without any parameters), constructor with 5 parameters (except for booksIssued and numBooksIssued – initialize nubBiiksIssued to 0), set and get methods for all attributes, equals and toString.

(3) Write a class called **LibrarySystem** having the following data members:

booksList:	list of books – LinkedList object, each element is of type **Book**,
membersList:	list of members – LinkedList object, each element is of type **LibMember**,
booksListSize:	actual number of objects of type **Book** in booksList,
membersListSize:	actual number of objects of type **LibMember** in membersList.

Include following methods in this class:

(i) Constructor without any parameters. Create booksList and membersList as empty linked lists and initialize booksListSize and membersListSize to 0.

(ii) addBook: inserts a new **Book**object at the end of the booksList. Object of type **Book** is passed as parameter. If the object already exists in the list, then do not add the object and return false, else return true after successfully adding the Book.

(iii) deleteBook: deletes a **Book** object from booksList. Accession number of the **Book** is passed as parameter. If the Book is issued to a member or the object is not found in the booksList, then the Book cannot be deleted and the method returns false, else the method returns true, after successfully deleting the object.

(iv) addMember: inserts a new **LibMember** object at the end of the membersList. Object of type **LibMember** is passed as parameter. If the object already exists in the list, then do not add the object and return false, else return true after successfully adding the member.

(v) deleteMember: deletes a **LibMember** object from membersList. CPR number of the **LibMember** is passed as parameter. If any Book is issued to the member or the object is not found in the membersList, then the member cannot be deleted and the method returns false, else the method returns true, after successfully deleting the object.

(vi) searchBook: searches the booksList by accessionNum, passed as parameter. If the object is found, it returns its location in the booksList, else returns -1.

(vii) searchMember: searches the membersList by cprNum, passed as parameter. If the object is found, it returns its location in the membersList, else returns -1.

(viii) isEmptyBooksList: returns true if the booksList is empty, else returns false.

(ix) isEmptyMembersList: returns true if the membersList is empty, else returns false.

(x) sizeBooksList: returns instance variable booksListSize.

(xi) sizeMembersList: returns instance variable membersListSize.

(xii) issueBook: accepts accession number of a Book as the first parameter and the CPR number of the member as the second parameter. The Book can be issued to a member only if:

(a) The Book exists in the booksList, (b)The member exists in the membersList,

(c) The Book is not issued to any member,

(d) The member has less than 10 books issued to him/her.

If the book can be issued to a member, then add the Book object in the booksIssued array for the member, increment numBooksIssuedby 1 for the member and also make issuedTo instance variable of the Book object reference to the member and return true. If the Book cannot be issued then return false.

Note that there will be cross referencing between the book and the member.

(xiii) returnBook: accepts accession number of a Book as the parameter. The Book can be returned only if:

(a) The Book exists in the Books list, (b) The Book is issued to a member.

If the Book can be returned, then delete it from booksIssued array for the member, decrement numBooksIssued by one for the member and also make issuedTo instance variable of the Book object to null and return true. If the Book cannot be returned, then return false.

(xiv) printBooksIssued: accepts CPR number of a member as a parameter and prints details of all the books issued to the member.

(xv) isBookIssued: accepts accession number of the Book as parameter. It returns true if the Book object exists in the bookList and is issued to a member, else returns false.

(4) Write a class called **LibraryMain** having only main method to test all the functionalities of class **LibrarySystem.**

Write main as a menu driven program using switch statement.

NOTE: For booksList and membersList, use **LinkedList** built-in class of java. It is a doubly linked list with ListIterator.Write detailed comments for each class and method. Write comments using Javadoc.