Cairo University
Faculty of Engineering
CMP(N) 306
Advanced Programming Techniques

Socket Programming in Java

Requirement:
In this requirement, we will implement a **connection-oriented** network communication with socket programming using server-client architecture for a library system. Clients can ask the server to borrow books from the library or return the borrowed books.

Define the following classes:

1. **Class Book:** contains the following data members: <u>isbn</u>, <u>title</u>, <u>author</u>, a boolean called <u>borrowed</u> indicating if the book is borrowed or not, and a string called <u>borrower</u> containing the client's name that currently borrowing the book.

2. **Class Library:** contains <u>an array of books</u> as data member and the following functions:
    a. **Constructor**: reads books' data from file and fills the array of books
    b. **Borrow**: takes the client's name and the isbn, and borrows this book to this client if possible. The function updates the book data members appropriately.
    c. **ReturnBack**: takes the client's name and the isbn, and returns this book to the library if it is borrowed by the passed client name. The function updates the book data members appropriately.
    d. **Print**: prints the <u>isbn</u> of all books and the name of the <u>borrower</u> of each

3. **Class LibClient**:
    a. Prompts the user to enter his client's name
    b. Opens two sockets with the server, one on the borrow port 6666, and the other on the return port 6667
    c. Sends the client's name to both sockets
    d. In a loop,
        i. Prompts and asks the user if he wants to borrow or return
        ii. Reads the isbn of the book to borrow (or return) from the user
        iii. Sends the isbn to the appropriate socket
        iv. Reads the response message from the server
        v. Prints the response message to the user on console

4. **Class LibServer**:
    a. It contains a library object that contains many books
    b. The server should be able to take requests from clients **in parallel** on both ports such that the server can borrow books for many clients in parallel, return books from many clients in parallel, borrow books and return books from many clients in parallel.
    Use multithreading for that.

c. **For serving a borrow request coming on the borrow port 6666:**
    i. Reads the client’s name from the client socket
    ii. Reads the isbn sent from the client socket
    iii. Calls function borrow of class library to borrow the required isbn to the client name
    iv. Sends a response message to the client socket with a descriptive message indicating whether:
        1. the borrow is done successfully (Success)
        2. the isbn is not found in library (Failure)
        3. the isbn is already borrowed (Failure)
    v. Calls function print of the library object to print it on the server console

d. **For serving a return back request coming on the return back port 6667:**
    i. Reads the client’s name from the client socket
    ii. Reads the isbn sent from the client socket
    iii. Calls function returnBack of class library to return back this isbn from the client name to the library
    iv. Sends a response message to the client with a descriptive message indicating whether:
        1. the return back is done successfully (Success)
        2. the isbn is not found in library (Failure)
        3. the isbn is not borrowed (Failure)
        4. the isbn is borrowed by a different client name (Failure)
    v. Calls function print of the library object to print it on the server console