

Data Structure Assignment 2

Library book Management System

```
import java.util.Scanner;
```

```
class Book {
```

```
    int bookID;
```

```
    String bookTitle;
```

```
    String AuthorName;
```

```
    String Status;
```

```
    Book next;
```

```
    public Book (int id, String title, String author,  
String status) {
```

```
        this .bookID = id;
```

```
        this .bookTitle = title;
```

```
        this .authorName = author;
```

```
        this .status = status;
```

```
        this .next = null;
```

```
}
```

```
}
```

```
class Booklist {
```

```
    private Book head;
```

```
    public void insertBook (int id, String title,  
String author, String status) {
```

```
        Book newBook = new Book (id, title,  
author, status);
```

```
        if (head == null) {
```

```
head = newBook;  
} else {  
    Book temp = head;  
    while (temp.next != null)  
        temp = temp.next;  
    temp.next = newBook;  
}
```

```
System.out.println("Book added  
successfully.");  
}
```

```
public void deleteBook (int id) {  
    if (head == null) {  
        System.out.println("No book in  
the library.");  
        return;  
    }  
    if (head.bookID == id) {  
        head = head.next;  
        System.out.println("Book deleted  
successfully.");  
        return;  
    }  
    Book temp = head;  
    while (temp.next != null && temp.next  
bookID != id)  
        temp = temp.next;  
    if (temp.next == null) {  
        System.out.println("Book deleted  
successfully.");  
    }  
}
```

```
public Book searchBook (int id) {  
    Book temp = head;  
    while (temp != null) {  
        if (temp.bookID == id)  
            return temp;  
        temp = temp.next;  
    }  
    return null;  
}
```

```
public void displayBooks () {  
    if (head == null) {  
        System.out.println ("No Books available.");  
        return;  
    }  
    Book temp = head;  
    System.out.println ("--- Current Books in  
library ---");  
    while (temp != null) {  
        System.out.println ("ID: " + temp.bookID  
+ " | Title: " + temp.bookTitle + " | Author:  
" + temp.authorName + " | Status: " + temp.  
status);  
        temp = temp.next;  
    }  
}
```

```
class TransactionStack {  
    private java.util.Stack < Transaction >  
    stack = new java.util.Stack<>();
```

```
public void push (String action, int bookID) {  
    stack.push (new Transaction (action,  
        bookID));  
}
```

```
public Transaction pop () {  
    if (stack.isEmpty ())  
        return null;  
    return stack.pop ();  
}
```

```
public void viewTransactions () {  
    if (stack.isEmpty ()) {  
        System.out.println ("No Transactions  
yet.");  
    }  
    return;  
}
```

```
System.out.println ("--- Transaction  
History ---");  
for (Transaction t : stack) {  
    System.out.println (t.action + "  
Book ID :" + t.bookID);  
}
```

```
}
```

class LibrarySystem {

```
    BookList bookList = new BookList ();
```

```
    TransactionStack transactionStack = new  
    TransactionStack ();
```

```
    public void issuebook (int id) {  
        book b = booklist.searchBook (id);
```

```
if ( b == null ) {  
    System.out.println("Book not found!");  
    return;  
}
```

```
if ( b.status.equalsIgnoreCase("Issued") ) {  
    System.out.println("Book is already  
    issued!");  
    return;  
}
```

```
b.status = "Issued";  
transactionStack.push("Issue", id);  
System.out.println("Book Issued  
Successfully.");
```

```
public void returnBook(int id) {
```

```
    Book b = bookList.searchBook(id);  
    if ( b == null ) {  
        System.out.println("Book not found!");  
        return;  
    }
```

```
    if ( b.status.equalsIgnoreCase("Available") )
```

```
{
```

```
    System.out.println("Book already Available");  
    return;
```

```
}
```

```
b.status = "Available";
```

```
transactionStack.push("Return", id);  
System.out.println("Book returned  
Successfully.");
```

```
}
```

```
public void undoTransaction() {
```

```
    Transaction last = transactionStack.pop();
```

```

if (last == null) {
    System.out.println("No transaction to
    undo.");
    return;
}

```

```

Book b =
booklist.searchBook(last.bookID);
if (b == null) {
    System.out.println("Book record not
    found.");
    return;
}

```

```

if (last.action.equalsIgnoreCase("Issue"))
    b.status = "Available";
else if
(last.action.equalsIgnoreCase("Return"))
    b.status = "Issued",

```

```

System.out.println("Last transaction
undone successfully.");
}

```

```

public class LibraryBookManagementSystem {
    public static void main (String [] args) {
        Scanner sc = new Scanner (System.in);
        LibrarySystem system = new LibrarySystem();
    }
}

```

```

while (true) {

```

```

    System.out.println ("In --- Library Book
    Management System ---");
    System.out.println ("1. Insert Book");
    System.out.println ("2. Delete book");
    System.out.println ("3. Search Book");
}

```

System.out.println ("4. Display Books");
System.out.println ("5. Issue Book");
System.out.println ("6. Return Book");
System.out.println ("7. Undo Last Transaction");
System.out.println ("8. View Transactions");
System.out.println ("9. Exit");
System.out.print ("Enter choice:");
int choice = sc.nextInt();

switch (choice) {

case 1 :

System.out.print ("Enter Book ID:");
int id = sc.nextInt();
sc.nextLine();
System.out.print ("Enter Title:");
String title = sc.nextLine();
System.out.print ("Enter Author:");
String author = sc.nextLine();
System.bookList.insertBook(id, title, author,
"Available");
break;

case 2 :

System.out.print ("Enter Book ID to delete:");
System.bookList.deleteBook(sc.nextInt());
break;

case 3 :

System.out.print ("Enter Book ID to search:");
Book b =
System.bookList.searchBook (sc.nextInt());
if (b == null)
System.out.println ("Found → " +
b.title + " by " + b.authorName + "(" +
b.status + ")");

② else :

```
System.out.println ("Book not found.");
break;
```

Case 4 :

```
System.bookList.displayBooks();
break;
```

Case 5 :

```
System.out.print ("Enter book ID to issue:");
System.issueBook (sc.nextInt ());
break;
```

Case 6 :

```
System.out.print ("Enter book ID to return:");
System.returnBook (sc.nextInt ());
break;
```

Case 7 :

```
System.undoTransaction ();
break;
```

Case 8 :

```
System.transactionStack.createNewTransactions ();
break;
```

Case 9 :

```
System.out.println ("Exiting ...");
return;
```

default :

```
System.out.println ("Invalid choices");
```

}

}

3

3