Report to assignment 3

Maryam Soleimani Shaldehi

In this report I will be explaining every single class and the methods that are used in them.

Class : Book

In this class first I defined the needed public variables and then I declared the Book method to ask and get these variables as inputs.

Class : User

Here I defined two string variables as username and password.

Then we have a array which its type is a Book class and it has 3 indices. This will be used later to limit the user for borrowing more than 3 books.

Then I defined a public variable called “booksCount” to count the number of books that the user borrows and when it reaches 3 it won’t let the user to borrow more, unless the user returns the books she has already borrowed. (we will see how this part works in the later classes.)

Then I declared a method called “User” to asks for username and password and get them as inputs.

Class : Librarian

It was exactly declared the same as the User class.

Class : Library

First I declared an array called “theBooks” from Book type and the size of “BOOKcapacity”. Each index of this array can store the full information of a book that are gotten as inputs in the Book class including the name of the book , the author, publishing year , ISBN (which is a unique number for each book to be distinguished by) and the quantity of that book. Then we set a variable to count (it will be explained in the method that is actually used in)

Method : askingForBookCapacity

This method is called in the very first position in the “main” method and asks for a capacity for books to declare the size of “theBook” array so the system knows the maximum number of books that can be added and won’t let more than that to be added.

Method : compareBookObjects

This method will get two Book class type variables as parameters “b1” and “b2”. This method will check if each two books have the same names or ISBNs and will avoid to add a repeated book to our book list in the next method.

Methos : addBook

In this method every time a book is added to our array the “count” will increase by one and its limit is the value of variable “BOOKcapacity”. When the value of “count” becomes more than the value of “BOOKcapacity” the system outputs that no more books can be added. In the for loop it is checked if a new book “b” which is a variable of type class “Book” wants to be added to our array is equal to any of them that already exist in our array(from index 0 to index “count”) so for checking that we call our “compareBookObjects” method and give it our parameters “b” and “theBooks[i]”(i is from 0 to “count”).

If the outcome is 0 it means that the ISBN or the name of those books are the same (because in t he compareBook method if any of those were the same the method returns 0 so when that method is equal to 0 it means that it returns 0 and the ISBN or the names of books are the same)

So if the book we want to add wasn’t repeated and the above statement didn’t happen it means that it can be successfully added to the array. So we assign the index number “count” of our array to this book and it will store the information related to the new book that is added and then the count variable will increase by 1 so later if any more books wanted to be added this book that just has been added will be compared with that to.

Method : searchByISBN

Here first we get an input which is the ISBN and define a variable called “flag” to keep track of the fact that the ISBN that the user enters actually exists or not. Then it goes through a loop and from the indices of array “theBooks” checks the “isbn” variables if the ISBN that is gotten from the output and the isbn in the array matches the whole information about that book will be shown at the order of: System.out.println("ISBN - Name - Author - Available Qty - Total Qty - publish year");

And then the value of “flag” increases by one. If it stays the same as default (0) it means that it did not go through the loop so there was no match for the ISBN so the following comment will be shown:

System.out.println("No Book for ISBN " + ISBN + " Found.");

Method : searchByAuthorName

It works the same as the previous method but we only need to replace “isbn” attribute with “authorName” because this method is search by author name not by ISBN.

Method : showAllBooks

It will loop through “theBooks” array and shows every information about the books by using (theBooks[i].attribute) for example : System.out.println(theBooks[i].isbn + "\t\t" + theBooks[i].bookName + "\t\t" + theBooks[i].authorName + "\t\t" + theBooks[i].bookQtyCopy + "\t\t" + theBooks[i].bookQty + "\t\t" + theBooks[i].publishYear);

Method : changeBookQty

This method will increase or decrease the number of books available. It gets a variable called ISBN as input and again loops through array the find a match for that. When it finds one then it inputs a number so the quantity of the books would be increased or decreased by (int addingQty = input.nextInt();)

And also : System.out.println("Enter Number of Books to be Added or Removed \n ATTENTION: if you want to increase just type a positive number like \n2 and if you want to decrease write a negative number like -2 ");

Method : doesBookExists

This method gets variable “ISBN” and loops through “theBooks” array and:

```
if (ISBN == theBooks[i].isbn){

            if (theBooks[i].bookQtyCopy > 0){

               System.out.println("Book exists.");

               return i;
            }

            System.out.println("Book does not exist");

            return -1;
         }
      }

      System.out.println("No Book of ISBN " + " Available in Library.");

      return -1;
   }
```

This part is obvious and I have already explained important parts of that .

Method : removeBook

Here I define a variable “bookIndex” : int bookIndex = doesBookExists(ISBN);

This is what doesBookExists returns. If it is equal to anything except -1 it means that the book exists and there are plenty of that available so and what it is equal to is the index of that book in the array.

theBooks[bookIndex].bookQtyCopy--; : here it removes the book from the array and in the next line it shows the whole information of the removed book.

Method : returnBook

Here if the book existed in the array bookQtyCopy will be increased and the book will be restored in the array again.

(this method is kind of the reverse form of the previous method)

Class : users

```
public int USERcapacity;
Scanner input = new Scanner(System.in);
User theUsers[] = new User[USERcapacity];
```

Works the same as:

```
public int BOOKcapacity;
Book theBooks[] = new Book[BOOKcapacity];
public static int count;
```

that is explained in Library class.

Method : askingForUsersCapacity

Works the same as askingForBookCapacity but instead of books it is for users.

Method : addUsers

Works the same as addBooks but instead of books it is for users.

Method : showAllUsers

Works the same as showAllBooks but instead of books it is for users.

Method : doesUserExist

Works the same as doesBookExist but instead of books it is for users.

Method : rentBook

Here first we declare a variable called “userIndex” and puts it equal to the return of “doesUserExist”.

If it is not equal to -1 it means that the user exists and can borrow book.Then it shows all the books :

```
book.showAllBooks();
```

the act of renting makes the book get removed so here we use: Book b = book.removeBook();

```
if (theUsers[userIndex].booksCount <= 3){

    System.out.println("adding book");

    theUsers[userIndex].borrowedBooks[theUsers[userIndex].booksCount] = b;

    theUsers[userIndex].booksCount++;
```

Book borrowedBooks[] = new Book[3]; : this line in the User class means that the user can not rent more than 3 books so this piece of code says:

If the number of books that the user rented are less than 3 the user can borrow the book and the number of books that user has will be increased by 1 : theUsers[userIndex].booksCount++;

And else the user can not rent more books.

Method : returnBook

It will loop through the array theUsers[userIndex]

```
for (int i = 0; i < u.booksCount; i++){

    System.out.println(u.borrowedBooks[i].isbn + " --- " +
u.borrowedBooks[i].bookName + " --- " + u.borrowedBooks[i].authorName);

}
```

here it shows the full info of borrowed books in the order.

```
int ISBN = input.nextInt();

for (int i = 0; i < u.booksCount; i++){

    if (ISBN == u.borrowedBooks[i].isbn){

        book.returnBook(u.borrowedBooks[i]);

        u.borrowedBooks[i] = null;

    }

}
```

here it gets the ISBN and try to find a match for that in the borrowedBooks array and if a match found

it will be returned to the theBooks and its place in borrowedBooks will be empty: u.borrowedBooks[i] = null;

Class : Main

```
Library bookCapacity = new Library();

bookCapacity.askingForBookCapacity();

users usersCapacity = new users();

usersCapacity.askingForUsersCapacity();
```

Here by calling these methods and using these classes we ask the user to enter the capacity of books and users for the size of related arrays.

```
Library ob = new Library();

users obUser = new users();
```

Here we make variables from these classes to call methods later by them in the switch statement.

Then as long as choice is not equal to 0 the program keeps showing menu and getting input choice.

And then in each case we call the related method as displayed in menu which is really obvious.