

The God We Trust



# 3rd Assignment Report

<Mitra Dehghan> <98222038>

## Introduction:

هدف کلی این پروژه ساخت یک برنامه جاوا می باشد که امکانات مختلف یک کتابخانه رو به نمایش گذاشته و قابل استفاده برای کاربرهای مختلف است .

کلیات برنامه به این شکل میباشد که توسط دو تایپ کاربر (معمولی و کتابدار) قابل اجراست که هرکدام باتوجه به نقششان دسترسی متفاوتی دارند مثلا کاربر معمولی میتواند کتاب قرض بگیرد یا پس بدهد (همچنین اگر از ثبل اکانت نداشته باشد ، قابلیت ساخت اکانت جدید نیز برای او فراهم است) و در طرف دیگر کتابدار میتواند کتاب ها و کاربران کتابخانه و حتی دیگر کتابداران را کنترل کند (میتواند به آنها افزوده یا از آنها کم کند ، بین آنها جستجو انجام دهد و یا در صورت نیاز بروز رسانی هم انجام دهد .)

## Design and Implementation:

ابتدای کار باتوجه به مورد ذکر شده در فایل read me برای هر کلاس متغیرهای مورد نظر را تعریف کردیم و getter setter های آنها را نیز ساختیم .

### کلاس Book :

ابتدا متغیرهای name, author, year of publish, isbn که در توضیحات سوال هم ذکر شده بودند و متغیر bookCounter برای استفاده در شمارش تعداد کتابها (که جلوتر در متود های increasing و decreasing به کارمان می آید) را تعریف کردیم و برای هریک setter و getter را نوشتیم.

بعد از آن دو متود increasing و decreasing را برای شمارش کتابها نوشتیم.

### کلاس Librarian:

ابتدا متغیرهای librarian user و librarian password (نام کاربری و رمز عبور کتابدار) را تعریف کردیم و طبق توضیحات مسئله متودهای adduser /

addLibrarian / addbook / removeuser / remove librarian /  
removebook را مشابه با متودهایی که در کلاس Library تعریف کرده بودیم  
نوشتیم و علاوه بر آن سه کلاس / searchuser / searchlibrarian /  
searchbook را هم داشتیم که هر کدام از آنها خروجی که متودهای search  
کلاس Library برای یک ابجکت خاص میگیرند را به ما برمیگردانند .

### کلاس User:

مشابه با کلاس Librarian ، متغیرهای username و password و علاوه بر آن  
لیستی به نام usersBooks برای ذخیره سازی کتابهای هر کاربر نوشتیم.  
کلاس Library :

در این کلاس سه لیست با نامهای users و librarians و booklist به ترتیب برای  
ذخیره سازی اطلاعات کاربران ، کتابدارها و کتابها ساختیم .  
بعد از تعریف کردن تمام getter و setter ها به سراغ بدنه متودهای خواسته شده  
رفتیم در این کلاس به طور کلی متودها برای یکی از سه دسته ی "کتاب ، کتابدار و  
کاربر" تعریف شدند ، مثلاً برای دسته اول (کتاب) متودهای addBook و  
removeBook و updateBook و searchBook را داشتیم که هر کدام به ترتیب  
برای اضافه کردن کتاب به کتابخانه ، حذف کردن کتاب از کتابخانه ، بروزرسانی کتاب  
در کتابخانه و جستجو کردن برای یک کتاب در کتابخانه به کار میروند که برای همه این  
متودها باید در آرگومانهای اول یک ابجکت از جنس همان کلاس Book میساختیم و  
بعد ادامه میدادیم برای دومتود اول یعنی اضافه کردن و حذف کردن که کافی بود  
لیست booklist را صدا زده و بلافاصله بعد آن با زدن add(). و قرار دادن ابجکت  
ساخته شده درون پرانتز کتاب مورد نظر را اضافه کنیم و مشابه آن برای  
removeBook با استفاده از remove(). انجام شد و در ادامه کار دقیقاً همین  
اضافه و حذف کردن را برای کاربر و کتابدار هم داشتیم که عملیات بطور مشابه تکرار  
میشد .

برای متود `searchBook` هم ابتدا ابجکتهی از جنس همان `Book` تعریف کردیم و با استفاده از یک حلقه `while` محتویات درون لیست `bookslist` ای که داشتیم را چک میکردیم به این شکل که تک به تک در عناصر داخل لیست جلو میرویم و آن را با ابجکتهی که ساختیم قیاس میکنیم و اگر برابری ای رخ داد آن را برمیگردانیم . و دوباره مشابه با این متود ، متودهای `searchUser` و `searchLibrarian` را هم کامل کردیم .

سه متود دیگر باقی ماند که آنها هم در ارتباط با کتاب بودند ، مورد اول `doesBookExist` که برای چک کردن وجود کتاب به کار میرود میباشد که حقیقتاً خیلی دلیل الزام وجود این متود را متوجه نشدم چون بنظرم میتوانستیم با همان متود `searchBook` به خواسته مطلوب خود در اینجا برسیم (مشابه این متود را برای کاربر و کتابدار هم داشتیم)

و در آخر دو متود `increaseBook` و `decreaseBook` که برای این کار استفاده شدند که بتوانیم به ترتیب زمانی که کاربری کتابی را به کتابخانه پس میدهد ، تعداد آن کتاب را مجدداً در کتابخانه افزایش دهیم و زمانی هم که کتابی قرض گرفته میشود تعداد آن را در کتابخانه کاهش دهیم که این کار توسط یک حلقه `for` به سادگی با استفاده از دو تابع `increasing` و `decreasing` که با استفاده از شمارنده `bookCounter` در کلاس `Book` تعریف شده بودند ، نوشته شد .

## کلاس Main :

در گام آخر به سراغ نوشتن کلاس `Main` رفتیم که اینجا ابتدا یک کتابدار پیش فرض در سیستم ثبت میکنیم که خودمان یک نام کاربری و رمزعبور برای او ست کرده ایم ، سپس برنامه شروع میشود و پیغامی نمایش داده میشود که از کاربر میخواهد هویت خودش را تعیین کند که یک کتابدار است یا یک کاربر معمولی و بعد از آن یک `switch case` کلی طراحی شده که دو `case` دارد (کاربر یا کتابدار) و یک `default` برای حالتی که کاربر ورودی نامفهومی غیر از موارد خواسته شده را بزند .

در case u (کاربر معمولی) از کاربر خواسته میشود که اگر اکانتی تا بحال ساخته پاسخ بله را تایپ کند و در غیر آن هرچه شد تایپ کند تا به مرحله بعدی رفته و اکانت خود را بسازد .

حال اگر کاربر یک کاربر جدید باشد وارد متود `creatAccount()` میشود تا بتواند اکانت خود را بسازد از آنجایی که ممکن است کاربر هم یک کاربر معمولی باشد که نیاز به کانت دارد و هم یک کتابدار این متود برای هر دو آنها طراحی شده برای همین مجدداً ابتدای کار پرسشی داریم که هویت کاربر مشخص شود و بعد از آن باتوجه به پاسخ مورد نظر از یکی از دو کلاس `User` و `Librarian` یک ابرجکت جدید ساخته میشود و به لیست مربوطه اضافه میگردد .

و پس از این مرحله در خط بعدی متود `runMenuUser` صدا زده میشود که نحتویات ان شامل دسترسی هایلیست که یک کاربر معمولی میتواند داشته باشد (گرفتن یا پس دادن کتاب) و باتوجه به نیاز کاربر متود مدنظر روی کتابی که کاربر آن را انتخاب کرده اعمال میشود .

مشابه با `case u` ، `case l` هم برای کتابدار تعریف میشود با این تفاوت که در اینجا با متود `runMenuLibrarian` سر و کار داریم مه شامل دسترسی های یک کتابدار است که توسط یک `switch case` کلی برای سه دسته کتابها و کتابدارها و کاربران طراحی شده که هر `case` ان مربوط به یکی از سه دسته بالاست و در هر `case` دوباره یک `switch case` جدید تعریف میشود که `case` های آن هر کدام برای یکی از قابلیت های `search / update / add / remove` می باشد .

چند مورد که بنظرم بهتر بود یادداشت کنم :

یه مشکل اساسی که توی `switch case` های `runMenu` به آن برخورد کردم این داستان بود که مثلاً وقتی کتابداری میخواست کتابی رو اضافه کند باید اسم کتاب رو وارد کرده و از طرفی این کتاب باید از جنس همان `Book` باشد تا به درستی لینک شود و بتوان کاری که مدنظر است را اعمال کرد اما موفق به انجام اینکار نشدم و اگر

هم بیاییم و کتاب ر فرضا از جنس String ذخیره سازی کنیم در کارکرد متودهای مربوطه به مشکل میخوریم و با خطای ناهماهنگی تایپ آنها مواجه میشویم .  
از طرفی ایده کلی برای ورود کاربر یا کتابدار (کسی که از قبل اکانت دارد) را اینگونه در ذهن داشتیم که با یک متود login بتوان اطلاعات را وارد کرد و به داده های مورد نظر وصل شد اما در نوشتن این متود به مشکلی خوردم برای همین محتویات آن خالیست .