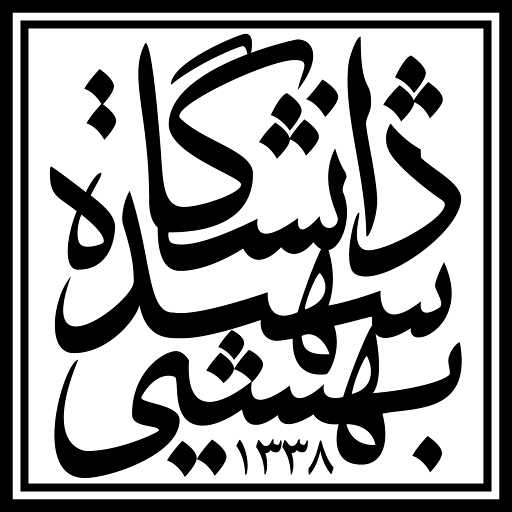
The God We Trust



3th Assignment Report

<محمد مهدی قلاوند> <98222121>

Introduction:

* یک برنامه جاوا شی گرایی که مدیریت یک کتابخانه را برعهده دارد.
* اهداف این تکلیف پیاده سازی مفاهیم ابتدایی شی گرایی می باشد
* این پروژه داری 4 کلاس اصلی به نام های Book و User و Library و Librarian میباشد .از طریق کلاس Librarian که در آن کلاس Library اضافه شده است میتوان به لیست Book و لیست User و... دسترسی پیدا کرد حتی می توان متد هایی برای اضافه کردن User و Book در آن پیاده سازی شده است که بتوان تغییراتی در آنها ایجاد کرد

Design and Implementation:

* در این تمرین دو کلاس Book و User وجود دارد که ارتباط یک به چند بین آنها برقرار است که هر User میتواند چندین Book داشته باشد از طرفی یک کلاس Library و جود دارد که در آن لیست Book و User و Librarian وجود دارد و خود کلاس Librarian دارای دو ویژگی نام کاربری و رمز عبور میباشد .
* یک چالشی که وجود داشت این بود که به چه صورت میتوان تمام این کلاس ها دسترسی پیدا کرد وقتی که نخواهیم از هر کلاس یک موجودیت ایجاد کنیم .برای اینکار در کلاس Librarian از کلاس Library یک صفت ایجاد کردیم که بتوانیم به کلاس های دیگر و متد های آن دسترسی پیدا کنیم و تغییرات را ایجاد کنیم
* همچنین یک چالش برای احراز هویت User و Librarian داشتم که با ایجاد یک Token در کلاس های آنها و با مقدار دهی آن در هنگامی که کاربر یا کتابدار لاگین کرده بتوانم بفهمم کدام کاربر یا کتابدار الان احراز هویت شده اند .

Testing and Evaluation:

* یکی از مشکلاتی که برای من وجود داشت نحوه پیاده سازی منو کاربدی برای کاربر بود که بعد از هر عملیات خارج نشود تا زمانی که خود درخواست خروج از سیستم را بدهد برای این کار متد زیر را پیاده سازی کردم.

String[] options = {  
 "0- Exist Menu",  
 "1- Login User",  
 "2- LogOut User",  
 "3- Borrow Book",  
 "4- Return Book",  
 "5- Login Librarians",  
 "6- LogOut Librarians ",  
 "7- Add Book ",  
 "8- Remove Book ",  
 "9- Create User ",  
 "10- Book Report ",  
 "11- Librarian Report ",  
};

Scanner scanner = new Scanner(System.*in*);  
int option = 1;  
while (option!=0){  
 *runMenu*(options);  
  
 try {  
 option = scanner.nextInt();

}  
catch (InputMismatchException ex){  
 System.*out*.println("Please enter an integer value between 1 and " + options.length);  
 scanner.next();  
}  
catch (Exception ex){  
 System.*out*.println("An unexpected error happened. Please try again");  
 scanner.next();  
}

Conclusion:

* قبل از ظهور برنامه نویسی شی گرا ، برنامه‌ها اغلب به شکل Procedural یا رویه ای نوشته می‌شد. در برنامه نویسی رویه ای برنامه‌ها به توابع تقسیم می‌شدند. یعنی برنامه از متغیرهایی که اطلاعات را ذخیره و توابعی که عملیات مورد نظر را روی اطلاعات انجام می‌دادند تشکیل می‌شد.

اما با بزرگ شدن نرم افزار برنامه نویسان مجبور به تکرار کدها به صورت مکرر می‌شدند. توابع در قسمت‌های مختلف به صورت نامرتب و تو در تو یکدیگر را صدا می‌زدند و برنامه به اصلاح به یک کد Spagetti تبدیل می‌شد. کد اسپاگتی به کدی اطلاق می‌شود که در آن توابع و قطعات کد مختلف به هم وابسته و در هم ریخته اند و در نتیجه ایجاد تغییر در یک تابع باعث به هم ریختن توابع زیادی می‌شود. پس از ظهور برنامه نویسی شی گرا توابع و متغیرهای مرتبط در واحد هایی به اسم کلاس (Class) تجمیع شدند. اشیا از روی این کلاس‌ها ساخته می‌شوند. و به این ترتیب صاحب متدها (Method) و ویژگی‌های (Attribute) آن کلاس‌ها می‌شوند.