

سلام! اومدم در مورد مسیری که برای نوشتن پروژه پایان ترم طی کردم بگم.

شروع.

خب اول از همه چند دور از روی فایل پروژه خوندم تا یه شکل ذهنی کلی ازش داشته باشم. ولی فهمیدم از چیزی که فکر میکردم پروژه طولانی‌تره! خب اول از همه یه فلوچارت کلی از پروژه نوشتم برای خودم.

۱. چه فایل‌هایی می‌خوایم؟ اینجا از فایل منظورم کلاسه. کلاس‌های داخل کد من حکم دیتا استراکچر دارن و تابع‌ها هم بک‌اند و فرانت قضیه داخل فایل اصلی مدیریت میکنن.

- کاربر
- فروشنده (ارث‌بری از کاربر)
- ادمین (ارث‌بری از کاربر)
- محصول
- فروشگاه
- دسته بندی

پس به طور کلی فهمیدم چه کلاس‌هایی نیاز دارم. پس فایل‌ها و کلاس‌ها ساختم. ولی خب فعلا توشون خبری نبود. باید آروم آروم کاملشون کنم.

۲. هر کلاس چه چیزایی ذخیره میکنه؟ این دومین سوال بود. چون باید آروم آروم کلاس‌ها بسازم.

- کاربر : یوزرنیم - پسورد - ایمیل - شماره تلفن - آدرس - سبد خرید - سبد انتظار - خریدهای نهایی - کیف پول
- فروشنده : اسم شرکت - پسورد - ایمیل - تلفن - آدرس کیف پول - محصولات - مجوز
- ادمین : یوزرنیم - پسورد - ایمیل
- محصول : اسم - توضیحات - موجودی - نظرات - دسته‌بندی
- فروشگاه : اسم - آدرس - شماره تماس - یوزرها - ادمین‌ها - فروشنده‌ها - فروش‌ها - دسته‌بندی‌ها و خیلی چیزها (:)

حالا با توجه به اینکه هر کلاس چی می‌خواد و چی از کاربر می‌گیره برای ساخته شدن و ... تابع اصلی کلاس و جا برای get و set و ... گذاشته میشه که هر وقت نیاز شد به اطلاعاتی اونجوری برنامه بگیره. پرایوت کردن داده‌ها برای کپسوله‌سازی هم داریم که این تسک پروژه هم انجام داده باشم مثلا (:)

خب بریم سراغ اصل کار!

۳. موقع شروع برنامه باید چه کارایی انجام بشه؟

- ساخت فروشگاه
- ساخت ادمین اصلی
- ساخت دسته‌بندی‌ها
- وارد منو شدن

خب با توجه به کلاس‌مون تا اینجا کار راحت نوشتیم. وارد بخش اصلی کار یعنی منوها شدیم. جایی که کاربر با برنامه تعامل میکنه و کلی تغییر و اتفاق روی دیتاها میوفته.

در کل این پروسه کد زدن سعی کردم از اسمای خوانا استفاده کنم تا خوانایی کد بیشترین باشه. چه برای کلاس چه تابع چه متغیر تا تمیز بودن کد ماکسیموم باشه! وقت نشد کامنت بذارم. ایشالا بعدا

۴. چه منوهایی داریم؟

- منوی اصلی شامل ورود و ساخت اکانت
- منوی ورود
- منوی ساخت اکانت
- منوی کاربر
- منوی ادمین
- منوی فروشنده

منوها و کلا کل پروسه برنامه برای اینکه تا ابد باز باشه داخل حلقه while همیشه درست قرار گرفت. البته گزینه بستن برنامه هم گذاشتم که حلقه شکسته شه. همچنین برای اینکه سر باگ‌ها برنامه بسته نشه و تست خراب نشه از try catch استفاده کردم!

با نوشتن هر بخش کد سعی میکردم که تستش کنم و جلو برم. پروسه تست کردن با اینکه تکرارش خسته کننده بود ولی خب اکثر مشکلات نشون میداد!

۵. بقیه منوها چی داشتن؟

- منوی اصلی : ورود - ثبت نام - خروج
- منوی ثبت نام : ثبت نام کاربر/ادمین/فروشنده
- منوی ورود : ورود کاربر/ادمین/فروشنده
- منوی کاربر: جست و جو - سبد خرید - نهایی کردن سبد خرید - شارژ اکانت - محصولات در انتظار تایید
- محصولات خریداری شده - موجودی - نوشتن کامنت - تغییر اطلاعات اکانت

- منوی ادمین: تایید فروشنده‌ها - تایید خریده‌ها - تایید شارژ اکانت - نمایش کاربرها/فروشنده‌ها/سود فروشگاه/فروش‌ها - اضافه کردن ادمین

- منوی فروشنده: اضافه کردن محصول - نمایش محصول‌ها - حذف محصول - کیف پول - فروش‌ها

برای طراحی منوها از منوی اول شروع کردم و به پایین اومدم. و برای هر بخش تابع طراحی میکردم و با پاس دادن فروشگاه و یا اطلاعات مد نظر به تابع کار رو انجام میدادم و به کاربر نشون میدادم نتیجه رو.

۶. پروسه خرید چگونه؟

کاربر سرچ میکنه. محصول مشاهده میکنه. به سبد خرید اضافه میکنه. اکانت رو شارژ میکنه. ادمین شارژ تایید میکنه. کاربر سبد خرید ثبت میکنه برای خرید. ادمین خرید تایید میکنه. خرید به محصولات خریداری شده برای فروشنده و کاربر و فروشگاه اضافه میشه.

این شکل کلی پروسه خرید بود. که نیاز داره قبلش فروشنده ساخته بشه. توسط ادمین تایید شه. محصول اضافه کنه و ...

حالا شروع میکنم از بالا اگر برای تابعی چالشی داشتم مینویسم!

۷. محصولات چگونه جست و جو میشن؟ اینکار با کمک رجکس انجام دادم. از قبل باهاش آشنا بودم و خب بهتر از بررسی دقیقاً برابر بودن. نتیجه بهتره هم داد.

۸. چک میشه که یوزرنیم تکراری ساخته نشه؟ بله. در بخش ثبت نام تک تک المان‌ها ورودی گرفته میشه دونه دونه و یوزرنیم هم چک میشه که برابر فرد قبلی نباشه. همین پروسه برای ادیت مشخصات اکانت هم انجام میشه. بهتره برای مقایسه دو String از equals استفاده بشه تا ==. چون به ارور میخوریم.

۹. کلاس‌ها موقع نوشتن تابع‌ها چی میشن؟ کلی اطلاعات از کلاس‌ها خونده میشه و استفاده میشه و بهشون اضافه میشه. برای همین کلی تابع add remove set get داخلشون مجبور شدم بسازم که بهم کمک کنن!

۱۰. از چه ساختارهایی مثل لیست و مپ و ... استفاده کردی؟ از لیست و مپ استفاده کردم. از لیست برای سری دیتاهای تکی. و برای مپ برای جفت‌ها. مثلاً لیست برای لیست

یوزرها/ادمین‌ها/فروشنده‌ها/محصولات/کامنت‌ها/فروش‌ها/در انتظار تاییدهای مختلف و مپ برای ریکوئست شارژ شامل دوتایی یوزر و مبلغ و برای سبدهای خرید شامل دوتایی‌های محصول و تعداد اون محصول.

۱۱. چرا کلاس Sale رو ساختی؟ برای فروش‌های نهایی شده دیدم که برای هر فروش ما نیاز به دیتای خریدار و فروشنده و محصول و تعداد محصول خریداری شده در کنار هم هستیم. از چیزی فراتر مپ هست و نمی‌خواستم با کثیف کاری پیش برم پس این کلاس ساختم. این کلاس از کلاس محصول ارث‌بری کرده! تا اینجای کار تقریباً منوی یوزر کامل شد برام. بقیه تابع‌ها چالش خاصی نداشتن و ساده بودن. اما تعدادشون کم نبود.

۱۲. درخواست‌های تایید فروشنده چطوری مدیریت میکردی؟ مدیریت تایید خرید و شارژ حساب چی؟ همه‌ای این تاییدها داخل لیست‌های مخصوص‌شون ذخیره شدن تا ادمین و داخل پنل ادمین بره سراغشون و تاییدشون کنه. این لیست‌ها زیرمجموعه کلاس فروشگاه بودن تا راحت کل برنامه در دسترس باشن که باهاشون کار کنم!

۱۳. سیستم ورودی گرفتنت از کاربر چطور دیدی و طراحی کردی؟ تا حد ممکن سعی کردم که صرفاً گزینه‌های ممکن به کاربر نشون بدم و با کمک گرفتن گزینه اون عدد عملیات انجام بشه. اینجوری امکان غلط کم میشه. البته سعی میکردم برای خوانایی بیشتر برنامه در کل پروسه تعامل کاربر با برنامه از دستورات با متن واقعی استفاده کنم تا گیج نشیم و پیش بریم!

۱۴. اضافه کردن ادمین جدید پس کجا رفت؟ اونو گذاشتم داخل پنل ادمین. چون فقط باید یه ادمین بهش دسترسی داشته باشه! اضافه کردن اون به پنل ثبت‌نام صرفاً یه پروسه لاگین قبلش اضافه میکرد که باعث پیچیدگی بی‌فایده میشد!

۱۵. دستور برگشت به عقب و کنسل کردن داری؟ اره. تا جای ممکن سعی کردم دستور BACK داشته باشم. معمولاً این اتفاق با فرستادن 0 در طول برنامه انجام میشه. بعد تموم شدن منوی یوزر سراغ منوی ادمین و بعدش سراغ منوی فروشنده رفتم. پروسه زدن کدام اینجوری بود.

۱۶. چطوری گرفتن چیزای جدید مثل کاربر جدید یا محصول جدید یا ادمین جدید انجام میدادی؟ اینا با مرحله مرحله ورودی گرفتن انجام میشدن. اگه تیکه‌ای شرط خاصی هم داشت بهش اضافه میشد. سعی میکردم تا جای ممکن پیچیده نباشن.

۱۷. هر محصول کجا ذخیره میشد؟ هر خریدنهایی شده؟ اکثر دیتاها به جا ذخیره میشدن. منظورم به کلاسه. ولی به چیزی مثل محصول هم داخل کلاس فروشندش ذخیره میشد هم فروشگاه. خریدنهایی هم همینطور. این کار دسترس پذیری این شکل از دیتاها بهتر میکرد و جاهایی باعث میشد دقیقا همون چیزی که میخوایم دم دست باشه و مجبور نشیم روی کل به لیست فور بزنینم و فیلتر کنیمش. نکتش اینه برای حذف اون چیز باید از دو جا پاک کنیمش. البته همیشه پاک کردن کلاس به داخل کلاس برد و باعث شد که دیگه کلاس کار نکنه. ایده خوبیه!

۱۸. آیا به محصول هرجایی ذخیره شده باشه همون به محصوله؟ یعنی کپی هم نیستن؟ این در مورد همه دیتاهای کلاس گونه سوال دارم که آیا دو جا ریختن کپی شون میکنه یا تغییر در به جا در همه جا تغییرش میشده؟ مثلا ادیت اکانت کاربر توسط کاربر اون کلاس کاربر داخل فروشگاه هم تغییر میده؟

خب امیدوارم سوال خوب رسونده باشم. این ترس خودمم بود که جاوا چطور عمل میکنه برای کلاسها. میتونید با گرفتن هش از به کلاس در جاهای مختلف متوجه بشید که یکی هست یا کپی شده. من اینکارو نکردم و ریسک کردم و فهمیدم که جاوا تمایلی به کپی کردن نداره و مهم نیست به کلاس جاهای مختلف بریزی. تهش آدرسی که در حافظه داره یکسانه. یعنی پوینتر تغییر نمیکنه!

۱۹. دسته بندی ها چی شد؟ اینو خودمم گذاشتم آخر کار. تصمیم گیری به شکل پیاده سازیش خیلی سخت بود. تهش تصمیم گرفتن به کلاس براش بسازم که دو حالت داره. به کلاس عادی و به کلاس که به ورودی از نوع کلاس خودم هم داره. در واقع پیاده شکلی از class و subclass داخل به کلاس! بنظرم خیلی جالب شد. به سری دیتای اختیاری مخصوص هم میگیره که از جنس رشته ان. داخل کد میتونید ببینید و لذت ببرید! ترجیح دادم به اینکه کلاس مثلا موبایل و کتاب و ... بسازم و هارد کد کنم. تصمیم گیری اینکه کدوم بهتره با شما! البته کلاسها ورودی نگرفتم و خودم به دلخواه قبل منوی اصلی ساختم. ولی به راحتی میشه اضافه کرد به منوی ادمین و به به! حقیقتا هرچی از باحالیش بگم کم گفتم.

فراموش شده ها :

۲۰. تایید فروشنده چطوره؟ درخواستش میره داخل لیست داخل شاپ. تا اون زمان به متغیر داخل کلاس فروشنده هست که ۰ عه و نمیداره باهش وارد منو شه و میگه اکانت تایید نشده. و وقتی تایید شد اون ۱ میشه و راحت میتونه بره و بیاد :)

۲۱. چطوری پول تقسیم میشه؟ زمانی که ادمین خرید تایید میکنه ۹۰ درصد مبلغ هر محصول میره برای فروشنده و ۱۰ درصد هم میره توی حساب فروشگاه. از منوی همه هم موجودی قابل چک هست!

۲۲. تلاش برای تمیز شدن بیشتر کد؟ یه تابع print تعریف کردم که جای متن طولانی و سخت
System.out.print بگیره. سر فرصت تک تک جایگزین میکنم. میگن کدی که درسته دست بهش نزن. میترسم
برای ارائه خراب شه یا replace all میزدم:))))

۲۳. موجودی نداشت خرید لغو شه؟ به ادمین موجودی و مبلغ خرید نشون میدم. خودش تصمیم بگیره تایید و رد شه. همین.

۲۴. کامنت گذاشتن چطوریه؟ هر کاربر میتونه برای محصولاتی که خرید کرده کامنت بذاره. داخل منوی کاربر
بخش اضافه کردن کامنت هست. حقیقتا اینکه برنامه رو برای تست به جایی برسونم که کامنت بتونم بذارم
خستم کرد:)))

خب تقریبا تابعهای دیگه برنامه که منوها کامل میکردن چالش خاصی نداشتن. کلاسها هم کامل و کاملتر
میشدن. صرفا زیاد و طولانی بودن. منم تا حد ممکن جهت تمیزی تیکه تیکه شون کردم تا منوها قابل خوانش
باشن!

اما یه چالش آخر دارم. برنامه خیلی زشت بود. درسته توی ترمینال ران میشد ولی اینکه دستورای قبلی تلنبار
میشدن به شدت روی مخ بود. از قبل تجربه خوندن و دیدن دستورایی که ترمینال رو clear میکنن داشتم داخل
پایتون و ++c. (خود clear یه دستور کامندلاین یونیکسه). حالا با یه سرچ به دستورای مربوط بهش برای تمیز
کردن صفحه برای جاوا رسیدم! حالا گذاشتمش داخلش یه تابع و هرجایی که نیاز بود ترمینال پاک شه اون تابع
صدا زدم! (انتخاب اینکه بگی کجا صفحه پاک کن سخته! اونقدر بدیهی نیست!)

قطعا کد در بهترین حالت خودش نیست. کلی ایده و بهینه‌سازی براش دارم ولی خب در این زمان تلاشمو کردم
بهترین نتیجه بده! و از همین خوشحالم.

مرسی که خوندین (اگه خوندین:)

با تشکر. حسن.