

به نام خدا



دانشگاه صنعتی شریف

دانشکده مهندسی برق

برنامه سازی شی گرا

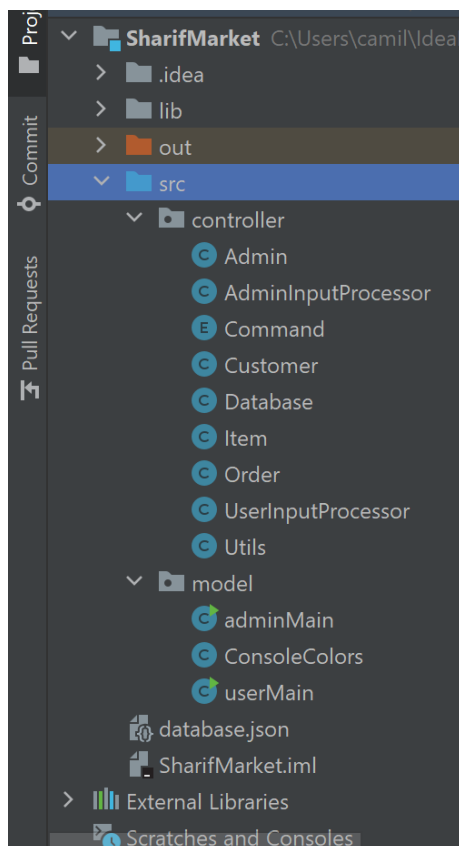
گزارش تمرین سوم – شریف مارکت

کاملیا کربلائی ۹۹۱۰۹۶۶۹

۱۴۰۰/۳/۲

کلاس‌ها:

ساختار برنامه به صورت زیر است:



برنامه دارای دو Main جداگانه برای برای فروشنده (Admin) و مشتری (Customer/User) می‌باشد.

کلاس Admin و کلاس Customer:

هر ادمین و همچنین هر مشتری دارای یک ID و یک password است که ID به صورت int و password به صورت استرینگ می‌باشد.

Command Enum:

Regex های دستورها در این enum قرار دارند.

کلاس Database:

کلاس Database کلاسی است که امور مربوط به خواندن از و نوشتن به فایل را انجام می‌دهد. برای ذخیره اطلاعات، از کتابخانه‌ی Gson استفاده شده است. فایل database.json در ابتدا به صورت زیر وجود دارد و با استفاده از برنامه، موارد درون آن بروزرسانی می‌شود.

```
database.json × AdminInputProcessor.java × Database.java ×
1 {"customers":[],"items":[],"deletedItems":[],"orders":[],"orderHistory":[],"admins":[]}
```

آنچه در فایل ذخیره می‌شود، ArrayList های customers, items, deletedItems, orders, orderHistory, admins و currentAdmin از جنس Admin و currentCustomer از جنس Customer است.

در کلاس Database، توابعی نیز برای اضافه کردن، حذف کردن و تغییر دادن اعضای ArrayList های مذکور وجود دارد.

* اگر فایل database.json تبدیل به null شده بود رشته‌ی زیر را در آن وارد کنید (نمی‌دانم این ارور چطور پیش می‌آید):

```
{"customers":[],"items":[],"deletedItems":[],"orders":[],"orderHistory":[],"admins":[]}
```

تابع‌های find:

تابع‌های find در کلاس‌های Admin, Customer, Item و Order با گرفتن ID یا در مواردی نام object، object منسوب به آن ID یا نام را return می‌کنند و در صورتی که objectی از آن کلاس با ID/نام مورد نظر وجود نداشته باشد null برمی‌گردانند.

کلاس Item:

هر Item دارای ویژگی های زیر می باشد:

```
public class Item {  
    private String name;  
    private int inStock;  
    private int buyingPrice;  
    private int sellingPrice;  
    private int ID;  
    private int ordersIn;  
    private int numberSold;  
    private int moneyMadeFrom;  
    private int itemProfit;  
}
```

- name: نام کالا
- inStock: تعداد موجود
- buyingPrice: قیمت خرید
- sellingPrice: قیمت فروش
- ID: شناسه ی کالا
- ordersIn: تعداد سفارش های این کالا
- numberSold: تعداد فروخته شده
- moneyMadeFrom: پول حاصل از فروش کالا
- itemProfit: سود حاصل از فروش کالا

کلاس Item علاوه بر تابع های find، دارای تابع های updateSales و چاپ کالاها نیز می باشد.

تابع updateSales در دستور checkout فرا خوانده می شود و اعمال زیر را انجام می دهد:

- تعداد orderهایی که از item خریداری شده وجود دارد را به علاوه ی ۱ می کند.
- تعداد فروخته شده در سفارش را به تعداد کل فروخته شده از آن کالا اضافه می کند.
- درآمد حاصل از آن سفارش را به درآمد کل کسب شده از فروش آن کالا اضافه می کند.
- سود کسب شده در سفارش را به سود کل حاصل از فروش آن کالا اضافه می کند.

نکته ی شایان توجه در مورد تابع های چاپ کالاها این است که عرض ستون ها بر اساس طول طولانی ترین عبارتی که قرار است در ستون نوشته شود تنظیم می شود.

کلاس Order:

هر Order دارای ویژگی‌های زیر است:

```
public class Order {  
    private int userID;  
    private String date;  
    private int itemID;  
    private int number;  
    private int orderID;  
    private String itemName;
```

- userID: شناسه‌ی خریدار
- date: تاریخ خرید
- itemID: شناسه‌ی کالای خریداری شده
- number: تعداد سفارش داده شده
- orderID: شناسه‌ی سفارش
- itemName: نام کالای خریداری شده

کلاس Order علاوه بر تابع‌های find، دارای تابع‌های چاپ سفارش‌های بررسی نشده و چاپ تاریخچه می‌باشد.

در تابع‌های چاپ سفارش‌ها و تاریخچه نیز مانند تابع‌های چاپ کالاها، عرض ستون‌ها بر اساس طول طولانی‌ترین عبارتی که قرار است در ستون نوشته شود تنظیم می‌شود.

کلاس Utils:

در کلاس Utils تعدادی تابع مفید قرار داده شده است که در دستورها و توابع مختلف برنامه، مورد استفاده قرار می‌گیرند.

کلاس ConsoleColors:

این کلاس قابلیت رنگی چاپ کردن متن در cmd را فراهم می‌آورد.

دستور لاگین، لاگوت و احراز هویت:

دستورهای login و logout برای Admin و Customer مشابه می باشد.

شناسه باید به صورت عدد صحیح وارد شود. سپس اگر شناسه قبلاً ثبت شده بود، برنامه از کاربر رمز درخواست می کند. کاربر ۳ بار اجازه ی وارد کردن رمز دارد و اگر هر سه بار غلط وارد کند، برنامه از دستور login خارج می شود و برای لاگین کردن بایستی مجدداً دستور login را وارد کند.

اگر شناسه قبلاً ثبت نشده بود (کاربر جدید)، برنامه از کاربر درخواست انتخاب رمز می کند. رمز به صورت String است. پس از وارد کردن رمز پیشنهادی توسط کاربر، برنامه از او درخواست می کند تا دوباره همان رشته را وارد کند (Confirm کردن رمز) و در صورتی که دو رشته برابر نبودند، کاربر باید مجدداً رمز پیشنهادی خود را وارد کند و تا وقتی رشته های اول و دوم برابر نشوند، این فرایند ادامه دارد.

متغیرهای currentCustomer/currentAdmin بعد از login، برابر کاربر وارد شده می شود.

در صورت وارد کردن دستور logout، اگر متغیرهای مذکور null نباشند، آنها برابر null می شوند.

برای ایجاد هر گونه تغییر در اطلاعات، Admin باید لاگین شده باشد.

دستور افزودن کالای جدید:

کالای جدید نمی تواند نامی برابر نام یکی از کالاهای موجود داشته باشد و همچنین قیمت فروش آن نباید کمتر از قیمت خرید آن باشد. در صورتی که این گونه input وارد شود، برنامه پیام خطا می دهد و کالای جدید افزوده نمی شود.

تابع حذف کالا و deletedItems ArrayList:

در صورت حذف کالا توسط ادمین، کالای حذف شده به deletedItems ArrayList اضافه می شود و از items ArrayList پاک می شود. ما به deletedItems نیاز داریم به این دلیل

که گاهی می‌خواهیم اطلاعات و آمار کالاهای حذف شده در اختیارمان باشد؛ مانند هنگامی که سود یا درآمد را می‌خواهیم محاسبه کنیم.

دستورهای ویرایش مشخصات کالا:

تابع‌های ویرایش مشخصات کالا به طرزی نوشته شده‌اند که نیازی به قابل قبول بودن تمام ورودی‌ها نمی‌باشد. برای مثال، اگر در تابع ویرایش نام و تعداد، تعداد وارد شده عدد قابل قبولی نباشد، بدون تغییر تعداد موجود، فقط نام آن تغییر می‌کند. این کار با استفاده از Boolean ها انجام شده است. به صورتی که برای مثال، یک `changeCount Boolean` داریم که در صورت نقض شدن شرایط عدد وارد شده برای موجودی برابر `false` می‌شود و عدد موجودی تغییر نمی‌کند.

دستور سفارش کالا:

برای سفارش کالا، مشتری باید لاگین شده باشد.

درون تابع سفارش کالا، یک Boolean به نام `setOrder` تعریف می‌شود که با نقض شدن شرایط ورودی، برابر `false` می‌شود.

دستور لغو سفارش:

برای لغو سفارش، همان مشتری‌ای که سفارش منسوب به اوست باید لاگین شده باشد. در هنگام لغو سفارش، تعداد کالاهایی که سفارش داده شده بود به موجودی کالا برگردانده می‌شود.

دستور `checkout`:

این دستور، تابع `updateSales` در کلاس `Item` را فرا می‌خواند. این تابع هم در `ArrayList` کالاهای پاک نشده هم پاک شده دنبال `Item` مورد خریداری می‌گردد تا تغییرات لازم بر اطلاعات کالا حتی در صورتی که توسط فروشنده `remove` شده باشد انجام گیرد. در نهایت نیز سفارش را به `orderHistory` اضافه می‌کند و از `orders` پاک می‌کند.

دستور چاپ سفارش‌های بررسی نشده و تاریخچه‌ی سفارشات:

این دستورها توابع `printOrders` و `printHistory` را از کلاس `Order` فرا می‌خوانند.

دستورهای محاسبه‌ی فروش و سود:

- دستور محاسبه‌ی سود کل: سود تک تک کالاهای موجود و حذف شده را جمع می‌کند.
- دستور محاسبه‌ی سود کالا: متغیر `itemProfit` کالای مورد نظر را چاپ می‌کند.
- دستور محاسبه‌ی آمار فروش کلی: آمار کالاهای موجود و حذف شده را چاپ می‌کند.
- دستور محاسبه‌ی آمار فروش یک کالا: آمار یک کالا را چاپ می‌کند.

دستور **exit**:

با وارد کردن کلمه‌ی “exit” برنامه به اتمام می‌رسد و `currentCustomer/` برابر `currentAdmin` قرار داده می‌شود.