

به نام خدا

تمرین ششم درس برنامه‌نویسی پیشرفته

نیم‌سال دوم ۱۴۰۱-۱۴۰۰

۱. تمامی فایل‌های کد را به همراه فایل متنی که در قالب pdf است (مورد سوم را بخوانید) به صورت یک فایل آرشیو zip (zip != rar) که به قالب زیر نام‌گذاری شده است، بارگذاری نمایید.

AP-HW6-FirstName_LastName-StudentNumber.zip

۲. در سوال‌هایی که ورودی و خروجی مطلوب آن‌ها مشخص شده است، رعایت نحوه ورودی گرفتن و نمایش خروجی اهمیت بسیاری دارد. دقیقاً همان‌طور که از شما خواسته شده است ورودی‌ها را خوانده و خروجی‌ها را تولید کنید. (سوالات دستی تصحیح می‌شوند اما لازم است که از فرمت ذکر شده تبعیت نمایید)

۳. پاسخ سوالات تشریحی را به صورت تایپ‌شده و در قالب یک فایل pdf (برای کل تمرین) تحویل دهید.

۴. در صورت مشاهده هرگونه تقلبی، طبق موارد گفته شده در قوانین درس برخورد خواهد شد.

۵. در صورت وجود هرگونه ابهام می‌توانید از طریق آدرس ایمیل «ap.winter2022@gmail.com» با تدریس‌یاران در ارتباط باشید.

۶. امکان آپلود تا دو روز پس از ددلاین، که با هر ساعت تاخیر یک درصد از سقف نمره قابل کسب، کسر می‌شود. برای مثال در صورت ارسال با ۱۰ ساعت تاخیر، سقف نمره قابل دریافت ۹۰ درصد است.

مهلت تحویل: تا چهارشنبه ۱۴ اردیبهشت ۱۴۰۱ ساعت ۲۳:۵۹ شب

صفحه

فهرست سوالات

سوال اول..... ۳

سوال دوم..... ۴

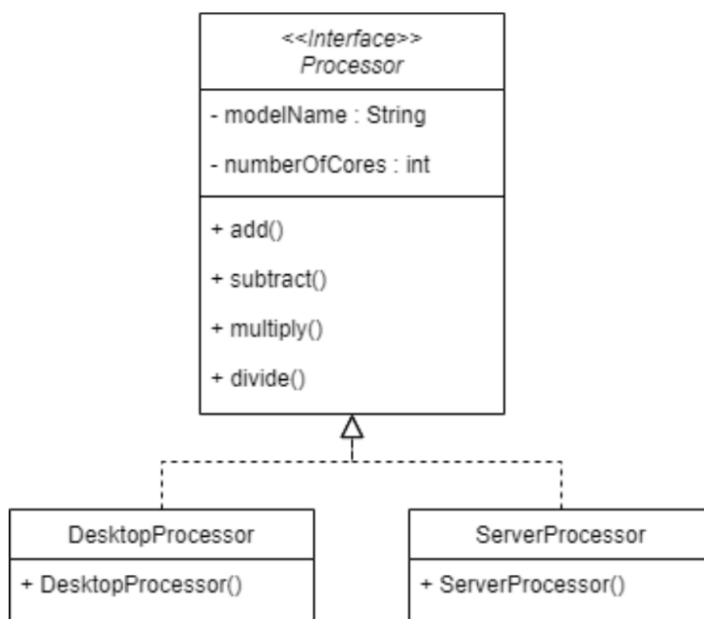
سوال سوم..... ۵

سوال چهارم..... ۶

سوال اول

به سوالات زیر پاسخ دهید:

- (۱) Design Pattern ها را به طور مختصر توضیح دهید و سه دسته کلی آن ها را نام ببرید.
 - (۲) برخی از فایده های استفاده از Design Pattern ها را توضیح دهید.
 - (۳) چند مثال از Design Pattern های استفاده شده در کتابخانه های Java JDK را همراه با کلاس مورد استفاده آن نام ببرید.
 - (۴) دو الگوی طراحی زیر را به اختصار توضیح دهید و نمودار کلاس^۱ آن ها را رسم کنید.
- Adapter
Factory Method
- (۵) نمودار کلاس زیر را در نظر بگیرید. این طراحی را با الگوی Factory Method پیاده سازی کنید و دیگرام جدید را رسم کنید (نوشتن کد لازم نیست).



¹ Class Diagram

سوال دوم

در این سوال، مراحل طراحی یک سیستم کتابخانه را بررسی می‌کنیم. با استفاده از مطالبی که در درس با آنها آشنا شدید (روش فعل‌ها/اسم‌ها، کارت CRC و ...) سعی کنید طراحی مناسبی برای پروژه ارائه داده و کارت‌های CRC هر قسمت را طراحی کنید. لازم به ذکر است که باید **Cohesion** و **Coupling** را رعایت کنید.

هر کتابخانه دارای اسم و آدرس است که می‌تواند لیستی از کتاب‌ها و اعضا داشته باشد و آن‌ها را چاپ کند. همچنین دارای لیستی از تمامی قرض‌های انجام شده است و می‌تواند آن‌ها را چاپ کند. کتابخانه می‌تواند یک عضو جدید (تکراری نبودن عضو باید چک شود) بپذیرد و یا یکی از اعضای قدیمی را از سیستم خود حذف کند.

کتابخانه می‌تواند کتاب جدیدی (تکراری نبودن کتاب باید چک شود) را اضافه و یا یک کتاب را از سیستم خود حذف کند. همچنین اگر عضوی کتابی را قرض بگیرد، این کتابخانه باید آن قرض را ذخیره کند و هنگامی که آن فرد کتاب را پس داد، قرض باید از سیستم حذف شود. همچنین اگر از مهلت قرض کتابی گذشته باید یک ایمیل برای فرد ارسال شود. کتابخانه قابلیت این را دارد که به دنبال کتاب خاصی بگردد. اطلاعات هر کتاب شامل عنوان و نام نویسنده است که می‌تواند آن‌ها را چاپ کند.

اطلاعات هر عضو شامل نام، ایمیل، شماره عضویت و تاریخ عضویت است که می‌تواند آن‌ها را چاپ کند. همچنین یک لیستی از قرض‌هایی که انجام داده است را نیز نگهداری می‌کند و اگر قرضی انجام یا تمام شود باید آن را ذخیره یا حذف کند. اطلاعات هر قرض شامل عضوی است که قرض متعلق به اوست، کتابی که قرض گرفته شده است و ۲ تاریخ که یکی زمان شروع قرض و دیگری زمان برگرداندن قرض است. برای نگهداری تاریخ تنها روز، ماه، سال و ساعت کافی است. می‌توان اطلاعات هر قرض را چاپ کرد.

پیاده‌سازی لازم نیست! (می‌توانید تنها کلاس‌ها و متدهای هر کدام را بنویسید و سپس از ابزار خودکار تولید نمودار کلاس استفاده کنید)

سوال سوم

در این سوال مراحل طراحی یک سیستم فروشگاه آنلاین را بررسی می‌کنیم. با استفاده از مطالبی که در درس با آن‌ها آشنا شدید (روش فعلها/اسمها، کارت CRC و ...) سعی کنید طراحی مناسبی برای پروژه ارائه داده و کارت‌های CRC هر قسمت را طراحی کنید. لازم به ذکر است که باید Cohesion و Coupling را رعایت کنید.

فروشگاه آنلاین دارای اسم و آدرس سایت است. این فروشگاه لیستی از محصولات و کاربران را دارد. هر کاربر یک سبد خرید و لیستی از تراکنش‌هایش را دارد. سبد خرید هر کاربر می‌تواند حاوی یک یا چند سفارش باشد و کاربر می‌تواند محتویات سبد خود را ببیند و یا محصولی را در صورتی که موجود است، سفارش دهد (به سبد خرید خود اضافه کند) و یا محصولی را از سبد خرید خود حذف کند و یا سبد خرید خود را نهایی و پرداخت کند که در صورت پرداخت، سبد خرید خالی می‌شود و محصول سفارش داده شده از محصولات فروشگاه کم می‌شود و یک تراکنش صورت گرفته و این تراکنش در صورت حساب‌های کاربر ثبت می‌شود و در نهایت ایمیلی مبتنی بر موفقیت خرید برای کاربر ارسال می‌شود.

اگر کاربری سبد خرید خود را نهایی و پرداخت نکرده باشد و در این مدت سفارشی از سبد خرید کاربر، در فروشگاه تمام شود این محصول باید از سبد خرید پاک شود.

این فروشگاه یک کاربر ادمین دارد که می‌تواند محصولی را در صورت تکراری نبودن اضافه کند و یا آن را از لیست محصولات حذف کند. همچنین کاربر ادمین می‌تواند کاربری را حذف کند. اطلاعات ادمین شامل نام، ایمیل و رمز عبور است.

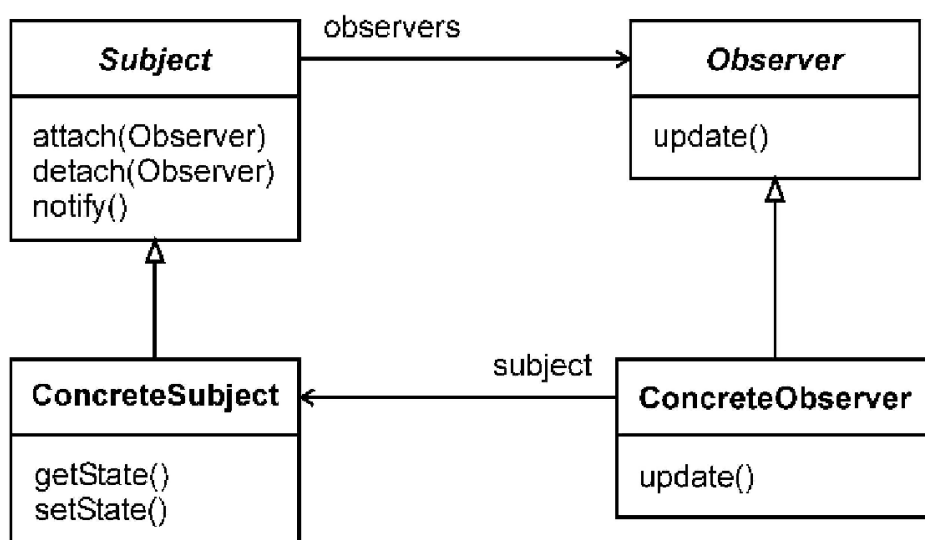
اطلاعات هر کاربر شامل نام، رمز عبور، ایمیل، شماره تماس، آدرس، تاریخ عضویت در فروشگاه، سبد خرید و لیست تراکنش‌ها است که می‌تواند آن‌ها را چاپ کند.

اطلاعات هر سفارش شامل محصول، سفارش‌دهنده، تاریخ ثبت سفارش می‌باشد و اطلاعات هر محصول شامل نام و قیمت آن محصول می‌باشد که این اطلاعات نیز می‌توانند چاپ شوند.

پیاده‌سازی لازم نیست! (می‌توانید تنها کلاس‌ها و متدهای هر کدام را بنویسید و سپس از ابزار خودکار تولید نمودار کلاس استفاده کنید)

سوال چهارم

در این سوال می‌خواهیم الگوی طراحی Observer را باهم تمرین کنیم. الگوی طراحی Observer یک روش معمول برای مدیریت ارتباط بین چندین کلاس است. این الگو به سازماندهی کد کمک می‌کند به گونه‌ای که مدیریت در صورت لزوم آسان‌تر می‌شود. دیاگرام زیر به فهمیدن این مفهوم کمک می‌کند:



برای مطالعه بیشتر این الگو طراحی از لینک‌های زیر کمک بگیرید.

<https://refactoring.guru/design-patterns/observer>

https://www.tutorialspoint.com/design_pattern/observer_pattern.htm

با توجه به توضیحات داده شده، می‌خواهیم یک سیستم خبرنگاری فوتبال طراحی کنیم. در این سیستم موجودیت‌های باشگاه (Club) و طرفدار (Follower) وجود دارد. به این صورت که طرفداران می‌توانند تیم‌های مورد علاقه خود را دنبال کنند و از اخبار جدید آگاه شوند. هر باشگاه مجموعه‌ای از بازی‌های پیشرو (Match) و بازیکنان (Player) دارد و می‌تواند مسابقه و یا بازیکن جدید به لیست خود اضافه کند. در هنگام عضویت در خبرخوان باشگاه، از افراد خواسته می‌شود مواردی که می‌خواهند اخبارشان را دنبال کنند، در سه کانال خبری خود باشگاه، بازی‌های پیشرو و یا اخبار بازیکنان باشگاه مشخص کرده و در صورتی که خبری از جانب هر یک از این بخش‌ها ارسال شود، دریافتش کنند.

به عبارت دیگر هر کدام از این موارد یک کانال خبری مستقل از هم هستند، و فرد تنها از کانال‌هایی که مشخص کرده است اخبار را دریافت می‌کند.

همچنین این افراد می‌توانند بعداً نیز در هر یک از حوزه‌های خبری عضو شوند و یا عضویت خود را لغو کنند.

هر خبر شامل عنوان و متن خبر است.

نحوه تست برنامه:

چند باشگاه مختلف بسازید و برای هر یک چند مسابقه و چند کانال خبری مختلف اضافه کنید.

چند طرفدار بسازید و توسط آنها باشگاه‌ها را دنبال کنید.

در ادامه خبرهایی را توسط باشگاه‌ها ارسال کنید تا طرفداران آنها را دریافت کنند.

نکات پیاده‌سازی:

- مسابقه‌ها حاوی اطلاعاتی مانند تاریخ مسابقه و نام تیم حریف هستند.
- بازیکنان حاوی اطلاعاتی مانند نام و نام خانوادگی، کد ملی و سن هستند.
- هر خبر شامل عنوان خبر (Title) و متن خبر است.
- هنگامی که فرد خبری را دریافت می‌کند در ترمینال نام فرد و متن خبر نیز نمایش داده شود.
- هدف این سوال تمرین و یادگیری الگوی طراحی Observer است و پیاده‌سازی‌ها تنها در چارچوب این الگوی طراحی پذیرفته می‌شود.
- در پیاده‌سازی اشیا از خلاقیت لازم بهره ببرید. به عنوان مثال از فیلدهای بیشتری در پیاده‌سازی اشیا استفاده کنید تا اطلاعات کامل‌تر باشد.
- دقت داشته باشید اجازه استفاده از کلاس آماده Observer و Observable را ندارید و در صورت نیاز آن‌ها را دستی پیاده‌سازی کنید.