

به نام خدا

## تمرین پنجم درس برنامه‌نویسی پیشرفته

نیم‌سال دوم ۱۳۹۹-۱۴۰۰

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-HW5-FirstName\_LastName-StudentNumber.zip

AP-HW5-Saman\_Hoseini-9731079.zip

۲. در سوال‌هایی که ورودی و خروجی مطلوب آن‌ها مشخص شده است، برنامه‌ی شما به صورت ماشینی تصحیح می‌شود. بنابراین 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.

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 گروه تلگرامی با تدریس‌یاران در ارتباط باشید.

۶. امکان آپلود تا دو روز پس از ددلاین و با ضریب‌های ۷۵ و ۵۰ درصد امکان‌پذیر است.

مهلت تحویل: تا جمعه ۲۴ اردیبهشت ۱۴۰۰ ساعت ۲۳:۵۹ شب

## فهرست سوالات

سوال اول.....۳

سوال دوم.....۴

سوال سوم.....۵

## سوال اول

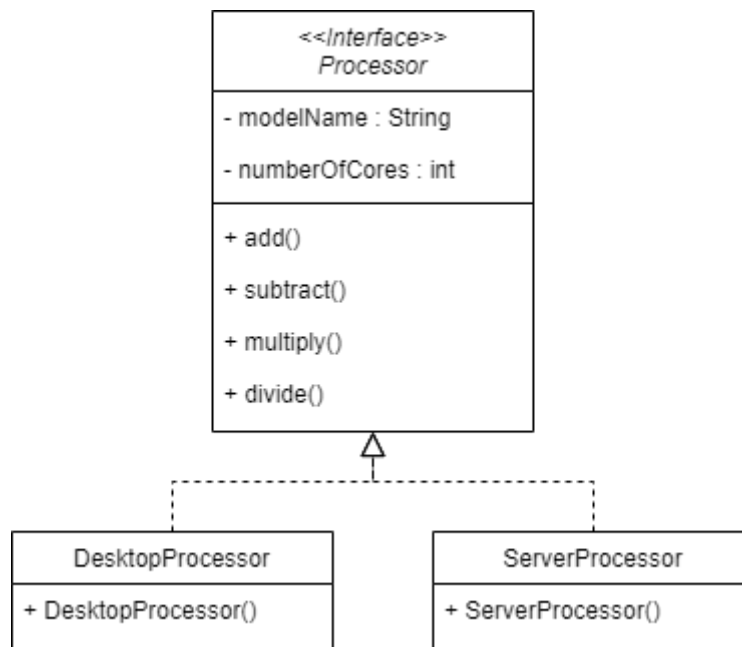
الف) مفاهیم زیر را به اختصار توضیح دهید:

Cohesion, Coupling, Encapsulation, Enumerated type

ب) دو الگوی طراحی زیر را به اختصار شرح دهید و دیاگرامی از آنها رسم کنید:

Factory Method, Adapter

ج) دیاگرام کلاس زیر را در نظر بگیرید. این طراحی را به الگوی Factory Method ببرید و دیاگرام جدید را رسم کنید:



برای مطالعه بیشتر این الگو طراحی از لینک زیر استفاده کنید:

<https://www.javatpoint.com/factory-method-design-pattern>

## 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 کاملاً رعایت کنید. پیاده‌سازی لازم نیست! (می‌توانید تنها کلاس‌ها و متدهای هر کدام را بنویسید و سپس از ابزار خودکار تولید کلاس دیاگرام استفاده کنید)

Class      Method

Field

کتابخانه دارای اسم و آدرس است. این کتابخانه لیستی از کتاب‌ها و اعضا دارد و می‌تواند آن‌ها را چاپ کند. همچنین دارای لیستی از تمامی قرض‌های انجام شده است و می‌تواند آن‌ها را چاپ کند. این 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.

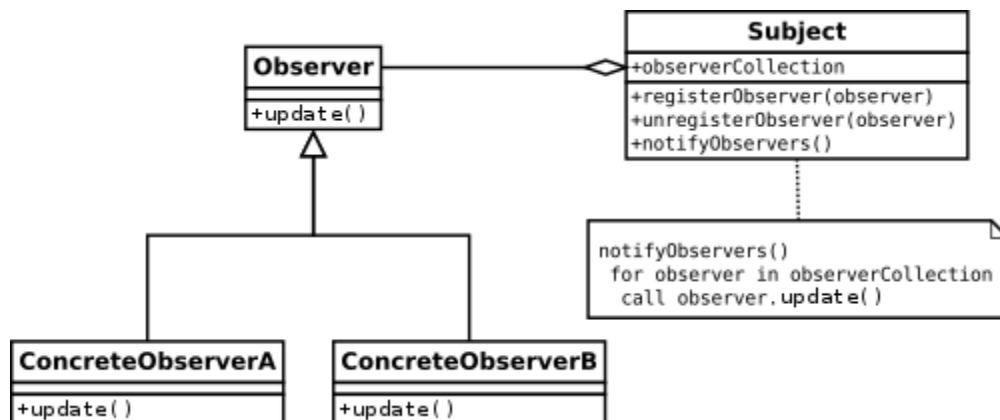
اطلاعات هر عضو شامل نام، ایمیل، شماره عضویت و تاریخ عضویت است که می‌تواند آن‌ها را چاپ کند. همچنین یک لیستی از قرض‌هایی که انجام داده است را نیز نگه‌داری می‌کند و اگر قرضی انجام یا تمام شود باید آن را ذخیره یا حذف کند.

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 می‌باشد. برای نگهداری تاریخ تنها روز، ماه، سال و ساعت کافی است. می‌توان اطلاعات هر قرض را چاپ کرد.

## سوال سوم

در این سوال می‌خواهیم الگو 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.

دیagram 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](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 حوزه اخبار خود باشگاه، بازی‌های پیش‌رو و یا اخبار بازیکنان باشگاه را مشخص کرده و در صورتی که خبری از جانب هر یک از این بخش‌ها ارسال شود، به آنها فرستاده شود.

به عبارت دیگر هر فرد تنها از بخش‌هایی که مشخص کرده اند اخبار را دریافت می‌کنند.

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 می‌باشد.

### نحوه تست برنامه:

چند باشگاه مختلف بسازید و برای هر یک چند مسابقه و چند کانال‌های خبری مختلف اضافه کنید. (همانطور که گفته شد کانال‌های خبری شامل اخبار خود باشگاه یا اخبار بازیکنان باشگاه و یا اخبار مسابقات است) چند طرفدار بسازید و توسط آنها باشگاه‌ها را دنبال کنید. در ادامه برنامه 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 می‌باشد.
- 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 اشیاء از خلاقیت لازم بهره ببرید. به عنوان مثال از فیلدهای بیشتری در پیاده‌سازی اشیاء 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 پیاده کنید

**توجه!** مستندسازی به کمک **Javadoc**، کامنت‌گذاری و رعایت اصول کد نویسی خوانا برای همه کلاس‌های پیاده‌سازی شده الزامی است. همچنین برای این تمرین علاوه بر فایل‌های کد، یک فایل متنی در قالب **PDF** برای قسمت تشریحی ارائه کنید.