



## با سلام و آرزوی سلامتی

در این تمرین می‌خواهیم مقداری با الگوهای طراحی ارائه شده، بیشتر آشنا شویم. در ادامه، توصیف یک سیستم ساده (به همراه یک کد اولیه) آورده می‌شود؛ سپس بر اساس آن از شما کارهایی خواسته شده‌است که لازم است انجام دهید.

## توصیف سیستم

در این تمرین، یک نسخه‌ی اولیه از یک سیستم نرم‌افزاری ([در این مخزن](#)) در اختیار شما قرار داده شده است.

این سیستم وظیفه‌ی مدیریت درخواست‌های پشتیبانی (Ticket) را بر عهده دارد و یک کد اولیه برای آن داده شده‌است. کد داده‌شده درست کار می‌کند، اما از نظر طراحی نرم‌افزاری کیفیت بالایی ندارد.

هدف این تمرین، درک، تحلیل، و بازطراحی سیستم با استفاده از الگوهای طراحی است، به گونه‌ای است که نگهداری، توسعه و فهم آن ساده‌تر شود. سیستم مدیریت درخواست‌های پشتیبانی باید رفتار زیر را داشته باشد:

### الف) دریافت درخواست

برای هر درخواست پشتیبانی جدید این اتفاقات رخ می‌دهد:

- از یک کانال ورودی به سیستم داده می‌شود (مانند WEB و یا EMAIL)
- در ابتدای کار، به عنوان یک درخواست جدید در نظر گرفته می‌شود.

### ب) پردازش درخواست

پس از دریافت درخواست، سیستم بر اساس نوع درخواست تصمیم می‌گیرد که : «کدام واحد مسئول رسیدگی به آن است؟» سپس بر اساس آن، عملیات رسیدگی به آن آغاز می‌شود.

### ج) پاسخ‌دهی

در طول رسیدگی به درخواست، برای آن یک پاسخ مناسب ارائه شده و ارسال می‌شود؛ نوع پاسخ می‌تواند بسته به نوع درخواست متفاوت باشد.

### د) وضعیت‌های درخواست

هر درخواست در طول عمر خود چند وضعیت متوالی را طی می‌کند، از جمله:

- ایجاد شدن

- ارجاع داده شدن
- در حال رسیدگی بودن
- حل شدن
- بسته شدن

رفتار سیستم در هر یک از این وضعیت‌ها می‌تواند متفاوت باشد.

## ه) ثبت رویداد

در پایان پردازش هر درخواست، یک پیام ثبت (Log) از انجام عملیات در سیستم ذخیره می‌شود.

## بخش ۱) تشخیص زیرمسائل و الگوها

با مشاهده‌ی کد داده شده، می‌توان دید که هیچ نشانی از یک طراحی شی‌گرایی مناسب در آن رعایت نشده‌است. بر اساس الگوهای مورد بحث در این آزمایش (که قبلاً گفته شده‌است)، می‌خواهیم این سیستم را بازطراحی کرده و بازآرایی کنیم.

برای این منظور، بر اساس شرح داده شده و کد موجود (و با توجه به الگوهای مفروض)، مسئله‌ی داده شده را به زیر مسائلی بشکنید و برای هر زیر مسئله، ضمن طراحی کلاس مناسب، یکی از الگوهای مفروض را پیشنهاد کنید؛ به عنوان نمونه می‌توانید به زیر مسائل زیر دقت کنید:

- آیا در جاهایی از سیستم رفتار وابسته به حالت دارید؟ در این صورت احتمالاً الگوی State برای آن بخش از سیستم مناسب باشد. در این صورت برای تحقق آن، چه کلاس‌هایی را تعریف می‌کنید؟
- آیا در جاهایی از سیستم، کارهایی را دارید که می‌تواند به چند روش انجام شود؟ در این صورت احتمالاً الگوی Strategy برای آن بخش از سیستم مناسب باشد. در این صورت برای تحقق آن، چه کلاس‌هایی را تعریف می‌کنید؟
- آیا در جاهایی از سیستم، نیازمند ایجاد آبجکت‌هایی پیچیده هستید که به صورت مرحله به مرحله و با یک الگوریتم مشخص ساخته می‌شوند؟ در این صورت احتمالاً الگوی Factory برای آن بخش از سیستم مناسب باشد. در این صورت برای تحقق آن، چه کلاس‌هایی را تعریف می‌کنید؟

برای این بخش، لازم است برای هر زیر مسئله موارد زیر را انجام دهید:

۱. به صورت مختصر آن را شرح دهید.
۲. الگو (یا الگوهای) مناسبی را برای آن پیشنهاد دهید (با ذکر دلیل).
۳. روش اعمال آن الگو را نیز شرح دهید.

## بخش ۲) ترسیم Class Diagram

پس از تشخیص زیرمسائل، یک نمودار ساختار کلاس برای آن ترسیم کنید تا مجموعه‌ی راهکار شما به خوبی دیده شود.

## بخش ۳) اعمال الگو

سپس لازم است که بر روی کد موجود، الگوهای پیشنهادی خود را اعمال کرده و کد را مطابق با الگوهای یافته شده، بازآرایی کنید.

## بخش ۴) تحلیل شی‌گرا

در نهایت، لازم است بدانیم که الگوی شما تا چه حدی باعث بهبود سیستم از منظر اصول شی‌گرایی می‌شود؟ برای این منظور، سیستم خود را از منظر اصول شی‌گرا (SOLID + PLK + CRP) با قبل از اعمال الگو مقایسه کرده و در سه خط توضیح دهید.

**مثال :** پس از اعمال این تغییرات، از نظر OCP چه تغییراتی در سیستم شما به وجود آمده؟ در چه جاهایی بهتر شده و در چه جاهایی بدتر شده؟

## روش تحویل

۱. لازم است تمامی اعضای گروه، در سایت گیت‌هاب یک اکانت داشته باشند. سپس یک مخزن عمومی در گیت هاب ایجاد کرده و

هم گروهی خود را به آن اضافه کنید. در صورتی که روش کار با گیت‌هاب را بلد نیستید، دیدن آموزش‌های مناسب در سطح

اینترنت می‌تواند کمک کننده باشد. در بین این آموزش‌ها، آموزش [دکتر آقامحمدی](#) را پیشنهاد می‌کنیم.

۲. تمرین می‌تواند در قالب گروه‌های ۲ نفره انجام شود.

۳. هر گروه باید دارای یک مخزن عمومی در گیت‌هاب باشد و محتویات پروژه، در آن موجود باشد.

a. پوشه‌ی حاوی کد اصلاح شده.

b. گزارش پروژه شامل گزارش‌های گام‌های ۱ تا ۴ که باید در قالب فایل README در مخزن موجود باشد.

۴. تا پیش از پایان زمان تحویل، لازم است که لینک مخزن را در یک فایل متنی ذخیره کرده و از طریق سامانه‌ی کوئرا ارسال

نمایید. توجه کنید که فاصله‌ی زمانی ارسال لینک در سامانه‌ی کوئرا، نباید اختلاف بیش از ۳۰ دقیقه با زمان آخرین Commit

داشته باشد. در غیر این صورت، زمان آخرین Commit معیار محاسبه‌ی تاخیر خواهد بود.

## هشدارهایی برای از دست ندادن نمره

به منظور جلوگیری از کاهش و یا صفر شدن نمره‌ی تمرین، به هشدارهای زیر توجه کنید:

۱. تمرین صرفا به روش گفته شده و با ساختار خواسته شده و از طریق GitHub تحویل گرفته می‌شود. در صورتی که با آن آشنا نیستید، از طریق منابع گفته شده آن را یاد بگیرید. در صورت تحویل تمرین از سایر روش‌ها، نمره‌ای به آن تعلق نخواهد گرفت.
۲. ارسال لینک مخزن صرفا از طریق سامانه‌ی کوئرا مقدور است.
۳. تمامی اعضای گروه باید به صورت مجزا در انجام پروژه مشارکت داشته‌باشند؛ در صورت عدم مشارکت (و یا کم بودن مشارکت)، نمره‌ی گزارش در درصد مشارکت همگروهی کم‌کار ضرب خواهد شد.
۴. معیار سنجش مشارکت، صرفا تاریخچه‌ی Commit‌های داخل مخزن عمومی و تغییر اعمال شده‌است. بنابراین در صورتی که به هر علتی اثری از مشارکت نفر دوم در مخزن موجود نباشد، برای آن فرد نمره‌ی صفر لحاظ می‌شود. برخی از مواردی که منجر به این اتفاق می‌شود در ادامه می‌آید (و در صورت رخداد، مدرس و دستیاران آموزشی هیچ مسئولیتی را در قبال تبعات آن بر عهده نخواهند داشت).
- a. برگزاری جلسات هماهنگی به صورت حضوری و مجازی و آپلود پروژه توسط یک فرد.
- b. جابجا شدن اکانت گیت‌هاب (ارسال کامیت توسط لب تاب همگروهی).
- c. آپلود یکجای تمام پروژه توسط یک نفر.
- d. ارسال گزارش در قالب فایل‌ی بجز README (نظیر pdf)
- e. نگارش گزارش به زبانی به جز زبان فارسی، قابل قبول نخواهد بود.
۵. پس از پایان یافتن زمان تحویل تمرین، به صورت تصادفی یک جلسه‌ی ارائه‌ی مجازی با اعضای برخی از گروه‌ها برگزار خواهد شد.
۶. در نهایت، انتظار می‌رود که دانشجویان به قوانین گفته شده در خصوص تقلب و استفاده‌ی نابجا از هوش مصنوعی توجه کنند.

**با آرزوی سلامتی برای شما**

**پورسلطانی**