

بہ نام خدا

هدف از طراحی این *Assignment*، آشنایی عملی با نحوه تحلیل، پیاده‌سازی و مواجه شما با مسئله ارائه شده، می‌باشد. پاره‌ای از مواردی که در راهکار ارائه شده شما برای مسئله، مورد بررسی قرار خواهند گرفت، در زیر بیان شده است.

Clean Code

Naming Convention

Data Type

Collections Framework

Efficiency

Modifiers

Contract

JDK Facility

Java 8+ Features

Design Pattern

OOP

با توجه به زبان سازمانی این شرکت، هدف این است که با زبان برنامه نویسی *Java* برای این مسئله راه‌حل ارائه شود. نسخه‌ی *JDK* که استفاده می‌شود بهتر است ۸ یا در صورت امکان نسخه‌های بالاتر باشد. ترجیح این است که تا حد امکان از قابلیت‌های داخلی *JDK* برای حل مسئله استفاده شود. اگر در بخشی احساس می‌شود که استفاده از یک کتابخانه خارجی لازم یا بهینه‌تر است، اضافه کردن آن بلامانع است.

هدف از این مسئله فقط حل کردن و قابل اجرا بودن آن نیست. تمامی مواردی که در فضای عملیاتی لازم است رعایت شوند، در راه حل ارائه شده شما بررسی می‌شوند. تمامی بخش‌های ذکر شده در این مسئله اجباری نیستند و می‌توانید فقط بخش‌هایی که لازم است را پیاده‌سازی نمایید. به موارد زیر دقت فرمائید:

- لطفاً تا حد امکان با اتکا به دانش خود به پیاده‌سازی صورت مسئله بپردازید.

- لطفاً پیاده‌سازی‌های هر بخش را به نحوی انجام دهید که قابل تمیز دادن باشند. گاهی یک بخش تمرین تکمیل کننده بخش قبلی است، در این مواقع بهتر است خروجی‌های حاصل از بخش قبل و بخش جدید قابل تفکیک باشند.
- ممکن است راه‌کارها و موارد بیشتری مدنظر شما باشند که به بهتر شدن نتیجه خروجی کمک می‌کنند، بسیار خوشحال می‌شویم که این موارد را در صورت لزوم در پیاده‌سازی شما ببینیم.
- توضیح‌های ارائه شده در بخش‌های مختلف این تمرین فقط کلیات را بیان می‌کنند، خود را محدود به این چارچوب نبینید.

تمرین

۱. داده‌های زیر را در قالب یک *Domain Model* (ها) و ... گروه‌بندی نمائید. در صورتی که متغیرهای دیگری جهت تکمیل یا تسهیل تمرین لازم است، اضافه فرمائید (اجباری).

- شماره کارت (۱۶ رقم)
- کد صادر کننده (۶ رقم)
- نوع کارت (نقدی، اعتباری)
- نام صادر کننده
- شماره ملی (۱۰ رقم)
- شماره تماس
- تاریخ انقضاء (ماه/سال)
- آدرس محل سکونت
- شماره حساب (۱۰ رقم)
- نام و نام خانوادگی شخص
- فعال بودن کارت

۲. خروجی‌های حاصل از بخش قبلی را به یک *Project* با مدیریت *Maven* تبدیل کنید. پیکربندی *POM* و استفاده از ویژگی‌های ابتدایی *Maven* در این بخش مدنظر می‌باشند (اجباری).

- یک نام مناسب برای *Project* انتخاب کنید.
- به نام‌گذاری *Package* ها دقت کنید.

۳. یک مخزن^۱ درون حافظه‌ای^۲ طراحی کنید که در آن به ازای هر شماره ملی منحصر به فرد، شماره کارت‌های منحصر به فرد مربوط به آن شخص وجود داشته باشند (اجباری).
- هر فرد فقط می‌تواند یک کارت از هر نوع و از هر صادر کننده داشته باشد.
 - این مخزن را با تعدادی داده‌های دلخواه و معتبر مقداردهی اولیه نمایید.
 - می‌توانید برای مقداردهی اولیه‌ی مخزن از داده‌های درون یک فایل متنی^۳ استفاده کنید (اختیاری).
۴. به بخش‌های قبلی، امکانی اضافه نمایید که به صورت یک گزارش دوره‌ای، میزان مصرف حافظه‌ی برنامه را در محیط *Console* چاپ نماید (اجباری).
- این بخش از پیاده‌سازی با اجرای برنامه کار خود را شروع می‌کند.
 - پس از اتمام برنامه فعالیت این بخش از پیاده‌سازی پایان می‌پذیرد.
۵. برای مخزن فوق چند *Unit Test* طراحی کنید که عملکرد آن را مورد ارزیابی قرار دهند. اگر لازم است کتابخانه‌های خارجی برای اجرای این بخش داشته باشید، از تنظیمات بخش ۲، این قابلیت‌ها را اضافه نمایید (اجباری).
۶. *Rest API* طراحی کنید که با ارسال شماره ملی و سایر متغیرهای مورد نیاز، بتواند اطلاعات یک کارت مورد نظر را از مخزن قبلی، واکنشی نماید. در این بخش لازم است ویژگی‌های زیر اعمال شوند (اجباری):
- داده‌ها به صورت *JSON* منتقل می‌شوند.
 - این *Service* باید قابلیت بازگرداندن یک آرایه از کارت‌ها را با توجه به متغیرهای ورودی داشته باشد.
 - در صورتی که شماره ملی موردنظر یافت نشد، *HTTP Status* مناسب بازگرداند.
 - بروز خطاهای ناشناخته را با استفاده از قابلیت‌هایی که *HTTP Protocol* در اختیار ما قرار می‌دهد، به کاربر گزارش نماید.
۷. اعتبارسنجی داده‌های ورودی را به *Service* طراحی شده در بخش ۵ اضافه نمایید (اختیاری).
۸. احراز هویت از طریق نام کاربری و گذرواژه به *Service* بخش ۵ اضافه گردد (اختیاری).
- احراز هویت را با استفاده از روش *Basic Auth* انجام دهید.

¹ Repository

² In-memory

³ Text

- برای نام‌کاری و گذرواژه‌های معتبر، می‌توان از تعریف مقادیر ثابت در بدنه برنامه استفاده کرد.
- ۹. برای رخدادنگاری (*Logging*) بخش ۶، ۷ و ۸ در مواردی همچون وجود نداشتن شماره ملی، نامعتبر بودن مقدار یا قالب داده‌های ورودی و تلاش‌های ناموفق برای ورود به سامانه، از *Log* های ذخیره شده در فایل‌های متنی استفاده کنید (اختیاری).
- در صورتی که برای این بخش وابستگی به کتابخانه خاصی لازم است، به فایل پیکربندی بخش ۲ اضافه فرمائید.
- ۱۰. *Domain Model*^۴ طراحی شده در مرحله اول را به یک *Relational Model* تبدیل کنید (اجباری).
- می‌توان از پایگاه داده‌های رابطه‌ای کوچک مانند *H2* یا *Derby* یا یکی پایگاه داده‌های رابطه‌ای متداول به دلخواه استفاده نمود.
- کلیه نکاتی که در تعریف *Domain Model* رعایت شده را در این بخش و در سطح پایگاه داده نیز رعایت فرمائید.
- می‌توان از قابلیت‌های *ORM* نیز استفاده نمود (اختیاری).
- ۱۱. *Service* طراحی شده در بخش ۶ را به نحوی تکمیل نمائید که امکان ثبت کارت جدید برای یک شخص وجود داشته باشد (اختیاری).
- داده‌های تبادل شده در قالب *JSON* می‌باشند.
- کارت‌های جدید در پایگاه داده ثبت شوند.
- به محدودیتی که برای کارت‌های یک شخص در بخش ۶ ذکر شده است دقت فرمائید.
- خطاهای محتمل، مدیریت شوند.
- در صورت امکان همگام‌سازی مخزن درون حافظه‌ای و پایگاه داده انجام شود. هدف از این بخش معتبر باقیماندن مخزن درون حافظه‌ای استفاده شده در بخش ۶، در صورت ثبت کارت جدید توسط کاربر می‌باشد.
- ۱۲. با استفاده از ابزارهای *API Documentation* مستند *Service* های طراحی شده در بخش ۶ و ۱۱ را ایجاد کنید (اختیاری).
- در این بخش می‌توانید از ابزار *Swagger* استفاده کنید.

^۴ منظور خروجی‌های طراحی شده در بخش اول است که باید به معادل رابطه‌ای خود تبدیل شوند.

۱۳. یک *Client* پیاده‌سازی کنید که امکان اتصال و فراخوانی *Service* های طراحی شده در بخش‌های قبلی را فراهم نماید (اختیاری).

- برای پیکربندی اولیه *Client* از قابلیت *Properties File* ها استفاده کنید.
- تا حد امکان فراخوانی *Service* از طریق *Client* را برای کاربر نهایی تسهیل کنید.
- خطاهای احتمالی را مدیریت کنید.

۱۴. برای سنجش عملکرد *Service* های ارائه شده در بخش ۶ و ۱۱ و همچنین *Client* طراحی شده در بخش ۱۳ تعدادی *unit test* پیاده‌سازی فرمائید (اختیاری).

- اگر به کتابخانه خاصی برای این بخش نیاز است، از طریق تنظیمات بخش ۲ آن را اضافه نمائید.
۱۵. پروژه نهایی را در یک *Repository* در وبگاه www.github.com بارگزاری کنید (اختیاری).

گزارش

پاسخ خود را به آدرس Me_Maleki@ISC.CO.IR با استفاده از راهکارهای زیر، ارسال فرمائید:

۱. در صورتی که بخش ۱۵ را پیاده‌سازی نموده‌اید، آدرس این *Repository* (در صورت نیاز دسترسی‌های لازم را پیش از ارسال آدرس فراهم نمائید) را جهت دسترسی به آن در اختیار ما قرار دهید.
۲. محتویات پروژه را پس از *zip* کردن، ارسال فرمائید.