

مکتب شریف

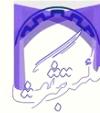
اولین بوت کمپ آموزشی - استخدامی ایران



C#/.Net

۱۴۰ مکتب





Cw-10

تمرین به صورت انفرادی می باشد. و اگر سوالی برآتون پیش امد اول از هم گروهی هاتون سپس از مرتبه میتوانید کمک بگیرد و حتما باید امروز تمرین تمام شود.

در این پروژه باید یک سیستم یکپارچه مدیریت کاربر و فروشگاه را در قالب یک پروژه Console App پیاده‌سازی کنید.

فاز ۱ - لایه Repository و ذخیره‌سازی JSON

برای هر موجودیت Generic Repository جدا (بدون) بسازید:

IUserRepository / JsonUserRepository

JsonProductRepository – IProductRepository

JsonOrderRepository / IOrderRepository

اطلاعات باید در فایل‌های زیر ذخیره شوند:

(وظیفه Repository فقط خواندن و نوشتن فایل باشد.) users.json - products.json - orders.json

فاز ۲ Service Layer + Dependency Injection

کلاس‌های زیر را ایجاد کنید:

UserService –

ProductService –

OrderService –

شرطی:

Repository –

- داخل new ها کردن منوع است.

- منطق تجاری (Business Logic) داخل Service ها باشد.

فاز ۳ - اعتبارسنجی حرفه‌ای رمز عبور + Custom Exception

در این فاز باید قوانین رمز عبور را از کلاس **UserService** جدا کنید تا ساختار پروژه تمیزتر، قابل توسعه‌تر و مطابق با اصول **SOLID** باشد.

مراحل انجام کار:

(۱) یک **IPasswordPolicy** به نام **Interface** طراحی کنید که متدی مانند زیر داشته باشد:

Validate(string username, string password)

یک کلاس **StrongPasswordPolicy** بسازید که **IPasswordPolicy** را پیاده‌سازی کند.

(۲) قوانین رمز عبور:

- حداقل ۸ کاراکتر داشته باشد.

- حداقل شامل ۱ حرف بزرگ (A-Z) باشد.

- حداقل شامل ۱ حرف کوچک (a-z) باشد.

- حداقل شامل ۱ عدد (۰-۹) باشد.

- حداقل شامل ۱ کاراکتر خاص مانند ! # % \$ @ باشد.

- رمز عبور نباید شامل **Username** کاربر باشد.

(۳) برای هر قانونی که رعایت نشد، یک **Custom Exception** جداگانه ایجاد کنید (نحوه **PasswordMissingDigitException**, **PasswordTooShortException**).

و در صورت نقض قانون، همان **Exception** را **throw** کنید + (به همراه خطای مخصوص خود).

(۴) در کلاس **UserService** هنگام ثبت‌نام کاربر، به جای نوشتن **if** های متعدد، فقط متد **Validate** را صدا بزنید.

(۵) کلاس **UserService** باید **IPasswordPolicy** را از طریق **Constructor Injection** دریافت کند.

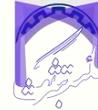
(۶) داخل **Service** مجاز به **new** کردن **StrongPasswordPolicy** نیستید.

فاز ۴ - سیستم پرداخت

کلاس **abstract Payment** به نام ایجاد کنید.

کلاس‌های مشتق:

CashPayment - CardPayment - OnlinePayment -



فاز ۵ - مدیریت خطا و Logger

در Repository

- اگر فایل وجود نداشت، ساخته شود. اگر فایل خالی بود، لیست خالی برگردانده شود. اگر JSON خراب بود، خطا ثبت شود و برنامه ادامه پیدا کند.
موارد زیر را ایجاد کنید:

ILogger - ConsoleLogger - FileLogger -

Logger نیز باید از طریق DI تزریق شود.

منوی نهایی برنامه (Console)

(۱) ثبت‌نام / ورود

(۲) مدیریت کاربران (CRUD)

(۳) مدیریت محصولات (CRUD)

(۴) ثبت سفارش

(۵) پرداخت سفارش

(۶) مشاهده سفارشات من

(۰) خروج