



Home Work

.Net

OOP - ObjectOriented Programming - 2



۱. فایل تمرین را در پنل خود آپلود کنید.

۲. **title** فایل تمرین به صورت (نام تمرین+نام و نام خانوادگی) به انگلیسی باشد.

۳. در صورتی که سوال و یا ابهامی دارید در گروه چت تلگرامی بپرسید.

۱. طراحی سیستم کتابخانه

هدف: استفاده از کلاس‌ها، متدها و اصول اولیه OOP

شرح: یک کلاس به نام **Book** طراحی کنید که شامل ویژگی‌های زیر باشد:

- Title
- Author
- ISBN
- IsAvailable

یک کلاس به نام **Library** ایجاد کنید که شامل موارد زیر باشد:

- یک لیست از کتاب‌ها (**List<Book>**)
- متدی به نام **AddBook** برای اضافه کردن کتاب به کتابخانه
- متدی به نام **BorrowBook** که با دریافت عنوان کتاب، وضعیت موجودی آن را تغییر دهد. اگر کتاب موجود نباشد، پیامی نمایش دهد.
- متدی به نام **ReturnBook** برای برگرداندن کتاب و تغییر وضعیت موجودی.

۲. سیستم مدیریت دانشگاه

هدف: استفاده از اصول ارث بری و چندریختی (**Polymorphism**)

شرح: یک کلاس پایه به نام **Person** طراحی کنید که ویژگی‌های زیر را داشته باشد:

- **Name**
- **Age**
- متدی به نام **GetDetails** که اطلاعات شخص را به صورت رشته برگرداند.

دو کلاس **Student** و **Professor** از کلاس **Person** ارث بری کنند و ویژگی‌های خاص خود را داشته باشند:

۱. کلاس **Student**:

- **StudentID**
- **Major**
- متد **GetDetails** که اطلاعات دانشجو را نمایش می‌دهد.

۲. کلاس **Professor**:

- **ProfessorId**
- **Subject**
- متد **GetDEtails** اطلاعات استاد را نمایش می‌دهد

*یک لیست از اشخاص مختلف (دانشجو و استاد) ایجاد کنید و اطلاعات هر کدام را نمایش دهید .

۳. طراحی سیستم مدیریت فروشگاه آنلاین

هدف : ترکیب مفاهیم کلاس‌ها، ارث‌بری، چندریختی و Interface

شرح : یک سیستم مدیریت فروشگاه آنلاین طراحی کنید.

یک کلاس پایه **Product** ایجاد کنید که ویژگی‌های زیر را داشته باشد:

- Name
- Price
- متد **GetProductDetails**

دو کلاس **Electronic** و **Clothing** از **Product** ارث‌بری کنند و ویژگی‌های خاص خود را اضافه کنند:

- **Electronic**: ویژگی **WarrantyPeriod**
- **Clothing**: ویژگی **Size** و **Material**

یک اینترفیس به نام **Idiscountable** ایجاد کنید که شامل متدی به نام **ApplyDiscount** باشد. هر کلاس که این اینترفیس را پیاده‌سازی کند، محصولات آن قابلیت تخفیف خواهد داشت (مثلا میتوانید در کلاس **Electronic** آن را پیاده سازی کنید و محصولات این کلاس را شامل تخفیف کنید).

سیستم را به گونه‌ای طراحی کنید که بتوانید محصولات مختلف را اضافه کنید و جزئیات محصولات را همراه با تخفیف نمایش دهید.

۴. سیستم مدیریت بیمارستان

شرح : یک سیستم برای مدیریت بیمارستان طراحی کنید که شامل موارد زیر باشد:

کلاس‌های پایه:

* کلاس پایه **Person** که شامل ویژگی‌های عمومی زیر باشد:

- Name
- Age
- NationalId

- متدی به نام **GetDetails** که اطلاعات شخص را برگرداند

کلاس‌های مشتق شده:

* کلاس **Patient** از کلاس **Person** ارث‌بری کند و شامل ویژگی‌های زیر باشد :

- **PatientId**
- **MedicalHistory** - لیستی از بیماری‌های گذشته بیمار
- متدی به نام **AddToMedicalHistory** برای اضافه کردن بیماری جدید به سابقه بیمار.

* کلاس **Doctor** از کلاس **Person** ارث‌بری کند و شامل ویژگی‌های زیر باشد:

- **DoctorId**
- **Specialization** - تخصص پزشک
- متدی به نام **Diagnose** که یک بیمار را به‌عنوان ورودی بگیرد و یک تشخیص بیماری به او اضافه کند.

* کلاس **Room** طراحی کنید که شامل موارد زیر باشد:

- **RoomNumber**
- **Capacity**
- لیستی از بیماران
- متدی به نام **AssignPatient** که یک بیمار را به اتاق اضافه کند. اگر ظرفیت پر شده باشد، **Exception** ایجاد کند .

* کلاس **Hospital** طراحی کنید که شامل موارد زیر باشد:

- لیستی از پزشکان و اتاق‌ها
- متدی به نام **AdmitPatient** که بیمار جدیدی را پذیرش کرده و به اولین اتاق خالی اضافه کند. اگر هیچ اتاقی ظرفیت نداشته باشد، پیام مناسب نمایش دهد.
- متدی به نام **DischargePatient** که بیمار را از سیستم حذف کند.

*مدیریت استثناها (Exception Handling)

یک استثنا به نام **RoomFullException** تعریف کنید که وقتی ظرفیت اتاق پر شده باشد، ایجاد شود.

بخش امتیازی :

اگر در کدتان از الگوهای طراحی زیر به روش خواسته شده استفاده کنید امتیاز جداگانه ای در نظر گرفته خواهد شد :

- **Singleton Pattern** برای مدیریت (Single Instance) از کلاس **Hospital** که تنها یک نمونه از آن در سیستم وجود دارد.
- **Factory Method Pattern** برای ایجاد بیماران و پزشکان به طور داینامیک با توجه به نوع آنها.
- **Strategy Pattern** برای مدیریت نوع تشخیص و رفتار پزشک بر اساس تخصص‌ها.