

# بیمارستان

## توضیحات:

در این سوال شما باید یک سیستم مدیریت بیمارستان را طراحی کنید که شامل اطلاعات مختلفی از جمله بیماران، پزشکان، پرستاران، و ملاقات‌های پزشک با بیماران باشد. این سیستم باید از اصول شی‌گرایی مانند abstraction، encapsulation، inheritance، polymorphism پیروی کند.

## وظایف:

### ۱. کلاس Person :

- این کلاس باید ویژگی‌های عمومی افراد بیمارستان (پزشک، بیمار، پرستار و ...) را داشته باشد.

- ویژگی‌ها:

- first\_name (نام)

- last\_name (نام خانوادگی)

- id (شناسه شخص، شماره یکتا)

- age (سن)

- gender (جنسیت)

- هدف: ایجاد یک کلاس پایه برای سایر انواع افراد بیمارستان.

### ۲. کلاس Patient :

- از کلاس Person ارث‌بری کند.

- ویژگی‌ها:

- disease (بیماری)

- doctor (پزشک معالج)

- متدها:

- get\_patient\_details : جزئیات بیمار را نمایش دهد.

- هدف: پیاده‌سازی ویژگی‌ها و رفتارهای خاص بیماران.

### ۳. کلاس Doctor :

- از کلاس Person ارث‌بری کند.

- ویژگی‌ها:

- specialization (تخصص پزشکی)

- patients (لیست بیماران تحت درمان)

- متدها:

- add\_patient : بیمار جدید به لیست بیماران پزشک اضافه شود.

- get\_doctor\_details : جزئیات پزشک را نمایش دهد.

- **هدف:** پیاده‌سازی ویژگی‌ها و رفتارهای خاص پزشکان و مدیریت بیماران تحت درمان.

#### ۴. کلاس Nurse :

- از کلاس Person ارث‌بری کند.

- ویژگی‌ها:

- department (بخش بیمارستانی)

- متدها:

- assist\_patient : کمک به بیمار در بخش‌های مختلف.

- get\_nurse\_details : جزئیات پرستار را نمایش دهد.

- **هدف:** پیاده‌سازی ویژگی‌ها و رفتارهای پرستاران.

#### ۵. کلاس Appointment :

- ویژگی‌ها:

- patient : بیمار

- doctor : پزشک

- appointment\_time : زمان ملاقات

- status : وضعیت ملاقات (تایید شده، لغو شده، در انتظار)

- متدها:

- schedule\_appointment : زمان ملاقات را تعیین کند.

- cancel\_appointment : ملاقات را لغو کند.

- get\_appointment\_details : جزئیات ملاقات را نمایش دهد.

- **هدف:** ایجاد سیستم مدیریت ملاقات‌ها و پیگیری وضعیت آن‌ها.

#### ۶. کلاس Hospital :

- این کلاس باید اطلاعات کلی بیمارستان را نگهداری کند و بتواند تمامی موجودیت‌ها را مدیریت کند.

- ویژگی‌ها:

- patients : لیست بیماران
- doctors : لیست پزشکان
- nurses : لیست پرستاران
- appointments : لیست ملاقات‌ها

◦ متدها:

- add\_patient : بیمار جدید به سیستم اضافه شود.
- add\_doctor : پزشک جدید به سیستم اضافه شود.
- add\_nurse : پرستار جدید به سیستم اضافه شود.
- schedule\_appointment : ملاقات جدید بین بیمار و پزشک ایجاد کند.
- get\_all\_patients : تمام بیماران را نمایش دهد.
- get\_doctor\_patients : بیماران هر پزشک را نمایش دهد.
- get\_patient\_appointments : ملاقات‌های یک بیمار را نمایش دهد.
- **هدف:** پیاده‌سازی کلاس اصلی بیمارستان برای مدیریت کلیه داده‌ها و تعاملات.

راهنمایی‌ها:

- استفاده از **وراثت**: شما باید از کلاس Person به عنوان کلاس پایه برای بیماران، پزشکان و پرستاران استفاده کنید تا ویژگی‌های مشترک را یکجا تعریف کنید.
- استفاده از **کپسوله‌سازی**: داده‌های داخلی هر کلاس را پنهان کنید و تنها از متدهای عمومی برای دسترسی به آن‌ها استفاده کنید.
- استفاده از **چندشکلی**: اطمینان حاصل کنید که کلاس‌ها و متدهای شما به درستی رفتارهای مختلف را بسته به نوع شیء مدیریت می‌کنند.
- استفاده از **انتزاع**: جزئیات اجرای توابع باید از نحوه نمایش و استفاده آن‌ها جدا باشد.

نکات:

۱. برای هر کلاس باید ویژگی‌ها و متدهای مناسب را طراحی کنید.
۲. در طراحی سیستم از روابط بین کلاس‌ها استفاده کنید، به طوری که اطلاعات بیمار، پزشک و ملاقات‌ها به درستی ارتباط داشته باشند.
۳. از متدهای get\_details برای نمایش اطلاعات افراد و ملاقات‌ها استفاده کنید.

۴. از متدهای مدیریت سیستم بیمارستان مانند `add_patient` و `schedule_appointment` برای اضافه کردن اطلاعات به سیستم استفاده کنید.

### نمونه خروجی:

در اینجا یک نمونه خروجی کامل برای سیستم مدیریت بیمارستان بر اساس ورودی‌های مختلف مانند اضافه کردن بیمار، پزشک، پرستار و ملاقات‌ها، آورده شده است:

### 1. اضافه کردن پزشک

وقتی که یک پزشک جدید به سیستم اضافه می‌شود، خروجی به شکل زیر خواهد بود:

Doctor added successfully:

Name: Dr. Amir Jalali

ID: D001

Age: 40

Gender: Male

Specialization: Cardiology

### 2. اضافه کردن بیمار

وقتی که یک بیمار جدید به سیستم اضافه می‌شود، خروجی به شکل زیر خواهد بود:

Patient added successfully:

Name: Parsa Mohebian

ID: P001

Age: 30

Gender: Female

Disease: Heart Disease

Assigned Doctor: Dr. Amir Jalali

### 3. اضافه کردن پرستار

وقتی که یک پرستار جدید به سیستم اضافه می‌شود، خروجی به شکل زیر خواهد بود:

Nurse added successfully:

Name: Sarah Mohammadi

ID: N001

Age: 28  
Gender: Female  
Department: Cardiology

#### 4. برنامه‌ریزی ملاقات بین بیمار و پزشک

وقتی که ملاقات بین بیمار و پزشک برنامه‌ریزی می‌شود، خروجی به شکل زیر خواهد بود:

Appointment scheduled successfully:  
Patient: Parsa Mohebian  
Doctor: Dr. Amir Jalali  
Date and Time: 2025-03-10 10:00  
Status: Scheduled

#### 5. نمایش جزئیات بیمار

وقتی که اطلاعات یک بیمار نمایش داده می‌شود، خروجی به شکل زیر خواهد بود:

Patient Details:  
Name: Parsa Mohebian  
ID: P001  
Age: 30  
Gender: Female  
Disease: Heart Disease  
Assigned Doctor: Dr. Amir Jalali

#### 6. نمایش جزئیات پزشک

وقتی که اطلاعات یک پزشک نمایش داده می‌شود، خروجی به شکل زیر خواهد بود:

Doctor Details:  
Name: Dr. Amir Jalali  
ID: D001  
Age: 40  
Gender: Male  
Specialization: Cardiology  
Patients:  
- Parsa Mohebian

## 7. لغو ملاقات

وقتی که ملاقات لغو می‌شود، خروجی به شکل زیر خواهد بود:

Appointment canceled successfully:

Patient: Parsa Mohebian

Doctor: Dr. Amir Jalali

Date and Time: 2025-03-10 10:00

Status: Canceled

## 8. نمایش لیست ملاقات‌ها

وقتی که لیست تمامی ملاقات‌های ثبت‌شده نمایش داده می‌شود، خروجی به شکل زیر خواهد بود:

All Appointments:

Patient: Parsa Mohebian, Doctor: Dr. Amir Jalali, Date: 2025-03-10 10:00, S

Patient: Alireza Goldoost, Doctor: Dr. Arian Fathi, Date: 2025-03-11 11:00,

## 9. نمایش لیست پزشکان

وقتی که لیست تمامی پزشکان نمایش داده می‌شود، خروجی به شکل زیر خواهد بود:

All Doctors:

Dr. Amir Jalali, ID: D001, Specialization: Cardiology

Dr. Arian Fathi, ID: D002, Specialization: Neurology

## جمع‌بندی نمونه خروجی‌ها:

- اطلاعات بیماران، پزشکان، پرستاران و ملاقات‌ها به طور واضح و ساختاریافته نمایش داده می‌شوند.
- در هر مرحله از فرآیند (اضافه کردن، ویرایش، لغو و مشاهده اطلاعات) یک خروجی متنی به صورت دقیق و خوانا برای کاربر نشان داده می‌شود.

## نکته مهم:

در صورت پیاده سازی متد یا کلاس اضافه در صورت نیاز واقعی به آن‌ها و مشکل نداشتنشان، مانعی برای پیاده سازی آن نیست.

