

به نام خدا

امتحان پایان ترم درس برنامه‌نویسی پیشرفته

پاییز ۱۴۰۲

نکات آزمون:

۱. پاسخ هر سوال را به صورت یک فایل py در قسمت مشخص شده در کوئرا آپلود کنید.
۲. استفاده از فایل‌های شخصی، سایت‌های اینترنتی و کتاب مجاز است. اما استفاده از هر گونه پیام رسان، هوش مصنوعی و همچنین کمک گرفتن از اشخاص دیگر مجاز نیست. در صورتی که در یک سوال از راهنمایی یک سایت اینترنتی استفاده کرده اید، لینک آن را در ابتدای کد کامنت کنید.
۳. سعی کنید همه سوالها را یک بار بخوانید و از سوالهایی شروع کنید که تسلط بیشتری دارید. نمره تمام سوالها برابر است.
۴. تمام وقت خود را صرف پاسخ به یک سوال نکنید. دقت کنید که زمان آزمون شما ۱۰۰ دقیقه است و در این مدت باید به ۳ سوال پاسخ دهید.
۵. در صورتی که در یک سوال ابهامی وجود دارد، برداشت خود را در ابتدای کدتان بنویسید. البته توجه داشته باشید که برداشت صحیح و معقول و در واقع فهم مساله خود حائز اهمیت است.

سوال یک) در سوال ماتریس (تمرین ۳) کلاس ماتریس و متودهای آن توضیح داده شده است. با استفاده از unittest یک برنامه برای تست متودهای این کلاس بنویسید. (حداقل ۱۰ تست خوب بنویسید).

سوال دو) شما مسئول طراحی یک شبیه‌سازی از یک فروشگاه حیوانات خانگی هستید. فروشگاه حیوانات خانگی انواع حیوانات خانگی مختلف مانند سگ، گربه و پرند را به فروش می‌رساند. هر حیوان خانگی نام، سن و قیمتی دارد. همچنین، فروشگاه حیوانات خانگی تعداد کل حیواناتی که فروخته شده را نیز نگهداری می‌کند. شما باید موارد زیر را پیاده‌سازی کنید:

۱. یک کلاس `Pet` را پیاده‌سازی کنید که به عنوان کلاس پایه برای همه انواع حیوانات خانگی عمل کند. این کلاس باید ویژگی‌ها و متدهای زیر را داشته باشد:

- ویژگی‌ها:

`name` - یک رشته که نام حیوان خانگی را نشان می‌دهد. (یک property فقط خواندنی)

`age` - یک عدد صحیح که سن حیوان خانگی را نشان می‌دهد. (یک property خواندنی/نوشتنی)

`price` - یک عدد اعشاری که قیمت حیوان خانگی را نشان می‌دهد. (یک property خواندنی/نوشتنی)

- متدها:

`__init__(self, name, age, price)` -

یک متود سازنده که ویژگی‌های `name`، `age` و `price` را مقداردهی اولیه می‌کند.

`make_sound(self)` -

متود صدایی که توسط حیوان خانگی تولید می‌شود. این متود باید توسط هر کلاس مشتق شده از این کلاس بازنویسی شود.

۲. سه کلاس به نام‌های `Dog`، `Cat` و `Bird` ایجاد کنید که از کلاس `Pet` مشتق شوند. هر کلاس مشتق شده باید صدایی خاصی که توسط آن حیوان تولید می‌شود را داشته باشد. متود `make_sound` را مطابق با این نیاز پیاده‌سازی کنید

۳. یک متغیر کلاس به نام `total_pets_sold` در کلاس `Pet` پیاده‌سازی کنید تا تعداد کل حیواناتی که فروخته شده را ردیابی کند. این متغیر را با مقدار ۰ مقداردهی اولیه کنید.

۴. یک متود کلاس به نام `get_total_pets_sold()` در کلاس `Pet` پیاده‌سازی کنید که مقدار فعلی متغیر `total_pets_sold` را برگرداند.

۵. یک متود نمونه به نام `sell_pet(self)` در کلاس `Pet` پیاده‌سازی کنید که هرگاه یک حیوان خانگی فروخته شود، مقدار متغیر `total_pets_sold` را یک واحد افزایش دهد.

۶. یک کلاس به نام `PetStore` ایجاد کنید که یک متود استاتیک به نام `calculate_discount(price)` داشته باشد. این متود قیمت یک حیوان خانگی را به عنوان ورودی دریافت کرده و تخفیفی بر اساس معیارهای زیر اعمال کند:

- اگر قیمت بیشتر یا مساوی با ۱۰۰ دلار باشد، یک تخفیف ۱۰٪ اعمال کنید.

- اگر قیمت کمتر از ۱۰۰ دلار باشد، یک تخفیف ۵٪ اعمال کنید.

- قیمت تخفیف‌دار را برگردانید.

۷. یک کلاس به نام `HybridPet` پیاده‌سازی کنید که همزمان از کلاس‌های `Dog` و `Bird` بهره‌برداری کند. متود `make_sound` را بازنویسی کنید تا صدایی ترکیبی از صداهای یک سگ و یک پرنده چاپ کند.

۸. یک کلاس به نام `PropertyPet` پیاده‌سازی کنید که از کلاس `Pet` بهره‌برداری کند و یک خاصیت به نام `price_per_month` داشته باشد. باید براساس قیمت حیوان خانگی تقسیم بر سن حیوان محاسبه شود. متدهای لازم برای پشتیبانی از این خاصیت را پیاده‌سازی کنید.

۹. حالا نمونه‌هایی از کلاس‌های مختلف ایجاد کنید و پیاده‌سازی خود را تست کنید. شما باید قادر به ایجاد حیوانات خانگی، فروش آن‌ها، محاسبه تخفیف، و دسترسی به تعداد کل حیوانات فروخته شده باشید.

توجه: ممکن است لازم باشد متدها یا ویژگی‌های دیگری اضافه کنید تا خواسته‌های سوال را برآورده کنید.

سوال سه) شما در حال کار بر روی یک پروژه هستید که شامل دانلود فایل‌های موسیقی از چند وبسایت مختلف به صورت همزمان است. برای ساده‌تر کردن این فرایند، تصمیم می‌گیرید از پکیج `requests` برای دانلود فایل‌ها و `BeautifulSoup` برای تجزیه صفحات وب استفاده کنید. علاوه بر این، می‌خواهید یک دکوراتور پیاده‌سازی کنید که زمان اجرای توابع دانلود را اندازه‌گیری کرده و آن را چاپ کند.

یک برنامه Python بنویسید که موارد زیر را انجام دهد:

۱. یک دکوراتور به نام `measure_execution_time` تعریف کنید که زمان اجرای یک تابع را اندازه‌گیری کرده و زمان گذشته به ثانیه را چاپ کند. این دکوراتور باید برای هر تابعی قابل استفاده باشد.

۲. از دکوراتور `measure_execution_time` برای اندازه‌گیری زمان اجرای تابع زیر استفاده کنید:

```
```python
@measure_execution_time
def download_file(url, save_path):
 # Implement the logic to download the file using the requests package
 # Save the downloaded file to the specified save_path```
```

۳. یک تابع به نام `get_files` پیاده‌سازی کنید که لیستی از آدرس‌های وب را به عنوان ورودی دریافت کرده و با استفاده از روش چندنخی، فایل‌ها را به صورت همزمان دانلود کند. این تابع برای هر آدرس یک نخ جداگانه ایجاد کرده و تابع `download_file` را برای دانلود فایل فراخوانی کند. فایل‌های دانلود شده باید با نام‌های منحصر به فرد ذخیره شوند، مثلاً با استفاده از هش آدرس وب یا یک برچسب زمانی. در نهایت، تابع باید یک لیست از مسیرهایی که فایل‌ها در آن‌ها ذخیره شده‌اند، را برگرداند.

```
```python
def get_files(urls):
    # Implement the multithreading logic here
    # Create threads and call download_file function for each URL
    # Download the files and save them with unique names
    # Return a list of the paths where the files are saved```
```

۴. با ارائه لیستی از آدرس‌ها به تابع `get_files`` برنامه خود را تست کنید و مسیرهایی که فایل‌ها در آن‌ها ذخیره شده را چاپ کنید. اطمینان حاصل کنید که آدرسها حاوی فایل‌های موسیقی مختلف هستند.