

پارسا فود

شرح تمرین

پارسا می‌خواهد از داشتن اسنپ فود خودکفا شود تا کمیسیون به اسنپ نپردازد پس از شما می‌خواهد که برایش سیستمی شبیه به اسنپ فود را برایش پیاده سازی کنید. علاوه بر آن برای اینکه این سیستم دقیقاً همان چیزی شود که می‌خواهد، به طور دقیق نحوه پیاده سازی آن را برای شما بیان کرده است

الزامات پیاده‌سازی

کلاس والد (Food)

یک کلاس انتزاعی (Abstract) به نام Food ایجاد کنید که تمامی غذاها از آن ارث‌بری کنند. این کلاس شامل موارد زیر است:

- **متغیر food_id** : هر غذا باید دارای یک شناسه‌ی یکتا باشد که به صورت خودکار مقداردهی شود.
- **متغیر name** : نام غذا را ذخیره کند.
- **متد calculate_price()** : یک متد انتزاعی که باید در کلاس‌های فرزند **بازنویسی (Override)** شود.
- **Overloading عملگر +** : امکان جمع کردن دو غذا و محاسبه قیمت مجموع آن‌ها را فراهم کند. همچنین، امکان ضرب قیمت غذا در یک عدد صحیح را فراهم کند.

کلاس Pizza

کلاس Pizza از Food ارث‌بری کرده و دارای ویژگی‌های زیر است:

- **اندازه‌ی پیتزا** (Small , Medium , Large) که روی قیمت تأثیر دارد.
- **نوع پیتزا** (مانند Pepperoni یا Margherita).
- **لیستی از افزودنی‌های اضافی** (Cheese , Extra Sauce , Olives), که قیمت نهایی را تغییر می‌دهند.
- **بازنویسی متد calculate_price()** برای محاسبه قیمت نهایی.

کلاس Burger

کلاس Burger از Food ارث‌بری کرده و دارای ویژگی‌های زیر است:

- **تعداد لایه‌های گوشت** (Triple , Double , Single) که بر قیمت تأثیر دارد.
- **نوع نان** (Brioche , Sesame , Regular) که روی قیمت تأثیر دارد.
- **لیستی از افزودنی‌های اضافی** (Egg , Bacon , Cheese) که قیمت را تغییر می‌دهند.
- **بازنویسی متد** calculate_price() برای محاسبه قیمت نهایی.

کلاس Drink

کلاس Drink از Food ارث‌بری کرده و شامل ویژگی‌های زیر است:

- **حجم نوشیدنی** (1L , 500ml , 300ml) که روی قیمت تأثیر دارد.
- **نوع نوشیدنی** (Water , Juice , Soda) که ممکن است روی قیمت اثر داشته باشد.
- **بازنویسی متد** calculate_price() برای محاسبه قیمت نهایی.

جدول قیمت غذاها

نوع غذا	حالت‌ها	قیمت (دلار)
Pizza	Small	8\$
	Medium	12\$
	Large	16\$
	Extra Cheese	2\$+
	Extra Sauce	1.5\$+
	Olives	1\$+
Burger	Single Layer	6\$

نوع غذا	حالت‌ها	قیمت (دلار)
	Double Layer	9\$
	Triple Layer	12\$
	Cheese	1\$+
	Bacon	2\$+
	Egg	1.5\$+
Drink	300ml	2\$
	500ml	3\$
	1L	5\$

کلاس Order

یک کلاس به نام Order پیاده‌سازی کنید که شامل ویژگی‌ها و قابلیت‌های زیر باشد:

- ذخیره‌ی چندین آیتم غذا همراه با تعداد آن‌ها در یک دیکشنری.
- متد `add_item(food, quantity)` برای اضافه کردن غذا به سفارش.
- متد `remove_item(food_id)` برای حذف یک آیتم از سفارش.
- متد `calculate_total()` برای محاسبه‌ی قیمت کل سفارش.
- متد `apply_discount(code)` برای اعمال کد تخفیف (مانند DISCOUNT10 برای ۱۰٪ تخفیف).
- متد `display_order()` برای نمایش جزئیات سفارش.
- دیکشنری کد های تخفیف را همراه با درصد تخفیف آن ها نیز باید در همین کلاس باشد.

نمونه کد که باید به درستی اجرا شود

```

pizza1 = Pizza("Large", extras=["Cheese", "Extra Sauce"])
pizza2 = Pizza("Medium", extras=["Olives"])

```

```

burger1 = Burger("Double", extras=["Bacon", "Cheese"])
burger2 = Burger("Triple", extras=["Egg"])
drink1 = Drink("500ml", "Soda")
drink2 = Drink("1L", "Juice")

test_order = Order()
test_order.add_item(pizza1, 2)
test_order.add_item(pizza2, 1)
test_order.add_item(burger1, 1)
test_order.add_item(burger2, 2)
test_order.add_item(drink1, 3)
test_order.add_item(drink2, 2)

test_order.display_order()
print(f"Total price after DISCOUNT10: ${test_order.apply_discount('DISCOUNT10')}")
print(f"Total price after DISCOUNT20: ${test_order.apply_discount('DISCOUNT20')}")
print(f"Total price after DISCOUNT30: ${test_order.apply_discount('DISCOUNT30')}")

```

مثال خروجی مورد انتظار

```

Order Summary:
2x Large Pizza with extras: Cheese, Extra Sauce (ID: 1) - $19.5 each
1x Medium Pizza with extras: Olives (ID: 2) - $13.0 each
1x Double Burger with extras: Bacon, Cheese (ID: 3) - $12.0 each
2x Triple Burger with extras: Egg (ID: 4) - $13.5 each
3x 500ml Soda (ID: 5) - $3 each
2x 1L Juice (ID: 6) - $5 each
Total Price: $110.00
Total price after DISCOUNT10: $99.0
Total price after DISCOUNT20: $88.0
Total price after DISCOUNT30: $77.0

```

نکات مهم هنگام پیاده‌سازی

۱. از کلاس‌های انتزاعی (Abstract Classes) برای ایجاد کلاس والد Food استفاده کنید.

۲. حتماً متد calculate_price() را در کلاس‌های فرزند بازنویسی (Override) کنید.

۳. برای مدیریت شناسه‌ی یکتا (food_id) از یک **متغیر استاتیک** در کلاس Food استفاده کنید.
۴. **Overloading عملگر +** را پیاده‌سازی کنید تا بتوانید جمع غذاها را انجام دهید.
۵. از یک **دیکشنری (dict)** برای ذخیره غذاها در Order استفاده کنید.
۶. تخفیف‌ها را در یک **دیکشنری از کدهای تخفیف** در نظر بگیرید.
۷. هنگام نمایش سفارش، **جزئیات هر آیتم** (مانند نام، تعداد و قیمت) را نمایش دهید.