



Home Work

Data Science & Machine Learning

Class, OOP

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. کلاس حساب بانکی

کلاسی با نام Account بسازید که به عنوان abstract برای سه کلاس دیگر مورد استفاده واقع می‌شود. این کلاس دو مشخصه balance و number را دارد که به ترتیب نشان دهنده موجودی و شماره حساب می‌باشند. همچنین این کلاس دارای مشخصه دیگری تحت عنوان transactions می‌باشد که لیست تراکنش‌های انجام شده را نشان می‌دهد. برای این کلاس سه متد برای واریز به حساب، برداشت از حساب و نمایش موجودی تعریف کنید. دقت کنید که پس از هر برداشت یا واریز مقدار مبادله شده در لیست تراکنش‌ها ذخیره شود (مقدار مثبت در صورت واریز و مقدار منفی در صورت برداشت).

پ.ن: متد __str__ این کلاس را به نحوی تعریف کنید که به صورت number: balance نمایش داده شود و balance را تا دو رقم اعشار گرد کنید.

۲. کلاس حساب جاری

کلاسی با نام CheckingAccount بسازید که از کلاس Account ارث بری می‌کند. یک مشخصه کلاسی برای این کلاس تحت عنوان min_amount تعریف کنید که حداقل مقدار موجودی لازم برای تشکیل حساب را نمایش دهد (به عنوان مثال ۱۰۰۰۰۰) و بر اساس این مقدار تشکیل حساب جدید انجام شود. همچنین متد __str__ آن را به نحوی بنویسید که خروجی آن به شکل زیر باشد:

```
"Checking Account, Number: number, Balance: balance"
```

۳. کلاس حساب پس انداز

کلاسی با نام SavingAccount بسازید که از کلاس Account ارث بری می‌کند. متدی برای این کلاس پیاده سازی نمایید که برای دارنده حساب نسبت دخل و خرج حساب را محاسبه نماید (نسبت مجموع واریزها به مجموع برداشت‌ها). متد __str__ آن را به نحوی بنویسید که خروجی آن به شکل زیر باشد:

```
"Saving Account, Number: number, Balance: balance"
```

۴. کلاس حساب ارزی

کلاسی با نام CurrencyAccount بسازید که از کلاس Account ارث بری می‌کند. یک متد ایستا برای این کلاس طراحی کنید که وظیفه آن تبدیل واحد ریال به دلار باشد به طوری که دارنده حساب ارزی قابلیت مشاهده موجودی خود بر اساس دلار قیمت روز را داشته باشد. متدی برای این کلاس طراحی کنید که با استفاده از آن دارنده حساب لیست تراکنش‌های خود براساس دلار را مشاهده نماید. متد __str__ آن را به نحوی بنویسید که خروجی آن به شکل زیر باشد:

```
"Business Account, Number: number, Balance: balance"
```

۵. کلاس مشتری - بخش اول

کلاسی با نام `Costumer` بسازید که ۳ مشخصه `ssn`، `name` و `acc_type` را داشته باشد که `acc_type` دارای مقادیر مجاز `B`، `S` و `C` است. همچنین `ssn` و `name` نیز نمایانگر نام و کد ملی مشتری می باشد. سه متد برای افتتاح حساب ایجاد کنید که هر یک از متدها متناظر با انواع حسابی است. که در بالا تعریف شده است. ورودی این سه متد موجودی و شماره حساب می باشد.

۶. کلاس مشتری - بخش دوم

برای کلاس `Costumer` متدهای زیر را تعریف کنید:

متد: `show_balance` این متد موجودی حساب مشتری را با فرمت مناسب نمایش می دهد.

متد: `make_deposit` این متد مقداری را به عنوان ورودی می گیرد و عملیات واریز به حساب را برای مشتری انجام میدهد.

متد: `make_withdraw` این متد مقداری را به عنوان ورودی می گیرد و عملیات برداشت از حساب را برای مشتری انجام میدهد.

متد: `make_transfer`: این متد ۳ مقدار حساب مبدا، حساب مقصد و مقدار انتقال را از ورودی متد می گیرد و عملیات انتقال را با استفاده از برداشت و واریز انجام میدهد و در نهایت مقادیر موجودی دو حساب مبدا و مقصد را نمایش میدهد.

پ.ن: است کردن مابقی `dunder method` ها (مانند `__add__`، `__lt__` و `__eq__`) و استفاده از آنها برای عملیات گفته شده دارای نمره امتیازی می باشد.

Happy Coding 😊