

۱. کتابخانه ی مرکزی شهرمان را مدل سازی کنیم! :

هر کتابخانه مرجعی دارد که با قفسه های زیاد و برچسب گذاری شده ، از کتاب ها با دسته بندی های گوناگون ، با نظم نگهداری میکند. کتابدار برای مدیریت و کمک به کتابخوانان در دسترس است.

قفسه ها:

هر قفسه باید رفتاری شبیه به لیست داشته باشد،
هر قفسه نهایتاً گنجایش ۱۰ هزار صفحه را دارد (جمع صفحات کتاب هایی که درون آن قرار میگیرند بیش از این تعداد نمیتواند باشد)
باید برچسبی داشته باشد با شناسه ی غیر قابل تکرار که ترکیبی از شناسه دسته بندی آن کتاب ها - قفسه ی nام است:

مثال :

برچسب قفسه ۱ : شناسه-دسته-بندی = ۱۲۳۴۳۱ - قفسه ی اول = ۱ —>
نتیجه : ۱-۱۲۳۴۳۱

کتابدار (از جنس انسان است) میتواند رفتار هایی هم داشته باشد :

- جستجو کتاب : با استفاده از گرفتن نام یا شناسه یا اطلاعاتی از نویسنده (کد نویسنده یا نام آن) از شما، لیستی از کتاب هایی که پیدا کرده است را برمیگرداند. (لیست حاوی کتاب هایی است که به صورت روبرو نمایش داده میشود: نام کتاب - قفسه کتاب - نویسنده - شناسه کتاب)
- گرفتن کتاب با استفاده از شناسه کتاب
- افزودن دسته بندی جدید در کلاس دسته بندی ها
- جستجو و برگرداندن کتاب در قفسه ی مورد نظر با استفاده از برچسب قفسه و شناسه یا نام یا اطلاعات نویسنده ی کتاب

دسته بندی ها:

باید از سمت کتابدار، دسته بندی ها افزوده شود!

هیچ دسته بندی تکراری وجود ندارد!

هر دسته بندی رفتاری دارد، تعداد کتاب های هر دسته بندی باید بصورت روبرو قابل درک باشد :
(len(actionCategory

کتاب:

هر کتاب اطلاعات مختلفی دارد:
اطلاعات اجباری ، غیر قابل تغییر (تعداد صفحات، شناسه ی یکتا، قیمت، دسته بندی، نویسنده)

کلاس هایی که باتوجه به دید خود میتوانید رفتار هایشان را ارتقا دهید، ولی استفاده از آنها اجباری است:

- کلاس مخزن - مرجع (رفتار های جامع : (len - تعداد کل کتاب ها برگشت داده شود!))
- نویسنده
و ...

از Mixin ، Abstraction, Encapsulation, Polymorphism, Inheritance باید بروش
صحیح استفاده کنید.

بکار گیری composition attr اجباری است (باتوجه به الگوی ذهنی خود ، از این قابلیت استفاده کنید! (مثال : هنگام تعریف مقادیر مربوط به نویسنده هر کتاب ، میتوانید از کلاس نویسنده (انسان در نقش نویسنده) استفاده کنید)

۲. میخواهیم کیف پول دیجیتالی خود را پیاده سازی کنیم:

- حداقل شی های مورد نیاز :
- پول با واحد های مختلف (دلار- یورو - تومان - ریال)
 - کیف پول (محلی برای نگهداری پول ها با شناسه ی اختصاصی و یکتا)
 - کارت های اعتباری (بانکی، سوخت، تلفن یا حتی ملی)
 - رسید ها (رسید های کارت های اعتباری)
 - اطلاعات بازایابی (اطلاعاتی که در صورت عدم دسترسی با کد ، کیف پول قابل بازایابی باشد .)

نکات :

- در کیف پول شما هر ورود و خروجی باید ذخیره شود و رفتاری بر مبنای نمایش تمام تراکنش ها بصورت لیست (منظور لیست پایتون نیست!) محور وجود داشته باشد!

- کیف پول شما اطلاعات ورودی خاصی جز موارد بازیابی و کد دسترسی ندارد (کد توسط برنامه ساخته میشود!) - میتوانید برای داده ی بازیابی پذیر از کد ملی استفاده کنید که نباید قابل تغییر باشد)
- رفتارهایی موجود باشد که موجودی کیف پول - موجودی تفکیکی هر مدل از پول - لیست رسید ها - حذف رسید دلخواه - کارت های اعتباری را ایجاب کند.

* امتیازی : برنامه را طوری بسازید که کلاس wallet امکان ساخت Instance را داشته باشد ولی هیچ instance برنگردد!
به مثال روبرو توجه کنید:

ali_wallet = Wallet(...)

قابل assign شدن نباشد و اررور دهد!
رفتار اینگونه باشد :

```
Wallet.create_wallet(...) # Code = 121212  
Wallet.get_wallet(121212) # return your wallet instance
```

در عمل اگر رفتاری برای دریافت تراکنش ها را در نظر بگیریم :
Wallet.get_wallet(121212).get_transactions()