CS50x Iran

Cookie Jar

Problem set

پاییز و زمستان ۱۴۰۴

فرض کنید که می‌خواهید یک ظرف کلوچه بسازید تا کلوچه‌ها را در آن نگه‌داری کنید. در فایلی به نام jar.py، یک class به اسم Jar با method هایی که در صفحه‌ی پیش رو خواهید دید، بسازید.

- `__init__`: این method باید یک ظرف کلوچه به نام capacity را مقداردهی اولیه کند که نشان دهنده‌ی حداکثر تعداد کلوچه‌هایی است که می‌تواند در یک ظرف (jar) قرار گیرد. اگر capacity یک مقدار غیر منفی int نبود، تابع `__init__` باید یک Error از نوع `ValueError` اعلام کند.
- `__str__` : این method باید یک str با n🍪 را در خروجی به ما برگرداند که در آن n🍪، تعداد کلوچه‌های داخل ظرف کلوچه است. برای مثال، اگر 3 تا کلوچه در داخل ظرف کلوچه است، بنابراین باید این متد در خروجی "🍪🍪🍪" را به ما بازگرداند.
- `deposit`: این method بایستی n تعداد کلوچه را به ظرف کلوچه اضافه کند. اگر اضافه کردن این تعداد باعث تجاوز از حداکثر ظرفیت ظرف کلوچه دارد، در این صورت این method باید یک Error از نوع `ValueError` اعلام کند.
- `withdraw`: این method بایستی n تعداد کلوچه را از ظرف کلوچه حذف کند. اگر حداقل به آن تعداد کلوچه در ظرف موجود نباشد، این method بایستی یک Error از نوع `ValueError` اعلام کند.
- `capacity`: این method بایستی مقدار ظرفیت ظرف کلوچه را در خروجی اعلام کند.
- `size`: این method باید کلوچه‌هایی که در اصل در ظرف کلوچه موجود است در خروجی به ما بازگرداند.

به فرم زیر، class خود را ساختاربندی کنید. شما نمی‌توانید پارامترهای متد ها را تغییر دهید، اما می‌توانید متدهای خود را اضافه کنید:

jar.py

```
class Jar:
    def __init__(self, capacity=12):

    def __str__(self):

    def deposit(self, n):

    def withdraw(self, n):

    @property
    def capacity(self):

    @property
    def size(self):
```

قبل یا بعد از پیاده سازی jar.py در فایلی به نام test_jar.py، حداقل 4 تابع دیگر در class خود قرار دهید تا به طور کامل پیاده سازی class خود را تست کنید. اسم تمامی توابع داخل فایل تست بایستی با test_ شروع شود تا بتوانید با اجرای دستور `pytest test_jar.py`، تست برنامه خود را انجام دهید!

به یاد داشته باشید که تست کردن instance methodها (methodها یک نمونه از یک class) به سادگی تست کردن یک تابع به تنهایی نیست، چرا که بعضی اوقات این Instance method باعث تغییر در همان state می‌شود. (مثلاً متغیر های یک نمونه از یک class) ممکن است برای تست کردن یک method (مثلا withdraw) نیاز باشد یک method دیگری را فراخوانی کنید (مثلاً deposit) اما ممکن است methodی که از ابتدا صدا می‌زنید درست نباشد.

بنابراین گاهی اوقات برنامه‌نویس ها، برای تست کردن methodها یک mock object درست می‌کنند که به وسیله آن عملکرد methodها را تست کنند. زبان پایتون یک کتابخانه mock object library دارد که شما می‌توانید فقط یک method را فراخوانی کنید و اول اون لایه متدی پایین را تغییر دهید بدون آن که method دیگری را فراخوانی کنید.

اگر چه برای ساده تر شدن، هیچ بخشی را نیاز نیست mock کنید. بلکه می‌توانید تست های خود را همانند قبل پیاده‌سازی کنید!

راهنمای تست

```
from jar import Jar


def test_init():
    ...


def test_str():
    jar = Jar()
    assert str(jar) == ""
    jar.deposit(1)
    assert str(jar) == "🍪"
    jar.deposit(11)
    assert str(jar) == "🍪🍪🍪🍪🍪🍪🍪🍪🍪🍪🍪🍪"


def test_deposit():
    ...


def test_withdraw():
    ...
```

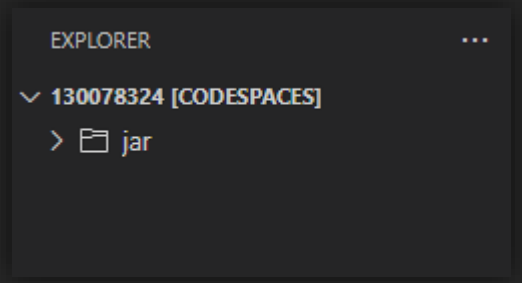
6 قبل از شروع توجه کنید:

وارد code.cs50.io شوید، سپس بر روی پنجره‌ی Terminal کلیک کنید. توجه داشته باشید که دستور پنجره‌ی Terminal شما باید به صورت زیر باشد:

```
$
```

سپس کد زیر را اجرا کنید تا یک پوشه به نام jar در codespace ایجاد شود:

```
$ mkdir jar
```



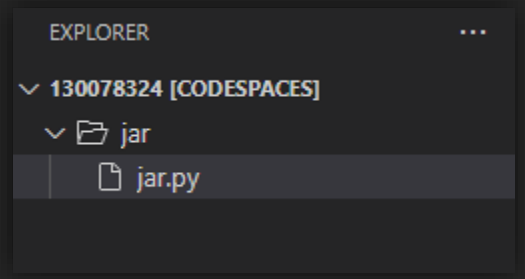
7 قبل از شروع توجه کنید:

سپس `cd jar` را اجرا کنید تا به پوشه‌ی jar منتقل شوید. اکنون شما باید آدرس `jar/ $` را در Terminal مشاهده کنید:

```
$ cd jar
jar/ $
```

حالا می‌توانید دستور `code jar.py` را اجرا کنید تا فایلی با نام jar.py ایجاد شود:

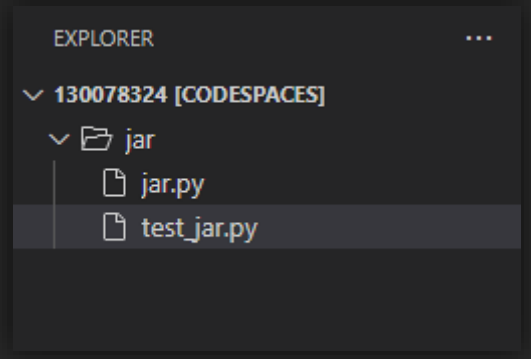
```
jar/ $ code jar.py
```



8 قبل از شروع توجه کنید:

یادتان باشد که دستور `code test_jar.py` را نیز اجرا کنید تا فایلی با نام test_jar.py ایجاد کنید:

```
jar/ $ code test_jar.py
```



اکنون می‌توانید در این فایل برنامه‌ی خود را بنویسید.

- فایل test_jar.py را باز کنید و کلاس Jar را با دستور from jar import Jar را import کنید. یک تابع به اسم `test_init` بسازید که در آن یک نمونه (instance) از کلاس `Jar` را با دستور `jar = Jar()` بگیرید. با دستور `assert` مطمئن شوید که این نمونه jar، ظرفیتی که باید داشته باشد را دارد، سپس تست‌های خود را با دستور `pytest test_jar.py` اجرا کنید.

- یک تابع دیگر به اسم `test_str` در فایل test_jar.py ایجاد کنید. در تابع `test_str`، یک نمونه از کلاس `Jar` ایجاد کنید و تعدادی cookie با استفاده از `deposit` ایجاد کنید. با دستور `assert`، مطمئن شوید که دستور `str(jar)`، هر تعداد کلوچه‌ای که با دستور `deposit` ایجاد شده است را در خروجی نمایش می‌دهد. سپس تست‌های خود را با استفاده از دستور pytest test_jar.py اجرا کنید. برنامه خود را با `pytest test_jar.py` اجرا کنید.

- یک تابع دیگر در فایل test_jar.py، به اسم `test_deposit` اضافه کنید. در این تابع، یک نمونه از کلاس `Jar` ایجاد کنید و تعدادی cookie با استفاده از `deposit` ایجاد کنید. با دستور `assert`، مطمئن شوید که دستور `str(jar)`، هر تعداد کلوچه‌ای که با دستور `deposit` ایجاد شده است را در خروجی نمایش می‌دهد. همچنین با دستور `assert` مطمئن شوید که اگر بیشتر از ظرفیت (capacity) کلوچه با تابع `deposit` ساخته‌اید، تابع `deposit` یک Error از نوع `ValueError` اعلام کند. تست های خود را با دستور `pytest test_jar.py` اجرا کنید.

- تابع دیگری در فایل test_jar.py، به اسم `test_withdraw` ایجاد کنید. در تابع `test_withdraw`، یک نمونه از کلاس `Jar` ایجاد کنید و تعدادی cookie با استفاده از `deposit` ایجاد کنید. با `assert` مطمئن شوید که با استفاده از `withdraw`، به اندازه مقدار size، کلوچه در ظرف (jar) باقی می‌ماند. همچنین دوباره با دستور `assert` مطمئن شوید که اگر بیشتر از مقدار size، تابع `withdraw` کلوچه حذف کند، این تابع بایستی یک Error از نوع `ValueError` اعلام کند. با استفاده از دستور `pytest test_jar.py` تست های خود را اجرا کنید.

❖ شما می‌توانید از آدرس زیر برای بررسی کردن کد خود استفاده کنید. CS50 از این برنامه برای آزمایش کد شما استفاده می‌کند. از این دستور استفاده کنید تا کدهایتان را امتحان کنید.

```
$ check50 cs50/problems/2022/python/jar
```

لبخند های سبز به این معنی هستند که برنامه‌ی شما در تست قبول شده و اخم های قرمز یعنی برنامه‌ی شما دارای ایراد هست. با مراجعه به check50 URL می‌توانید خروجی مورد انتظار و خروجی برنامه‌ی خود را بررسی کنید.

با اجرا کردن صورت زیر در Terminal پاسخ خود را ارسال کنید.

```
$ submit50 cs50/problems/2022/python/jar
```

CS50x Iran

Harvard's Computer Science 50x Iran