

#W5PA: LEGB Mock

Ungraded programming assignment

1. Описание задания	2
2. Библиотека по работе с активами (Asset)	2
3. Задания	2



1. Описание задания

В этом задании вам нужно настроить Mock (в enclosing namespace), который будет иметь динамическое поведение в зависимости от числа вызовов. Цель задания:

1. Научить строить Mock'и;
2. Научить пользоваться правилом LEGB.

2. Библиотека по работе с активами (Asset)

Базовая версия библиотеки по работе с активами и ее консольный интерфейс доступны на github учебного курса:

- <https://github.com/big-data-team/python-course>
- см. asset.py, asset_example.txt

Инструмент предоставляет возможность получить оценки инвестиционной прибыльности по продукту за интересующий период в будущем (в годах).

Например, вызов:

```
asset.py -f asset_example.txt -p 1 2 5
```

выведет информацию о доходности продукта из указанного файла (asset_example.txt) за 1, 2 и 5 лет в будущем.

3. Задания

Prerequisites:

1. Настройка окружения: <https://github.com/big-data-team/python-course>

С учетом примеров и лайфхаков, показанных на этой неделе:

1. Напишите тесты и создайте Mock'и, показанные в примерах на этой неделе. С целью наработки практических навыков по определению правильного namespace и указания порядка @patch - copy-paste решения тестов не выкладываются. Чеклист:
 - a. Сделать @patch на load_asset_from_file и проверить число обращений и передаваемые аргументы;
 - b. Расширение функциональности Asset для имитации бурной деятельностью с помощью sleep (три варианта) и наложение @patch на "сон";



- c. Наложить `@patch` на метод класса `Asset`, чтобы возвращать интересующее значение и проверить это с помощью `pytest fixture capsys` или `caplog`;
 - d. Расширить функциональность `Asset` для перевода курса валют: добавьте вызов нереализованной функции из другой библиотеки (e.g. `cbr`) и замокайте его, чтобы возвращать какое-то осмысленное значение.
2. Бонусный трек: в функции `print_asset_revenue` вы можете заметить закомментированные строки. Заменить имеющийся `print` на `print`, который закомментирован. Посмотрите какие тесты сломаются, поймите почему и исправьте проблему.
3. Задание “никогда такого не было, и вот опять”. В рамках пункта задания “1.d” у вас появилась внешняя зависимость, которая возвращает курс USD по отношению к рублю. Создайте `Mock`, которой при обращении за курсом USD будет каждый раз немного увеличивать возвращаемое значение. Для усвоения правила LEGB добавляется обязательное требование - `Mock` и вспомогательные функции должны быть реализованы полностью внутри одной тест-функции.
4. С целью привития хорошей культуры написания кода в дополнение к уже изученным ранее PEP-257 и PEP-008 на этой неделе рекомендуется изучить `Google Python Style Guide`. Изучите рекомендации по хорошему написанию кода:
 - a. <https://google.github.io/styleguide/pyguide.html>
 - b. У автора курса точно есть свои привычки и аргументы, которые расходятся с рекомендациями `Google Python Style Guide`. Предлагаем в чате курса обсудить с какими аргументами вы например не согласны, а что интересного и нового почерпнули из этого документа. Хештеги для удобства поиска - `#google-style-guide` `#discussion`.

Материалы для погружения:

1. `Mock`, официальная документация:
 - <https://docs.python.org/dev/library/unittest.mock.html>