Динамические атрибуты объекта

issue-01

Даны объявления в формате JSON (https://ru.wikipedia.org/wiki/JSON).

В объявлении могут присутсвовать различные поля любой вложенности.

Пример объявления с атрибутом 'ближайшие станции метро' (`metro_stations`):

```
{
  "title": "iPhone X",
  "price": 100,
  "location": {
    "address": "город Самара, улица Мориса Тореза, 50",
    "metro_stations": ["Спортивная", "Гагаринская"]
  }
}
```

Пример объявления с атрибутом 'категория' (`class`):

```
{
  "title": "Вельш-корги",
  "price": 1000,
  "class": "dogs",
  "location": {
    "address": "сельское поселение Ельдигинское, поселок санатория Тишково, 25"
  }
}
```

Напишите класс `Advert`, который:

- динамически создает атрибуты экземпляра класса из атрибутов JSON-объекта
    - не нужно фиксировать атрибуты в классе
      пример ниже **НЕВЕРНЫЙ**

      ```
      # НЕВЕРНЫЙ ПРИМЕР! Создавайте атрибуты динамически
      class Advert:
          def __init__(self, mapping):
              self.title = mapping['title']
              self.price = mapping['price']
              ...
      ```

    - к атрибутам любой вложенности можно обращаться через точку

```
# создаем экземпляр класса Advert из JSON
lesson_str = """{
    "title": "python",
    "price": 0,
    "location": {
        "address": "город Москва, Лесная, 7",
        "metro_stations": ["Белорусская"]
        }
    }"""
lesson = json.loads(lesson_str)
lesson_ad = Advert(lesson)

# обращаемся к атрибуту location.address
lesson_ad.location.address

# Out: 'город Москва, Лесная, 7'
```

- к названия атрибутов, являющихся ключевыми словами python, в конце название добавляем _

```
# создаем экземпляр класса Advert из JSON
dog_str = """{
    "title": "Вельш-корги",
    "price": 1000,
    "class": "dogs"
    }"""
dog = json.loads(dog_str)
dog_ad = Advert(dog)

# обращаемся к атрибуту `dog_ad.class_` вместо `dog_ad.class`
dog_ad.class_

# Out: 'dogs'
```

- подсказка-01: создайте отдельный класс, который будет преобразовывать JSON-объекты в python-объекты с доступом к атрибутам через точку и используйте его для разбора полей объявления и вложенного поля `location`
- подсказка-02: в python есть функция, которая проверяет, является ли строка ключевым словом

- имеет свойство price
    - проверяет, что значение не отрицательное и при создании объекта и при присваивании

```
lesson_str = '{"title": "python", "price": -1}'
lesson = json.loads(lesson_str)
lesson_ad = Advert(lesson)

# Out: ValueError: must be >= 0

lesson_str = '{"title": "python", "price": 1}'
lesson = json.loads(lesson_str)
lesson_ad = Advert(lesson)
lesson_ad.price = -3

# Out: ValueError: must be >= 0
```

- в случае отстувия поля price в JSON-объекте возвращает 0

```
lesson_str = '{"title": "python"}'
lesson = json.loads(lesson_str)
lesson_ad = Advert(lesson)
lesson_ad.price

# Out: 0
```

issue-02

Добавьте текстовое представление объектов класса Advert.

Пример ожидаемого использования:

```
iphone_ad = Advert({'title': 'iPhone X', 'price': 100})
print(iphone_ad)

# Out: iPhone X | 100 ₽
```

Напишите миксин ColorizeMixin, который:

- меняет цвет текста при выводе на консоль
- задает цвет в атрибуте класса repr_color_code

```
class Advert(ColorizeMixin, ?, ?):
    repr_color_code = 32  # green
```

- использование комбинации __repr__ и __str__ **НЕВЕРНО**. Нужно использовать или __repr__ или __str__.
- подсказка-01: про изменение цвета можно почитать тут - Add Colour to Text in Python (http://ozzmaker.com/add-colour-to-text-in-python/)
- подсказка-02: данный миксин похож на EmojiMixin с лекции. Реализацию можно переиспользовать.
- подсказка-03: для "красивого" решения можно воспользоваться методом __init_subclass__ (https://docs.python.org/3/reference/datamodel.html#object.__init_subclass__)

DoD (Definition of Done) - критерии, позволяющие понять, что задача сделана, как ожидается:

- написан класс, экземпляры которого позволяют обращаться к полям через точку: `iphone_ad.price`
- класса `Advert` не содержит атрибуты при объявлении. Исключение: реализация свойства `price`
- атрибут `title` - обязательный. При его отсутствии выбрасываем исключение `ValueError`
- экзмепляры класса `Advert` инициалируются из **словаря**
- поле `Advert.price` выбрасывает исключение при установке отрицательного значения и при создании объекта и при присваивании
- выводится адрес при обращении к атрибуту через точки: `iphone_ad.location.address`
- выводит категорию при обращении через точку: `corgi.class_`
- при выводе обяъвления в консоли `print(corgi)` получаем надпись 'Вельш-корги | 1000 ₽' желтым цветом
- нет замечаний от `flake8`