

Python (BSU FAMCS Fall'18)

Домашнее задание 4

Преподаватели: Дмитрий Косицин, Светлана Боярович

Реализуйте умный дескриптор `smart_property` (можно другое имя), который действует аналогично свойству `property`, но выполняет дополнительные действия благодаря мета-информации.

Пример

```
class A(object):
    @smart_property
    def path_to_docs(self):
        """
        <property description>

        :type: str
        :default: ./some_folder/another_folder

        :check_exists
        :abspath
        """
        pass

a = A()
assert a.path_to_docs == os.path.abspath("./some_folder/another_folder")
```

Декорированный и описанный таким образом метод должен привести к созданию свойства (с *getter*-ом и *setter*-ом) с именем `path_to_docs`. Значение по умолчанию, которое примет свойство, будет иметь тип `str` и значение «./some_folder/another_folder». При установке нового значения, оно будет проверено на соответствие типу `str`.

Указанный спецификатор `check_exists` говорит о том, что в некоторый момент (продумайте: при каждом вычислении или только при установке значения?) будет проверяться существование пути. Спецификатор `abspath` преобразует установленный путь в абсолютный.

Предусмотрите возможность взятия значения без применения спецификаторов, например как `a.smart_property.raw_value`. В таком случае в примере выше должен быть возвращен именно относительный путь.

Продумайте, как указать и проверять несколько допустимых типов, например, `int` и `float`. Отдельно следует обработать случай значения `None` – логично, чтобы некоторые свойства (или все) допускали установку такого значения.

Продумайте формат для указания некоторого допустимого выражения в качестве значения по умолчанию, например, `2 ** 22` (можно для этого использовать `ast.literal_eval`). Помните, что выполнение произвольного кода недопустимо! Более сложные конструкции реализуйте самостоятельно, ограничив возможности с помощью спецификаторов.

Предусмотрите возможность легкого добавления *новых* спецификаторов, не требующее изменения логики кода.

Добавьте еще несколько спецификаторов:

- `create_if_not_exists` – создает файл или папку, если ее не существует (не забудьте про корректную обработку ошибок и возможный race condition);
- `readonly` – создает свойство только с *getter*-ом;
- `mandatory` – к некоторому моменту времени обязано быть установлено в значение, отличное от `None` (продумайте, как проверить все *mandatory* поля).

Придумайте и реализуйте один или несколько своих спецификаторов.

Приложите к вашему коду *юнит-тесты*, покрывающие основной функционал.