

3.2 Метод `__call__`. Функторы и классы-декораторы

5 out of 11 steps passed 8 out of 25 points received

Видео-разбор подвига (решение смотреть только после своей попытки):

<https://youtu.be/e1U6SoX1ChI> (<https://youtu.be/e1U6SoX1ChI>)

Подвиг 4. Предположим, мы разрабатываем класс для обработки формы авторизации на стороне сервера. Для этого был создан следующий класс с именем `LoginForm`:

```
class LoginForm:
    def __init__(self, name, validators=None):
        self.name = name
        self.validators = validators
        self.login = ""
        self.password = ""

    def post(self, request):
        self.login = request.get('login', "")
        self.password = request.get('password', "")

    def is_validate(self):
        if not self.validators:
            return True

        for v in self.validators:
            if not v(self.login) or not v(self.password):
                return False

        return True
```

Здесь `name` - это заголовок формы (строка); `validators` - список из валидаторов для проверки корректности поля. В методе `post` параметр `request` - это словарь с ключами `'login'` и `'password'` и значениями (строками) для логина и пароля соответственно.

Пример использования класса `LoginForm` (в программе не писать):

```
from string import ascii_lowercase, digits

lg = LoginForm("Вход на сайт", validators=[LengthValidator(3, 50),
CharsValidator(ascii_lowercase + digits)])
lg.post({"login": "root", "password": "panda"})
if lg.is_validate():
    print("Дальнейшая обработка данных формы")
```

Вам необходимо в программе объявить классы валидаторов:

`LengthValidator` - для проверки длины данных в диапазоне `[min_length; max_length]`;

`CharsValidator` - для проверки допустимых символов в строке.

Объекты этих классов должны создаваться командами:

```
lv = LengthValidator(min_length, max_length) # min_length - минимально допустимая
длина; max_length - максимально допустимая длина
cv = CharsValidator(chars) # chars - строка из допустимых символов
```

Для проверки корректности данных каждый валидатор должен вызываться как функция:

```
res = lv(string)
res = cv(string)
```

и возвращать True, если string удовлетворяет условиям валидатора и False - в противном случае.

P.S. В программе следует только объявить два класса валидаторов, на экран выводить ничего не нужно.

To solve this problem please visit
<https://stepik.org/lesson/701987/step/5>