

PYTHON329 HW №20

Домашнее задание

Необходимо создать два декоратора для валидации пользовательских данных перед их сохранением в CSV формате.

Первый декоратор (`password_validator`) проверяет:

- сложность пароля
- включая его длину
- количество букв верхнего и нижнего регистра
- и количество спец-знаков.

Второй декоратор (`username_validator`) проверяет, что в имени пользователя **отсутствуют** пробелы. Создайте функцию (`register_user`), которая принимает имя пользователя и пароль, и дозаписывает их в CSV файл, обернув ее **обоими** декораторами.

Требования:

1. Декоратор `password_validator`:

- Принимает параметры: `min_length` (минимальная длина пароля, по умолчанию 8), `min_uppercase` (минимальное количество букв верхнего регистра, по умолчанию 1), `min_lowercase` (минимальное количество букв нижнего регистра, по умолчанию 1), `min_special_chars` (минимальное количество спец-знаков, по умолчанию 1).
- Проверяет, соответствует ли пароль заданным критериям.
- Если пароль не соответствует критериям, выбрасывает `ValueError` с описанием того, что именно не выполнено.

2. Декоратор `username_validator`:

- Не принимает параметров.
- Проверяет, что в имени пользователя отсутствуют пробелы.
- Если в имени пользователя есть пробелы, выбрасывает `ValueError` с подробным описанием проблемы.

3. Функция `register_user`:

- Принимает две строки: `username` (имя пользователя) и `password` (пароль).
- Дозаписывает имя пользователя и пароль в CSV файл.
- Оборачиваем функцию **обоими** декораторами для валидации введенных данных.

Пример использования:

```
import csv

def password_validator(length: int = 8, uppercase: int = 1, lowercase: int = 1, special_chars: int = 1):
    """
    Декоратор для валидации паролей.

    Параметры:
        length (int): Минимальная длина пароля (по умолчанию 8).
        uppercase (int): Минимальное количество букв верхнего регистра (по умолчанию 1).
        lowercase (int): Минимальное количество букв нижнего регистра (по умолчанию 1).
        special_chars (int): Минимальное количество спец-знаков (по умолчанию 1).

    Пример использования:
    @password_validator(length=10, uppercase=2, lowercase=2, special_chars=2)
    def register_user(username: str, password: str):
        pass
    """

def username_validator():
    pass

@password_validator(length=10, uppercase=2, lowercase=2, special_chars=2)
@username_validator
def register_user(username: str, password: str):
    """
    Функция для регистрации нового пользователя.

    Параметры:
        username (str): Имя пользователя.
        password (str): Пароль пользователя.

    Raises:
        ValueError: Если пароль или юзернейм не соответствует заданным условиям.
    """
    # Запись имени пользователя и пароля в CSV файл
```

```
# Тестирование успешного случая
try:
    register_user("JohnDoe", "Password123!")
    print("Регистрация прошла успешно!")
except ValueError as e:
    print(f"Ошибка: {e}")

# Тестирование неудачного случая по паролю...

# Тестирование неудачного случая по юзернейму...
```

Требования к коду:

- Проект должен быть оформлен согласно стандарту PEP-8.
- Все функции и **переменные** должны быть аннотированы.
- Должны присутствовать док-стринги для всех функций и декораторов.
- Код должен успешно проходить проверку типов с использованием Муру.
- Все должно работать и все проверки в коде присутствовать.
- Опционально: И да прибудет с вами Замыкание 🌟😊