

Класс User

Реализуйте класс `User`, описывающий интернет-пользователя. При создании экземпляра класс должен принимать два аргумента в следующем порядке:

- `name` — имя пользователя. Если `name` не является непустой строкой, состоящей только из букв, должно быть возбуждено исключение `ValueError` с текстом:

```
Некорректное имя
```

- `age` — возраст пользователя. Если `age` не является целым числом, принадлежащим отрезку `[0; 110]`, должно быть возбуждено исключение `ValueError` с текстом:

```
Некорректный возраст
```

Экземпляр класса `User` должен иметь два атрибута:

- `_name` — имя пользователя
- `_age` — возраст пользователя

Класс `User` должен иметь четыре метода экземпляра:

- `get_name()` — метод, возвращающий имя пользователя
- `set_name()` — метод, принимающий в качестве аргумента значение `new_name` и изменяющий имя пользователя на `new_name`. Если `new_name` не является непустой строкой, состоящей только из букв, должно быть возбуждено исключение `ValueError` с текстом:

```
Некорректное имя
```

- `get_age()` — метод, возвращающий возраст пользователя
- `set_age()` — метод, принимающий в качестве аргумента значение `new_age` и изменяющий возраст пользователя на `new_age`. Если `new_age` не является целым числом, принадлежащим отрезку `[0; 110]`, должно быть возбуждено исключение `ValueError` с текстом:

```
Некорректный возраст
```

Примечание 1. Если при создании экземпляра класса `User` имя и возраст одновременно являются некорректными, должно быть возбуждено исключение, связанное с именем.

Sample Input 1:

```
user = User('Гвидо', 67)

print(user.get_name())
print(user.get_age())
```

Sample Output 1:

```
Гвидо
67
```

Sample Input 2:

```
user = User('Гвидо', 67)

user.set_name('Тимур')
user.set_age(30)

print(user.get_name())
print(user.get_age())
```

Sample Output 2:

```
Тимур
30
```