Консольное клиент-серверное приложение для игры в «слова»

Приложение должно быть построено по архитектурной модели *Model View Controller* и может быть как сервером, так и клиентом — это определяется аргументами командной строки. Примеры запуска:

```
> python abc.py
```
запуск в режиме клиента, адрес сервера по-умолчанию localhost

```
> python abc.py 192.168.1.98
```
запуск в режиме клиента, адрес сервера 192.168.1.87

```
> python abc.py server
```
запуск в режиме сервера, адрес localhost

```
> python abc.py server 192.168.1.98
```
запуск в режиме сервера, сетевой адрес 192.168.1.98

Обмен данными между клиентом и сервером должен происходить в формате JSON по сетевому протоколу UDP.

1. Структура

Исходный код приложения должен содержать, как минимум, следующие компоненты (классы):

- **`AbcApplication`** — реализует логику приложения и игры (предоставлен вам в готовом виде в файле **`app.py`**);
- **`AbcModel`** — реализует хранение и проверку использованных слов;
- **`AbcView`** — ведёт диалог с пользователем в командной строке;
- **`AbcServer`** — реализует диалог с другим игроком по сети в режиме сервера;
- **`AbcClient`** — реализует диалог с другим игроком по сети в режиме клиента.

2. Класс AbcModel

Конструктор этого класса можно оставить пустым.
Класс содержит методы, позволяющие:

- проверить, использовалось ли слово ранее (`is_used`);
- добавить слово в список использованных (`add`);

3. Класс AbcView

Конструктор этого класса можно оставить пустым.
Класс содержит методы, позволяющие:

- отобразить любой текст (`show`);

- отобразить слово, названное оппонентом (`show_word`);
- отобразить сообщение о выигрыше (`show_win`);
- отобразить сообщение о проигрыше (`show_loose`);
- отобразить сообщение, что слово не подходит, и оно нолжно начинаться на другую букву (`show_mistake`);
- отобразить сообщение, что такое слово уже использовалось (`show_used`);
- отобразить сообщение, что такого слова не существует (`show_unknown`);
- принять очереднot слово от пользователя (`ask_word`).

Примечание: Метод должен запрашивать слово повторно, если пользователь ничего не ввёл или ввёл больше одного слова.

4. Класс AbcServer

Конструктор класса должен создать серверный UDP сокет и принимат обязательные параметры для этого: имя хоста и номер порта.
Класс содержит методы, позволяющие:

- принять запрос на подключение от клиента (`connect`);
- отправить клиенту информацию о только что сделанном ходе (`send`);
- отправить клиенту информацию о завершении игры (`end`);
- дождаться хода клиента и получить информацию о нём, т.е. строку и столбец (`receive`).

5. Класс AbcClient

Конструктор класса должен создать клиентский сокет для коммуникации с сервером, и принимает обязательные параметры для этого: имя хоста сервера и номер порта.
Класс содержит методы, позволяющие:

- отправить серверу запрос на подключение (`connect`);
- отправить серверу информацию о только что сделанном ходе (`send`);
- отправить серверу информацию о завершении игры (`end`);
- дождаться хода сервера и получить информацию о нём, т.е. строку и столбец (`receive`).

6. Проблема класса AbcApplication

В логике метода **`my_turn`** необходимо добавить проверку того, что слово существует, с помощью предоставленного вам модуля **`wiktionary`**.