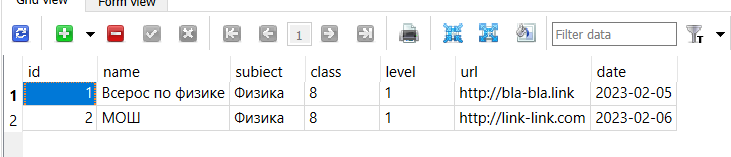
Проект состоит из 4-х модулей

1. Config.py – модуль, содержащий основные настройки бота
2. Db.py – модуль для работы с БД (эту часть надо переделывать на имеющиеся доработки с БД). Текущий вариант был сделан с приложенным sqlite, чтобы упростить отладку и была понятна примерная структура таблиц
3. Keyboards.py – модуль, содержащий клавиатуры бота. Отвечает за выбор пользователя
4. Bot.py – основной модуль. Отслеживает и реагирует на действия пользователя.

Таблицы БД

1. Таблица olymp2 (использовалась для отладки). Содержит поля:

* Id
* Name – наименование
* Subject – предмет
* Class – класс, для которого проводится олимпиада
* Level – уровень олимпиады
* url – ссылка на сайт олимпиады
* date – дата проведения олимпиады

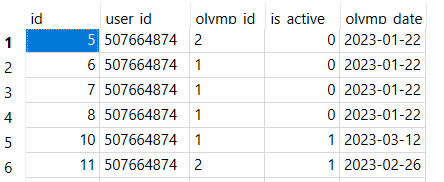


Данная структура кривовата. Мысли по улучшению:

* По уму надо SUBJECT выводить в отдельную таблицу. В таблице olymp2 в поле subject должен быть id записи. Примерные поля таблицы SUBJECT: id (автоинкремент), наименование. Рекомендуется так сделать, чтобы избежать проблем с разным написанием, т.к. Физика, физика и т.п. – это разные строковые значения
* Также, наверное, стоит добавить краткое описание олимпиад

1. Таблица reminder. Содержит поля:

* Id
* User\_id – id пользователя, которому пишет бот
* Olymp\_id – id олимпиады из таблицы olymp2
* Is\_active – активно ли напоминание
* Olymp\_date – дата олимпиады



По таблице remind под вопросом необходимость там поля olymp\_date. По уму данное значение должно браться с олимпиады. С другой стороны, это приведет к лишней связке таблиц и усложнению запросов. Разумнее думаю будет оставить так как есть.

Логика работы бота:

1. Запуск /start
2. Далее идет последовательный выбор с помощью выпадающих клавиатур:

* Класс обучения
* Предмет (выбирается с определенного набора кнопок). Нужно определиться со списком. Все предметы добавлять смысла, наверное, не имеет. Стоит брать основные
* Уровень олимпиады

1. Происходит запрос к БД. В случае если нет ни одной записи, подходящей под критерии, то выходит сообщение, что данные не найдены и нужно сформировать новый запрос. Если данные найдены, то они выводятся списком.
2. При желании можно добавить напоминания по интересующим олимпиадам

Напоминания

Для каждой выведенной записи существует возможность добавления напоминания. Для этого под описанием есть соответствующая кнопка. При нажатии на кнопку добавляется напоминание в БД и выводится сообщение пользователю. У добавленных напоминаний поле is\_active = 1, что означает, что они активны.

На текущий момент нет проверки на уникальность записи в БД по связке ID олимпиады и ID пользователя. Сейчас сделана простая реализация – добавление без проверок.

Обработка напоминаний выполняется ежедневно в заданное переменной REMIND\_TIME время. Для отладки, соответственно, удобно менять данное значение и вносить свои изменения.

Переменная REMIND\_DAYS отвечает за период (кол-во дней), за которое пользователю будет выводиться напоминание.

Если текущая дата >= (Дата олимпиады – REMIND\_DAYS), то по расписанию будет выполнено оповещение для активных напоминаний. После отправки оповещения пользователю поле is\_active = 0.

Используемые библиотеки:

* Aiogram – основная библиотека для работы бота
* Aioschedule – библиотека, отвечающая за планирование