**Пояснительная записка**

**к телеграм-боту “ЕГЭ-справочник/тренажёр”**

**Крюкова Дарья**

**Оренбург**

**2024**

**СОДЕРЖАНИЕ**

[Введение 3](#_Toc165414363)

[Практическая часть 4](#_Toc165414364)

[Создание бота 4](#_Toc165414365)

[Библиотеки: 4](#_Toc165414366)

[Запрос API 5](#_Toc165414367)

[Работа с БД 6](#_Toc165414368)

[Загрузка на сервер 9](#_Toc165414369)

# Введение

Задачей проекта было создание телеграмм бота со справочными материалами по ЕГЭ, командами, которые используют сторонние API для запроса и базы данных для записи данных, введенных пользователем .

Техническое задание:

1. Создать телеграм-бота на aiogram

2. Создать команды, которые отсылают картинки и документы в ответ на сообщения пользователя

3. Создать команды, которые получают на ввод слов(-а), с которыми создается API-запрос, результат отсылается ботом в ответ на сообщение пользователя.

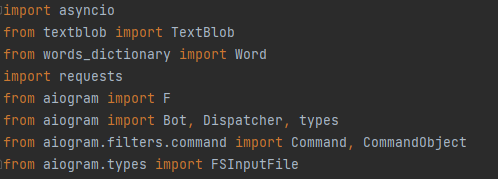
4. Создать команду, которая получают информацию от пользователя и записывает ее в БД.

5. Создание ORM-модели для БД

# Практическая часть

## Создание бота

### Библиотеки:



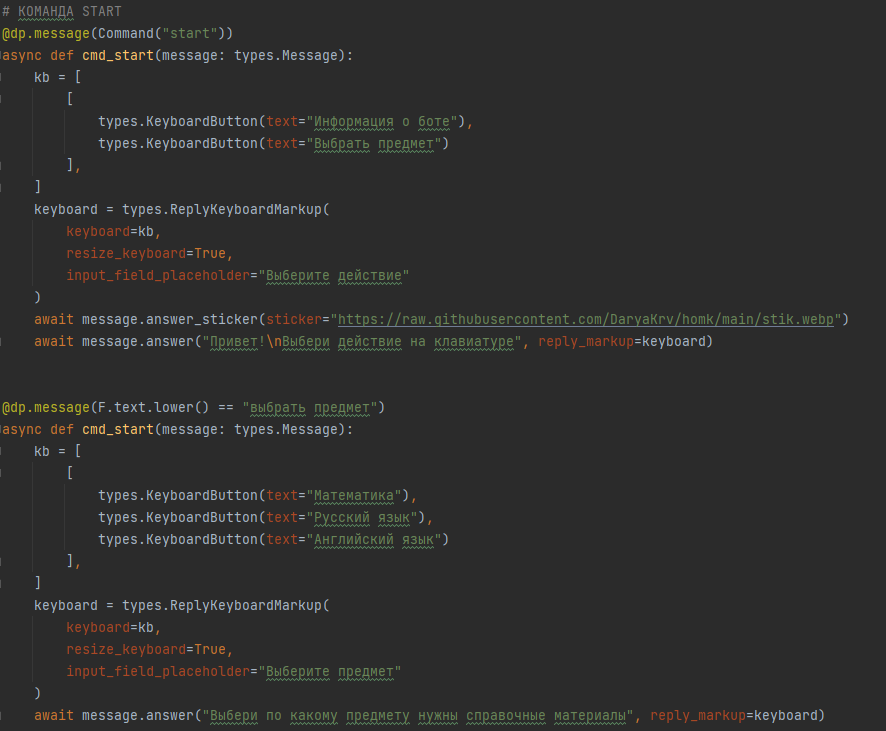
Asyncio – для асинхронного выполнения запросов пользователя

Aiogram – библиотека для создания и работы с тг-ботом

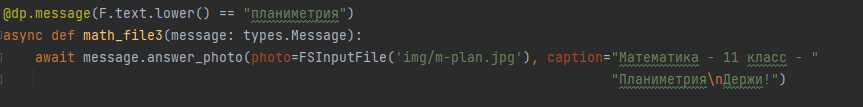
Textblob – просто интересная библиотека, связанная с правописанием

Содержание main.py:

Задача-команда /start – стикер, текст, клавиатура с выбором  
Через метод F(из aiogram) можно работать с текстом, отправленным в сообщении



Работа с кнопками, клавиатурой описывается однотипно, так что разберу отдельные новые части:

Отправление существующего в папке img файла происходит через FSInputFile, текст для отправки файла обрабатывается методом F. 

### Запрос API

Запрос – это обычный перевод слова, но просто используется по-разному

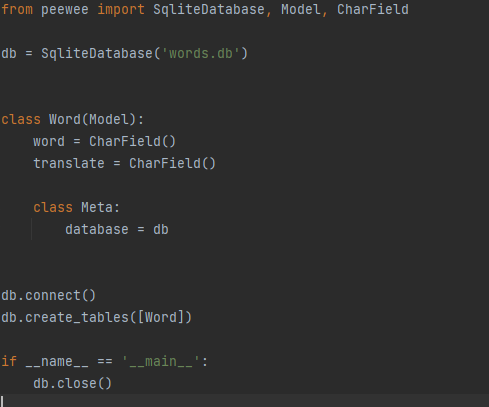
На примере команды /say. Передается слово, слово идет в запрос переводчику, возвращается перевод, при желании можно поменять настройки переводчика.



### Работа с БД

Задача – получить слово и записать его с переводом в “словарь” (БД – words.db) для последующего вывода.

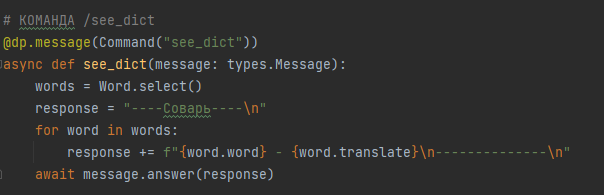
ORM-модель простая.



В нее передается, во-первых, само слово, потом его перевод, полученный через запрос, описанный выше. Пример:



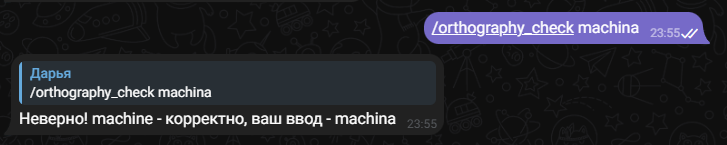
Вывод словаря:



Работа с TextBlob:

Библиотека вписывалась в тему проекта, а API словаря я не нашла, так что проверку орфографии делала именно через нее. Библиотека проверяет правильность написанного, путем подбора ближайшего похожего слова. Пример:





## Загрузка на сервер

Загрузку на сервер делала через replit.com и сервис UpTimerRobot.

На replit.com загрузила все файлы бота, создала дополнительный файл background.py с Flask-сервером, который работает одновременно с ботом, однако он засыпает если бот не активен несколько минут. Для решения пользовалась UpTimerRobot, сервис каждые 5 минут обращался к Flask-серверу, чтобы бот и сервер не спали. Когда проверяла в течении 30 минут все работало, потом уже начинает валиться. Посмотрела, почему так происходит. Сам сервис replit.com закрыл возможность такой реализации бота. Сами файлы бота в порядке, при запуске бот полностью выполняет поставленные задачи.