# Структура и логика проекта

Этот документ описывает структуру и логику проекта, реализованного в предоставленном скрипте. Скрипт предназначен для работы с Telegram-ботом, который предоставляет пользователю информацию о погоде и функции управления списком избранных городов.

## 1. Структура проекта

Проект состоит из следующих основных модулей:

### 1.1 Модуль работы с базой данных

Функция `init\_db` отвечает за создание и инициализацию базы данных SQLite. Она создает две таблицы:

- `weather`: хранит данные о погоде (город, температура, влажность, описание, дата).

- `cities`: хранит список избранных городов пользователя.

### 1.2 Модуль взаимодействия с OpenWeatherMap API

Функция `get\_weather\_data` отправляет HTTP-запрос к API OpenWeatherMap для получения информации о погоде. Данные сохраняются в таблице `weather`.

### 1.3 Модуль управления избранными городами

Функции `save\_city` и `get\_favorites` обеспечивают добавление городов в избранное и получение списка избранных городов из таблицы `cities`.

### 1.4 Модуль Telegram-бота

Telegram-бот управляется библиотекой `telegram`. Основные компоненты:

- Команда `/start`: Приветствует пользователя и предлагает меню.

- Обработчик сообщений: Реализует логику обработки пользовательского ввода и взаимодействия с другими модулями.

## 2. Логика приложения

Приложение работает следующим образом:

### 2.1 Запуск приложения

Скрипт запускается, создается база данных, и Telegram-бот начинает работу.

### 2.2 Обработка команд

Пользователь вводит команды или текст через интерфейс Telegram. В зависимости от ввода:

- Если пользователь выбрал "Посмотреть погоду", бот запрашивает название города, получает данные о погоде и отправляет их пользователю.

- Если пользователь выбрал "Добавить в избранное", бот сохраняет город в базу данных.

- Если пользователь выбрал "Показать избранное", бот отображает список сохраненных городов и позволяет получить данные о погоде для любого из них.

### 2.3 Взаимодействие с базой данных и API

При каждой операции, связанной с погодой или избранными городами, скрипт взаимодействует с базой данных SQLite для сохранения и получения данных, а также с OpenWeatherMap API для обновления информации о погоде.