Пояснительная записка к проекту:

**Telegram-бот для управления задачами (To-Do List)**

**Авторы**: Маркин Даниил, Кузин Сергей, Иванова Альмира

### 1. Введение

Проект представляет собой **многофункционального Telegram-бота** для управления задачами (To-Do List) с использованием **SQLite** в качестве базы данных. Бот позволяет пользователям создавать, редактировать, удалять задачи, устанавливать напоминания, добавлять описания к задачам и отслеживать их статусы.

**Цель проекта:**Разработать удобный инструмент для планирования задач с расширенным функционалом, включая хранение данных в SQLite, добавление описаний задач и систему напоминаний.

**Задачи:**

* Реализовать интерактивное меню с кнопками для управления задачами.
* Настроить систему хранения данных в **SQLite** с поддержкой нескольких пользователей и общих списков задач.
* Добавить функционал **напоминаний** с асинхронной проверкой времени.
* Внедрить возможность добавления **описаний** к задачам.
* Обеспечить обработку ошибок и обратную связь для пользователя.
* Оптимизировать производительность бота для стабильной работы.

### 2. Техническая реализация

#### 2.1. Используемые технологии:

* **Язык программирования:** Python 3.10+
* **Библиотеки:**
  + aiogram — для работы с Telegram API и асинхронного взаимодействия.
  + sqlite3 — хранение данных в SQLite.
  + datetime и asyncio — обработка времени и асинхронных задач.
  + uuid — генерация уникальных идентификаторов для списков задач.
* **Хранение данных:** Локальная база данных SQLite (ToDo.db).

#### 2.2. Структура базы данных:

* **Таблица users:**
  + ***user\_id*** (TEXT, PRIMARY KEY) — ID пользователя в Telegram.
  + ***list\_id*** (TEXT) — ID списка задач пользователя.
* **Таблица task\_lists:**
  + ***list\_id*** (TEXT, PRIMARY KEY) — уникальный идентификатор списка задач.
* **Таблица tasks:**
  + ***task\_id*** (INTEGER, PRIMARY KEY) — уникальный идентификатор задачи.
  + ***list\_id*** (TEXT) — ID списка задач.
  + ***task*** (TEXT) — название задачи.
  + ***description*** (TEXT) — описание задачи.
  + ***status*** (TEXT) — статус задачи ("Не начата", "В процессе", "Выполнена").
  + ***created\_at*** (TEXT) — дата создания задачи.
  + ***reminder\_time*** (TEXT) — дата и время напоминания.
  + ***reminded*** (INTEGER) — флаг отправки напоминания.
  + ***completed\_at*** (TEXT) — дата завершения задачи.
* **Таблица list\_members:**
  + ***list\_id*** (TEXT) — ID списка задач.
  + ***user\_id*** (TEXT) — ID пользователя.

#### 2.3. Основные компоненты:

1. **Система состояний (FSM):**
   * Реализована для поэтапного взаимодействия (например, добавление задачи → ввод названия → описание → установка напоминания).
2. **Меню бота:**
   * Основные кнопки:
     + ➕ Добавить — создание новой задачи.
     + 📋 Список — просмотр активных задач.
     + ✅ Выполненные — просмотр завершенных задач.
3. **Работа с задачами:**
   * Добавление, редактирование (название, описание), удаление задач.
   * Изменение статуса ("Не начата", "В процессе", "Выполнена").
   * Установка напоминаний в формате DD-MM-YYYY HH:MM.
4. **Обратная связь:**
   * Уведомления об успешных действиях и ошибках (например, некорректный формат даты).

### 3. Особенности реализации

* **Многопользовательская поддержка:**
  + Каждый пользователь имеет свой список задач, но может быть участником нескольких списков.
* **Асинхронность:**
  + Проверка напоминаний выполняется в фоновом режиме без блокировки основного потока.
* **Гибкость:**
  + Возможность редактирования названия и описания задачи, установки и отмены напоминаний.
* **Обработка ошибок:**
  + Защита от некорректного ввода (например, запрет использования текста кнопок в названии задачи).

### 4. Сценарии использования

1. **Добавление задачи:**
   * Пользователь нажимает ➕ Добавить → вводит название → описание (опционально) → устанавливает напоминание (опционально).
2. **Просмотр задач:**
   * Активные задачи отображаются с статусами, описаниями и датами напоминаний.
   * Завершенные задачи выводятся отдельным списком с датой выполнения.
3. **Управление задачами:**
   * Редактирование, изменение статуса, удаление через интерактивные кнопки.
4. **Напоминания:**
   * Бот отправляет уведомление в указанное время и отмечает напоминание как выполненное.

### 5. Этапы разработки

1. **Проектирование:**
   * Создание схемы базы данных SQLite с поддержкой многопользовательского режима.
   * Дизайн FSM-состояний для взаимодействия с пользователем.
2. **Реализация:**
   * Написание ядра бота (обработчики команд, меню, FSM).
   * Интеграция SQLite и функционала напоминаний.
3. **Тестирование:**
   * Проверка корректности сохранения данных.
   * Тестирование напоминаний и обработки ошибок.
4. **Деплой:**
   * Запуск бота на сервере.

### 6. Заключение

Telegram-бот для управления задачами успешно реализован как удобное и надежное решение для повышения личной продуктивности. Проект сочетает простой интерфейс с мощным функционалом, включая гибкие напоминания и детальное описание задач. Использование SQLite обеспечивает стабильное хранение данных, а модульная архитектура позволяет легко расширять функционал. Бот готов к использованию и имеет потенциал для дальнейшего развития, например, через добавление категорий задач или интеграцию с календарями.