

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
Факультет прикладной математики – процессов управления

ОТЧЕТ
по лабораторной работе №5
по дисциплине «Функциональное программирование»
на тему «Создание бота в Телеграмм»

Студент гр. 22.Б15

Суворов Н.В.

Преподаватель

Киямов Ж.У.

Санкт-Петербург

2023 г.

Содержание

1. Цель работы	2
2. Задача.....	2
3. Теоретическая часть	2
4. Алгоритм метода	3
5. Описание программы	3
6. Рекомендации пользователю	4
7. Рекомендации программисту	4
8. Контрольный пример	5
9. Заключение	5
10. Литература	5

1. Цель работы

Создать чат-бота в телеграмм.

2. Задача

- Разработать архитектуру чат-бота
- Интегрировать стороннее API

3. Теоретическая часть

1. API (Интерфейс прикладного программирования):

API - это набор инструкций и протоколов, которые определяют, как различные компоненты программного обеспечения должны взаимодействовать друг с другом. В случае Telegram, Telegram API предоставляет разработчикам доступ к различным функциям мессенджера, таким как отправка сообщений, управление ботами и т.д.

2. Токен (API-ключ):

Токен - это уникальная строка символов, предоставляемая сервисом, в данном случае Telegram, для идентификации вашего бота при обращении к их API. Это своего рода пароль, который позволяет вашему коду взаимодействовать с Telegram API от имени вашего бота.

3. Бот в Telegram:

Бот в Telegram - это автоматизированный аккаунт, способный взаимодействовать с пользователями через сообщения. Он может выполнять различные действия по запросу пользователя или автоматически на основе запрограммированной логики.

4. Webhooks и Polling:

Webhooks и Polling - два способа, с помощью которых бот может получать обновления от Telegram API. Polling представляет собой циклический опрос сервера Telegram на предмет новых обновлений, тогда как Webhooks позволяют Telegram отправлять обновления вашему серверу в реальном времени.

5. Python-Telegram-Bot:

python-telegram-bot - это библиотека на Python, которая упрощает создание чат-ботов для Telegram. Она предоставляет удобные классы и методы для взаимодействия с Telegram API, обработки сообщений и управления

4. Алгоритм метода

Настройка бота и подключение к API Telegram:

Используется telebot.TeleBot для создания бота с помощью токена, полученного от BotFather при регистрации бота.

Работа с базой данных SQLite:

Программа использует SQLite для хранения информации о задачах и напоминаниях пользователей.

Обработчики сообщений от пользователя:

- /start и /help: Предоставляют пользователю информацию о боте.
- /addtask: Запрашивает пользователя добавить задачу и дедлайн.
- /tasks: Отправляет пользователю список его задач и дедлайнов.
- /news: Отправляет пользователю последние новости о Tesla.
- /remind: Запрашивает у пользователя время для напоминания о задаче.

Бесконечный цикл для получения сообщений от пользователей:

Бот ожидает новые сообщения от пользователей и реагирует на них в соответствии с написанной логикой обработчиков.

5. Описание программы

В программе не используются специальные классы. Описание функций приведено в таблице 5.2.

Таблица 5.1. Описание классов

Имя класса	Описание
Нет	Нет

Таблица 5.1. Описание функций

Имя	Описание
add_task_to_db	Добавляет задачу в БД
get_user_tasks	Получает задачу из БД
add_reminder_to_db	Добавляет напоминание в БД
get_user_reminders	Получает напоминание из БД
delete_user_reminders	Удаляет напоминание из БД
send_welcome	Обрабатывает начальные сообщения
echo_all	Обрабатывает все сообщения

Код программы выложен на github:

https://github.com/AlexShinalov/functional_programming/tree/main/laba%205

6. Рекомендации пользователю

Войдите в телеграмм и найдите бота. Напишите /help для получения справки о командах.

7. Рекомендации программисту

Для запуска программы необходима 64-битная операционная система Windows и Python версии не ниже 3.1. Для корректной работы программы рекомендуется использовать IDE PyCharm версии 2023.21 и pip install версии 23.1.0. Для корректной работы необходимо установить библиотеки: **sqlite3**, **requests**, **telebot**, **time**. Их установку рекомендуется производить через **pip install**. Пример **pip install requests**

8. Контрольный пример

В данном разделе представлен контрольный пример, демонстрирующий способность бота выполнять отправку сообщений.

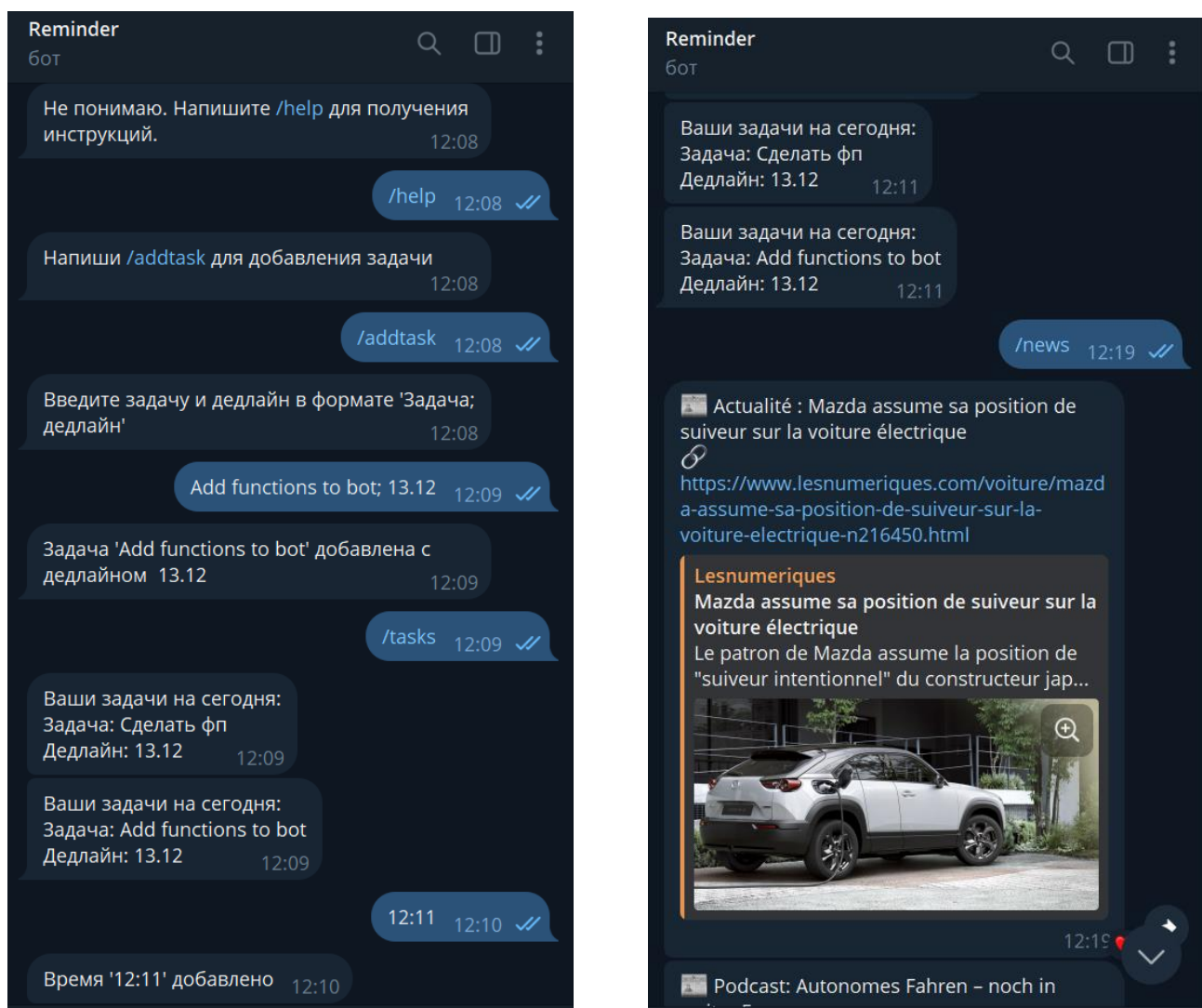


Рисунок 8.4–8.5. Пример работы бота

9. Заключение

В рамках представленной работы был разработан и успешно реализован бот для обработки и отправки сообщений. Данный бот может создавать напоминания для пользователей и отправлять их.

10. Литература

<https://habr.com/ru/articles/442800/>

<https://newsapi.org/>

<https://python-telegram-bot.org/>