

**муниципальное бюджетное общеобразовательное учреждение города
Ростова-на-Дону "Школа № 6 имени Героя Советского Союза Самохина
Н.Е."**

ИНДИВИДУАЛЬНЫЙ ПРОЕКТ

на тему:

"Разработка Telegram бота"

Выполнил учащийся 10 класса:

Шарохин Дмитрий Вячеславович

Научный руководитель:

Басков Николай Николаевич

Допуск к защите: _____

г.Ростов-на-Дону 2024 г

Оглавление

Введение	
Глава 1.....	
1.1 Что такое Telegram и его возможности.	
1.2 Основы Telegram-ботов.....	
1.3 Языки программирования для разработки ботов.	
1.4 Библиотеки для разработки Telegram-бота.....	
1.5 Принципы работы с API погоды.	
1.6 Источники информации о погоде.....	
Глава 2.....	
2.1 Изучение языка программирования.	
2.2 Среда разработки.....	
2.3 Создание токена в Telegram и OpenWeatherMap	
2.4 Скрипт бота	
2.5 Запуск бота на сервере.	
2.6 Тестирование бота.	
Заключение.	
Список использованной литературы.....	
Приложение 1	
Приложение 2	
Приложение 3	

Введение

В современном информационном обществе, где технологии становятся неотъемлемой частью нашей повседневной жизни, разработка Telegram-ботов приобретает все большую актуальность. Особенно важной становится эта тема, когда речь идет о создании бота для прогноза погоды.

Актуальность данного проекта проявляется в современных потребностях пользователей в быстром и удобном доступе к информации о погоде. В сфере информационных технологий растет интерес к созданию инновационных инструментов, способных решать повседневные задачи более эффективно и комфортно. Разработка телеграм-бота для прогноза погоды отвечает этим требованиям, предлагая новый способ получения погодной информации, который удовлетворяет потребности пользователей в простоте использования и оперативности получения данных.

Проблема, перед которой стоит разработка телеграм-бота для прогноза погоды, заключается в неудобстве и ограниченности традиционных способов получения информации о погоде. Пользователи часто сталкиваются с необходимостью переключения между различными приложениями или веб-сайтами для получения прогноза, что может быть времязатратным и неэффективным. Также люди часто сталкиваются с проблемой выбором подходящей одежды под погоду.

Целью данного проекта является разработка Telegram бота, который предоставляет пользователям актуальные данные о погоде и рекомендации по выбору подходящей одежды на основе текущих климатических условий.

Задачи:

1. Изучение основ работы с телеграм-ботами и их интеграции с внешними сервисами.
2. Изучение языков программирования и их библиотеки
3. Анализ и выбор подходящего API для получения погодной информации.
4. Написание кода для обработки запросов пользователей и получения погоды.
5. Запуск бота на отдаленном сервере
6. Тестирование бота.

Гипотеза. Я считаю, что Telegram является очень удобной платформой для реализации своих ботов.

Практическая значимость данного проекта заключается в создании инструмента, который повысит уровень удобства и доступности получения информации о погоде для пользователей мессенджера Telegram.

В ходе решения поставленных задач были использованы следующие **методы исследования**: анализ, обобщение, описание, моделирование, программирование.

Объектом исследования является возможности интернет-ботов.

Предметом исследования является Telegram-бот

Глава 1.

1.1 Что такое Telegram и его возможности.

Telegram (Telegram) - это мессенджер, разработанный 14 августа 2013 года Павлом Дуровым, основателем социальной сети ВКонтакте и задумывался как мессенджер для конфиденциального общения. Он представляет собой кроссплатформенное приложение, которое позволяет пользователям обмениваться сообщениями, файлами, изображениями и другими медиа-файлами через интернет. С 2013 по 2014 год аудитория Telegram выросла до 100 тысяч активных пользователей в месяц, а в феврале 2016 года Павел Дуров заявил, что мессенджером пользуются более 100 миллионов человек. В 2024 году Telegram посещают 800 миллионов человек в месяц. По количеству аудитории Telegram входит в пятерку самых популярных мессенджеров в мире. При сравнении мессенджеров, таких как Telegram, WhatsApp, и Viber, можно обнаружить много общих черт. В то же время, Telegram выделяется среди них благодаря :

- **Защиты конфиденциальности:** Telegram предлагает мощное шифрование для защиты переписки. Это включает в себя функцию "секретных чатов", где сообщения могут быть удалены после определенного времени или признаны самоуничтожающимися.

- **Кроссплатформенности:** Приложение доступно для широкого спектра платформ, включая iOS, Android, Windows, macOS и Linux. Это обеспечивает пользователей возможностью общаться между различными устройствами и операционными системами.

- **Богатым функциональным возможностям:** В Telegram доступно множество функций, включая групповые чаты, каналы, возможность создания ботов, голосовые и видеозвонки и другие.

- **Telegram открытая платформа для разработчиков:** он предоставляет API (интерфейс программирования приложений), позволяющее разработчикам создавать собственные боты и интегрировать их с мессенджером.

Павел и Николай Дуровы предоставляют поддержку для Telegram. Павел обеспечивает финансовую и идеологическую поддержку, в то время как Николай занимается технической стороной проекта. Для Telegram был специально разработан уникальный протокол, который является открытым, безопасным и способным эффективно взаимодействовать с несколькими дата-центрами. В результате Telegram стал самым быстрым и безопасным мессенджером в мире.

Понимание основ Telegram важно для разработки Telegram-ботов и понимания их функциональности и возможностей в контексте данного проекта.

1.2 Основы Telegram-ботов.

Telegram-боты - это приложения, разработанные для работы в мессенджере Telegram, которые автоматизируют различные задачи и предоставляют разнообразный функционал для пользователей. Они представляют собой программные агенты, способные обрабатывать сообщения от пользователей, выполнять команды и взаимодействовать с другими сервисами и приложениями через API. Функциональные возможности: Telegram-боты могут выполнять множество функций, включая отправку сообщений, ответы на запросы пользователей, выполнение команд, работу с базами данных, автоматизацию процессов и многое другое. Интерфейс взаимодействия: Пользователи могут взаимодействовать с ботами через обычные чаты в Telegram, отправляя им текстовые сообщения или команды. Боты могут отвечать на эти сообщения автоматически или с помощью специально разработанных сценариев. Использование в различных областях: Telegram-боты широко применяются в различных областях, таких как образование, развлечения, бизнес, медицина, финансы и другие. Они могут быть использованы для создания информационных сервисов, бронирования услуг, организации конкурсов и опросов, обработки заказов и многое другое.

Разработка и использование Telegram-ботов открывает перед разработчиками и пользователями много новых возможностей для автоматизации задач, улучшения пользовательского опыта и создания новых сервисов и приложений.

1.3 Языки программирования для разработки ботов.

Для создания ботов для Телеграм существует несколько языков программирования, которые зарекомендовали себя как надежные и эффективные инструменты. Одни из самых популярных языков — это Python, JavaScript, Java, Ruby, C#

Каждый язык программирования имеет свои преимущества и давайте рассмотрим них:

Python:

Преимущества:

- Простота и лаконичность кода, что делает его привлекательным для начинающих и опытных разработчиков.
- Большое количество библиотек, таких как `python-telegram-bot`, `aiogram`, упрощающих разработку и интеграцию с Telegram API.

- Большое сообщество разработчиков и обширная документация, что облегчает получение помощи и решение проблем.

Недостатки:

- Медленная скорость выполнения по сравнению с некоторыми другими языками.
- Некоторые ограничения в работе с многопоточностью, что может затруднить масштабирование.

JavaScript:

Преимущества:

- Широкое распространение и популярность, особенно на стороне клиента веб-приложений.
- Асинхронная природа языка, что делает его эффективным для обработки большого количества запросов.
- Богатая экосистема фреймворков и библиотек для разработки как серверной, так и клиентской части приложений.

Недостатки:

- Ограниченные возможности в работе с многопоточностью.
- Некоторые разработчики считают сложным согласование клиентской и серверной части приложений.

Java:

Преимущества:

- Кроссплатформенность и высокая производительность, что делает его привлекательным для разработки крупных и масштабируемых приложений.
- Богатая стандартная библиотека и множество фреймворков для разработки различных видов приложений.
- Поддержка многопоточности из коробки, что важно для обработки множества одновременных запросов.

Недостатки:

- Больше кода по сравнению с Python или JavaScript, что может усложнить разработку.
- Меньшая гибкость и лаконичность синтаксиса по сравнению с другими языками.

Ruby:

Преимущества:

- Простота и красивый синтаксис, что делает его приятным для разработки.
- Богатая экосистема фреймворков, таких как Ruby on Rails, упрощающих создание веб-приложений и API.
- Мощные средства метапрограммирования, позволяющие создавать выразительный и гибкий код.

Недостатки:

- Меньшая производительность по сравнению с некоторыми другими языками.
- Меньшее сообщество разработчиков и меньшее количество библиотек и фреймворков, чем у более популярных языков.

C#:

Преимущества:

- Высокая производительность и эффективное управление ресурсами, что делает его подходящим для разработки крупных приложений.
- Интеграция с платформой .NET и широкая поддержка инструментов и технологий Microsoft.
- Широкий набор библиотек и фреймворков для разработки приложений различного назначения.

Недостатки:

- Ограниченная кроссплатформенность, поскольку C# чаще всего используется в экосистеме Microsoft.
- Меньшая популярность и меньшее количество разработчиков по сравнению с более распространенными языками.

Каждый из этих языков имеет свои сильные и слабые стороны, и выбор зависит от конкретных требований проекта, предпочтений разработчика и контекста использования бота.

В итоге бота можно написать на любом языке программирования, но самым популярным считается Python, потому что в его экосистеме много открытых библиотек и готовых решений. К тому же он легок в изучении. Python позволяет создавать ботов для Telegram с минимальными усилиями, что делает его идеальным выбором для начинающих разработчиков.

1.4 Библиотеки для разработки Telegram-бота.

В данном разделе рассмотрены основные библиотеки на языке Python, которые широко используются для создания Telegram ботов. Ниже представлен краткий обзор каждой из них:

Начнем с `python-telegram-bot`. Эта библиотека имеет богатую историю и широкое сообщество пользователей.

`python-telegram-bot` - это библиотека Python для создания телеграм-ботов, разрабатываемая сообществом. Она предоставляет обширный набор функций и простой интерфейс для взаимодействия с Telegram Bot API. Эта библиотека активно поддерживается и обновляется сообществом разработчиков.

Преимущества:

- **Активное сообщество:** `python-telegram-bot` поддерживается активным сообществом разработчиков, что гарантирует обновления и поддержку.
- **Документация:** Библиотека имеет хорошо составленную документацию, которая помогает разработчикам быстро начать работу.
- **Поддержка асинхронности:** Поддержка `async/await` позволяет эффективно управлять асинхронными операциями.
- **Обработка медиа:** `python-telegram-bot` предоставляет мощные средства для обработки изображений, аудио и других медиа-файлов.
- **Интеграция с внешними сервисами:** Библиотека упрощает интеграцию бота с другими сервисами и базами данных.

Недостатки:

- **Может показаться избыточной для простых ботов:** Для простых ботов с ограниченным функционалом, `python-telegram-bot` может показаться избыточной.

`Aiogram` - это высокопроизводительная и интуитивно понятная библиотека для создания телеграмм-ботов на Python. Она была разработана с учетом активного участия сообщества.

`aiogram` предоставляет множество инструментов и функций, что делает ее идеальным выбором для создания ботов с расширенной функциональностью. Она активно поддерживается и обновляется

сообществом разработчиков, и это гарантирует актуальность и надежность библиотеки.

Преимущества:

- Высокая производительность: aiogram разработана с учетом производительности, что делает ее идеальным выбором для проектов с высокой нагрузкой.
- Асинхронное программирование: Поддержка асинхронности позволяет эффективно управлять большим количеством одновременных запросов.
- Богатый функционал: Библиотека предоставляет богатый набор функций для обработки различных типов сообщений и медиа-файлов.
- Активное сообщество: aiogram имеет активное сообщество разработчиков, что обеспечивает быструю поддержку и обновления.
- Интеграция: aiogram легко интегрируется с внешними сервисами и ресурсами.

Недостатки:

- Может потребовать более глубокого понимания асинхронного программирования: Для некоторых разработчиков, особенно начинающих, использование асинхронности может потребовать изучения.
- Более детальная настройка: Иногда создание сложной функциональности может потребовать более глубокой настройки библиотеки.

Telebot - это популярная библиотека Python для создания телеграм-ботов. Она была разработана для простоты и удобства использования, и быстро завоевала популярность среди разработчиков. Эта библиотека предоставляет простой и интуитивно понятный способ создания ботов, что делает ее привлекательным выбором как для начинающих, так и для опытных разработчиков.

Преимущества:

- Простота в использовании: Telebot предоставляет простой и интуитивно понятный способ создания ботов, что упрощает начало работы.
- Обширные примеры кода: Библиотека сопровождается множеством примеров кода и хорошо структурированной документацией.
- Модульность и расширяемость: Telebot позволяет создавать модульные боты и легко расширять их функциональность.
- Быстрый старт: Создание базового бота с Telebot может занять всего несколько минут.

Недостатки:

- Ограниченный функционал: Telebot может оказаться несостоятельным для проектов, требующих сложной функциональности.

- **Меньше активное сообщество:** По сравнению с некоторыми другими библиотеками, сообщество вокруг Telebot может быть менее активным.

Благодаря своей простоте и удобству использования, Telebot является отличным выбором как для начинающих, так и для опытных разработчиков, которые хотят быстро и легко создать Telegram бота.

1.5 Принципы работы с API погоды.

Регистрация и получение ключа API: Перед началом работы с API необходимо зарегистрироваться на соответствующем сервисе и получить ключ API, который будет использоваться для запросов погодных данных.

- **Формирование запроса:** Для получения погодных данных необходимо сформировать HTTP-запрос к API, указав необходимые параметры, такие как город или координаты, дата и время, тип данных (текущая погода, прогноз и т. д.).

- **Отправка запроса:** Сформированный запрос отправляется на сервер API погоды с использованием HTTP-методов, таких как GET или POST.

- **Обработка ответа:** Полученный от сервера ответ обычно представляет собой структурированные данные в формате JSON или XML, содержащие информацию о погоде для запрошенного местоположения и времени.

- **Использование полученных данных:** Полученные погодные данные могут быть обработаны и использованы в приложении или сервисе для отображения текущей погоды, прогноза, аналитики и т. д.

- **Обработка ошибок:** При работе с API необходимо учитывать возможность возникновения ошибок, таких как неправильный формат запроса, неверный ключ API, временная недоступность сервера и т. д. Обработка таких ошибок позволяет обеспечить стабильную работу приложения.

1.6 Источники информации о погоде.

API на Python позволяет взаимодействовать с различными сервисами и ресурсами через код. Он используется для интеграции приложений, автоматизации задач, разработки приложений и получения данных. Это упрощает работу с внешними сервисами, такими как базы данных, веб-сервисы и внешние приложения. В целом, API на Python играет важную роль в создании распределенных систем и приложений, обеспечивая эффективное взаимодействие с внешними ресурсами.

Несколько популярных источников данных о погоде с открытым API:

- OpenWeatherMap API: OpenWeatherMap предоставляет доступ к погодным данным для различных географических местоположений. Их API позволяет получить текущие условия, прогнозы, данные о ветре, осадках и другие метеорологические параметры.
- Weather.com API: Weather.com (The Weather Channel) предоставляет широкий спектр погодных данных и прогнозов. Их API позволяет получить текущие условия, прогнозы, астрономические данные и многое другое.
- AccuWeather API: AccuWeather предлагает точные прогнозы погоды и данные о погоде. Их API позволяет получить доступ к текущим условиям, прогнозам, уровням осадков, температуре, влажности и другим параметрам.
- Weather Underground API: Weather Underground предоставляет данные о погоде от тысяч метеостанций по всему миру. Их API позволяет получить доступ к текущим условиям, прогнозам, историческим данным и многое другое.
- The Weather Company API (IBM Weather): The Weather Company, подразделение IBM, предоставляет доступ к погодным данным, прогнозам, глобальным картам погоды и другим метеорологическим данным через свое API.

OpenWeatherMap считается одним из лучших сервисов для получения погоды благодаря широкому охвату географии, большому объему данных, разнообразию источников информации, API для разработчиков, доступу к геопространственным данным, а также возможности выбора между бесплатной и платной версией.

Я подробно рассмотрел основы Telegram-ботов, изучив их суть и принципы работы. Я углубился в понимание того, как боты взаимодействуют с пользователями через мессенджер Telegram, а также рассмотрели популярные языки программирования, библиотеки и источники информации о погоде.

Глава 2.

2.1 Изучение языка программирования.

Для начала нужно изучить язык программирования для разработки бота. Но программировать на python я уже начал давно . Я его изучал при помощи различных видеороликов на ютуб , на сайте metanit.com и , когда ходил на школьные курсы. Он действительно оказался самым простым языком программирования , которым я пользовался.Он напоминает обычный английский язык, что делает код более понятным и легко читаемым.А также Python обычно требует меньше кода для достижения тех же целей, чем многие другие языки программирования. Это позволяет программистам быстрее писать код и упрощает его понимание.

Для работы с Telegram я выбрал библиотеку под названием Telebot, так как это простой и понятный фреймворк, который подходит для начинающих..Для моей задачи он идеально подойдет.

2.2 Среда разработки

Среду разработки для программирования я использовал Visual Studio Code (VS Code) - бесплатный и открытый текстовый редактор с поддержкой множества языков программирования, обладающий мощными функциями, такими как подсветка синтаксиса, автодополнение и интеграция с Git. Он расширяем и кроссплатформенн, что делает его удобным инструментом для разработки на различных операционных системах. VS Code также имеет обширную систему расширений, позволяющую настраивать его под индивидуальные потребности разработчика.

2.3 Создание токена в Telegram и OpenWeatherMap

Определившись с платформой разработки, необходимо зарегистрировать бота. Для этого, как ни странно, нужен другой бот. BotFather — официальный сервис для настройки и управления созданными телеграм-ботами. Он не занимается конкретными функциями и командами — только косметическими изменениями вроде названия, описания и аватарки.Но главное — он выдает уникальный токен, который необходим для авторизации бота и работы с Bot API. Любой, кто получит токен, сможет управлять ботом. Как я получил токен в BotFather:

1. Отправил в чат с BotFather команду /newbot.
2. Ввел название бота — в этой категории особых ограничений нет.
3. Ввел юзернейм бота — его техническое имя, которое будет отображаться в адресной строке. К нему уже больше требований: юзернейм должен быть уникальным, написан на латинице и обязательно заканчиваться на bot. Так «Telegram» защищается от злоумышленников, которые могут выдавать ботов за реальных людей.

4. Готово. BotFather пришлет токен бота — его можно использовать для настройки в сторонних сервисах.

Я создал бота с названием “Погодный друг”. Дал ему айди @Weather_school_project_bot . Также определил ему аватарку и описания при помощи команд /setuserpic и /setabouttext соответственно (Рис.1). Добавил собственные команды как /start (Приветственную команду) и /w (Чтобы узнать погоду в указанном городе) , далее я реализую их.

Ресурс для определения погодных условий:

Для своего проекта я выбрал сервис OpenWetherMap ,чтобы получать данные о погоде. Я провёл не долгую регистрацию ,зашёл в свой профиль и далее в раздел MyAPI keys , и создал индивидуальный токен, с помощью которого буду получать информацию. Также ознакомился с дополнительными командами.

2.4 Скрипт бота

Приступлю к написанию самого кода. Для начала создаём папку проекта Project_for_school. Далее помещаем туда 2 файла: config.py- для хранения токенов, ключей и main.py - основной файл (Рис 3). Заходим в config.py и создаём две переменные под названием tg_tokken - для токена только что созданного Telegram бота и openweather_apikey - для ключа с сервиса OpenWeatherMap (Рис 4).

Далее заходим в основной файл main.py, скачиваем и импортируем нужные библиотеки : telebot (Для управления ботом) , requests (Для парсирования данных о погоде) и datetime (Для обозначения времени) , а также импортируем с config.py ключи доступа (Рис 5). После создаём объект бота и пишем обработчик на команду /start при помощи декоратора (@bot.message_handler(commands=['start'])) (Рис.6) . На эту команду он будет отвечать "Привет! Я бот погоды. Для получения информации о погоде введите /w и название города.Например /w Москва." . Затем пишем обработчик текстовых сообщений при вводе “/w и названия города” (Рис.7). Обработчик будет принимать и сохранять название города , введенное пользователем, и далее при помощи функции requests.get() отправляет запрос на OpenWeatherMap (Рис.8). Разбираю полученный JSON-ответ и извлекаю необходимую информацию о погоде, такую как температура, ощущаемая температура , влажность воздуха, скорость ветра, давление , погодные условия , время захода солнца , заката солнца, продолжительности дня.Всё это сохраняю в отдельные переменные (Рис.9).

Дополнительно в зависимости от температуры воздуха пишу вывод собственных рекомендаций по выбору одежды (Рис.10).

Например если температура выше 30 градусов цельсия то бот пишет:

- Носите очень легкую и дышащую одежду, из льна, хлопка, тонких вискозных тканей, натурального шелка.

- Одевайтесь в светлые цвета, отражающие солнечные лучи.

- Обязательно используйте головные уборы и очки для защиты от солнца.

- Подбирайте открытую обувь, например сандали и шлёпки.

если температура от 20 до 30 градусов цельсия то бот пишет:

- Одевайтесь в легкую одежду из натуральных материалов .

- Футболки, рубашки, лёгкие джинсы , шорты , юбки, платья - отличный выбор для этой температуры.

- Предпочитайте обувь с хорошей продуваемостью, например кроссовки с сеточкой.

если температура выше от 10 до 20 градусов цельсия то бот пишет:

- Вы можете выбрать ветровку в сочетании с лёгким свитером или футболкой и длинными брюками.

- Обувайтесь удобно и не слишком тепло, например кеды или туфли.

если температура от 0 до 10 градусов цельсия то бот пишет:

- При выборе одежды отдавайте предпочтение утепленным материалам, таким как шерсть или пух.

- Не забудьте надеть шапку, ведь большая часть тепла теряется через голову.

- Используйте непродуваемую обувь.

если температура меньше 0 до 10 градусов цельсия то бот пишет:

- Выбирайте теплую и хорошо изолированную одежду.

- Одевайтесь в слои: нательное теплое белье, свитер или джемпер, а затем пальто или куртка

- Не забудьте о теплых носках и обуви.

- Используйте теплые шапки, шарфы и перчатки для защиты от холода.

А если будет дождь, то бот обязательно напомним :

“Не забудьте взять зонтик!”

Также я написал проверку , если пользователь вдруг не правильно написал название города , то бот сообщит это .

После я пишу вывод всех данных при помощи функции `reply_to` (Рис.11).

В конце я пишу команду `bot.polling(non_stop=True)`, чтобы бот бесперерывно работал.

2.5 Запуск бота на сервере.

Подошло время запустить бота на отдаленном сервере , чтобы он работал даже , когда не запущен мой компьютер. Для этого я выбрал сайт `pythonanywhere` , который бесплатно предоставляет сервер. Я зарегистрировался , загрузил все нужные файлы и при помощи консоли Bash запускаю файлы (Рис.12).Теперь бот запущен на сервере.

2.6 Тестирование бота.

Захожу в Telegram и нахожу своего бота по поиску . Проверяю его на работоспособность. Ввожу команду `/start`, и он даёт свой ответ (Рис.13).Далее ввожу `/w Ростов-на-Дону` и бот выдаёт всю информацию о погоде (Рис.14).

Бот готов и отлично выполняет свои функции.

Заключение.

В заключение, проект по разработке Telegram бота, предназначенного для предоставления информации о погоде, оказался не только увлекательным и интересным, но и очень полезным с точки зрения получения ценного опыта в области программирования.

В ходе данного проекта я исследовал различные языки программирования и библиотеки, применяемые для создания чат-ботов. Это дало мне возможность сравнить их особенности, преимущества и недостатки, что помогло мне принять обоснованные решения в выборе лучших средств решения проблем.

Разработка этого бота позволила мне погрузиться в процесс создания программного обеспечения с использованием языка Python и библиотеки telebot. Я приобрел уверенность в работе, научился обрабатывать пользовательские запросы, интегрировать сторонние сервисы (в данном случае - сервис прогноза погоды), а также улучшил навыки планирования и структурирования проекта.

Особенно стоит отметить, что Telegram оказался идеальной платформой для реализации своего бота. Его простота в использовании, обширные возможности по интеграции с внешними сервисами, а также удобный интерфейс для пользователя делают его отличным выбором для создания разнообразных ботов. Кроме того, активное сообщество разработчиков и обширная документация значительно облегчают процесс создания и развития бота.

В целом, разработка Telegram бота на Python не только позволила мне познакомиться с новыми технологиями и инструментами, но и дала ценный опыт, который я смог применить в будущих проектах. Telegram действительно является удобной и эффективной платформой для реализации своих идей, и я с уверенностью могу рекомендовать ее для создания ботов различной направленности.

Список использованной литературы.

1. uchet-jkh / [Электронный ресурс] - URL: <https://uchet-jkh.ru/i/nacem-pisutsya-boty-dlya-telegram/> (Дата обращения: 18.12.2023)
2. habr / [Электронный ресурс] - URL: <https://habr.com/ru/companies/otus/articles/771110/> (Дата обращения: 18.12.2023)
3. inclient / [Электронный ресурс] - URL: <https://inclient.ru/telegram-stats/> (Дата обращения: 03.03.2024)
4. metanit / [Электронный ресурс] - URL: <https://metanit.com/> (Дата обращения: 03.03.2024)
5. journal.tinkoff / [Электронный ресурс] - URL: <https://journal.tinkoff.ru/guide/howto-telegram-bot/#:~:text=%D0%91%D0%BE%D1%82%D0%B0%20%D0%BC%D0%BE%D0%B6%D0%BD%D0%BE%20%D0%BD%D0%B0%D0%BF%D0%B8%D1%81%D0%B0%D1%82%D1%8C%20%D0%BD%D0%B0%20%D0%BB%D1%8E%D0%B1%D0%BE%D0%BC,%D0%BE%D1%82%D0%BA%D1%80%D1%8B%D1%82%D1%8B%D1%85%20%D0%B1%D0%B8%D0%B1%D0%BB%D0%B8%D0%BE%D1%82%D0%B5%D0%BA%20%D0%B8%20%D0%B3%D0%BE%D1%82%D0%BE%D0%B2%D1%8B%D1%85%20%D1%80%D0%B5%D1%88%D0%B5%D0%BD%D0%B8%D0%B9> (Дата обращения: 03.03.2024).

Приложение 1

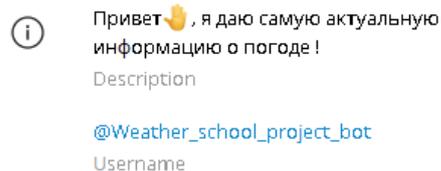
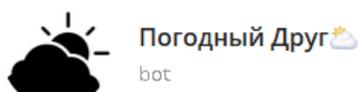


Рис.1 “Создание бота”



Рис.2 “Команды”

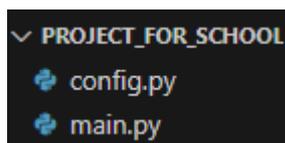


Рис.3 “Папка”

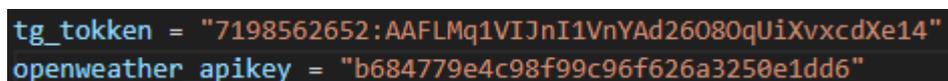


Рис.4 “Переменные для токенов”

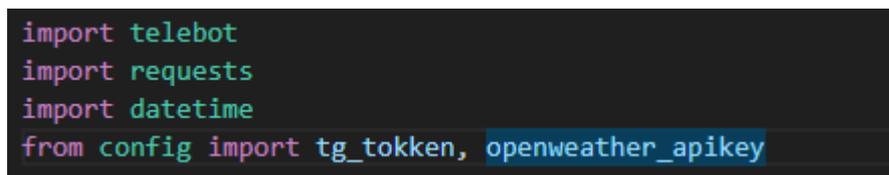


Рис.5 “Импорт библиотек”

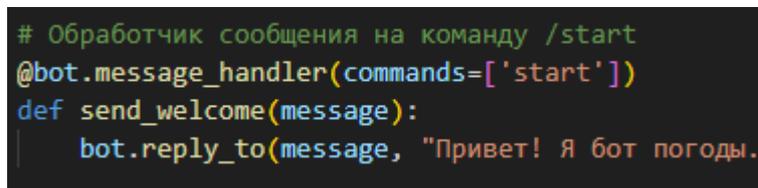


Рис.6 ”Создание объекта бота и обработчик на команду /start в скрипте”

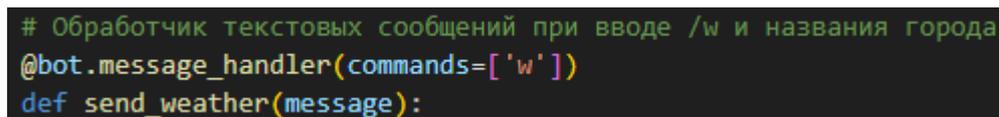


Рис.7 “Обработчик на команду /w и названия города в скрипте”



Приложение 2

Рис.8 “Отправка get запроса в скрипте”

```

temp = data["main"]["temp"] # Переменная со значением температуры
fells_temp = data["main"]["feels_like"] # Переменная со значением ощущаемой температуры
wind_speed = data["wind"]["speed"] # Переменная со значением скорости ветра
pressure = int(data["main"]["pressure"]) * 0.75006375541921 # Переменная со значением давления
humidity = data["main"]["humidity"] # Переменная со значением влажности
weather_description = data["weather"][0]["main"] # Переменная со значением погоды

```

Рис.9 “Назначение переменных с данными о погоде в скрипте”

```

def choise_cloth(fells_temp):
    fells_temp = float(fells_temp)
    if fells_temp > 30: # температура > 30
        return (" - Носите очень легкую и дышащую одежду, изо льна, хлопка, тонких вискозных тканей, натурального шелка.\n"
                " - Одевайтесь в светлые цвета, отражающие солнечные лучи.\n"
                " - Обязательно используйте головные уборы и очки для защиты от солнца.\n"
                " - Подбирайте открытую обувь, например сандали и шлёпки.\n")
    elif 20 < fells_temp <= 30: # температура от 20 до 30
        return (" - Одевайтесь в легкую одежду из натуральных материалов .\n"
                " - Футболки, рубашки, лёгкие джинсы , шорты , юбки, платья - отличный выбор для этой температуры.\n"
                " - Предпочитайте обувь ☑ хорошей продуваемостью, например кроссовки ☑ сеточкой.\n")
    elif 10 < fells_temp <= 20: # температура от 10 до 20
        return (" - Вы можете выбрать ветровку в сочетании ☑ лёгким свитером или футболкой и длинными брюками.\n"
                " - Обувайтесь удобно и не слишком тепло, например кеды и туфли.\n")
    elif 0 < fells_temp <= 10: # температура от 0 до 10
        return (" - При выборе одежды отдавайте предпочтение утепленным материалам, таким как шерсть или пух.\n"
                " - ☑ забудьте надеть шапку, ведь большая часть тепла теряется через голову.\n"
                " - Используйте непродуваемую обувь.\n")
    elif fells_temp <= 0: # температура меньше 0
        return (" - Выбирайте тёплую и хорошо изолированную одежду.\n"
                " - Одевайтесь в слои: нательное теплое белье, свитер или джемпер, ☑ затем пальто или куртка\n"
                " - ☑ забудьте ☑ теплых носках и обуви.\n"
                " - Используйте теплые шапки, шарфы и перчатки для защиты от холода.\n")

```

Рис.10 “Рекомендации по одежде в зависимости от температуры в скрипте”

```

# Вывод информации о погоде пользователю
response = (f"***{datetime.datetime.now().strftime('%Y-%m-%d %H:%M')}***\nПогода в городе: {st
            f"\nТемпература: {round(temp)}°C, ощущается как: {round(fells_temp)}°C\n"
            f"Влажность: {humidity}%\nДавление: {round(pressure)} мм.рт.ст\n☑ скорость в
            f"Восход солнца: {sunrise_timestap} \n☑ закат солнца: {sunset_timestap}\nПрод
            f"{choise_cloth(fells_temp)}\n☑ помните, что эти рекомендации могут варь
            "***Хорошего дня!***")
bot.reply_to(message,response)

```

Рис.11 “Вывод информации о погоде в скрипте”

```

bash: ~/home/naakta/.bashrc: Permission denied
05:18 ~ $ cd bot
05:18 ~/bot $ python main.py

```

Рис.12 “Запуск бота на сервере”

Приложение 3

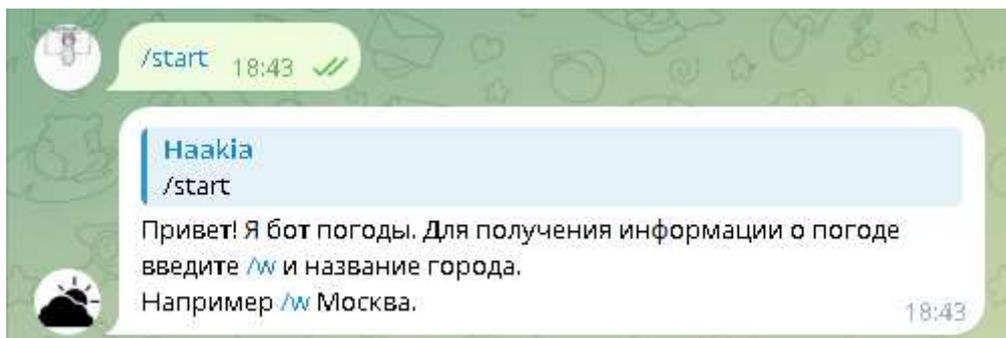


Рис.13 "Ответ на команду /start в Telegram"

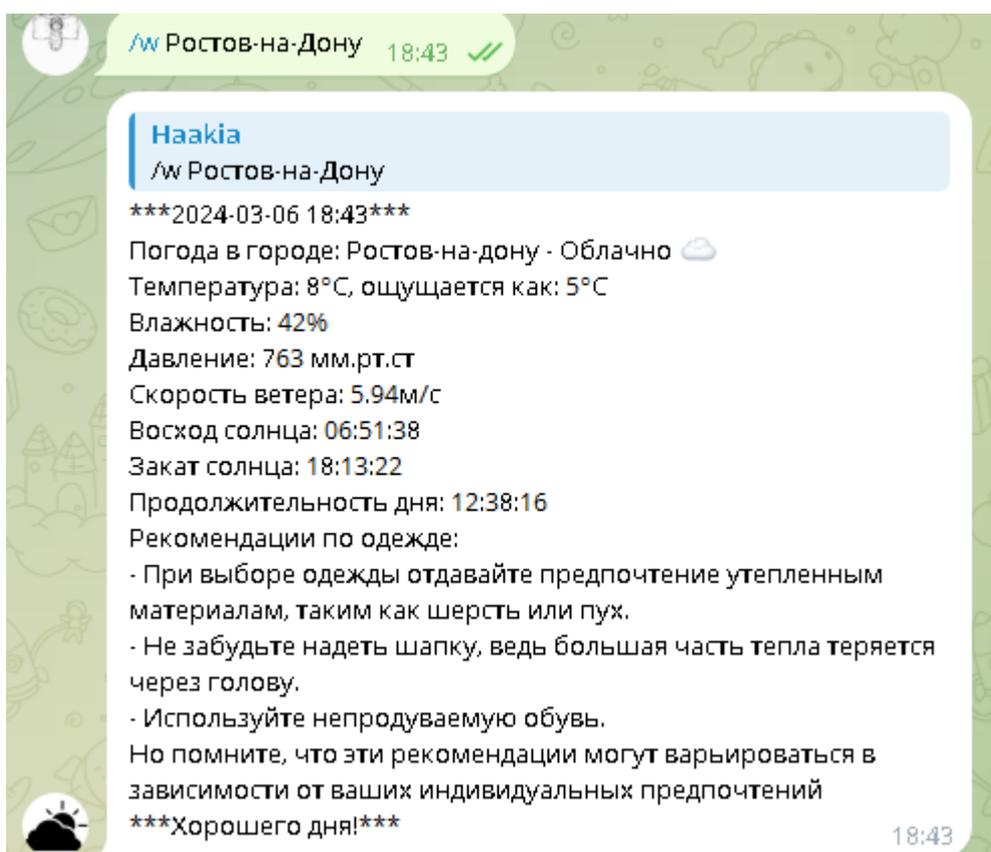


Рис.14 "Вывод информации о погоде в Telegram"