

## СОДЕРЖАНИЕ

ВЕДЕНИЕ .....	3
ГЛАВА 1. ОБЗОР И СРАВНИТЕЛЬНЫЙ АНАЛИЗ СРЕДСТВ ДЛЯ РАЗРАБОТКИ СЕРВИСА .....	5
1.1 Общие понятия, функции и классификация Telegram-бота .....	5
1.2 Выбор программного обеспечения для разработки Telegram-бота .....	9
1.3 Описание методологии разработки сервиса .....	17
ГЛАВА 2. ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ СЕРВИСА .....	23
2.1 Описание процесса создания Telegram-бота.....	23
2.2 Описание реализации Telegram-бота на языке программирования Python с помощью фреймворка aiogram .....	32
2.3 Практическая реализация Telegram-бота «Расписание Сыктывкара» ...	40
ЗАКЛЮЧЕНИЕ .....	48
СПИСОК ЛИТЕРАТУРЫ .....	50

## ВЕДЕНИЕ

Сегодня Telegram является одной из самых популярных мессенджеров в мире, успешно конкурируя со своими главными конкурентами - WhatsApp и Viber. Миллионы людей выбирают Telegram благодаря его простоте использования, безопасности и возможности общаться с друзьями и близкими, не стесняясь защиты своей личной информации.

В данном проекте рассмотрим о Telegram и его ботах, обозначим их функции и объясним, почему они стали так популярны. Нельзя отрицать, что они привлекательны для множества бизнесов. Разработчики теперь имеют возможность применять свои навыки в создании ботов для увеличения своей прибыли. Рассмотрим на создании проекта социально-направленного бота для пользователей Telegram и общественным транспортом в г. Сыктывкаре.

Целью данного проекта является разработка асинхронного Telegram-бота на языке Python для социального проекта «Расписание Сыктывкара».

Задачами проекта будут:

1. Рассмотреть общие понятия, функции и классификацию Telegram-бота.
2. Определить оптимальное программное обеспечения для разработки Telegram-бота.
3. Описать методологию разработки сервиса Telegram-бот.
4. Описать процесс создания Telegram-бота.
5. Описать реализации Telegram-бота на языке программирования Python с помощью фреймворка aiogram.

С помощью данного проекта будет рассмотрена теория и отработка практических навыков создания Telegram-бота на основном языке специализации автора на курсе обучения программированию Python с помощью фреймворка aiogram.

В проекте применялись методы анализа данных, обобщения сведений и описание практического опыта написания программного кода.

При создании проекта автор разработал и внедрил социального Telegram-бота [https://t.me/RaspSykt\\_bot](https://t.me/RaspSykt_bot) или @RaspSykt\_bot под названием «Расписание Сыктывкара». Проект рассчитан на людей, которые пользуются общественным транспортом и используют Telegram в своей повседневной жизни.

Проект разрабатывался на устройстве с операционной системой Linux с применением интегрированной среда разработки PyCharm на языке Python с основной библиотекой для создания Telegram-ботов aiogram.

# ГЛАВА 1. ОБЗОР И СРАВНИТЕЛЬНЫЙ АНАЛИЗ СРЕДСТВ ДЛЯ РАЗРАБОТКИ СЕРВИСА

## 1.1 Общие понятия, функции и классификация Telegram-бота

За последние полгода количество людей, использующих мессенджер, увеличилось на 66%. Это значительный рост, который привлекает внимание как новых пользователей, так и бизнес-сектор. Одна из ключевых особенностей Telegram, которой изучают пользователи, - это наличие тысяч местных ботов.

Бот - это маленькое приложение, которое может выполнять задачи без участия пользователя.

Сам термин «бот» - это сокращение от привычного слова «робот». Боты Telegram — это именно они. Это специальные аккаунты, которые создаются автоматически и позволяют пользователям выполнять различные задачи через мессенджер.

Бот может выполнять операции под руководством пользователя, который вводит текстовые команды. После нажатия кнопки «Старт» происходит диалог человека и машины по принципу «вопрос-ответ». С помощью таких виртуальных помощников пользователь может, например, узнать прогноз погоды или заказать суши.

Чтобы понять, что бот — это другая сторона, достаточно посмотреть на его имя. В названии всегда должно присутствовать слово «бот». Кроме того, утилита никогда не инициирует диалог и не имеет статуса «онлайн» или «оффлайн».

Перечислим основные преимущества ботов в Telegram:

- Работают круглосуточно, 7 дней в неделю;
- Быстрое время отклика;
- Простота использования путем выбора текстовых команд;

- Взаимодействие с ботом происходит через мессенджер, поэтому нет необходимости устанавливать дополнительное программное обеспечение;
- Нет доступа к личным данным, если только пользователь не даст прямого указания;
- Боты могут отправлять новости, рассказывать анекдоты, напоминать о важной информации, искать места, бронировать рестораны, заказывать билеты и выполнять множество других функций.

Он обладает функциональностью, позволяющей вести диалог с пользователями, отвечать на вопросы, предоставлять информацию и выполнять различные действия. Одним из примеров такого бота является Telegram-бот. Он способен выполнять все те же функции, что и человек в чате. Бот может отправлять текстовые сообщения, картинки, видео и файлы, как автоматически, так и по запросу пользователя. Однако, боты в Telegram могут делать гораздо больше, чем просто отвечать на вопросы и отправлять контент. Они могут выполнять действия, которые нельзя настроить на обычном канале. Например, бот может быть настроен на продажу товаров и принимать оплату от клиентов. Он может взаимодействовать с пользователями, отвечать на их запросы, предоставлять информацию о товарах или услугах.

Боты могут быть интегрированы с различными системами, такими как CRM (система управления взаимоотношениями с клиентами), платформы обмена сообщениями или системы продажи билетов. Это открывает широкие возможности для использования ботов в бизнесе. Например, бот может собирать потенциальных клиентов, подключая их к CRM-системе, чтобы упростить процесс управления клиентской базой данных. Бот может автоматически отправлять информацию о новых продуктах или акциях, а также отвечать на вопросы потенциальных клиентов. Также, бот может быть

использован для продажи билетов на мероприятия или предоставления информации о них.

Боты в Telegram становятся все более популярными и широко используются в различных сферах деятельности. Они облегчают взаимодействие с клиентами, автоматизируют рутинные задачи и помогают улучшить качество обслуживания. Благодаря своей гибкости и возможностям интеграции, боты становятся незаменимыми инструментами для бизнеса и повседневной жизни.

Боты в Telegram выполняют различные задачи, будь то команды самого мессенджера, бизнес-компаний или обычных людей. Они стали популярными не только среди пользователей, но и на рынке разработки. На платформе Kwork уже более тысячи фрилансеров предлагают свои услуги по созданию ботов. Цены на разработку ботов варьируются в широком диапазоне: от 500 до 500 тысяч рублей, в зависимости от сложности задачи.

Этот рост в популярности ботов отражает растущий интерес к автоматизации и удобству, которые они предоставляют. Боты становятся неотъемлемой частью нашей повседневной жизни, помогая людям во многих аспектах. Они могут отвечать на вопросы, предоставлять информацию, помогать в покупках и даже развлекать нас.

С ростом числа пользователей мессенджера и увеличением количества ботов, можно ожидать, что их функциональность и возможности будут продолжать развиваться. Таким образом, создание и использование ботов становится все более доступным и привлекательным для широкой аудитории. Это открывает новые возможности для бизнеса и обычных людей, которые хотят воспользоваться преимуществами автоматизации и удобства, которые предоставляют боты.

В этой социальной сети используется распространенная форма бота. Единственное отличие от реальных пользователей -приставка «бот» в названии.

Социальные боты, также известные как социальный искусственный интеллект или социальные алгоритмы, представляют собой программные агенты, которые самостоятельно взаимодействуют в социальных сетях. Сообщения, отправляемые этими ботами, просты и могут работать как в группах, так и в различных настройках, при этом имея определенную человеческую контролируемую функцию через алгоритмы. Кроме того, социальные боты могут использовать искусственный интеллект и машинное обучение для более естественного взаимодействия с людьми при передаче сообщений.

Все основные плюсы использования Telegram-ботов уже были описаны выше. Если говорить о проектах, которые ориентированы на общение и социальное взаимодействие, то боты в них работают режиме 24/7, с огромной скоростью реагируют на команды, отправленные людьми в виде текстовых сообщений. И само общение происходит прямо через мессенджер, без необходимости устанавливать еще какое-то дополнительное программное обеспечение. Создав такого бота однажды, он может прекрасно справляться со своими обязанностями очень долго, до тех пор, пока не потребуется внести изменения в его логику работы.

Telegram-боты в социально-направленных проектах — это большое преимущество среди программных комплексов имитирующих поведение реальных пользователей.

Telegram-боты социальных сетей разрабатываются для персонализированных сообщений, учитывая предварительно определенные пользовательские критерии. Например, бот может использоваться для анализа автобусных маршрутов и планирования времени поездки. Таким образом, осведомленность целевых групп, которые перемещаются на общественном городском транспорте, повышается благодаря проектам, основанным на социальной сети Telegram. Создание ботов с рассылкой сообщений, которые предварительно настраиваются в соответствии с заданными критериями

пользователей, является одним из способов анализа расписания движения общественного городского транспорта.

Следует выделить следующие боты для социальной сферы:

- Информационные боты. Основная задача таких ботов - информировать пользователей о тех или иных событиях.
- Бот-помощник. Это дополнения к веб-версии онлайн-сервисов.

Четких критериев классификации ботов для социальной направленности не существует. Дело в том, что, как правило, они сочетают в себе сразу несколько механизмов для эффективного выполнения ряда задач. Они помогают в переводе текста, обучении, тестировании, поиске информации, играх и даже в использовании других сервисов и взаимодействии с теми, кто имеет доступ к глобальной сети Интернет.

Таким образом, Telegram-бот можно назвать социально-ориентированным программным обеспечением, выполняющим свои функции круглосуточно, независимо от человеческого ресурса. Один из социально-направленных ботов, созданный автором проекта рассмотрим в следующем разделе.

## **1.2 Выбор программного обеспечения для разработки Telegram-бота**

В современной эпохе информационных технологий огромное количество процессов зависит от развития IT-сферы. Практически каждый работник предприятий в наши дни использует разнообразные приложения с целью повышения эффективности и качества работы. Мир постоянно движется вперед, и создание Телеграм-бота в мессенджере Telegram становится актуальным. Однако для реализации этой идеи необходимо учитывать скорость написания кода, возможности языка программирования и другие факторы. Таким образом, выбор языка программирования должен основываться на их способности написания кода, а также на возможностях их предоставления.



Язык программирования – это своего рода правила, которые определяют внешний вид компьютерной программы и ограничения для ее выполнения. Программа представляет собой набор кода, который пишется в соответствии с данными правилами. Этот код называется «исходным кодом». Языки программирования – формальные искусственные языки, аналогичные естественным языкам, включая алфавит, словарный запас, грамматику, синтаксис и семантику.

Создание бота в Telegram – это задача, которая требует времени и усилий. Чтобы успешно разработать Telegram-бота, необходимо обладать определенными знаниями, читать о соответствующих темах и разработать стратегию. Однако, можно упростить этот процесс, предоставив информацию о наиболее удобных и простых языках программирования для создания Telegram-ботов. Многие эксперты сходятся во мнении, что выбор языка программирования для создания бота в Telegram не имеет принципиального значения. Главное – иметь доступ к API мессенджера.

Согласно статистике, наиболее часто Telegram-ботов создают на языке Python. Это достаточно простой язык, легкий для изучения, и он предлагает множество готовых библиотек. Кроме Python, Telegram-бота можно написать на Java, Ruby, JavaScript, Node.js, Go и других языках. Также существуют списки языков с неофициальными библиотеками, но рекомендуется выбирать язык, который соответствует всем требованиям.

Однако, стоит отметить, что самым популярным и современным языком программирования является Python. Он особенно хорош для программистов благодаря простому синтаксису и обширной стандартной библиотеке функций.

Для создания бота в Telegram программисту понадобится пакет для создания Telegram-бота, который поддерживает оболочкой для API Telegram. Таким образом, выбор языка программирования для создания бота в Telegram зависит от целей.

Языки программирования можно разделить на две крупные категории: высокоуровневые и низкоуровневые. Высокоуровневые языки, такие как Java, Python и C++, обладают более простым синтаксисом и позволяют программисту более удобно работать, так как они ориентированы на удовлетворение потребностей разработчика, а не на определение возможностей компьютера. Однако программы, написанные на высокоуровневых языках, требуют перекодирования в машинный язык, чтобы компьютер мог их исполнить. Для этого используются интерпретаторы или компиляторы, встроенные в системы программирования на этих языках.

В отличие от высокоуровневых языков, таких как Python или Java, низкоуровневые языки, например, Assembler, более тесно связаны с машинным языком и дают возможность создавать программы, которые выполняются быстрее и используют ресурсы компьютера более эффективно. Но, естественно, использование таких языков требует более глубокого понимания аппаратной части компьютера, а их синтаксис является сложнее для программиста.

Существует обширное разнообразие языков программирования, каждый из которых находит своё применение в определенной сфере. Если задуматься о выборе языка программирования для создания Telegram-бота с доступом в интернет, необходимо обратить внимание на статистику популярности различных языков. Такой анализ поможет определиться с наиболее подходящим языком для реализации данного проекта.

Сегодня наиболее популярными языками программирования являются Java, Python, JavaScript, C++ и C#. Они широко применяются в различных областях, включая веб-разработку, мобильные приложения, научные исследования и многое другое. Каждый из этих языков имеет свои преимущества и особенности, поэтому выбор зависит от требований проекта и предпочтений разработчика.

Java является одним из самых популярных языков программирования и широко используется для создания кросс-платформенных приложений. Он

обладает богатой стандартной библиотекой, множеством инструментов и поддержкой большого сообщества разработчиков.

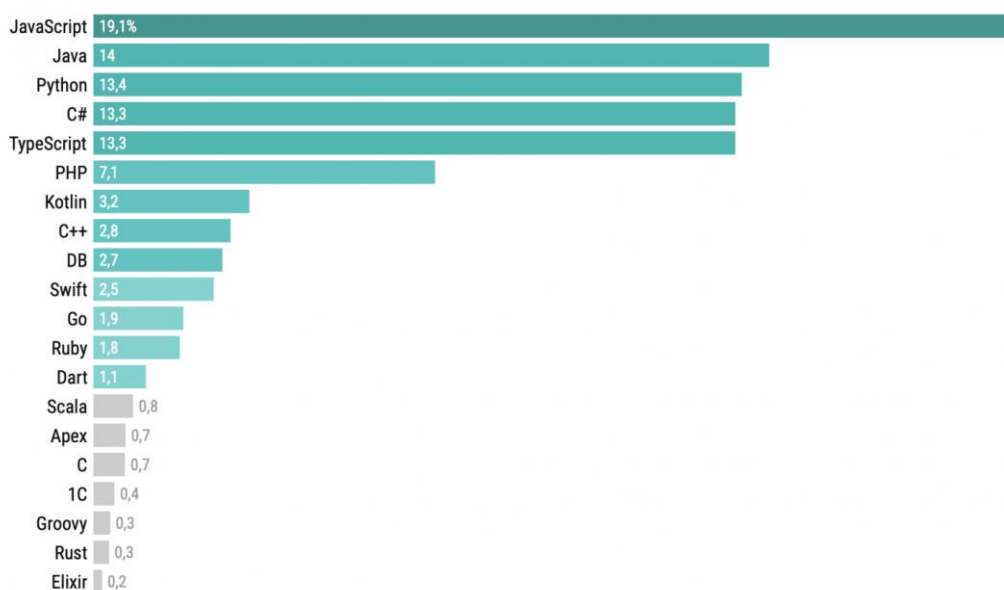
Python также пользуется большой популярностью благодаря своей простоте и удобству в использовании. Он обладает широким спектром библиотек и фреймворков, что делает его отличным выбором для разработки Telegram-бота.

JavaScript в основном используется для разработки интерактивных веб-приложений и может быть полезен, если планируется создать Telegram-бота, интегрированный в веб-сайт.

C++ и C# являются более низкоуровневыми языками и обеспечивают высокую производительность и эффективное использование ресурсов компьютера. Они часто используются для разработки сложных систем и игр.

При выборе языка программирования для Telegram-бота важно учитывать требования проекта, уровень знаний разработчика и доступность необходимых инструментов и библиотек.

По данным, представленным на GitHub результаты ежегодного опроса, в этом году самым популярным языком остается JavaScript — 19% разработчиков пишут на нем коммерческие проекты. Далее — Java (14%) и Python (13%). Последний впервые среди самых популярных. С минимальным отрывом за Python идут C# и TypeScript.



По данным, представленным выше самыми популярными языками на данный момент, являются JavaScript, C# и Python, поэтому для анализа и сравнения выберем именно эти языки.

JavaScript — мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили. Преимущества JavaScript: ни один современный браузер не обходится без поддержки JavaScript. С использованием написанных на JavaScript плагинов и скриптов справится даже не специалист. Полезные функциональные настройки. Постоянно совершенствующийся язык – сейчас разрабатывается бета-вариация проекта, JavaScript2. Взаимодействие с приложением может осуществляться даже через текстовые редакторы – Microsoft Office и Open Office. Перспектива использования языка в процессе обучения программированию и информатике. Недостатки JavaScript: Пониженный уровень безопасности ввиду повсеместного и свободного доступа к исходным кодам популярных скриптов. Множество мелких раздражающих ошибок на каждом этапе работы. Большая часть из них легко исправляется, но их наличие позволяет считать этот язык менее профессиональным, сравнительно с другими. Повсеместное распространение. Своеобразным недостатком можно считать тот факт, что часть активно используемых программ (особенно приложений) перестанут существовать при отсутствии языка, поскольку целиком базируются на нем.

C# — язык программирования, сочетающий объектно-ориентированные и контекстно-ориентированные концепции. Преимущества: для малых предприятий и некоторых отдельных разработчиков бесплатные инструменты включают Visual Studio, Azure Cloud, Windows Server, Parallels Desktop для Mac Pro и многие другие; большое количество синтаксических конструкций, разработанных для лучшего понимания написания кода; очень прост в изучении; после покупки Xamarin на C# можно писать программы и приложения для операционных систем, таких как iOS, Android, MacOS и Linux; имеется целое сообщество из опытных программистов. Недостатки:

приоритетная ориентированность на Windows платформу; язык бесплатен только для небольших фирм, индивидуальных программистов, стартапов и учащихся. Крупной компании покупка лицензионной версии этого языка обойдется в значительную сумму; в языке осталась возможность использования оператора безусловного перехода.

Python — это высокоуровневый язык программирования общего назначения, который также используется для разработки веб-приложений. Язык нацелен на повышение производительности труда разработчиков и читаемости кода. Python поддерживает различные парадигмы программирования: структурную, объектно-ориентированную, функциональную, императивную и аспектно-ориентированную. Язык включает в себя динамическую типизацию, автоматическое управление памятью, полное самонаблюдение, механизм обработки исключений, поддержку многопоточных вычислений и практические структуры данных высокого уровня.

Преимущества Python: открытая разработка; довольно прост в изучении, особенно на начальном этапе; особенности синтаксиса стимулируют программиста писать хорошо читаемый код; предоставляет средства быстрого прототипирования и динамической семантики; имеет большое сообщество, позитивно настроенное по отношению к новичкам; множество полезных библиотек и расширений языка можно легко использовать в своих проектах благодаря предельно унифицированному механизму импорта и программным интерфейсам; механизмы модульности хорошо продуманы и могут быть легко использованы; абсолютно всё в Python является объектами в смысле ООП, но при этом объектный подход не навязывается программисту. Недостатки Python: не слишком удачная поддержка многопоточности; на Python создано не так уж много качественных программных проектов по сравнению с другими универсальными языками программирования, например, с Java; отсутствие коммерческой поддержки средств разработки (хотя эта ситуация со временем

меняется); изначальная ограниченность средств для работы с базами данных; бенчмарки показывают меньшую производительность Python по сравнению с основными Java VM, что создаёт этому языку репутацию медленного.

Сейчас рассмотрим наглядно синтаксис языков по написанию равнозначного по результату кода:

## Hello World на Java

```
1 class HelloWorld {
2     public static void main(String[] args) {
3         System.out.println("Hello, World!");
4     }
5 }
```

Рисунок 1. Пример кода на семействе Java

## Hello World на C

```
1 #include <stdio.h>
2
3 int main()
4 {
5     printf("Hello, World!\n");
6     return 0;
7 }
```

рисунке 2. Пример кода на семействе C

## Hello World на Python

```
1 print("Hello, World!")
```

Рисунок 3. Пример кода на Python

По рисункам четко видно, когда семейство «Java» и «C» требуют больше четырех строчек кода, Python использует всего одну.

Таким образом, в короткие сроки можно овладеть навыками программирования на Python. Этот язык также совместим со всеми современными операционными системами. Опытные разработчики признают Python хорошим выбором для разработки программ.

Сравним Python с другими языками программирования.

Python и Java являются противоположностями в плане скорости запуска и выполнения программ. Python - интерпретируемый язык с динамической типизацией, в то время как Java - компилируемый язык со статической типизацией. Программы на Python запускаются быстрее, но выполняются медленнее, в то время как приложения на Java открываются медленнее, но работают быстрее.

Java позволяет разрабатывать кроссплатформенные приложения. Однако Python также совместим со всеми основными операционными системами и используется для разработки системных программ. Изучение Python требует меньше времени, чем изучение Java.

Python и C или C++. Все особенности Java (языка, похожего на C) также относятся и к C или C++. Отличие Python состоит в стоимости разработки и качестве программного обеспечения. Создание кода на C или C++ требует большей квалификации, поэтому ПО, созданное на этих языках, обладает высоким качеством и, соответственно, стоит дороже.

Python и C или C++ отличаются длиной кода и простотой восприятия. Программы, написанные на Python, могут быть в 10 раз короче, чем аналогичные на C или C++, что существенно ускоряет процесс разработки.

Python и JavaScript занимают первые две позиции в рейтинге TIOBE. Это два языка программирования, ориентированных на объекты. Синтаксис Python проще, чем у JavaScript. Однако программы, написанные на Python, выполняются медленнее. JavaScript обладает высокой производительностью и используется для разработки приложений, работающих в онлайн-режиме. Таким образом, программирование на Python популярно в сфере Data Science и машинного обучения, а JavaScript используется для веб-разработки.

Python и PHP. Оба языка не вызывают больших сложностей при изучении. Однако Python имеет более понятный и читаемый синтаксис. PHP специализируется на создании сайтов, в то время как Python является

универсальным языком программирования, предназначенным для решения широкого круга задач.

Python является одним из наиболее популярных вариантов благодаря своей простоте и обширной поддержке сообщества. Однако, важно помнить, что каждый язык имеет свои особенности и преимущества, поэтому следует выбирать то, что лучше всего соответствует потребностям и целям разработки бота в Telegram.

Автором работы был выбран именно язык программирования Python, который более понятен в изучении и легко читаем. Создаваемое приложение Telegram-бота существенно не зависит от скорости языка и не влияет на производительность приложения.

### **1.3 Описание методологии разработки сервиса**

В тексте описанном выше было отмечено, что Python - отличный выбор для создания Telegram-бота для Telegram. Теперь рассмотрим необходимые библиотеки этого языка для разработки такого Telegram-бота.

В Python есть несколько полезных модулей при создании Telegram-бота, которые рассмотрим ниже.

Фреймворков для написания Telegram-ботов очень много. Из популярных – есть aiogram, telebot, telethon, python-telegram-bot. Есть и по code решения по созданию Telegram-ботов.

Здесь разберём какие библиотеки или решения существуют, их плюсы и минусы и границы применимости.

Первой рассмотрим библиотеку python-telegram-bot. Библиотека поддерживает сразу 2 версии: синхронную и асинхронную. Синхронная – это версии 13.X. Асинхронная – в том же репозитории, просто для неё нужно установить версию библиотеки 20.X и старше.

За счёт этого библиотека дружелюбнее для начинающих, пусть это и создаёт некоторую путаницу для новичков.



Библиотека обладает более подробной документацией в сравнении с `aiogram`, есть статьи об архитектуре, спам-лимитах Telegram. Функционал, кажется, более насыщенным.

Из минусов – библиотека не обладает таким же активным комьюнити, как `aiogram`, в репозитории которого есть ссылки на сообщества по странам.

Второй библиотекой рассмотрим `telethon`. Эта библиотека нужна уже если есть желание сделать не обычного бота в Telegram, а подключить Python уже к аккаунту в Telegram и роботизировать его. Получится, что Python-скрипт как бы управляет аккаунтом, в то время как библиотеки выше управляют ботами, которые в рамках Telegram являются отдельными сущностями, так же, как и каналы или группы.

Это подходит далеко не под все проекты и нужно не во всех ситуациях. Пригодится, например, если есть желание писать пользователям первыми, так как Telegram-боты так делать не умеют. Но и попасть под запрет из-за спама так тоже куда легче, чем если работать с Telegram-ботом.

Документация библиотеки тоже весьма подробная и иногда даже раскрывает дополнительные темы, вроде философии библиотеки или чёрной магией, не предусмотренной библиотекой.

`Telebot` – простая в использовании библиотека для создания ботов для Telegram. Она предоставляет множество функций, которые позволяют легко создавать Telegram-ботов, включая поддержку клавиатур, обработку сообщений и многое другое. `Telebot` также имеет большое сообщество пользователей и разработчиков, которые могут помочь решить любые проблемы, с которыми может столкнуться программист.

Рассматривая преимущества `Telebot` можно выделить следующие: простота использования, большое сообщество пользователей и разработчиков, хорошая документация. Из недостатков выделяются: ограниченный функционал, не поддерживает работу с базами данных, могут возникнуть проблемы с производительностью при обработке большого количества запросов

В завершении опишем aiogram. Самый популярный вариант, который фигурирует практически в каждой статье о Telegram-боте. Самое распространённое ошибочное мнение него – что асинхронность ускорит Telegram-бота. На самом деле, aiogram делает это, но для этого недостаточно просто писать `async` и `await` в начале объявления или вызова функций. Для асинхронной работы Telegram-бота все операции внутри неё не должны быть блокирующими (или хотя бы какая-то осязаемая их часть). Это значит, что больше нельзя пользоваться библиотекой `requests`, ведь она синхронная, а нужен её асинхронный аналог, например, `httpx` или `asks`. Нельзя пользоваться `redis`, нужен `aioredis`. И так далее.

Библиотека хороша для написания асинхронного кода.

Создание ботов для Telegram – это важная задача для многих программистов и бизнес-владельцев. Когда дело доходит до выбора библиотеки для создания Telegram-ботов, две наиболее популярные опции, которые можно рассмотреть, это Aiogram и Telebot.

Создание ботов для Telegram – это одна из важных задач, которую многие программисты и предприниматели ставят перед собой. Когда речь заходит о выборе библиотеки, существуют две самые популярные опции – Aiogram и Telebot. У обеих библиотек есть свои плюсы и минусы, и окончательный выбор зависит от индивидуальных потребностей и предпочтений. Так что рассмотрим все варианты внимательно, чтобы сделать правильный выбор.

Aiogram – более новая библиотека для создания ботов для Telegram, которая предоставляет больше гибкости и возможностей, чем Telebot. Aiogram имеет более сложный API, который позволяет создавать более сложные функции, такие как обработка аудио, видео и файлов, а также поддерживает работу с базами данных. Aiogram также имеет активное сообщество пользователей и разработчиков, которые могут помочь решить любые проблемы, с которыми может столкнуться разработчик.

Выбор между Aiogram и Telebot обусловлен собственными требованиями и предпочтениями. Если нужна простая в использовании библиотека с базовым функционалом, то лучшим выбором будет Telebot. Если же использовать более гибкую и функциональную библиотеку, которая поддерживает работу с базами данных, тогда Aiogram может оказаться оптимальным решением для выбора. В любом случае, обе библиотеки предоставляют огромное количество возможностей для создания ботов для Telegram, и выбор зависит от потребностей и уровня опыта в данной сфере.

Автор проекта выбрал для использования библиотеку Aiogram, так как в проекте есть базы данных, что не поддерживает библиотека Telebot.

Далее опишем какие модули были использованы при написании проекта.

Asyncio - этот модуль для написания параллельного кода с использованием синтаксиса `async` или `await`. Asyncio используется в качестве основы для нескольких асинхронных модулей Python, которые предоставляют высокопроизводительные сетевые и веб-серверы, библиотеки подключений к базе данных, распределенные очереди задач и т.д.

Основные характеристики aiohttp: поддержка HTTP-протокола как на стороне клиента, так и на стороне сервера. Поддерживает как клиентские, так и серверные веб-сокеты «из коробки» и позволяет избежать чрезмерного запроса обратного вызова.

Signal представляет собой список зарегистрированных асинхронных обратных вызовов. Жизненный цикл сигнала состоит из двух этапов: после создания его содержимое может быть заполнено с помощью стандартных операций со списком.

Async-timeout - менеджер контекста тайм-аута, совместимый с asyncio.

Asyncpg - это библиотека интерфейса базы данных, разработанная специально для PostgreSQL и Python. Asyncpg - это эффективная, чистая реализация двоичного протокола сервера PostgreSQL для использования с asyncio фреймворком Python.

Attrs - это пакет Python, который избавив от рутинной работы по реализации объектных протоколов.

Dnspython предоставляет как высокоуровневый, так и низкоуровневый доступ к DNS. Классы высокого уровня выполняют запросы к данным с заданным именем, типом и классом и возвращают набор ответов. Классы низкого уровня позволяют напрямую манипулировать зонами DNS, сообщениями, именами и записями.

Gino - это асинхронная ORM (Объектно-реляционное отображение) библиотека для работы с базами данных в Python. Она обеспечивает высокую производительность и простой интерфейс для работы с базой данных.

Основные особенности Gino:

- Асинхронная поддержка основана на асинхронном подходе, позволяя приложениям работать быстро и эффективно с базами данных.
- Простота использования предоставляет интуитивно понятный API для работы с базой данных. Легко выполняет запросы, создаёт, изменяя и удаляет записи.
- Поддержка различных баз данных совместима с различными базами данных, такие как PostgreSQL, MySQL, SQLite и другими.
- Миграции базы данных предоставляет инструменты для управления миграциями базы данных, облегчая изменение структуры базы данных.

Пакет greenlet является дополнительной компонентой для версии CPython под названием Stackless, который обеспечивает поддержку микропотоков, известных как «тасклеты». Тасклеты запускаются псевдо-одновременно в одном или нескольких потоках операционной системы и синхронизируются с помощью обмена данными через «каналы».

Multidict — это специальный вид словаря, который предназначен для удобной работы с несколькими значениями одного и того же ключа. Такая функциональность особенно полезна в синтаксическом анализе, который широко используется в различных оболочках. Зачастую в подобных задачах

требуется работать с различными значениями, связанными с одним ключом, и именно здесь подкласс словаря `multidict` приходит на помощь. Он облегчает манипуляции с данными и позволяет эффективно работать с различными значениями, обеспечивая удобство и гибкость в процессе анализа.

`Pymongo`, представляющий собой пакет для Python, является набором инструментов, предназначенных для работы с MongoDB. Это является оптимальным выбором для взаимодействия с базой данных MongoDB из Python.

Пакет `python-dotenv` предоставляет удобный способ считывания пар ключ-значение из файла `.env` и загрузки переменных окружения, необходимых для вашего приложения. Этот небольшой инструмент значительно облегчает работу с окружением и упрощает настройку переменных для вашего проекта.

`Pytz`, как библиотека, осуществляет функции преобразования даты и времени, что делает ее незаменимой для работы с международными клиентами.

`SQLAlchemy` — это программное обеспечение с открытым исходным кодом для работы с базами данных при помощи языка SQL. Оно реализует технологию программирования ORM (Object-Relational Mapping), которая связывает базы данных с концепциями объектно-ориентированных языков программирования. `SQLAlchemy` позволяет описывать структуры баз данных и способы взаимодействия с ними прямо на языке Python.

Таким образом, главным для создания и подключения Telegram-бота необходимо использовать модуль `aiogram` - это современный и полностью асинхронный фреймворк для Telegram Bot API, написанный на Python с `asyncio` и `aiohttp`. Эта библиотека предоставляет удобный и высокоуровневый интерфейс для взаимодействия с API Telegram, что позволяет разработчикам легко создавать и настраивать функциональность Telegram-ботов.

## ГЛАВА 2. ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ СЕРВИСА

### 2.1 Описание процесса создания Telegram-бота

Перед тем, как приступить к разработке, важно полностью понять принципы работы Telegram-бота. Давайте рассмотрим пример на компьютере с установленным Telegram-клиентом. Вот несколько важных моментов, на которые следует обратить внимание:

1. Для работы с Telegram-ботом на компьютере необходимо установить интерпретатор Python, а также сервер и клиент Telegram. Это позволит автору проекта создать программу-бота, в которой будет определена логика, шаблоны и возможные операции.

2. Внутри интерпретатора Python будет функционировать программа-бот, в которой опишем логику работы и зададим необходимые операции. Это позволит разработчику управлять Telegram-ботом и обрабатывать запросы пользователей.

3. В приложении, написанном на Python, будем использовать библиотеку для связи с сервером Telegram. В эту библиотеку нужно вшить секретный ключ, чтобы сервер Telegram мог идентифицировать Telegram-бота и установить связь с ним.

4. Когда пользователь в Telegram делает запрос расписания, Telegram-бот загружает необходимые данные на сервер, а сервер выводит результат на компьютер.

5. Запрос пользователя проходит обработку через утилиту на Python, и на сервере Telegram формируется и возвращается необходимый ответ пользователю.

6. Внедрение Telegram-бота в мессенджер происходит без особых трудностей. Описанный принцип действий подходит не только для расписания, но и для любого другого типа Telegram-ботов в мессенджере.

Теперь давайте рассмотрим краткий план действий для разработки Telegram-бота:

1. Сначала требуется зарегистрировать нового бота в Telegram. Для этого в поисковой строке мессенджера найдите @BotFather и перейдите к нему. Затем вызовем команду «/Start» в нижней части окна и выберите команду «/newbot» из списка. Telegram-бот попросит указать имя для создаваемого бота (юзернейм). Обратите внимание, что после создания имя будет сложно изменить. Также потребуется указать короткое имя для Telegram-бота. Надежно сохраните токен Telegram-бота — в будущем это понадобится для авторизации и работы с Telegram-ботом.

Название должно быть уникальным, определить его с первого раза не всегда получается.

Очень многие юзернеймы уже заняты. Свободных коротких юзернеймов осталось очень мало.

Юзернейм Telegram-бота выглядит как обычный юзернейм, но должен заканчиваться на «bot».

Есть несколько Telegram-ботов с именами @pic, @vid, @sticker, @gamee — это официальные боты Telegrama, которые не содержат в имени «bot»

Какие существуют лимиты в Telegram для ботов:

- Количество Telegram-ботов (созданных через @BotFather) до 20 штук
- Длина @username для Telegram-бота от 5 до 32 символов
- Длина информации о Telegram-боте (/setabouttext) до 120 символов
- Длина описания Telegram-бота до 512 символов
- Файлы отправляемые Telegram-ботом без локального сервера Bot API до 50 МБ
- Лимит выгрузки файлов с локальным сервером Bot API до 2000 МБ
- Файлы принимаемые Telegram-ботом без локального сервера Bot API до 20 МБ

- Лимит загрузки файлов с локальным сервером Bot API до 2000 МБ
- Количество кнопок до 100 штук
- Markup-данные сообщения например, inline-кнопки до 10 KB
- Частота обычной отправки сообщений в групповых чатах до 20 сообщений в минуту в одном чате
- Количество команд (в BotFather) от 0 до 100 команд
- Длина команды (в BotFather) от 1 до 32 символов
- Длина описания команды до 256 символов, должно быть не пустым, если установлено через Bot API
- Длина метаданных для /start используется для deep linking до 64 байт
- Длина предупреждения Telegram-бота до 200 символов
- Количество inline-результатов до 50 элементов на страницу

Когда создается Telegram-бот, BotFather даст его токен. Токен выглядит примерно так: 110201543:AAHdqTcvCH1vGWJxfSeofSAs0K5PALDsaw. Именно с помощью токена можно управлять Telegram-ботом.

В BotFather удобно управлять Telegram-ботами своими командой /mybots

Теперь будет доступно редактирование Telegram-бота. С помощью команд можно изменить аватарку, общее название, описания и т.д. Также в боковую кнопку «Меню» можно зашить ссылку на сайт или социальную сеть.

Основные команды в BotFather:

- /setname — изменить имя Telegram-бота
- /setdescription — изменить описание Telegram-бота
- /setuserpic — изменить аватарку Telegram-бота
- /setcommands — задать команды для Telegram-бота
- /deletebot — удалить Telegram-бота

Telegram-бот создан и отображается в поиске, но пока еще ничего не умеет.



Таким образом, первый этап подготовки завершен и можно переходить к следующему.

Открыв Telegram-бота, пользователи могут увидеть его профиль.

Описание (Description) — это текст, который пользователи будут видеть в начале диалога с Telegram-ботом под заголовком «Что может делать этот бот?»

Информация (About) — это текст, который будет виден в профиле Telegram-бота.

Аватарка. Аватарки Telegram-ботов, в отличие от аватарок пользователей и чатов, не могут быть анимированными. Только картинки.

Команды — тут имеются ввиду подсказки команд в Telegram-боте. Подробнее о командах ниже.

Необходимо потратить время и заполнить описание и информацию Telegram-бота, чтобы пользователям было понятнее и проще его использовать. Можно оставить там свои контакты и поставьте аватарку, чтобы Telegram-бота было проще отличать от других чатов в списке.

Понятно, что главная функция Telegram-бота — отправлять и получать сообщения.

И то, и другое можно делать со всеми видами сообщений (фото и видео, файлы, опросы, голосовые сообщения и т. д.).

В Telegramе можно делиться файлами до 2 ГБ, но в Bot API более жесткие лимиты: Telegram-боты могут скачивать файлы до 20 МБ и отправлять файлы до 50 МБ.

Если Telegram-бот уже загрузил файл на сервер Telegramа, то может использовать `file_id`, чтобы отправлять этот файл.

Загружать файл на сервер можно в том числе и по URL файла.

Рассмотрим куда может писать Telegram-бот. Telegram-бот может писать в личку только тем пользователям, которые его запустили. Пользователь может заблокировать Telegram-бота, и тогда Telegram-бот снова не сможет ему писать. Telegram-боты не могут писать другим Telegram-ботам.

Telegram-бота можно добавить в группу (если в BotFather включена соответствующая настройка). По умолчанию видит не все сообщения.

В группе Telegram-боту можно дать права администратора, чтобы он мог выполнять действия админов. В одной группе может быть до 20 Telegram-ботов. В публичные группы (группы с юзернеймом) Telegram-ботов могут добавлять только админы.

Также Telegram-бота можно добавить в канал, причем только как администратора. Самый частый способ использования Telegram-ботов в каналах — добавление кнопок под постами («лайки», ссылки и прочее).

На самом деле многие группы в Telegramе являются супергруппами.

Раньше было четкое разделение на группы и супергруппы. По задумке, супергруппы — это группы для сообществ. Супергруппы могут иметь больше участников, публичные ссылки и другие улучшения.

Со временем, видимо, решили, что это неудобная концепция. Теперь обычная группа становится супергруппой, когда у группы меняются какие-нибудь настройки.

Супергруппу нельзя обратно превратить в группу. С точки зрения API супергруппа устроена так же, как и канал. Важное отличие супергрупп от обычных групп состоит в нумерации сообщений.

Опишем id пользователей и чатов. У каждого пользователя, Telegram-бота, группы, канала в Telegramе есть собственный id. Различать чаты в коде Telegram-бота следует именно по id, потому что он никогда не меняется.

В токене Telegram-бота первая часть — это его id. Например, токен 110201874:AAHdqTcvCH1vGWJxfSeofSAs0K5PALDsaw принадлежит Telegram-боту с id 110201874.

В Bot API перед id супергрупп и каналов пишется -100. Так, id 1356415630 превращается в -1001356415630.

Не рекомендуется хранить id пользователей и чатов в 32-битном типе числа: теперь id могут превышать  $2^{31}-1$ .

Каждое сообщение в Telegramе имеет свой id. Это относится и к системным сообщениям (пользователь зашел в группу, изменилось название группы и т. д.). Через Telegram API Telegram-боты могут получать по запросу сообщения в любом чате по их id. У сообщений id в супергруппах и каналах уникальны для чата: первое сообщение в чате имеет номер 1, второе имеет номер 2 и так далее.

У сообщений id в личных сообщениях и обычных группах работают по другому. Там, можно сказать, нумерация сквозная: id сообщения уникально для каждого отправившего его пользователя. Так, первое сообщение от пользователя во всех личных переписках и группах имеет номер 1, второе сообщение от того же пользователя имеет номер 2 и так далее.

Видимость сообщений в группах для Telegram-ботов обычно начинается именно на команды. Telegram не уведомляет Telegram-бота об остальных сообщениях, и это гарантирует приватность переписки.

Но если Telegram-боту нужно видеть все сообщения в группе (например, если это Telegram-бот или антиспам-бот), для него можно отключить Privacy mode.

Privacy mode — настройка в BotFather, которая по умолчанию включена. В таком режиме Telegram-бот в группах видит только такие сообщения:

- Сообщения с упоминанием Telegram-бота;
- Ответы на сообщение Telegram-бота, ответы на ответы и так далее;
- Системные сообщения;
- Команды для Telegram-бота.

Если Privacy mode выключен, то Telegram-бот видит все сообщения в группе, Telegram-бот-админ в группе, в любом случае видит все сообщения.

Telegram-бот, работающий через Bot API, в любом случае не будет видеть сообщения от других Telegram-ботов.

В группах могут быть не только сообщения от пользователей, так же могут быть:

- Сообщения в группе из привязанного канала (с точки зрения API это пересланные сообщения)
- Сообщения от лица группы от анонимных администраторов той же группы
- Сообщения от лица публичных каналов от любых пользователей

Это стоит учитывать при разработке Telegram-ботов для групп.

Часто используемый способ «общения» пользователей с Telegram-ботом — команды. Команды начинаются на «/» и состоят из латинских букв (можно использовать цифры и нижние подчеркивания).

Команды подсвечиваются как ссылки: нажатие отправляет команду в чат.

В группах, чтобы различать команды от разных Telegram-ботов, Telegram предлагает ставить в конце команды юзернейм Telegram-бота. Например: «/start@examplebot».

В BotFather можно указать подсказки команд для Telegram-бота. Он будут отображаться при вводе «/» и команд. Если есть подсказки, рядом с кнопкой «Отправить» появляется кнопка для открытия меню команд.

Если в подсказках команд «/help», в профиле Telegram-бота появляется кнопка «Помощь с ботом». Нажатие на кнопку отправляет эту команду.

Если в подсказках команд есть «/settings», в профиле Telegram-бота появляется кнопка «Настройки бота». Нажатие на кнопку отправляет эту команду.

С 2021 года Telegram-боты могут показывать разные меню команд для разных пользователей и групп, а также меню команд может зависеть от языка пользователя и того, является ли участник группы админом.

Как можно заметить, сообщения в Telegramе могут содержать не только обычный текст, но и жирный, курсив и др. В Bot API разметку сообщений можно делать в HTML и Markdown.

В Telegram API для разметки надо вместе с сообщением передавать entities (MessageEntityBold, MessageEntityItalic и так далее). Хорошие библиотеки сами превращают HTML или Markdown в текст и entities.

Способы выделения текста:

- Жирный текст
- Курсив
- Подчёркнутый текст
- Зачёркнутый текст
- Моноширинный текст («в строке» и «блоком»)
- Ссылка (встроенная в текст)
- Спойлер — текст, который показывается только после нажатия на плашку.

Упоминание пользователя — это текст, похожий на ссылку, клик по которому открывает профиль пользователя. Если упомянуть в группе её участника, он получит уведомление. Чтобы вставить в сообщение упоминание пользователя, в Bot API нужно встроить ссылку на «tg://user?id=123456789».

Telegram-бот может оставлять кнопки под своими сообщениями. Кнопки под сообщениями (они же inline keyboards или inline buttons) в основном бывают трёх видов:

- URL button — кнопка с ссылкой.
- Callback button. При нажатии на такую кнопку Telegram-боту придёт апдейт. С созданием кнопки можно указать параметр, который будет указан в этом апдейте (до 64 байтов). Обычно после нажатий на такие кнопки Telegram-боты изменяют исходное сообщение или показывают notification или alert.

- Switch to inline button. Кнопка для переключения в инлайн-режим. Кнопка может открывать инлайн в том же чате или открывать меню для выбора чата. Можно указать в кнопке запрос, который появится рядом с никнеймом Telegram-бота при нажатии на кнопку.

Есть другой тип кнопок: keyboard buttons. Они отображаются вместо клавиатуры как подсказки. При нажатии на такую кнопку пользователь просто отправит этот текст. При этом в личных чатах с помощью кнопки можно:

- Запросить номер телефона пользователя,
- Запросить геолокацию пользователя,
- Открыть у пользователя меню создания опроса.

Есть опция `resize_keyboard`, которая отвечает за то, изменять ли высоту этой «клавиатуры из кнопок». По умолчанию она, обычно, выключена, и тогда высота клавиатуры стандартная большая.

Инлайн-режим (inline mode) — это специальный режим работы Telegram-бота, с помощью которого пользователь может использовать Telegram-бота во всех чатах. Пользователь вводит юзернейм Telegram-бота в поле для ввода сообщения. После юзернейма можно ещё записать запрос (текст до 256 символов). Появляется меню с результатами. Выбирая результат, пользователь отправляет сообщение.

Инлайн-режим можно включить в BotFather, там же можно выбрать плейсхолдер вместо стандартного «Search...»

В группе можно запретить использовать инлайн всем или некоторым участникам. В официальных приложениях Telegram это ограничение объединено с ограничением на отправку стикеров и GIF.

Таким образом, процесс создания Telegram-бота на языке программирования Python складывается из многочисленных особенностей Telegram, который предписывает определенные команды и ограничения, создающих Telegram-ботов, как по функционалу, так и по оформлению.

## **2.2 Описание реализации Telegram-бота на языке программирования Python с помощью фреймворка aiogram**

Для начала реализации создадим файл с базовым шаблоном Telegram-бота на aiogram.

Основная особенность этой библиотеки заключается в том, что она асинхронная, что означает, что все хэндлеры должны быть асинхронными, а вызовы методов API должны содержать ключевое слово `await`, так как они возвращают корутины.

Корутины - это паттерн проектирования, который позволяет писать асинхронные программы, способные выполнять несколько задач одновременно. В отличие от традиционных потоков, где задачи выполняются параллельно и могут блокировать друг друга, корутины позволяют приостанавливать выполнение одной задачи и переключаться на другую. Это позволяет более эффективно использовать ресурсы процессора и памяти. Корутины - это легковесные вычисления, которые работают поверх потоков. Они могут быть приостановлены и возобновлены в любой момент, что позволяет эффективно использовать ресурсы процессора. Когда корутина приостанавливается, связанные с ней вычисления сохраняются в памяти и освобождают поток для выполнения других задач. Это позволяет выполнять несколько корутин одновременно в рамках ограниченного пула потоков, что снижает нагрузку на системные ресурсы.

В aiogram, чтобы создать асинхронного Telegram-бота, необходимо использовать декоратор «`@dp.message_handler`» для обработки входящих сообщений. Этот декоратор принимает различные параметры, такие как текстовые фильтры и типы сообщений, которые должны быть обработаны. Затем определяем асинхронную функцию, которая будет вызываться, когда подходящее сообщение будет получено. Внутри этой функции можем выполнять различные действия, такие как отправка сообщений, обработка данных и т. д. Кроме того, aiogram предоставляет удобный интерфейс для работы с различными асинхронными методами API Telegram. Можно

использовать ключевое слово `await` для ожидания ответа от API и дальнейшей обработки полученных данных. Например, можем отправить сообщение с помощью метода `@bot.send_message` и дождаться ответа, чтобы убедиться, что сообщение было успешно отправлено.

В целом, `aiogram` предоставляет мощный инструментарий для разработки асинхронных ботов на платформе Telegram. Благодаря использованию корутин и эффективному управлению ресурсами, можем создавать Telegram-ботов, которые могут обрабатывать большое количество запросов одновременно, минимизируя нагрузку на систему.

На что стоит обратить внимание по сравнению с другой библиотекой для Telegram, например, `pyTelegramBotAPI`, концепция хэндлеров (обработчиков событий), разница лишь в том, что в `aiogram` хэндлерами управляет диспетчер.

Диспетчер регистрирует функции-обработчики, дополнительно ограничивая перечень вызывающих их событий через фильтры. После получения очередного апдейта (события от Telegram), диспетчер выберет нужную функцию обработки, подходящую по всем фильтрам, например, «обработка сообщений, являющихся изображениями, в чате с ID `икс` и с длиной подписи `игрек`». Если две функции имеют одинаковые по логике фильтры, то будет вызвана та, что зарегистрирована раньше.

Чтобы зарегистрировать функцию как обработчик сообщений, нужно сделать одно из двух действий:

1. Навесить на неё декоратор, как в примере выше.
2. Напрямую вызвать метод регистрации у диспетчера или роутера.

Для того чтобы сделать код чище и читабельнее, `aiogram` расширяет возможности стандартных объектов Telegram. Например, вместо `@bot.send_message(...)` можно написать `message.answer(...)` или `message.reply(...)`. В последних двух случаях не нужно подставлять `chat_id`, подразумевается, что он такой же, как и в исходном сообщении.



Разница между `answer` и `reply` простая: первый метод просто отправляет сообщение в тот же чат, второй делает «ответ» на сообщение из `message`.

Более того, для большинства типов сообщений есть вспомогательные методы вида «`answer_{type}`» или «`reply_{type}`».

Существует ряд ситуаций, когда при запуске Telegram-бота необходимо передать несколько дополнительных значений. Может быть, это будет какая-нибудь переменная или же объект конфигурации, список чего-то, отметка времени и так далее. Для этой цели достаточно передать указанные данные в качестве наименованных (`kwargs`) аргументов в диспетчере или присвоить значения, как если бы работали со словарем. Это особенно удобно для передачи объектов, которые должны существовать в единственном экземпляре и не изменяться в процессе работы Telegram-бота (то есть быть только для чтения). Если предполагается, что значение будет изменяться со временем, необходимо помнить, что это будет работать только с изменяемыми объектами. Чтобы получить значения в хэндлерах, просто указывайте их как аргументы.

Для обеспечения безопасности и избежания хранения токена прямо в коде, можно поместить такие данные в отдельный конфигурационный файл. В нашей работе будем использовать отдельные файлы формата `.env`. Это позволит легко управлять конфиденциальными данными и обеспечить безопасность информации.

Очень важно отметить, что передача информации в виде аргументов и использование файлов конфигурации `.env` являются эффективными методами для работы с данными в Telegram-боте. Эти подходы позволяют гибко управлять информацией и обеспечивают безопасность и конфиденциальность данных. В этой части подробно рассмотрим применение различных форматирований к сообщениям и работу с мультимедийными файлами.

Одна из ключевых задач большинства Telegram-ботов - это обработка текстовых сообщений. Текст может быть использован для выражения практически любой информации, и важно представлять ее красиво.

Разработчику доступны три способа разметки текста: HTML, Markdown и MarkdownV2. Среди них наиболее продвинутыми считаются HTML и MarkdownV2, в то время как «классический» Markdown имеет ограниченные возможности и больше не используется в aiogram.

Перед тем, как приступить к изучению возможностей работы с текстом в aiogram, важно отметить одно важное отличие между версией aiogram 3.x и 2.x: в «двойке» по умолчанию обрабатывались только текстовые сообщения, тогда как в «тройке» можно обрабатывать сообщения любого типа. Одним из аргументов, отвечающих за выбор форматирования при отправке сообщений, является параметр `parse_mode`.

Если в Telegram-боте часто применяется одно и то же форматирование, то каждый раз указывать аргумент `parse_mode` может быть неудобно. Однако, в aiogram есть возможность передать необходимый тип прямо в объект Bot. И если в какой-то конкретной ситуации не требуется использовать форматирование, просто укажите `parse_modeNone`.

Нередко бывают ситуации, когда окончательный текст сообщения Telegram-бота заранее неизвестен и формируется исходя из каких-то внешних данных: имя пользователя, его ввод и т.д. Напишем хэндлер на команду «/hello», который будет приветствовать пользователя по его полному имени (`first_name + last_name`), например: «Hello, Иван Иванов».

Разработчики могут воспользоваться удобствами Telegram, которые значительно упрощают их жизнь. Предварительная обработка сообщений пользователей на стороне Telegram позволяет избежать необходимости использования регулярных выражений для извлечения определенных сущностей, таких как адрес электронной почты, номер телефона, имя пользователя и другие. Вместо этого, можно получить эти данные непосредственно из объекта Message и его поля `entities`, содержащего массив объектов типа MessageEntity. Чтобы проиллюстрировать это, напомним обработчик, который извлекает ссылку, адрес электронной почты и моноширинный текст из сообщения, по одному элементу каждого типа.

Здесь кроется важный подвох. Telegram возвращает не сами значения, а их начало в тексте и длину. Более того, текст считается в символах UTF-8, а entities работают с UTF-16, из-за этого, если просто взять позицию и длину, то при наличии UTF-16 символов (например, эмодзи) ваш обработанный текст просто сместится.

Telegram предоставляет пользователям множество способов ввода информации. Одним из них являются команды: ключевые слова, начинающиеся со слэша, например, «/new» или «/ban». Иногда Telegram-бот может быть спроектирован так, чтобы ожидать после самой команды какие-то *аргументы*, вроде «/ban 2d» или «/settimer 20h This is delayed message». В составе aiogram есть фильтр Command(), упрощающий жизнь разработчика.

Telegram предлагает не только текстовые сообщения, но и возможность обмениваться различными медиафайлами, такими как фото, видео, гифки, геолокации и стикеры. Каждый из этих медиафайлов имеет свой уникальный идентификатор файла (file\_id) и уникальный идентификатор файла (file\_unique\_id). Используя file\_id, можно многократно отправлять один и тот же файл, при этом отправка будет мгновенной, так как сам файл уже хранится на серверах Telegram. Этот способ является наиболее предпочтительным.

В отличие от file\_id, идентификатор file\_unique\_id нельзя использовать для повторной отправки или скачивания медиафайла, но зато он одинаковый у всех Telegram-ботов для конкретного медиа. Нужен file\_unique\_id обычно тогда, когда нескольким Telegram-ботам требуется знать, что их собственные file\_id относятся к одному и тому же файлу.

Когда файл ещё не присутствует на сервере Telegram, Telegram-бот имеет возможность загрузить его по трем различным путям: как файл из файловой системы, используя ссылку или напрямую предоставив набор байтов. Это позволяет ускорить процесс отправки и обеспечить более аккуратное обращение к серверам мессенджера. Важно правильно выполнить загрузку (upload) файлов на Telegram только один раз и затем использовать полученный file\_id, который будет доступен после первой загрузки медиа.

В aiogram 3.x присутствуют 3 класса для отправки изображений - FSInputFile, BufferedInputFile, URLInputFile.

В этой части погрузимся в описание свойства Telegram-ботов, как использование кнопок. Прежде всего, чтобы не путаться, давайте определимся с терминологией. Одни, которые закрепляются внизу экрана вашего устройства, будем называть простыми кнопками, а те, что присоединяются к сообщениям, - инлайн-кнопками.

Вместе с появлением Bot API в далеком 2015 году появился и этот вариант кнопок, которые представляют собой шаблоны сообщений. Принцип работы простой - то, что написано на кнопке, будет отправлено в текущий чат. Таким образом, для обработки нажатия кнопки Telegram-боту необходимо распознавать входящие текстовые сообщения.

В Telegram существует шесть специальных видов обычных кнопок, не являющихся обычными шаблонами сообщений. Они предназначены для:

- отправки текущей геолокации;
- отправки своего контакта с номером телефона;
- создания опроса или викторины;
- выбора и отправки Telegram-боту данных пользователя с нужными критериями;
- выбора и отправки Telegram-боту данных (супер) группы или канала с нужными критериями;
- запуска веб-приложения (WebApp).

Опишем них подробнее.

Отправка текущей геолокации. Здесь всё просто: где пользователь находится, те координаты и отправляет. Это будет статическое гео, а не Live Location, который обновляется автоматически.

Отправка своего контакта с номером телефона стало проще, благодаря кнопке «Отправить контакт» с предварительным подтверждением. Когда пользователь нажимает на эту кнопку, Telegram-бот получает его контакт с

номером телефона. Кроме того, для предотвращения мошенничества, была реализована проверка наличия совпадений между отправленным контактом и профилем пользователя. Таким образом, даже если некоторые пользователи попытаются обойти кнопку и отправить другой контакт, сможем обнаружить такие случаи и принять соответствующие меры: достаточно проверить в хэндлере или в фильтре равенство «`message.contact.user_id == message.from_user.id`».

Создание опроса или викторины. По нажатию на кнопку пользователю предлагается создать опрос или викторину, которые потом отправятся в текущий чат. Необходимо передать объект `KeyboardButtonPollType`, необязательный аргумент `type` служит для уточнения типа опроса (опрос или викторина).

Выбор и отправка Telegram-боту данных пользователя с нужными критериями. Показывает окно выбора пользователя из списка чатов юзера, нажавшего на кнопку. Необходимо передать объект `KeyboardButtonRequestUser`, в котором надо указать сгенерированный любым способом айди запроса и критерии, например, «бот», «есть подписка Telegram Premium» и т.д. После выбора юзера Telegram-бот получит сервисное сообщение с типом `UserShared`.

Выбор и отправка Telegram-боту чата с нужными критериями. Показывает окно выбора пользователя из списка чатов юзера, нажавшего на кнопку. Необходимо передать объект `KeyboardButtonRequestChat`, в котором надо указать сгенерированный любым способом айди запроса и критерии, например, «группа или канал», «юзер — создатель чата» и т.д. После выбора юзера Telegram-бот получит сервисное сообщение с типом `ChatShared`.

Запуск веб-приложения (WebApp). При нажатии на кнопку открывает WebApp. Необходимо передать объект `WebAppInfo`.

В отличие от обычных кнопок, инлайновые цепляются не к низу экрана, а к сообщению, с которым были отправлены. В этой части рассмотрим два типа таких кнопок: `URL` и `Callback`.

Самые простые инлайн-кнопки относятся к типу URL, т.е. «ссылка». Поддерживаются только протоколы HTTP(S) и «tg://».

В марте 2019 года разработчики Telegram добавили возможность отключать переход к профилю пользователя у пересланного сообщения. При попытке создать URL-кнопку с ID юзера, у которого отключен переход по форварду, Telegram-бот получит ошибку `Bad Request: BUTTON_USER_PRIVACY_RESTRICTED`. Соответственно, прежде чем показывать такую кнопку, необходимо выяснить состояние упомянутой настройки. Для этого можно вызвать метод `getChat` и в ответе проверить состояние поля `has_private_forwards`. Если оно равно `True`, значит, попытка добавить URL-ID кнопку приведёт к ошибке.

Telegram-бота — это приложение, которое должно работать круглосуточно. Для этого есть хостинги, которые работают на постоянной основе. Конечно, можно развернуть доступ и на личном компьютере, но в этом случае компьютер будет выполнять роль сервера и его нельзя будет отключать от сети. Еще одна особенность — это пропускная способность Интернет-соединения, как правило, у хостинга эти показатели очень отличаются от обычного доступа.

Для развертывания бота для Telegram лучше всего подходит VDS/VPS. Он не требует наличия домена в обязательном порядке, и к нему можно будет обратиться по выделенному IP-адресу, что удобно при развертывании Telegram-бота. К тому же такие серверы лучше поддерживают увеличение масштабов проекта, так что при увеличении потока посетителей вам не придется переходить на другой сервер — достаточно будет просто сменить тариф или его настройки.

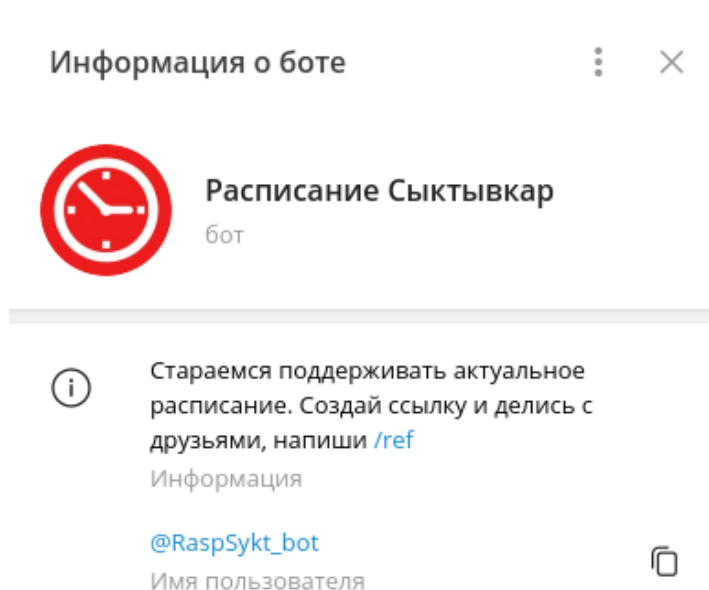
Таким образом при соблюдении особенностей применения модулей и знания языка программирования Python с помощью фреймворка `aiogram` можно разработать Telegram-бота и разместить его на хостинге с круглосуточным режимом работы. Саму разработку Telegram-бота «Расписание Сыктывкара» опишем ниже.

## 2.3 Практическая реализация Telegram-бота «Расписание Сыктывкара»

Для реализации проекта, автором была осуществлена регистрация Telegram-бота с помощью системного Telegram-бота @BotFather и зафиксированы имя пользователя @RaspSykt\_bot и название Telegram-бота «Расписание Сыктывкара». Telegram-боту присвоена ссылка, по которой можно перейти в него из любого браузера [https://t.me/RaspSykt\\_bot](https://t.me/RaspSykt_bot).

Создано короткое описание для привлечения подписчиков и с призывом делиться этим Telegram-ботом со своими друзьями и знакомыми через реферальную ссылку, которую они могут создать самостоятельно с помощью команды Telegram-боту «/ref», что позволяет проявить интерес участника к самому Telegram-боту.

Оформление Telegram-бота приведено на рисунке ниже:



В процессе регистрации был получен и главный секретный токен, который необходим для настройки API.

Далее воспользуемся IDE PyCharm и установим на него дополнительное расширение aiogram с помощью меню IDE или командной строки по команде:

```
pip install aiogram
```

Для конфиденциальности все секретные данные были вынесены в отдельный системный файл .env

Файл .env - это текстовый файл, который содержит пары ключ-значение всех переменных, требуемых приложением. Этот файл включен в проект локально, не сохраняется в системе управления версиями, чтобы не подвергать риску потенциально конфиденциальную информацию.

В этом файле указаны следующие ключи:

- BOT\_TOKEN — полученный токен при регистрации
- ADMIN — ID администратора Telegram-бота
- PGUZER — имя пользователя базы данных
- PGPASSWORD — пароль доступа к базе данных
- DATABASE — название базы данных

```
1 import os # библиотека для работы с ОС
2
3 from dotenv import load_dotenv
4
5 load_dotenv()
6
7 BOT_TOKEN = str(os.getenv("BOT_TOKEN"))
8
9 admins_id = [
10     str(os.getenv("ADMIN"))
11 ]
12
13 support_ids = [
14     str(os.getenv("ADMIN"))
15 ]
16
17 ip = os.getenv('ip')
18 PGUZER = str(os.getenv('PGUZER'))
19 PGPASSWORD = str(os.getenv('PGPASSWORD'))
20 DATABASE = str(os.getenv('DATABASE'))
21
22 POSTGRES_URI = f'postgresql://{PGUZER}:{PGPASSWORD}@{ip}/{DATABASE}'
23
24
```



Далее, как уже выше писалось, для начала надо создать файл с базовыми настройками самого Telegram-бота, где прописывается TOKEN, ID администратора, данные для подключения к базе данных.

Следующий файл описывает функцию, запускаемую вместе с запуском Telegram-бота, которая выполняет подключение к базе данных, созданием необходимых таблиц в базе данных (при необходимости можно удалить таблицы) и сообщение админам, что подключение к базе данных произошло успешно, таблицы созданы и Telegram-бот запущен и готов к работе. Это облегчает диагностику начальной работы Telegram-бота.

```
1  async def on_startup(dp): # Создаем функцию, которая будет запускаться при запуске бота
2
3      import filters
4      filters.setup(dp)
5      import middlewares
6      middlewares.setup(dp)
7
8      from loader import db # эта команда подключает к БД PSQL
9      from utils.db_api.db_gino import on_startup
10     print('Подключение к PSQL')
11     await on_startup(dp)
12     #
13     # print('Удаление базы данных')
14     # await db.gino.drop_all()
15     #
16     print('Создание таблиц')
17     await db.gino.create_all()
18     print('Готово')
19
20     from utils.notify_admins import on_startup_notify
21     await on_startup_notify(dp) # эта команда отправляет всем админам что бот запущен
22
23     from utils.set_bot_commands import set_default_commands
24     await set_default_commands(dp)
```

Теперь требуется создать меню с командами самого Telegram-бота, для этого делаем асинхронную функцию, где прописываем меню с командами. Эти команды можно вызвать через меню или просто отправить текстом следующего вида, например: «/bus».

```

1  from aiogram import types
2
3
4  2 usages
5  async def set_default_commands(dp):
6
7      types.BotCommand(command='bus', description='Автобусы'),
8      types.BotCommand(command='trains', description='Поезда'),
9      types.BotCommand(command='plane', description='Самолеты'),
10     types.BotCommand(command='artlife', description='Арт Лайф'),
11     types.BotCommand(command='poster', description='Афиша'),
12     types.BotCommand(command='help', description='Помощь')
13
14 ]]
```

Создадим файл с кодом для главного запуска — команды «/start». В нем пропишем функцию, которая улавливает команду и ID пользователя, а так же

```

9  |
10 @rate_limit(limit=3)
11 @dp.message_handler(*custom_filters: IsPrivate(), CommandStart()) # создаем месенджер который ловит команду
12 async def commands_start(message: types.Message, user_id=None): # создаем асинхронную функцию
13     args = message.get_args() # принимает сообщение /start и id кто пригласил
14     new_args = await quick_commands.check_args(args, message.from_user.id)
15
16     try:
17         user = await quick_commands.select_user(message.from_user.id)
18         if user.status == 'active':
19             await message.answer(f'Привет <b>{user.first_name}</b>\n'
20                                 f'Какое расписание тебя интересует?\n'
21                                 f'👉 Открой меню:')
22         elif user.status == 'banned':
23             await message.answer('Ты заблокирован')
24     except Exception:
25         await quick_commands.add_user(user_id=message.from_user.id,
26                                       first_name=message.from_user.first_name,
27                                       last_name=message.from_user.last_name,
28                                       username=message.from_user.username,
29                                       referral_id=int(new_args),
30                                       status='active')
31
32     try:
33         count_refs = await quick_commands.count_refs(int(new_args)) # Считает сколько уже зарегистрировано
34         await dp.bot.send_message(chat_id=int(new_args),
35                                   text=f'По твоей ссылке зарегистрировался {message.from_user.first_name}\n'
36                                         f'У тебя уже зарегистрировалось: {count_refs} ')
37     except Exception:
38         pass
39     await message.answer(f'Привет {message.from_user.full_name}! \n'
40                           f'Мы рады приветствовать тебя!\n'
41                           f'Выбери в меню, что тебя интересует:\n'
42                           f'👉 ')
43
```

ID человека, пригласившего его, если пользователь перешёл по реферальной ссылке. Сделаем проверку, есть этот человек в базе данных или нет, если это новый человек, значит сообщим ему, что он уже зарегистрирован. Если этого пользователя нет в базе данных, то отправим ему приветственное сообщение с именем, если это имя указано у него.

Так как с Telegram-ботом работает администратор, который может отправлять сообщения конкретному пользователю, требуется установить приватность этих сообщений. Создадим проверку на приватность сообщений отправляемых администратором пользователю и наоборот.

```
1  from aiogram import types
2  from aiogram.dispatcher.filters import BoundFilter
3
4
5  class IsPrivate(BoundFilter):
6      async def check(self, message: types.Message):
7          return message.chat.type == types.ChatType.PRIVATE
```

В разработанном Telegram-боте предусмотрена возможность отправки рассылки с информированием всех пользователей Telegram-бота об

```
12
13  @dp.message_handler(*custom_filters: IsPrivate(), text='/mailing', chat_id=admins_id)
14  async def start_mailing(message: types.Message, state: FSMContext):
15      await message.answer(f'Введите текст рассылки:')
16      answer02 = message.text
17      await state.update_data(text=answer02)
18      await BM.text.set()
19
20  @dp.message_handler(*custom_filters: IsPrivate(), state=BM.text, chat_id=admins_id)
21  async def mailing_text(message: types.Message, state: FSMContext):
22      answer = message.text
23      markup03 = InlineKeyboardMarkup(row_width=2, inline_keyboard=[
24          [
25              InlineKeyboardButton(text='Добавить фотографию', callback_data='add_photo'),
26              InlineKeyboardButton(text='Далее', callback_data='next'),
27              InlineKeyboardButton(text='Отменить', callback_data='quit')
28          ]
29      ])
30      await state.update_data(text=answer)
31      await message.answer(text=answer, reply_markup=markup03)
32      await BM.stete.set()
33
```

изменениях в расписании. Для этих целей предусмотрена функция создания рассылки с возможностью прикрепления фотографии.

В самом начале, при создании Telegram-бота была заложена возможность приглашения в Telegram-бот по реферальной ссылке. Для этих целей пропишем код создание реферальной ссылки с помощью команды «/ref». Команда создает реферальную ссылку, отправляет ее пользователю, запросившему её и сообщает о количестве подписавшихся по этой ссылке пользователей.

```
1  from aiogram import types
2      from loader import dp
3      from aiogram.utils.deep_linking import get_start_link
4
5  from utils.db_api import quick_commands
6
7
8  @dp.message_handler(commands=['ref']) #
9  async def commands_hello(message: types.Message):
10      ref_link = await get_start_link(payload=message.from_user.id)
11      count_refs = await quick_commands.count_refs(message.from_user.id)
12      await message.answer(f'Привет {message.from_user.full_name}\n'
13                           f'У тебя {count_refs} подписчиков\n'
14                           f'Твоя реферальная ссылка\n'
15                           f'{ref_link}')
```

Как и в любой программе нужно предусмотреть действия пользователей, которые пытаются совершить непредсказуемые команды. В

```
1  from aiogram import types
2      from loader import dp
3      from utils.misc import rate_limit
4
5
6  @rate_limit(limit=10)
7  @dp.message_handler()
8  async def error_op(message: types.Message):
9      await message.answer(f'Команды "{message.text}" нет в боте')
10
```

этом случае пользователь получит предупреждение, что такой команды нет в Telegram-боте.

Если пользователь будет повторять несуществующие команды многократно в течение короткого времени, то будет заблокирован на определенное время, о чем получит сообщение. Это предусмотрено написанным кодом, который устанавливает временное ограничение на неизвестные команды, которые не прописаны в Telegram-боте.

```
1 def rate_limit(limit: int, key=None):
2     def decorator(func):
3         setattr(func, 'throttling_rate_limit', limit)
4         if key:
5             setattr(func, 'throttling_key', key)
6         return func
7     return decorator
```

В Telegram-боте предусмотрено, если требуется помощь или

```
1 from aiogram import types
2 from loader import dp
3 from aiogram.types import ReplyKeyboardMarkup, KeyboardButton
4 from utils.misc import rate_limit
5
6
7 route51 = ReplyKeyboardMarkup(
8     keyboard=[
9         [
10             KeyboardButton(text="Есть неточность"),
11             KeyboardButton(text="Есть предложение")
12         ],
13     ],
14     resize_keyboard=True, one_time_keyboard=True
15 )
16 @rate_limit(limit=10)
17 @dp.message_handler(text='/help')
18 async def commands_help(message: types.Message):
19     await message.answer(text=f'Привет {message.from_user.full_name}! \n'
20                           f'Какая тебе нужна помощь?', reply_markup=route51)
```

дополнительная информация, а как же он нашёл неточность в расписании, которое размещено в Telegram-боте, то пользователь имеет возможность об этом сообщить администратору Telegram-бота. Для этих целей предусмотрена команда «/help».

В этой команде есть два варианта общения: просто сообщить о неточности или же вступить в переписку с администратором Telegram-бота. Есть ограничение по времени для вступления в прямую переписку с администратором.

```
1 from datetime import datetime, time
2
3 if time( hour: 17, minute: 00) < datetime.utcnow().time() > time( hour: 6, minute: 0):
4     print(datetime.utcnow().time())
5 else:
6     print(time( hour: 17, minute: 00))
```

Основные настройки программирования Telegram-бота описаны выше. Для его круглосуточного функционирования автор проекта разместил Telegram-бот на хостинге <https://timeweb.cloud>, что позволяет Telegram-боту работать вне зависимом от человека режиме.

Из выше написанного, следует, что разработка Telegram-бота «Расписание Сыктывкара» произведена успешно, Telegram-бот развернут на хостинге и осуществляет свою работу в круглосуточном режиме, помогая пользователям Telegram, осуществляющим передвижение на общественном транспорте узнавать своевременно расписание автобусов. Найти действующий Telegram-бот можно по имени @RaspSykt\_bot и названию Telegram-бота «Расписание Сыктывкара». Telegram-боту присвоена ссылка, по которой можно перейти в него из любого браузера [https://t.me/RaspSykt\\_bot](https://t.me/RaspSykt_bot).

## ЗАКЛЮЧЕНИЕ

Таким образом, Telegram-бот можно назвать социально-ориентированным программным обеспечением, выполняющим свои функции круглосуточно, независимо от человеческого ресурса. Один из социально-направленных ботов, созданный автором проекта, был обоснован и описан с приложением скриншотов кода.

Автором работы был выбран именно язык программирования Python, который более понятен в изучении и легко читаем. Создаваемое приложение Telegram-бота существенно не зависит от скорости языка и не влияет на производительность приложения.

Python является одним из наиболее популярных вариантов благодаря своей простоте и обширной поддержке сообщества. Однако, важно помнить, что каждый язык имеет свои особенности и преимущества, поэтому следует выбирать то, что лучше всего соответствует потребностям и целям разработки бота в Telegram.

Главным для создания и подключения Telegram-бота, автором проекта был использован модуль `aiogram` - это современный и полностью асинхронный фреймворк для Telegram Bot API, написанный на Python с `asyncio` и `aiohttp`. Эта библиотека предоставляет удобный и высокоуровневый интерфейс для взаимодействия с API Telegram, что позволяет разработчикам легко создавать и настраивать функциональность Telegram-ботов.

Процесс создания Telegram-бота на языке программирования Python складывается из многочисленных особенностей Telegram, который предписывает определенные команды и ограничения для разработчиков, создающих Telegram-ботов, как по функционалу, так и по оформлению. Автор проекта придерживался этих правил.

При соблюдении особенностей применения модулей и знания языка программирования Python с помощью фреймворка `aiogram` был разработан

Telegram-бот и размещен на хостинге с круглосуточным режимом работы. Само приложение Telegram-бота «Расписание Сыктывкара» описано подробно в данном проекте.

Из выше написанного, следует, что разработка Telegram-бота «Расписание Сыктывкара» произведена успешна, Telegram-бот развернут на хостинге и осуществляет свою работу в круглосуточном режиме, помогая пользователям Telegram, осуществляющим передвижение на общественном транспорте узнавать своевременно расписание автобусов. Найти действующий Telegram-бот можно по имени @RaspSykt\_bot и названию Telegram-бота «Расписание Сыктывкара». Telegram-боту присвоена ссылка, по которой можно перейти в него из любого браузера [https://t.me/RaspSykt\\_bot](https://t.me/RaspSykt_bot).

Делая вывод из выше написанного, что цель проекта достигнута, задачи выполнены.

Проект подлежит дальнейшему развитию. Автор проекта намерен реализовать базу данных движения автобусов по остановкам с указанием конкретного времени прибытия и выбором в Telegram-боте конкретной остановки для более точного информирования пользователей о прибытии автобусов на остановку. Возможно применение геоданных для автоматического определения остановки с предложением автобусных маршрутов, проходящих по ближайшей остановке.



## СПИСОК ЛИТЕРАТУРЫ

1. Автоматизация бизнес-процессов как необходимое условие эффективности компании [Электронный ресурс]. – URL: <https://www.kp.ru/guide/avtomatizatsija-biznesa.html> (Дата обращения: 21.09.2023).
2. Автоматизированные информационные системы [Электронный ресурс]. – URL: [https://spravochnick.ru/informacionnye\\_tehnologii/setevye\\_informacionnye\\_sistemy/avtomatizirovannye\\_informacionnye\\_sistemy/](https://spravochnick.ru/informacionnye_tehnologii/setevye_informacionnye_sistemy/avtomatizirovannye_informacionnye_sistemy/) (Дата обращения: 21.09.2023).
3. Алгоритм [Электронный ресурс]. – URL: <http://www.machinelearning.ru/wiki/index.php?title=Алгоритм> (Дата обращения: 13.09.2023).
4. Василькова В. В., Легостаева Н. И., Радусhevский В. Б. Социальные боты как инструмент развития гражданского участия // Мониторинг общественного мнения: Экономические и социальные перемены. 2019. № 5. С. 19—42
5. Василькова В. В., Легостаева Н. И., Радусhevский В. И. Коммуникативные стратегии социальных ботов: модели воздействия на аудиторию // Вестник Санкт-Петербургского университета. Социология. 2019. Т. 12. Вып. 2. С. 153-163.
6. Введение в язык Java [Электронный ресурс]. – URL: <http://www.codenet.ru/webmast/java/03.php> (Дата обращения: 17.09.2023).
7. Виды языков программирования [Электронный ресурс]. – URL: <http://csaa.ru/vidy-jazykov-programmirovanija/> (Дата обращения: 17.09.2023).
8. Википедия [Электронный ресурс]. – URL: <https://ru.wikipedia.org/wiki/JavaScript> (Дата обращения: 17.09.2023).

9. Всё о парсинге сайтов на Python [Электронный ресурс]. – URL: [https://pikabu.ru/story/vsyo\\_o\\_parsinge\\_saytov\\_na\\_python\\_5384166](https://pikabu.ru/story/vsyo_o_parsinge_saytov_na_python_5384166) (Дата обращения: 20.09.2023).

10. ГОСТ 19.701-90 ЕСПД Схемы алгоритмов, программ, данных и систем. Обозначения условные и правила выполнения [Электронный ресурс]. – URL: <http://docs.cntd.ru/document/gost-19-701-90-esp> (Дата обращения: 14.09.2023).

11. Доступ к Интернету в Python с использованием Urllib.Request и urlopen() [Электронный ресурс]. – URL: <https://webformyself.com/python-urllib-request-i-urlopen/> (Дата обращения: 19.09.2023).

12. Изучаем программирование на Python / Пол Бэрри; [пер. с англ. М.А. Райтман]. - Москва: Издательство «Э», 2018. - 624 с.

13. Использование MS Project для управления проектами [Электронный ресурс]. – URL: <https://habr.com/post/151593/> (Дата обращения: 09.09.2023).

14. Как использовать чат-боты в бизнесе: 5 идей и 5 кейсов [Электронный ресурс]. – URL: <https://vc.ru/services/93850-kak-ispolzovat-chat-boty-vbiznese-5-idey-i-5-keysov> (Дата обращения: 18.09.2023).

15. Как раскрутить социальный стартап: опыт создателей чат-бота по профориентации школьников [Электронный ресурс]. – URL: <https://vc.ru/education/151148-kak-raskrutit-socialnyy-startap-opyt-sozdateley-chat-bota-po-proforientacii-shkolnikov> (Дата обращения: 21.11.2023).

16. Как самостоятельно сделать бота в «Telegram» без навыков программирования [Электронный ресурс]. – URL: <https://journal.tinkoff.ru/guide/howto-telegram-bot> (Дата обращения: 01.12.2023).

17. Лимиты Telegram [Электронный ресурс]. – URL: [limits.tginfo.me](https://limits.tginfo.me)

18. Модуль datetime [Электронный ресурс]. – URL: <https://pythonworld.ru/moduli/modul-datetime.html> (Дата обращения: 19.09.2023).

19. Модуль json [Электронный ресурс]. – URL: <https://pythonworld.ru/moduli/modul-json.html> (Дата обращения: 19.09.2023).
20. Модуль random [Электронный ресурс]. – URL: <https://pythonworld.ru/moduli/modul-random.html> (Дата обращения: 19.09.2023).
21. Модуль sqlite — Работаем с базой данных [Электронный ресурс]. – URL: <https://python-scripts.com/sqlite> (Дата обращения: 19.09.2023).
22. Модуль subprocess [Электронный ресурс]. – URL: <https://pythonworld.ru/moduli/modul-subprocess.html> (Дата обращения: 19.09.2023).
23. Модуль sys [Электронный ресурс]. – URL: <https://pythonworld.ru/moduli/modul-sys.html> (Дата обращения: 19.09.2023).
24. Модуль time [Электронный ресурс]. – URL: <https://pythonworld.ru/moduli/modul-time.html> (Дата обращения: 19.09.2023).
25. Осваиваем парсинг сайта: короткий туториал на Python [Электронный ресурс]. – URL: <https://proglib.io/p/web-scraping> (Дата обращения: 20.09.2023).
26. Понятие языка программирования [Электронный ресурс]. – URL: <https://ibrain.kz/informatika/ponyatie-yazyk-programmirovaniya> (Дата обращения: 17.09.2023).
27. Понятие языка программирования [Электронный ресурс]. – URL: <https://ibrain.kz/informatika/ponyatie-yazyk-programmirovaniya> (Дата обращения: 17.09.2023).
28. Примеры использования чат-ботов в бизнесе [Электронный ресурс]. – URL: <https://vc.ru/flood/25197-business-bot> (Дата обращения: 18.09.2023).
29. Развитие моделирования бизнес-процессов [Электронный ресурс]. – URL: <https://www.scienceforum.ru/2015/802/15636> (дата обращения: 11.09.2023).

30. Рейтинг языков программирования 2023. JavaScript/TypeScript завоевывают мир, Python вошел в топ-3 [Электронный ресурс]. – URL: <https://habr.com/ru/articles/730954/> (Дата обращения: 21.11.2023).

31. Роботы для борьбы с рутинной [Электронный ресурс]. – URL: [https://terralink.ru/rpa/?utm\\_source=google&utm\\_medium=cpc&utm\\_term=автоматизация%20бизнес%20процессов&utm\\_content=automatizato\\_in\\_message2&utm\\_campaign=RPA\\_RU\\_LP\\_RPA\\_TRG\\_GA&gclid=CjwKCAjw8J32BRBCEiwApQEKgeKRmP9f9lun1C5mY4MKg0hddoZFakNzSA\\_GV8l-VRxZwYzWmYewZRhoCSkgQAvD\\_BwE](https://terralink.ru/rpa/?utm_source=google&utm_medium=cpc&utm_term=автоматизация%20бизнес%20процессов&utm_content=automatizato_in_message2&utm_campaign=RPA_RU_LP_RPA_TRG_GA&gclid=CjwKCAjw8J32BRBCEiwApQEKgeKRmP9f9lun1C5mY4MKg0hddoZFakNzSA_GV8l-VRxZwYzWmYewZRhoCSkgQAvD_BwE) (Дата обращения: 21.09.2023).

32. Самоучитель PYTHON [Электронный ресурс]. – URL: <http://pythoshka.ru/p1138.html> (Дата обращения: 20.09.2023).

33. Снимет – Википедия [Электронный ресурс]. – URL: <https://ru.wikipedia.org/wiki/Снимет> (Дата обращения: 17.09.2023).

34. Социальные сети в России: цифры и тренды, осень 2023 [Электронный ресурс]. – URL: <https://vc.ru/social/932129-socialnye-seti-v-rossii-cifry-i-trendy-osen-2023> (Дата обращения 03.12.2023)

35. Текстовый редактор – Википедия [Электронный ресурс]. – URL: [https://ru.wikipedia.org/wiki/Текстовый\\_редактор](https://ru.wikipedia.org/wiki/Текстовый_редактор) (Дата обращения: 17.09.2023).

36. Хабр, учебный процесс в IT [Электронный ресурс]. – URL: <https://habr.com/ru/company/skillbox/blog/443412/> (Дата обращения: 17.09.2023).

37. Чат-бот в Telegram: что важно знать и как создать [Электронный ресурс]. – URL: <https://umnico.com/ru/blog/chatbot-for-telegram> (Дата обращения: 28.11.2023).

38. Что лучше pytelegrambotapi или aiogram [Электронный ресурс]. – URL: <https://iakpp.ru/articles/chto-luchshe-pytelegrambotapi-ili-aiogram.html> (Дата обращения: 26.10.2023).

39. Что такое бот в Telegram: виды и функции [Электронный ресурс]. – URL: <https://gb.ru/blog/chto-takoe-telegram> (Дата обращения: 10.10.2023).

40. Что такое программное обеспечение [Электронный ресурс]. – URL: <http://procomputer.su/comp-gramotnost/79-chto-takoe-programmnoeobespechennie> (Дата обращения: 17.09.2023).

41. Язык программирования Python [Электронный ресурс]. – URL: <https://web-creator.ru/articles/python> (Дата обращения: 17.09.2023).

42. Hello World на разных языках программирования [Электронный ресурс]. – URL: <https://vscode.ru/articles/hello-world-na-raznyh-yazykah-programmirovaniya.html>

43. SQL Lite [Электронный ресурс]. – URL: <https://lecturesdb.readthedocs.io/databases/sqlite.html> (Дата обращения: 21.09.2023).

44. Tkinter.messagebox — Tkinter message prompts [Электронный ресурс]. – URL: <https://docs.python.org/3.9/library/tkinter.messagebox.html> (Дата обращения: 19.09.2023).

45. With ... as - менеджеры контекста [Электронный ресурс]. – URL: <https://pythonworld.ru/osnovy/with-as-menedzhery-konteksta.html> (Дата обращения: 19.09.2023).