МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

ФАКУЛЬТЕТ АВТОМАТИКИ И ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

КАФЕДРА ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

**Отчет ПО ПРАКТИКЕ**

Создание Telegram-бота для путешественников по городам России

тема

Место проведения практики: Новосибирский государственный технический

полное наименование предприятия (организации)

университет

Сроки практики по учебному плану: с 1.09.22. по 31.12.22.

|  |  |
| --- | --- |
| Выполнил:  Студент: Ванин Константин Евгеньевич\_\_\_\_\_\_  (Ф. И. О.)  Группа: *\_\_АВТ-918\_\_\_\_\_\_\_\_\_*  Факультет: \_\_\_АВТ\_\_\_\_\_\_\_\_\_\_\_\_.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  подпись  «\_\_\_»\_\_декабря\_\_ 2022 г. | Проверил:  Руководитель: к.т.н., доцент Зыбарев В.М.  Балл: \_\_\_\_\_\_\_\_\_\_  Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *(«отлично», «хорошо», «удовлетворительно», «неуд.»)*  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  подпись  «\_\_\_» \_\_декабря\_\_ 2022 г. |

НОВОСИБИРСК

2022

**Содержание**

[Введение 2](#_Toc122469238)

[1. Организационная структура кафедры, направления научной и научно-исследовательской работы 4](#_Toc122469239)

[1.1 Современное состояние материально-технической базы кафедры 4](#_Toc122469240)

[1.2 Организация учебного процесса на кафедре 6](#_Toc122469241)

[1.3 Организация НИР на кафедре 7](#_Toc122469242)

[2. Статистический анализ 9](#_Toc122469243)

[2.1 Статистический анализ мессенджера Telegram 9](#_Toc122469244)

[2.2 Статистический анализ чат-ботов и поисковых запросов пользователей 12](#_Toc122469245)

[3. Выбор технологий и среды разработки 16](#_Toc122469246)

[3.1 Выбранные технологии разработки 16](#_Toc122469247)

[3.1.1 Программная платформа «Node.js» 16](#_Toc122469248)

[3.1.2 Telegram Bot API 17](#_Toc122469249)

[3.1.3 Облачная платформа «Heroku» 19](#_Toc122469250)

[3.2 Среда разработки 19](#_Toc122469251)

[3.2.1 PyCharm Community Edition 19](#_Toc122469252)

[4. Разработка Telegram-бота 21](#_Toc122469253)

[4.1 Разработка чат-бота для Telegram Bot API 21](#_Toc122469254)

[4.2 Тестирование Telegram-бота 25](#_Toc122469255)

[Заключение 28](#_Toc122469256)

[Список использованной литературы 29](#_Toc122469257)

[Приложение 31](#_Toc122469258)

# **Введение**

В соответствии с целями практики и заданием на учебную практику необходимо было разработать Telegram-бота для путешественников по городам России.

Интернет в современных условиях – это универсальная среда для общения, развлечений и обучения. В настоящее время в мире существует большое количество средств, форм и способов общения, и немалая часть из них так или иначе связана с современными техническими возможностями, которые, в частности, представлены использованием глобальной компьютерной сети. Интернет кроме источника разнообразной и полезной для пользователей информации также является основной формой виртуального общения. Мессенджеры, как форму общения, стали использовать чаще, чем социальные сети. Мессенджер — это эффективный канал, который стоит использовать в качестве формы получения информационных сообщений, выполнения информационных запросов и т.д.

Актуальность научно-исследовательской работы обусловлена высокой популярностью мессенджеров и таких средств автоматизации как чат-боты среди пользователей сети Интернет. Чат-боты позволяют упростить ежедневные рутинные задачи, такие как получение информации о погоде, пробках, последних новостях и другие. Главным достоинством относительно классических приложений является возможность совмещения всех возможностей на платформе одного мессенджера. Согласно статистике основной потребностью путешественников является поиск направлений для путешествий; поиск отелей, гостиниц, хостелов. Следующие по популярности темы — достопримечательности и развлечения, туристические компании, транспорт. Реже упоминают конкретные варианты отдыха, спрашивают про погоду и оформление виз.

Мы предполагаем, что актуальным будет создания информационного Telegram-бота, который сможет показывать прогноз погоды в разных городах России, что поможет путешественникам подготовиться к поездке более точно. В России Telegram-боты разнообразны по выполняемому функционалу, но ограниченны одной задачей, а туристические запросы разнообразны, поэтому мы реализуем прогноз погоды в разных городах России с использованием возможности выбора языка.

Большинство ботов в России и стран СНГ имеют узкую специализацию, например: бот, ориентированный только на дальнобойщиков помогающий найти АЗС по следуемому маршруту; информация о стоимости аренды квартиры в выбранном городе. А в других странах Telegram-боты имеют более проработанные параметры поиска и поддержку разных языков, а также распознавание голосового запроса. В связи с этим было решено создать информационный Telegram-бот для путешественников по городам России, отвечающим современным запросам.

Целью научно-исследовательской работы является разработка Telegram-бота для путешественников по городам России.

Исходя из поставленной цели, были поставлены следующие задачи:

− анализ выбранной предметной области;

− выбор технологий и среды разработки;

− разработка чат-бота на платформе Telegram на языке Python.

Структура работы: данная работа состоит из введения, в которой описывается актуальность данной работы, цели и задачи. Имеет четыре главы, заключение, список использованной литературы и приложение.

1. **Организационная структура кафедры, направления научной и научно-исследовательской работы**
   1. **Современное состояние материально-технической базы кафедры**

В настоящее время кафедра вычислительной техники располагает парком персональных компьютеров из более 70-ти машин, имеет свой кафедральный сервер с автономным энергообеспечением для защиты имеющихся информационных материалов, а также 1-й сегмент суперкомпьютерной грид-системы с включением в нее терминальных классов «Тонкий клиент» (ауд 301, 302) и лабораторий (ауд. 304, 307, 320).

Все аудитории кафедры обеспечены мультимедийными проекторами, как стационарными, так и переносными, а аудитории 303, 318 – интерактивными досками. В дополнение к существующему оборудованию для занятий по дополнительному образованию (подготовка «Web-дизайнеров») кафедрой закуплено и используется специальное художественное оборудование в ауд. 313. Всего в настоящее время установлено оборудования на сумму, превышающую 21 млн. рублей.

В перспективе предстоит расширение лабораторных комплексов на базе «Grid System» и «Cisco Systems», что позволит значительно улучшить качество преподавания и расширить возможности научно-исследовательской работы.

Приказом №162 от 11 июня 2010 г. на кафедре ВТ образован научно-образовательный центр «Прикладной анализ данных». Основные его задачи: оказание различных научно-исследовательских и образовательных услуг, не предусмотренных университетскими планами научных работ и основными учебными планами направлений и специальностей, по которым ведется обучение в НГТУ. За время работы центра подготовлено 56 специалистов в области «Web-дизайн» и в сетевых специалистов «Cisco Systems».

По итогам выполнения ИОП «Высокие технологии» приказом по университету №205 от 2 марта 2009 г. на базе средств «Grid System» был создан центр коллективного пользования «Распределенные вычислительные системы». Основными направлениями его деятельности являются: предоставление доступа сотрудников, студентов и сторонних лиц к высокопроизводительным вычислительным ресурсам, развитие совместно с другими подразделения университета информационно-вычислительной среды и связь ее с глобальными грид-сетями, обучение и переобучение студентов и специалистов суперкомпьютерной технике, распределенным и параллельным информационным технологиям, проведение НИР в этом направлении.

Приказом №2129 от 27 декабря 2011 г. в рамках инновационного гранта на кафедре был создан «Информационно-аналитический центр информационных технологий и цифровых систем», основными задачами которого являются: обеспечение инновационных подразделений НГТУ и других внутренних и внешних потребителей профильной аналитической научно-технической информацией, касающейся мониторинга и анализа текущего состояния дел по тематике центра, прогноз ее развития, экспертиза инноваций в отрасли информационных технологий и цифровых систем, оказание помощи в участии в профильных платформах и образовательных услуг.

* 1. **Организация учебного процесса на кафедре**

Богатая материально-техническая база кафедры позволяет обеспечить студентов возможностями обучения разнородным средствам вычислительной техники и телекоммуникаций. Так, например, установленный сегмент грид-системы позволяет студентам обучаться и работать на типовых персональных компьютерах, рабочих станциях, как отдельных элементах, так и встроенных в грид-систему.

В рамках образовательной работы кафедра осуществляет следующую деятельность:

1. Обеспечение лабораторных работ, практических занятий, семинаров, лекций, презентаций по проблемам сетевых технологий и защите информации в рамках образовательных программ. В настоящее время существует 252 учебных курса.
2. Адаптация учебных и сертификационных материалов Cisco для образовательных целей.
3. Разработка учебно-методических комплексов для дисциплин, по которым в лабораториях будут проводиться занятия. В связи с появлением в университете двухуровневой системы подготовки выпускников (бакалавр - магистр), кафедра проводит большую работу по разработке соответствующих учебных планов, рабочих программ и учебно-методических комплексов по всем трем уровням обучения: бакалавриат, специалитет, магистратура.
4. Создание филиала академии Cisco Systems с целью подготовки сертифицированных специалистов в соответствии с требованиями уровня CCNA.
5. Организация академии Microsoft и создание международного центра сертификации VUE.
6. Подготовка кадров высшей квалификации. Центр «Прикладной анализ данных», созданный при кафедре в 2010 г., осуществляет повышение квалификации по программам: «Компьютерная графика и веб-дизайн» (240 часов), «Сетевой специалист, сертифицированный Cisco» (72 часа), «Средства и технологии презентации в образовательном процессе» (72 часа).
7. Переподготовка кадров. В 2003 г. на кафедре была открыта подготовка по 2-годничной программе дополнительного к высшему образованию «Специалист в области компьютерной графики и веб-дизайна», а с 2011 г. также «Системный инженер (специалист по эксплуатации аппаратно-программных комплексов, персональных ЭВМ и сетей на их основе)» (срок обучения 1,5 года), «Разработчик профессионально ориентированных компьютерных технологий» (2,5 года), «Преподаватель информатики» (1,5 года).

За период 10 лет кафедрой было опубликовано 16 монографий, 125 учебников и учебных пособий.

## **1.3 Организация НИР на кафедре**

НИР студентов подразделяются на:

а) учебно-исследовательскую работу студентов – работу, включаемую в учебный процесс;

б) собственно НИРС – работу, выполняемую во внеучебное время.

Научно-исследовательская работа студентов, включаемая в учебный процесс, осуществляется кафедрой в следующих формах:

а) выполнение лабораторных работ, домашних работ, курсовых и дипломных работ, содержащих элементы научных исследований;

б) введение элементов научного поиска в практические и семинарские занятия;

в) выполнение конкретных нетиповых заданий научно-исследовательского характера в период производственной и преддипломной практик;

г) ознакомление с теоретическими основами методики, постановки, организации и выполнения научных исследований, планирования и проведения научного эксперимента и обработки полученных данных;

д) участие в работе студенческих научных семинаров.

Учебная научно-исследовательская работа студентов (УИРС) является одним из важнейших средств повышения качества подготовки и воспитания специалистов с высшим образованием, обладающих навыками исследования и способных творчески применять в практической деятельности.

Научно-исследовательская работа студентов (НИРС), организуемая во внеучебное время, включает следующие формы:

а) участие в работе студенческих научных коллективов;

б) участие в работе проблемных научных групп на профилирующих (выпускающих) кафедрах;

в) участие в выполнении хоздоговорной тематики кафедры.

Формы и методы НИРС зависят от уровня подготовки студентов. На младших курсах преобладают такие формы НИРС как написание рефератов, выполнение расчетных работ, перевод литературы и др. На старших курсах – реальное курсовое и дипломное проектирование, постановка и модернизация лабораторных работ, участие студентов в подготовке и проведении научных экспериментов, выполнение хоздоговорных научно-исследовательских работ.

Первый договор на выполнение научно-исследовательской работы был заключен кафедрой с Сибирским филиалом Всесоюзного института механизации сельского хозяйства (СибВИМ). Целью работы являлась разработка электронного коррелятора, чем было положено начало научному направлению по разработке электронной аппаратуры для статистической обработки информации (научные руководители: Ю.К.Постоенко, В.В.Губарев).

В 1969 г. на кафедре сложилось научное направление по исследованию и построению распределенных вычислительных систем (научный руководитель - к.т.н., доцент В. И. Жиратков), а в 1974г. - направление по разработке специализированных вычислительных устройств для автоматизации научных исследований (научный руководитель - к.т.н., доцент В.И.Соболев).

# **2. Статистический анализ**

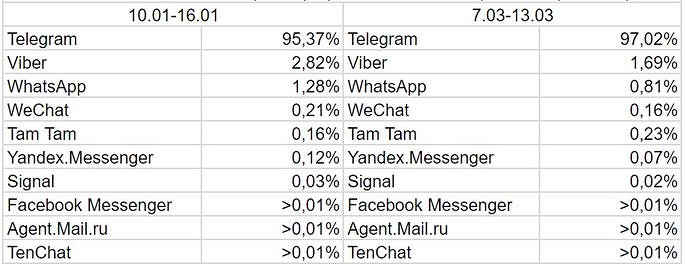
**2.1 Статистический анализ мессенджера Telegram**

В 2021 году Telegram стал самым быстрорастущим мессенджером в России: за год объём мобильного интернет-трафика в сервисе вырос более чем в два раза. К таким выводам пришли специалисты МегаФона на основе анализа данных пользователей [3].

Telegram в 2021 году впервые выбился в лидеры в 50 регионах страны. На декабрь 2021 года он опережал Whatsapp в Москве, Санкт-Петербурге, Самарской и Челябинской областях, Забайкальском крае, Чувашской Республике. Меньше всего Telegram пользовались в республиках Северного Кавказа, где лидером традиционно являлся WhatsApp, доля которого составляет 67%. За год совокупный объем мобильного интернет-трафика в мессенджерах вырос на 87%. Все больше абонентов используют их не только для отправки сообщений, но и для аудио и видео вызовов [7].

В 2022 году лидером среди мессенджеров стал Telegram, он обогнал по популярности лидера 2021 года WhatsApp. На втором месте Viber, затем WhatsApp. За рассматриваемый период Telegram увеличил свою долю в общем трафике на мессенджеры с 95,37% в январе 2022 г. До 97,02% в марте 2022 г. Остальные мессенджеры потеряли в доле общего трафика по сравнению с данными за январь 2022 г. Telegram становится таким популярным, не только как мессенджер, но и как информационный ресурс, где люди могут узнавать новости. Информацию о деятельности групп, каналов, записываться на прием к врачу и многое другое.

«После блокировки некоторых социальных сетей люди перешли на общение в мессенджерах. Также часть людей предпочитает получать информацию из тематических каналов в Telegram. Скорее всего, именно этим объясняется повышение трафика на мессенджеры в марте», — отметила Ирина Пюскюлян, директор по маркетингу Yota [2].



*Рис.1-Статистические данные использование интернет-трафика различными мессенджерами.*

Telegram – это кроссплатформенное приложение, разработанное на языке программирования C++, позволяющее обмениваться сообщениями и файлами большинства форматов [3].

Мессенджер использует специально разработанную серверную часть с закрытым кодом, работающим на серверах Германии и США. Telegram имеет ряд преимуществ [11]:

− приватность – все чаты зашифровываются, а сообщения уничтожаются через обозначенное время;

− быстрота – скорость доставки сообщений выше, чем у аналогов;

− распределение – сервера Telegram расположены по всему миру, что повышает отказоустойчивость;

− открытость – использование открытого протокола MTProto и API, бесплатных для всех;

− отсутствие ограничений на размер сообщений и вложенных файлов.

Основателями мессенджера Telegram являются братья Павел и Николай Дуровы. Павел является финансовой и идеологической опорой проекта, а Николай занят техническими аспектами. Особенностями данного мессенджера является специально разработанный протокол шифрования MTProto [7].



*Рис.2-Протокол шифрования.*

Мессенджер Telegram доступен для всех популярных платформ, например, Android OS, iOS, Windows Phone, Linux и другие. Так же в большинстве версия для разных операционных систем доступна функция выбора языка [1].

В последнее время идет Telegram ведет активную разработку своей блокчейн-платформы Telegram Open Network и криптовалюты Gram на ее основе.

* 1. **Статистический анализ чат-ботов и поисковых запросов пользователей**

В последнее время в мессенджерах набирают популярность такие сервисы как чат-боты. Сам термин «чат-бот» был придуман Майклом Молдина в 1994 г. для описания разговорных программ. Изучив современное состояние использования чат-ботов в мессенджерах, можно прийти к выводу, что чат-боты являются универсальными средствами, способными к решению разнообразных задач – от общения до развлечений, от предоставления медицинской консультации до заказа товаров и услуг посредством специализированных прикладных решений, от распознавания эмоций до решения сложных консалтинговых задач в службах поддержки клиентоориентированных информационных систем. В любом случае, в независимости от платформы, чат-бот – это прикладная программа, которая, получая информацию от пользователя, формирует корректные, логически обоснованные ответы [2].

На сегодняшний момент существует огромное разнообразие чат-ботов для путешествий. Некоторые из возможных вариантов представлены ниже [4]:

- новостные и информационные чат-боты по городам России;

- игровые или квестовые чат-боты для некоторых городов России;

- информационные чат-боты по городам России;

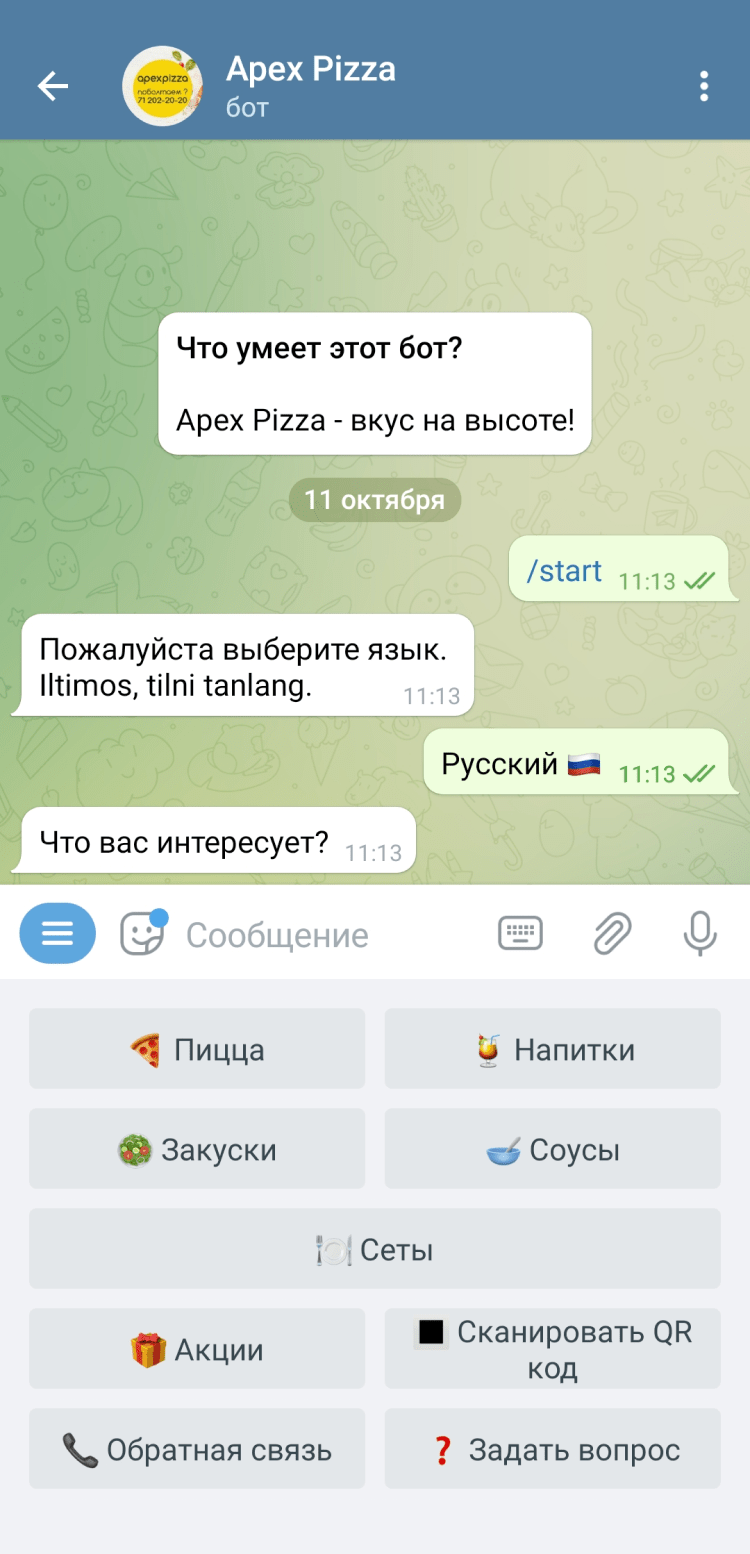
- рекламные чат-боты;

- чат-боты для доставок, магазинов, сервисов (такси «Максим», пиццерия «Папа Джонс», бронирования поездов и самолетов «Туту.рф»);

− чат-боты для консультации и поддержке клиентов (банк «Точка», каршеринг «UrentCar») и другие.

Стоит также отметить актуальные запросы путешественников в чат-ботах: направления и отели, гостиницы, хостелы и прочее — их упоминают в 75% всех запросов про путешествия. Следующие по популярности темы — достопримечательности и развлечения (17% запросов), туристические компании (14%), транспорт (11%). Реже упоминают конкретные варианты отдыха (6%), спрашивают про погоду (3%) и оформление виз (2%) [1].

Судя по формулировкам поисковых запросов, с прошлого года на туристическом рынке произошли два важных изменения: во-первых, люди стали чаще самостоятельно организовывать путешествия, во-вторых, начали искать более дешёвые варианты размещения [5].



*Рис. 3-Пример чат-бота доставки еды.*

В последнее время большие возможности открывают информационные технологии, связанные с чат-ботами. Они настолько прочно вошли в жизнь людей, что применяются во всех сферах человеческой деятельности, и с каждым днем их роль все больше увеличивается. В первую очередь это объясняется тем, что основную часть своего времени люди проводят за смартфоном в email-клиентах и мессенджерах.

Боты – это программы, которые выполняют различные задачи для пользователя, находящегося в мессенджере. Бот выглядит как обычный чат, однако общение происходит не с человеком, а с программой, которая может принять заказ на вызов машины, если это – бот такси, или прислать свежие статьи, если это – новостной бот, или заказать доставку еды на дом, если это – ресторанный бот [12].

Не так давно чат-боты обрели большую популярность: во-первых, чат-боты – это «платформы» для решения бизнес-задач; во-вторых, чат-бот – это приложение, которое поддерживает диалог с пользователем, выбирая ответы из базы данных: вы спрашиваете, где пообедать и тут же получаете мгновенный ответ. Кроме того, чат-боты выполняют множество полезных функций по исполнению рутинных операций, поиску информации, объединению данных, работе с клиентурой.

Чат-бот как виртуальный собеседник имеет базу знаний, которая представляет собой наборы возможных вопросов пользователя и соответствующих им ответов. Наиболее распространенными вариантами для получения нужного ответа являются ключевые слова, совпадение фразы, совпадение контекста. Всегда существуют какие-то простые и легкие по выполнению дела, на которые не хочется тратить время. Тут на помощь всегда могут прийти чат-боты [13].

Большинство ботов в России и стран СНГ имеют узкую специализацию, например: бот, ориентированный только на дальнобойщиков помогающий найти АЗС по следуемому маршруту; информация о стоимости аренды квартиры в выбранном городе, бронирование номеров, заказ такси, запись к врачу, доставка еды в выбранном городе. А в других странах Telegram-боты имеют более проработанные параметры поиска и поддержку разных языков, а также распознавание голосового запроса. В связи с этим было решено создать информационный Telegram-бот прогноза погоды для путешественников по городам России, отвечающим современным запросам.

# **3. Выбор технологий и среды разработки**

## **3.1 Выбранные технологии разработки**

### **3.1.1 Программная платформа «Node.js»**

Node.js – программная платформа на движке V8, который транслирует JavaScript-код в машинный. Этот движок превращает узкоспециализированный язык JavaScript в язык общего назначения.

В Node.js появилась возможность взаимодействовать с устройствами ввода-вывода через свой API, написанный на C++, использовать сторонние библиотеки, созданные при помощи разных языков программирования, вызывая их из JavaScript-кода. Node.js используется преимущественно как back-end язык программирования, выполняя роль сервера, но также есть возможность разработки оконных десктопных приложений при помощи специальных фреймворков, например, NW.js, AppJS или Electron [6].

Из особенностей можно отметить возможность использовать язык Node.js для разработки приложений в сфере «интернета вещей» (Internet of Things, IoT). В последнее время мы все чаще встречаемся с смарт-браслетами, колонками и любой другой «умной» техникой, которая в теории может быть запрограммирована на Node.js. История Node.js началась с разработки нового JavaScript-движка V8 в датском отделении компании Google. Ведущий разработчик Lars Bak выделил основные проблемы, которые должны были решены в движке, это производительность и масштабируемость. Первая лабораторная версия была готова 3 июля 2008 года, а позже был представлен первый браузер Chromium в публичный релиз которого входил новый движок. Помимо браузеров на основе Chromium, например, Chrome, Comodo Dragon, Opera, Амиго и другие, существует ряд других продуктов, использующих V8 [10]:

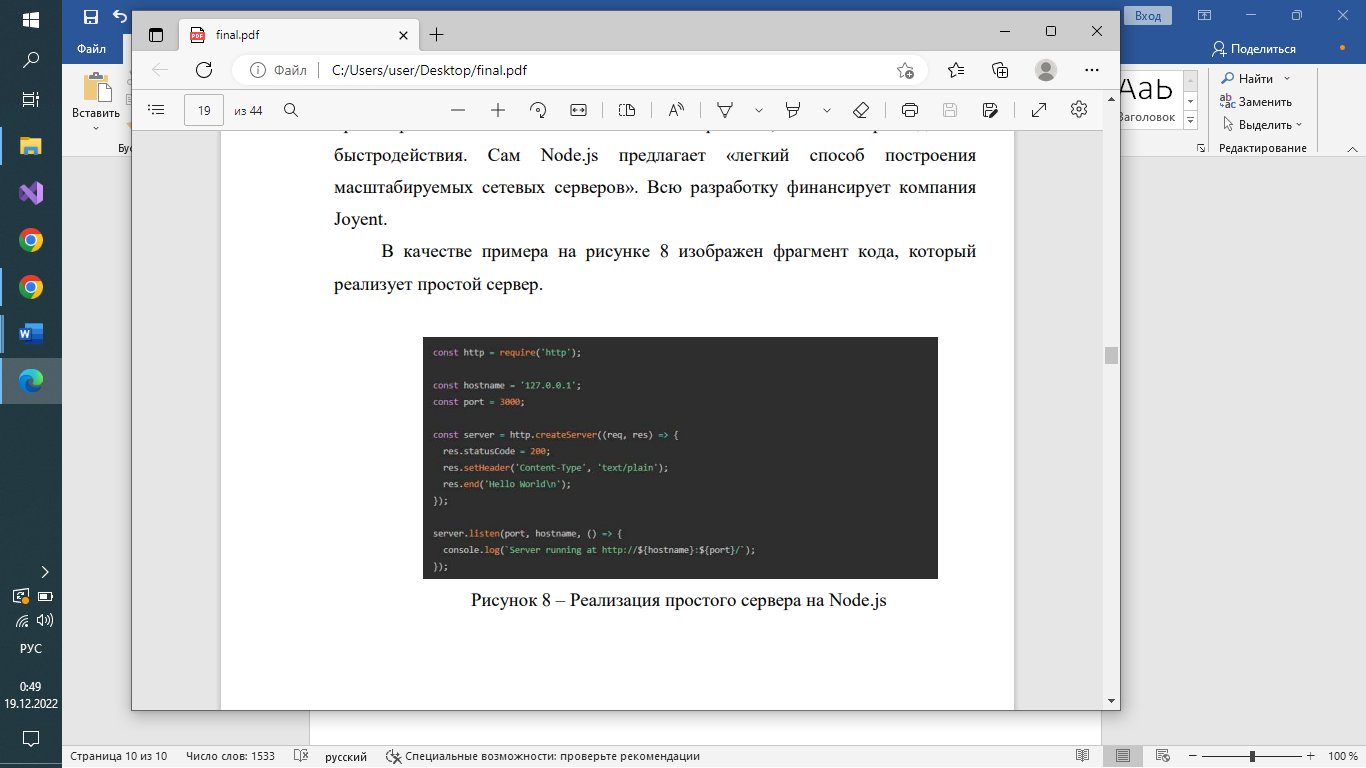
− Maxthon – веб-браузер со встроенным блокиратором рекламы, использующий два движка рендеринга: WebKit и Trident;

− стандартный браузер Android; − операционная система Android;

− HP webOS – операционная система от Hewlett-Packard для коммуникаторов, нетбуков и планшетов;

− Google Chrome OS – операционная система от Google на базе проекта Chromium, ориентированная на облачные сервисы.

В 2009 году программист Rayn Dahl, после двух лет экспериментов над созданием серверных веб-компонентов, представляет Node.js. В ходе исследований разработчик пришел к выводу, что вместо традиционной модели параллелизма на основе потоков следует обратиться к событийноориентированным системам из-за их простоты, низких расходов и быстродействия. Сам Node.js предлагает «легкий способ построения масштабируемых сетевых серверов». Всю разработку финансирует компания Joyent.

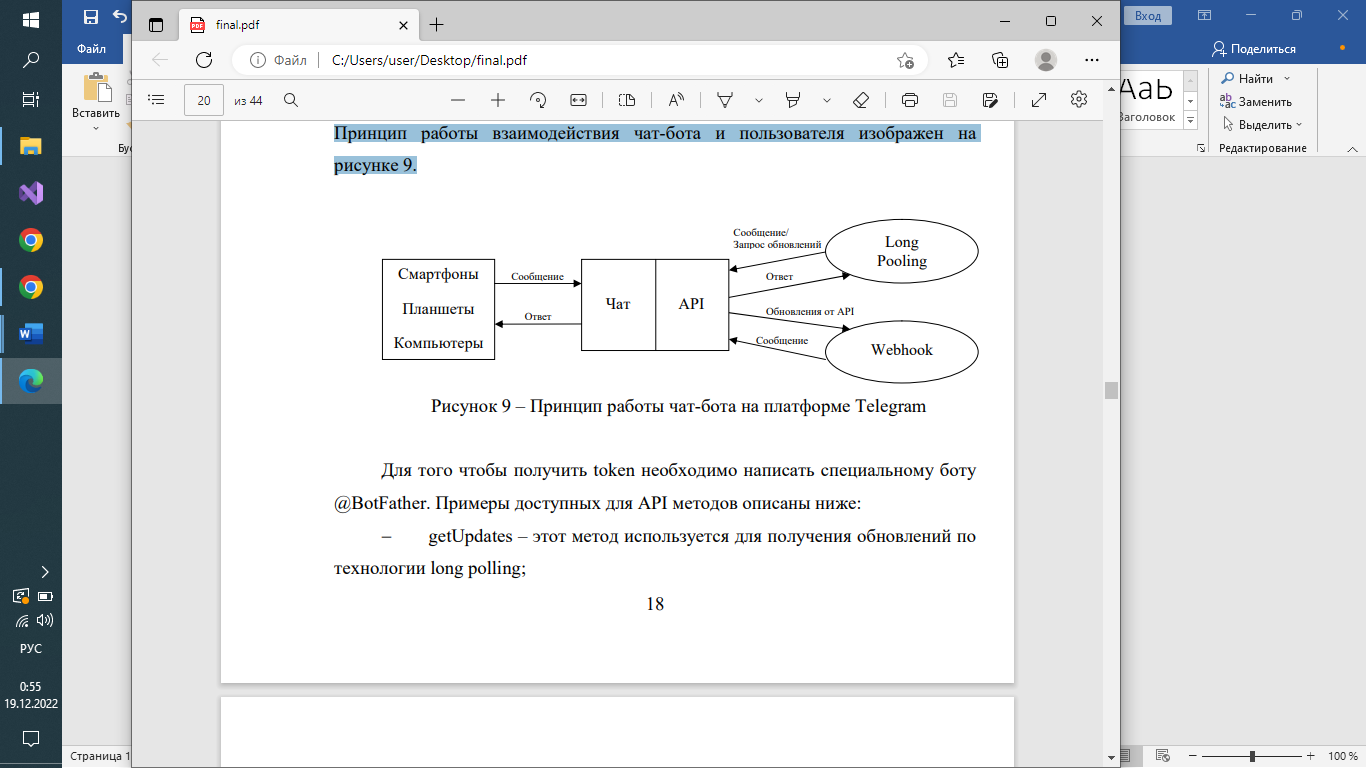


*Рис.4-Пример кода на Node.js для создания простого сервера.*

### **3.1.2 Telegram Bot API**

Bot API представляет собой HTTP-интерфейс для работы с ботами в Telegram. Каждый бот – это специальный аккаунт, созданный для автоматического обрабатывания и отправления сообщений [8].

Существует два противоположных по логике способа получения обновлений от бота: − long pulling – приложение автоматически опрашивает сервера Telegram на наличие каких-либо обновлений для бота. Сервера Telegram сами оповещают приложение на сервере как только появятся какие-либо обновления. Входящие обновления будут храниться на сервере до тех пор, пока их не обработают, но не дольше 24 часов. Независимо от способа получения обновлений, в ответ отправляется объект Update, сериализованный в JSON. Все запросы к Telegram Bot API должны осуществляться через HTTPS в следующем виде: https://api.telegram.org/bot/НАЗВАНИЕ\_МЕТОДА. Принцип работы взаимодействия чат-бота и пользователя можно изобразить следующим образом [11]:



*Рис.5-Механизм взаимодействия чат-бота и пользователя.*

Для того чтобы получить token необходимо написать специальному боту @BotFather. Примеры доступных для API методов описаны ниже [9]:

− getUpdates – этот метод используется для получения обновлений по технологии long polling;

− setWebhook – метод привязывает к боту url домена, где содержится запущенный бот;

− sendMessage – метод отправляет текстовое сообщение в клиент Telegram;

− sendLocation – метод отправляет сообщение с координатами в клиент Telegram;

− getFile – метод возвращает загруженный файл по его имени и др.

Допускаются POST и GET запросы. Для передачи параметров в Bot API есть 4 способа:

− Запрос в URL − application/x-www-form-urlencoded;

− application/json (не приемлем для загрузки файлов);

− multipart/form-data (для загрузки файлов).

### **3.1.3 Облачная платформа «Heroku»**

Heroku – облачная PaaS-платформа, поддерживающая ряд языков программирования. С 2010 года является дочерней компанией Salesforce. Heroku, одна из первых облачных платформ, появилась в июне 2007 года и изначально поддерживала только язык программирования Ruby, но на данный момент список поддерживаемых языков также включает в себя Java, Node.js, Scala, Clojure, Python, Go и PHP [13].

На серверах Heroku используются операционные системы Debian или Ubuntu. Heroku основали в 2007 году Джеймс Линденбаун, Адам Уиггинс и Орион Генри для поддержки проектов, основанных на Rack. 8 декабря 2010 года Heroku была куплена компанией Salesforc, сделав её своей дочерней компанией. В 2011 году Ю. Мацумото, создатель языка программирования Ruby, пришёл в компанию в роли ведущего инженера. В этом же месяце была внедрена поддержка Node.js и Clojure. Также платформа предоставляет поддержку таких СУБД, как Cloudant, Membase, MongoDB и Redis, помимо PostgreSQL. Приложения, работающие на платформе Heroku, используют также DNS-сервер (обычно приложения имеют доменное имя вида «имя\_приложения.herokuapp.com»). Для всех приложений выделяются несколько независимых виртуальных процессов, названные «dynos». Они распределены по специальной виртуальной сетке, которая состоит из нескольких серверов. На Heroku имеется собственную систему контроля версий.

### **3.2 Среда разработки**

### **3.2.1** **PyCharm Community Edition**

PyCharm — это кроссплатформенная интегрированная среда разработки для языка программирования Python, разработанная компанией JetBrains на основе IntelliJ IDEA. Предоставляет пользователю комплекс средств для графических отладчиков и работы с кодом. Продукт доступен в двух версиях: PyCharm Community Edition - бесплатная версия, находится под лицензией Apache License, и PyCharm Professional Edition - расширенная версия продукта, обладающая дополнительной функциональностью, является проприетарным ПО.

Основные возможности [10]:

* Отладка кода при помощи PyDev;
* Рефакторинг кода;
* Поддержка Git, SVN, Mercurial и других систем контроля версиями;
* Автодополнение кода.

Пользователи могут сами писать свои плагины, тем самым расширять возможности PyCharm. Некоторые плагины из других JetBrains IDE могут работать с PyCharm. Существует более тысячи плагинов, совместимых с PyCharm.PyCharm был выпущен на рынок интегрированных сред разработки для создания конкуренции с PyDev (однако, на данный момент PyCharm использует PyDev для отладки кода) и более распространённой среды разработки Komodo IDE. Бета-версия была выпущена в июле 2010 года, версия 1.0 была выпущена тремя месяцами позже. Версия 2.0 вышла 13 декабря 2011 года. Версия 3.0 была выпущена 24 сентября 2013 года.

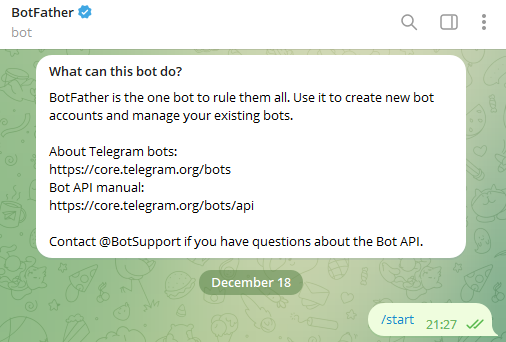
PyCharm Community Edition, бесплатная версия с открытым исходным кодом, была опубликована 22 октября 2013 года.

В марте 2016 года JetBrains перешла на подписную модель лицензирования, а вместе с этим изменилась и нумерация версий. Мы будем работать на PyCharm Community Edition 2022.3.

# **4. Разработка Telegram-бота**

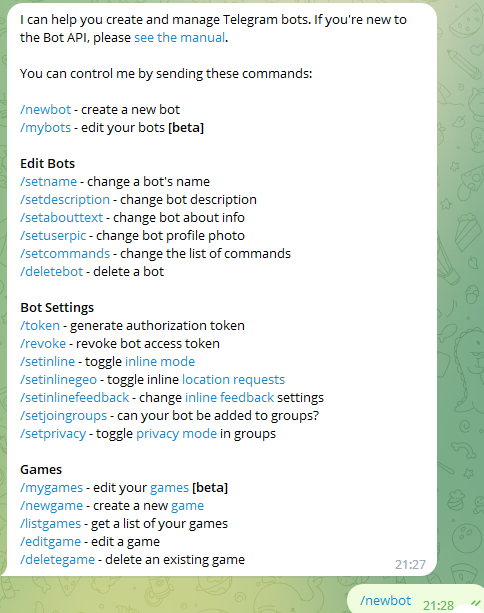
## **4.1 Разработка чат-бота для Telegram Bot API**

Первым шагом разработки приложения нужно начать диалог с ботом «BotFather» для этого нужно ввести команду в чат «/start» как на Рис.6.



*Рис.6 - Команда выполнения «/start».*

Следующий шаг – это перечисление команд, который может выполнять данный Telegram-бот, Рис.7.



*Рис.7-список команд*

Приведем пример некоторых команд чат-бота. Для установки дополнительных параметров, таких как иконка чат-бота, приветственное сообщение, описание чат-бота, а также удаление имеющихся чат-боты, существуют следующие команды.

Таблица 1 – Доступные команды для изменения чат-ботов

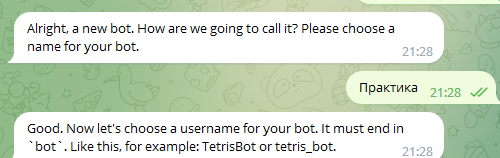
|  |  |
| --- | --- |
| Команда | Описание |
| /setdescription | Присваивает текст, который будет отображаться при первом открытии чат-бота |
| /setname | Изменяет существующее имя |
| /setabouttext | Присваивает текст в поле «О чат-боте» |
| /setcommands | Позволяет создать список доступных команд |
| /setuserpic | Присваивает выбранную картинку |
| /deletebot | Удаляет выбранного чат-бота |

Помимо команд для изменения основных параметров чат-бота существует ряд команд, которые позволяют выводить неизменяемые параметры (токен), а также присваивать значения.

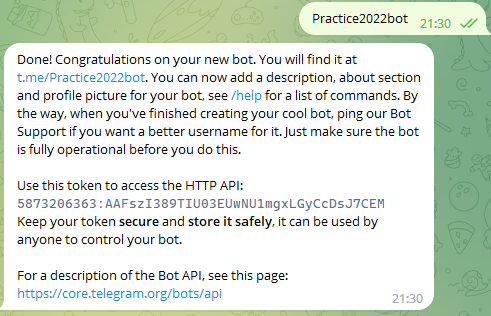
Таблица 2. – Доступные команды для дополнительной настройки чат-бота

|  |  |
| --- | --- |
| Команда | Описание |
| /token | Возвращает полученный ранее токен у выбранного бота. |
| /revoke | Аннулирует токен доступа к боту. |
| /setinline | Включает или выключает возможность вызывать бота из других чатов. |
| /setinlinegeo | Включает или выключает возможность передавать местоположение бота из другого чата. |
| /setinlinefeedback | Позволяет получать информацию о количестве выбранных пользователями команд. |
| /setjoingroup | Определяет может ли быть добавлен в групповые диалоги бот. |
| /setprivacy | Включает режим конфиденциальности. В этом режиме бот получает, обрабатывает и отсылает обратно информацию отдельно для каждого пользователя в чате. |

Следующим этапом является регистрация бота. Регистрация начинается с команды «/newbot», после чего предлагается ввести название чат-бота с обязательным условием: в конце название должно быть указано «Bot» или «bot». Если все условия были удовлетворены, то «BotFather» выдает токен (специальный набор символов для доступа к HTTP API Telegram Bot) и URL-адрес для доступа к чат-боту. Присвоим Telegram-боту имя Рис.8 и никнейм бота Рис.9.



*Рис.8 - Имя бота*.



*Рис.9 – Никнейм бота.*

После настроек на стороне Telegram и получении токена можно приступить к разработке программной части чат-бота.

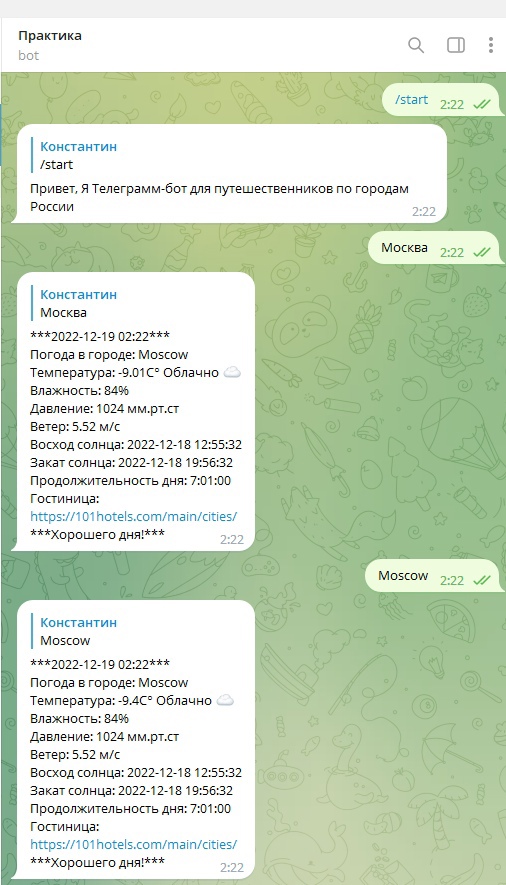
Следующий шаг, это регистрация на сайте <https://openweathermap.org/api>, создаем токен по которому идёт подключение к сайту. На этом сайте чат-бот берет данные о погоде о различных городах России.

Для программирования чат-бота нам необходимо установить среду разработки PyCharm Community Edition 2022.3. После этого, необходимо установить язык программирования Python версии 3.11.1. Также нам понадобиться установить дополнительные пакеты angiogram и pyTelegramBotAPI.

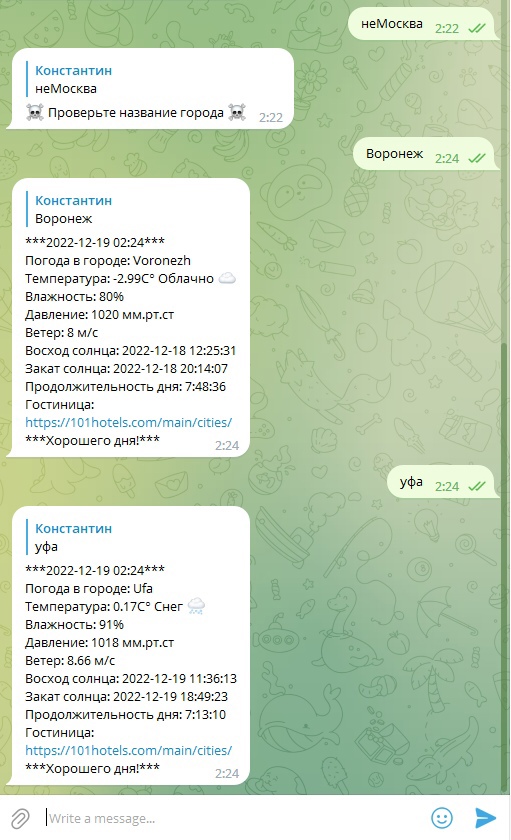
Весь листинг кода будет прикреплен в приложении.

## **4.2 Тестирование Telegram-бота**

Таким образом, пользователь с помощью нашего Telegram-бота вводит название города, который он планирует посетить, в случае корректного ввода и ему выдается информация о дате и времени запроса, название города, температура, типы осадков, уровень влажности, давление, скорость ветра, время восхода и заката солнца, продолжительность светового дня и ссылка на сайт бронирование гостиницы, для удобства путешественников по России Рис.10. Также он выполняет запросы как на английском, так и на русском языке. Распознает запросы с маленькой и большой буквы. В случае некорректного ввода названия выводит сообщение: «Проверьте название города», при этом работоспособность программы не нарушена Рис.11.



*Рис.10 – работа чат-бота.*



*Рис.11 – работоспособность чат-бота после некорректного ввода.*

# **Заключение**

В настоящее время популярность мессенджеров как средств общения неизменно растет. Так же стоит отметить рост популярности такого вида программных продуктов как чат-боты, которые работают на платформах мессенджеров.

Пользователям не придется скачивать множество приложений для решения узконаправленных задач, потому что достаточно иметь лишь мессенджер и необходимый набор чат-ботов, которые не занимают место в памяти смартфона.

В рамках производственной практики: научно-исследовательской работы были выполнены поставленные задачи. Во-первых, были изучены мессенджеры. Согласно теме индивидуального задания был выбран мессенджер Telegram как самый удобный и доступный в плане документации Telegram Bot API.

Во-вторых, были изучены имеющиеся аналоги чат-бота на платформе Telegram, поисковые запросы путешественников России. В рамках последней выполненной задачи были выбраны технологии и среда для разработки чат-бота среди которых Python 3.11.1, PyCharm Community Edition 2022.3. На последнем этапе мы разработали и протестировали Telegram-бота для путешественников по городам России. Полный код находится в разделе «Приложение».

Задачи, которые может решать бот:

- ему выдается информация о дате и времени запроса, название города, температура, типы осадков, уровень влажности, давление, скорость ветра, время восхода и заката солнца, продолжительность светового дня;

- ссылки на сайт бронирование гостиницы;

- он выполняет запросы как на английском, так и на русском языке, распознает запросы с маленькой и большой буквы. В случае некорректного ввода названия выводит сообщение: «Проверьте название города», при этом работоспособность программы не нарушена.

# **Список использованной литературы**

1) Messina, C. 2016 will be the year of conversational commerce [Электронный ресурс] / C. Messina // Medium. – 2016. – Режим доступа: <https://medium.com/chris-messina/2016-will-be-the-year-of-conversational-commerce-1586e85e3991>

2) Telegram Bot API [Электронный ресурс]: Telegram Documents. – Режим доступа: <https://core.telegram.org/bots/api>

3) Аванесян Н. Л., Telegram, как пример мессенджера: возможности и перспективы развития. [Электронный ресурс] / Н. Л. Аванесян // Научный потенциал XXI века. – 2017. – Режим доступа: <https://elibrary.ru/download/elibrary_29653726_34734017.pdf>

4) Агальцов, В. Базы данных: В 2-х книгах. Книга 2: Распределенные и удаленные базы данных : Учебник / В. Агальцов – Москва : Форум, 2018. – 271 с.

5) Архаков, Д. NodeJS: Делаем кнопки в Telegram API (inlinekeyboards) [Электронный ресурс] / Д. Архаков // Блог о программировании. – 2016. – Режим доступа: <https://archakov.im/post/nodejs-make-buttons-on-telegram-api.html>

6) Иванов А. Д., Чат-бот в Telegram и ВКонтакте, как новый канал распространения новостей. [Электронный ресурс] / А. Д. Иванов // Волжский университет имени В.Н. Татищева. – 2016. – №3 – с. 126-132. – Режим доступа: <https://elibrary.ru/download/elibrary_26673675_34058358.pdf>

7) Козлов А. А., Телеграм-бот как простой и удобный способ получения информации [Электронный ресурс] / А. А. Козлов, А. В. Батищев // Территория науки. – 2017. – №5. – с. 55-64. – Режим доступа: <https://cyberleninka.ru/article/v/telegram-bot-kak-prostoy-i-udobnyy-sposob-polucheniya-informatsii>

8) Матвеева Н. Ю., Технологии создания и применения чат-ботов [Электронный ресурс] / Н. Ю. Матвеева, А. В Золотарюк . // Научные записки молодых исследователей. – 2018. – №1. – с. 28-30. – Режим доступа: <https://cyberleninka.ru/article/v/tehnologii-sozdaniya-i-primeneniya-chat-botov>

9) Официальный сайт Node.JS [Электронный ресурс]: About – Режим доступа: <https://www.nodejs.org/>.

10) Официальный сайт Telegtam [Электронный ресурс]: API – Режим доступа: https://core.telegram.org/api.

11) Рейтинг месседжеров 2018 [Электронный ресурс]: Сервис сравнения и выбора приложений и сервисов в сфере IT. – 2018. – Режим доступа: <https://coba.tools/compilation/reiting-messendzherov-2018>

12) Холмс, С. Стек MEAN. Mongo, Express, Angular, Node. : Учебное пособие / С. Холмс – Санкт-Петербург : Питер Пресс, 2017. – 496 с.

13) Хэррон, Д. Node.js. Разработка серверных веб-приложений на JavaScript. : Учебное пособие / Д. Хэррон – Москва : ДМК Пресс, 2014. – 144 с.

# **Приложение**

Приложение 1. Листинг программы:

#файл main.ry

import requests

import datetime

from pprint import pprint

from config import open\_weather\_token

def get\_weather(city, open\_weather\_token):

code\_to\_smile = {

"Clear": "Ясно \U00002600",

"Clouds": "Облачно \U00002601",

"Rain": "Дождь \U00002614",

"Drizzle": "Дождь \U00002614",

"Thunderstorm": "Гроза \U000026A1",

"Snow": "Снег \U0001F328",

"Mist": "Туман \U0001F32B"

}

try:

r = requests.get(

f"http://api.openweathermap.org/data/2.5/weather?q={city}&appid={open\_weather\_token}&units=metric"

)

data = r.json()

pprint(data)

city = data["name"]

cur\_weather = data["main"]["temp"]

weather\_description = data["weather"][0]["main"]

if weather\_description in code\_to\_smile:

wd = code\_to\_smile[weather\_description]

else:

wd = "Посмотри в окно, не пойму что там за погода!"

humidity = data["main"]["humidity"]

pressure = data["main"]["pressure"]

wind = data["wind"]["speed"]

sunrise\_timestamp = datetime.datetime.fromtimestamp(data["sys"]["sunrise"])

sunset\_timestamp = datetime.datetime.fromtimestamp(data["sys"]["sunset"])

length\_of\_the\_day = datetime.datetime.fromtimestamp(data["sys"]["sunset"]) - datetime.datetime.fromtimestamp(

data["sys"]["sunrise"])

print(f"\*\*\*{datetime.datetime.now().strftime('%Y-%m-%d %H:%M')}\*\*\*\n"

f"Погода в городе: {city}\nТемпература: {cur\_weather}C° {wd}\n"

f"Влажность: {humidity}%\nДавление: {pressure} мм.рт.ст\nВетер: {wind} м/с\n"

f"Восход солнца: {sunrise\_timestamp}\nЗакат солнца: {sunset\_timestamp}\nПродолжительность дня: {length\_of\_the\_day}\n"

f"Хорошего дня!"

)

except Exception as ex:

print(ex)

print("Проверьте название города")

def main():

city = input("Введите город: ")

get\_weather(city, open\_weather\_token)

if \_\_name\_\_ == '\_\_main\_\_':

main()

#файл config.ry

open\_weather\_token = "38b85901140fd580a0baccf2d04ec976"

tg\_bot\_token = '5873206363:AAFszI389TIU03EUwNU1mgxLGyCcDsJ7CEM'

#файл test.ry

smile = "\U0001F600"

print(smile)

#файл main\_weather\_tg\_bot.py

import requests

import datetime

from config import tg\_bot\_token, open\_weather\_token

from aiogram import Bot, types

from aiogram.dispatcher import Dispatcher

from aiogram.utils import executor

bot = Bot(token=tg\_bot\_token)

dp = Dispatcher(bot)

@dp.message\_handler(commands=["start"])

async def start\_command(message: types.Message):

await message.reply("Привет, Я Телеграмм-бот для путешественников по городам России")

@dp.message\_handler()

async def get\_weather(message: types.Message):

code\_to\_smile = {

"Clear": "Ясно \U00002600",

"Clouds": "Облачно \U00002601",

"Rain": "Дождь \U00002614",

"Drizzle": "Дождь \U00002614",

"Thunderstorm": "Гроза \U000026A1",

"Snow": "Снег \U0001F328",

"Mist": "Туман \U0001F32B"

}

try:

r = requests.get(

f"http://api.openweathermap.org/data/2.5/weather?q={message.text}&appid={open\_weather\_token}&units=metric"

)

data = r.json()

city = data["name"]

cur\_weather = data["main"]["temp"]

weather\_description = data["weather"][0]["main"]

if weather\_description in code\_to\_smile:

wd = code\_to\_smile[weather\_description]

else:

wd = "Посмотри в окно, не пойму что там за погода!"

humidity = data["main"]["humidity"]

pressure = data["main"]["pressure"]

wind = data["wind"]["speed"]

sunrise\_timestamp = datetime.datetime.fromtimestamp(data["sys"]["sunrise"])

sunset\_timestamp = datetime.datetime.fromtimestamp(data["sys"]["sunset"])

length\_of\_the\_day = datetime.datetime.fromtimestamp(data["sys"]["sunset"]) - datetime.datetime.fromtimestamp(

data["sys"]["sunrise"])

await message.reply(f"\*\*\*{datetime.datetime.now().strftime('%Y-%m-%d %H:%M')}\*\*\*\n"

f"Погода в городе: {city}\nТемпература: {cur\_weather}C° {wd}\n"

f"Влажность: {humidity}%\nДавление: {pressure} мм.рт.ст\nВетер: {wind} м/с\n"

f"Восход солнца: {sunrise\_timestamp}\nЗакат солнца: {sunset\_timestamp}\nПродолжительность дня: {length\_of\_the\_day}\n"

"Гостиница:\nhttps://101hotels.com/main/cities/"

f"\n\*\*\*Хорошего дня!\*\*\*"

)

except:

await message.reply("\U00002620 Проверьте название города \U00002620")

if \_\_name\_\_ == '\_\_main\_\_':

executor.start\_polling(dp)

Приложение 2. Перечень прилагаемых документов:

1. Индивидуальное задание практики;
2. Аннотация;
3. Дневник по практике;
4. Отчет по практике.

Отчет имеет следующую структуру:

Введение

1. Организационная структура кафедры, направления научной и научно-исследовательской работы

1.1 Современное состояние материально-технической базы кафедры

1.2 Организация учебного процесса на кафедре

1.3 Организация НИР на кафедре

2. Статистический анализ

2.1 Статистический анализ мессенджера Telegram

2.2 Статистический анализ чат-ботов и поисковых запросов пользователей

3. Выбор технологий и среды разработки

3.1 Выбранные технологии разработки

3.1.1 Программная платформа «Node.js»

3.1.2 Telegram Bot API

3.1.3 Облачная платформа «Heroku»

3.2 Среда разработки

3.2.1 PyCharm Community Edition

4. Разработка Telegram-бота

4.1 Разработка чат-бота для Telegram Bot API

4.2 Тестирование Telegram-бота

Заключение

Список использованной литературы

Приложение

Приложение 3. Практическая значимость работы и дальнейшие возможности разработки:

Это первый в России работающий Telegram-бот, который может как на русском, так и на английском языке распознавать запросы города в России, находить актуальный прогноз погоды на сегодняшней день, описывать подробную характеристику осадков, влажности, ветра, длину светового дня и т.д. Также, он к каждому запросу городу, он отправляет актуальную ссылку на поиск ночлега для Российского путешественника. Также он может распознавать вопросы как с большой, так и с маленькой буквы. При неверном вводе, бот просит проверить корректность ввода названия города.

В дальнейшем возможности бота можно расширить, показывая ссылки на достопримечательности того города, в который собирается путешественник, ссылки на билеты как авиатранспорта, так ж/д транспорта. Telegram-бот можно привязывать к различным базам данных, сделать широкий рабочий интерфейс меню, возможных команд бота и т.д.