МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ

ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ им. А. И. ГЕРЦЕНА»



Основная профессиональная образовательная программа

Направление подготовки 09.03.01 Информатика и вычислительная техника

Направленность (профиль) «Технологии разработки программного обеспечения»

форма обучения – очная

**Выпускная квалификационная работа**

Разработка туристического программного веб-агента

Обучающейся 4 курса

Слуцкой Екатерины Андреевны

Научный руководитель:

д. п. н., проф., зав. каф. КТиЭО

Власова Елена Зотиковна

Санкт-Петербург

2019

Оглавление

[ВВЕДЕНИЕ](#_heading=h.tyjcwt) **3**

[ГЛАВА 1](#_heading=h.n6tfpr1qb1q3) **6**

[1.1. Выбор платформы](#_heading=h.1t3h5sf) 6

[1.2. Выбор функционала](#_heading=h.4d34og8) 8

[1.3. Выбор инструментов](#_heading=h.2s8eyo1) 10

[1.4. Обзор существующих решений](#_heading=h.17dp8vu) 14

[1.5. Вывод к первой главе](#_heading=h.3rdcrjn) 17

[ГЛАВА 2](#_heading=h.44sinio) **18**

[2.1. Модель](#_heading=h.2jxsxqh) 18

[2.2. Этапы разработки](#_heading=h.z337ya) 20

[2.3. Подготовка к разработке](#_heading=h.3j2qqm3) 20

[2.4. Написание кода](#_heading=h.4i7ojhp) 24

[2.5. Обработка исключений](#_heading=h.2xcytpi) 32

[2.6. Перспективы развития](#_heading=h.1ci93xb) 34

[2.7. Вывод ко второй главе](#_heading=h.3whwml4) 34

[ЗАКЛЮЧЕНИЕ](#_heading=h.2bn6wsx) **35**

[СПИСОК ЛИТЕРАТУРЫ](#_heading=h.qsh70q) **36**

# 

# 

# 

# **ВВЕДЕНИЕ**

Современный мир невозможно представить себе без интернета. По данным Internet World Stats его используют более 3,8 миллиардов пользователей или более половины всех жителей планеты. Степень проникновения интернета во все сферы жизни человека очевидна для любого образованного человека. Люди используют его дома, на работе или учебе, и, конечно, в путешествиях: интересуются достопримечательностями, курсом валют, погодой, отелями, транспортом, мероприятиями. Эту информацию они получают в информационных центрах, у других людей через социальные сети и из различных интернет-сервисов. Особой популярностью пользуются именно веб-ресурсы, за счет своей полноты и доступности. Основная идея данной работы заключается в том, чтобы интегрировать различные функции, которые могут пригодиться в путешествии, в одно место, к которому пользователь сможет получить доступ находясь в любой точке мира, имея при себе лишь устройство с выходом в сеть интернет.

Безусловно, все функции, необходимые в путешествии, уже не раз были реализованы в различных вариациях, но для того, чтобы ими воспользоваться, необходимо загружать приложения из всемирной сети или заходить на сайты, что в свою очередь не всегда удобно из-за необходимости выделения дополнительной памяти и расхода трафика не смотря на то, что в путешествиях это весьма нежелательно. Поэтому для реализации сформулированной выше задачи в качестве платформы для интеграции функций был выбран программный веб-агент, который отправляет пользователю только необходимую информацию.

Программный агент — это программа, которая вступает в отношение посредничества с пользователем или другой программой. То есть данная программа будет получать для пользователя необходимую информацию и возвращать её пользователю в удобном формате. Программные агенты могут работать на базе уже установленного приложения, что существенно экономит память в устройстве. Достаточно скачать одно приложение и подключить на нем несколько программных агентов, а не скачивать для каждой цели свое приложение.

Целевой аудиторией данного продукта являются молодые люди в возрасте от 18 до 30 лет, которые активно используют мобильные устройства и много путешествуют. У них всегда под рукой смартфон, и они предпочитают с его помощью получать всю необходимую информацию, так как для них это наиболее привычный способ.

Актуальность данной работы заключается в том, что на данный момент представлено множество различных сервисов, которые помогают пользователю в поездке, но они все находятся в абсолютно разных местах, в этой работе предложен вариант интегрирования их в одно место для облегчения пользователю дальнейшей работы с ними. Кроме того решение, предложенное в данной работе, позволит пользователю получать всю информацию в формате текстовых сообщений, что является наиболее привычным и удобным способом для современного человека, и экономит трафик, а это, как было сказано выше, в путешествии очень важно.

Цель выпускной квалификационной работы — разработка программного веб-агента, выполняющего функции туристического ассистента.

Для достижения указанной цели были определены следующие задачи.

1. Выбрать платформу для разработки программного агента. Так как одна из целей разработки — экономия памяти в устройстве, было решено разрабатывать агента на базе уже существующего приложения.
2. Проанализировать использование в путешествиях различных интернет-ресурсов. Это необходимо для выбора функций, которые будет выполнять программный агент.
3. Осуществить обоснованный отбор технологий для разработки программного веб-агента. На данный момент существует большое количество различных технологий для разработки, а следовательно, есть необходимость в обоснованном их анализе и отборе наиболее оптимальной.
4. Разработать модель будущего программного агента. Модель должна удовлетворять требованиям, связанным с функциональностью разрабатываемого веб-агента.
5. Реализовать программного веб-агента и апробировать его.

# 

# 

# 

# 

# 

# 

# 

# 

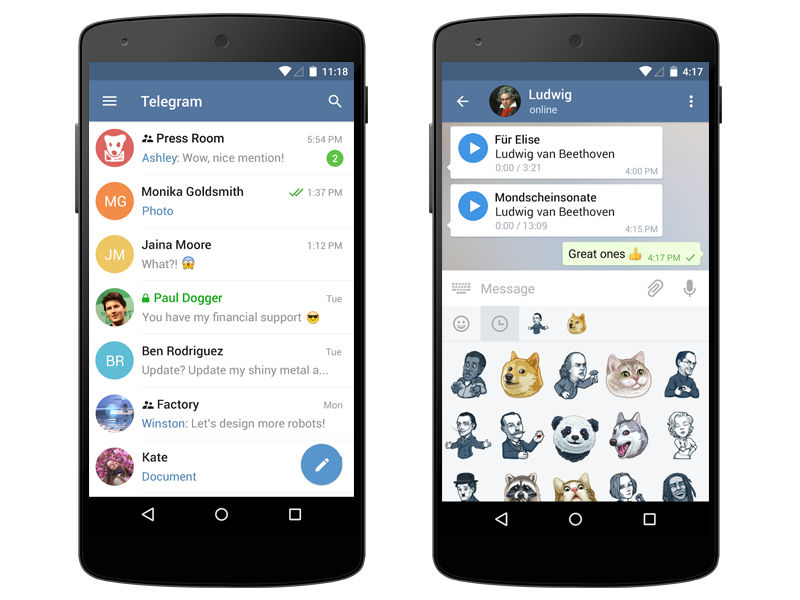
# **ГЛАВА 1**

## **1.1. Выбор платформы**

Веб-агенты могут базироваться на различных платформах или быть отдельным приложением. Вариант с отдельным приложением не соответствует одной из поставленных целей, а именно экономии памяти, поэтому было решено разрабатывать веб-агента на базе уже существующего приложения, которое уже есть у большинства пользователей. После анализа различных сервисов для размещения веб-агента, платформой для разработки веб-агента был выбран мессенджер Telegram. В нем роль программных веб-агентов выполняют боты. Бот (сокращение от робот) в Telegram — это программа, которая взаимодействует с пользователем в отдельном чате, выполняя различные функции: развлечение, обучение, сбор и анализ данных и многих других. Telegram был выбран по ряду причин:

1. Доступность. Telegram является полностью бесплатным приложением и доступен для скачивания в большинстве стран мира в Google Play, AppStore и Windows Store.
2. Кроссплатформенность. Помимо того, что Telegram можно использовать на смартфоне с любой операционной системой, на официальном сайте он доступен для загрузки и установки во всех операционных системах для компьютера и имеет веб-версию для работы в браузере.
3. Популярность. Telegram входит в тройку самых популярных мессенджеров в России и мире. Согласно исследованию, которое провёл в своё время автор канала SMM в Telegram, наибольшую популярность Telegram имеет у мужчин в возрасте от 25 до 34 лет и женщин от 12 до 34. Эти люди составляют 77% от общей аудитории. Данная группа людей совпадает с целевой аудиторией разрабатываемого продукта.
4. Простота в использовании. Веб-агенты в Telegram выглядят как обычная переписка, поэтому их использование не должно вызывать трудностей у пользователя. Кроме того, с 2017 года в Telegram доступен русский язык, что еще сильнее упрощает его использование для русскоязычных пользователей, которые являются целевой аудиторией программного агента.
5. Экономия трафика. На чат в мессенджере тратиться значительно меньше трафика, чем на загрузку сайта, в связи с отсутствием лишней графической информации. Так как в поездках пользователь находится в роуминге, где интернет не входит в стандартный пакет услуг и значительно дороже, чем в домашнем регионе, этот критерий очень важен для потенциальных пользователей.
6. Инструменты для разработки. API Telegram бесплатен и имеет подробную документацию как на английском, так и на русском языке. У большинства современных языков программирования есть библиотеки для написания Telegram ботов, которые существенно облегчают задачу разработки. Кроме того, существуют различные сервисы для создания ботов без знания какого-либо языка программирования. А сам Telegram имеет открытый исходный код, ознакомиться с которым может любой желающий.

Помимо этого, к преимуществам Telegram можно отнести приватность, за счет уникального способа шифрования, скорость и отсутствие рекламы. В связи с обозначенными выше причинами его можно считать наиболее оптимальной платформой для разработки туристического программного веб-агента. C интерфейсом Telegram для ОС Android можно ознакомиться на рисунке 1. Интерфейс для других систем не сильно отличается от представленного.

**

*Рисунок 1* — *интерфейс Telegram*

## **1.2. Выбор функционала**

Для того чтобы определиться с функциями, которые будет выполнять программный агент, необходимо проанализировать спрос будущих пользователей на использование различных интернет-сервисов в путешествии, узнать какая информация в поездке для них наиболее важна. Кроме того, надо учитывать ограниченные возможности программного веб-агента и ограниченного трафика, так как, как говорилось выше, одна из целей создания этого бота — экономия трафика, поэтому было принято решение избегать графическую, аудио и видео информацию.

В ходе исследования запросов пользователей и возможности их реализации в разрабатываемом Telegram боте было принято решение включить в него следующие функции:

1. Курс валют. Так как он постоянно меняется, пользователям нужна актуальная информация о нем для понимания того, выгодно ли менять валюту в данном месте, для проверки цен на товары и услуги и для расчета бюджета поездки. Для получения информации об актуальном курсе валют, пользователю нужно зайти на сайт ЦБ РФ и найти там нужную валюту, а программный агент может сделать все это сам, используя XML с этого сайта, возвратив пользователю только актуальный курс интересующей его валюты на сегодняшний день. Все это делается без особых усилий со стороны пользователя.
2. Погоду. Она, как и курс валют, нестабильна. А находясь в туристическом месте, пользователь непременно будет проводить много времени на улице, поэтому информация о погоде для него важна. Градусник за окном в отелях и хостелах устанавливают далеко не всегда, сайты с погодой очень массивные и тратят много трафика и времени, в то время как в боте данный запрос займет пару секунд и практически не истратит трафик, так как выдаст только нужную информацию, без рекламы и графики. Для получения информации о погоде, был выбран сайт openweathermap.org, так как у него есть бесплатный API с подробной инструкцией для доступа к данным о погоде в любой точке мира. Информация с этого сайта прошла проверку на точность и актуальность, поэтому ей можно верить.
3. Информацию о достопримечательностях. Обзорные экскурсии по городу сейчас пользуются меньшим спросом из-за развития интернета, откуда можно бесплатно почерпнуть всю информацию. Программный агент в этом случае заменит гида и брошюры и расскажет о достопримечательностях основную информацию, при этом, пользователю не придется самому искать что-либо в интернете, тратя на это время и трафик. Достаточно сделать короткий запрос в чате. Кроме того, вся информация останется в переписке, и к ней можно будет вернуться в офлайн-режиме в удобное время. Это может быть полезно в тех случаях, когда доступа к интернету в районе интересующего объекта нет, и пользователь может получить всю информацию заранее, а вернуться к ней уже у интересующего объекта. Информацию о достопримечательностях было решено использовать из интернет-энциклопедии Википедии, так как у нее есть своя библиотека для большинства популярных языков, в ней содержится актуальная и достоверная информация о большинстве культурно значимых объектах мира и доступ к ней бесплатен и открыт для всех.

Все эти функции могут быть реализованы в Telegram боте, так как имеют текстовый запрос, который легко обрабатывается ботом, и текстовый ответ в виде сообщения, которое доступно во всех версиях Telegram.

## **1.3. Выбор инструментов**

Следующим и важнейшим шагом является выбор языка программирования, на котором будет реализован продукт. Для начала были отобраны три языка, которые входят в пять самых популярных языков программирования: Php, Java и Python. Эти языки бесплатны, их документация находится в открытом доступе, по ним написано множество учебников, существует огромное количество сред разработки для работы на них, то есть проблем в их использовании быть не должно. Ниже приведен аннотированный список этих языков, именно на основе которого будет проводится отбор языка для разработки программного агента.

«PHP (рекурсивный акроним словосочетания *PHP: Hypertext Preprocessor*) — это распространенный язык программирования общего назначения с открытым исходным кодом. PHP специально сконструирован для веб-разработок и его код может внедряться непосредственно в HTML»[2].

Java — объектно-ориентированный язык программирования, разрабатываемый компанией Sun Microsystems с 1991 года и официально выпущенный 23 мая 1995 года. Изначально новый язык программирования назывался Oak (James Gosling) и разрабатывался для бытовой электроники, но впоследствии был переименован в Java и стал использоваться для написания апплетов, приложений и серверного программного обеспечения. Программы на Java могут быть транслированы в байт-код, выполняемый на виртуальной java-машине (JVM) — программе, обрабатывающей байт-код и передающей инструкции оборудованию, как интерпретатор, но с тем отличием, что байт-код, в отличие от текста, обрабатывается значительно быстрее.

Python — высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Синтаксис ядра Python минималистичен. В то же время стандартная библиотека включает большой объём полезных функций. Python поддерживает структурное, объектно-ориентированное, функциональное, императивное и аспектно-ориентированное программирование. Основные архитектурные черты — динамическая типизация, автоматическое управление памятью, полная интроспекция, механизм обработки исключений, поддержка многопоточных вычислений, высокоуровневые структуры данных. Поддерживается разбиение программ на модули, которые, в свою очередь, могут объединяться в пакеты.

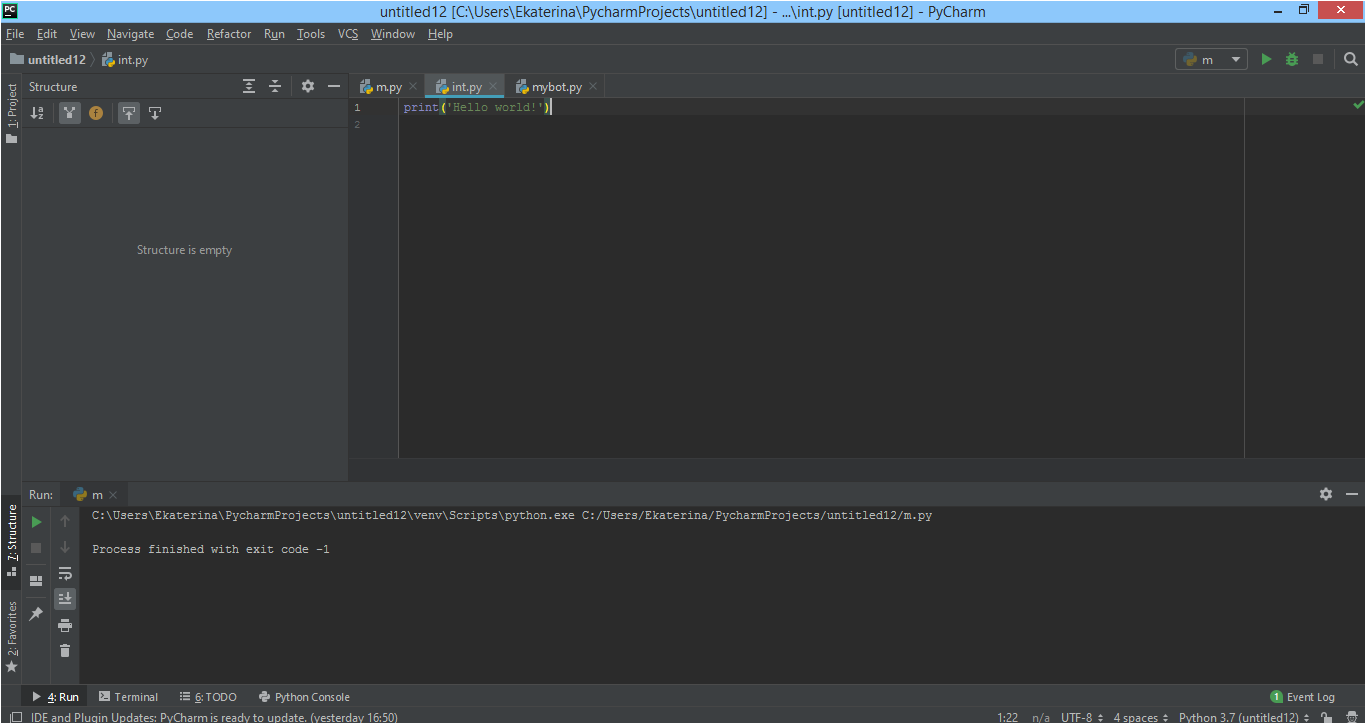
После обзора этих языков, было принято решение реализовывать программного агента на python, так как данный язык программирования имеет понятный и лаконичный код, и множество разных библиотек, что существенно облегчает разработку и дальнейшую работу с кодом, кроме того, этот язык, как было сказано выше является одним из самых популярных языков программирования на данный момент, что является несомненным плюсом, популярность обеспечивает постоянные обновления и усовершенствования языка, а также, благодаря популярности языка, существует множество справочной литературы, которая облегчает процесс разработки на нем.

После выбора языка важно выбрать основную библиотеку, с использованием которой будет реализован бот. Python имеет большое количество библиотек для различных целей, в частности, для создания Telegram ботов. Одной из таких библиотек является pyTelegramBotAPI, она была выбрана из-за ее популярности, у нее больше 2000 звезд на github.com, доступности, она абсолютно бесплатна и доступна на всех ныне действующих версиях Python, и наличия в ней декораторов, что существенно облегчает работу и делает код более читабельным. Данная библиотека имеет различные методы для отправки, принятия и обработки запросов, что делает ее еще более привлекательной. Эти методы существенно облегчают разработку и благодаря им разработчику достаточно запрограммировать лишь бизнес-логику будущего веб-агента.

Следующим шагом является выбор среды разработки, так как Python —популярный язык, он имеет множество различных сред разработки. Ниже приведен список наиболее популярных из них.

1. PyCharm — это интеллектуальная Python IDE с полным набором средств для эффективной разработки на языке Python. PyCharm выполняет инспекцию кода на лету, автодополнение, в том числе основываясь на информации, полученной во время исполнения кода, навигацию по коду, обеспечивает множество типов рефакторинга. Разработана компанией JetBrains.
2. PyDev — это плагин Python для Eclipse. Поддерживается всеми известными операционными системами. Источники в свободном доступе по публичной лицензии Eclipse. Включает в себя следующие возможности: обработка доработки кода, интеграция отладки Python, добавление токен-браузера, инструменты рефакторинга и многое другое
3. Komodo IDE — это коммерческая среда разработки, которая построена на Komodo Edit и имеет ряд дополнительных функций, как например, отладка кода (включая удаленную), рефакторинг, работа с VCS, HTTP Inspector, Rx Toolkit, интегративный шелл, юнит-тесты, интеграцию с другими продуктами компании (Stackato, Perl Dev Kit и TCL Dev Kit) и другие мелочи.
4. PyScripter — это свободная интегрированная среда разработки (IDE) языка Python для операционной системы Windows. PyScripter разработан на компилируемом языке Object Pascal.

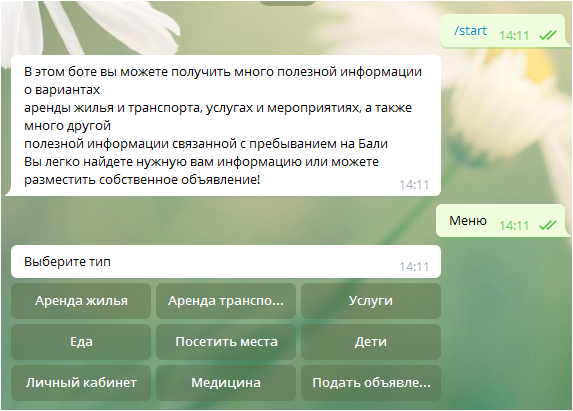
После анализа данных сред разработки выбор был сделан в пользу PyCharm, так как эта среда универсальна, разработана крупной и известной компанией, у нее есть подсветка синтаксиса, что делает код более читабельным, и она имеет бесплатную лицензию для студентов. С его интерфейсом можно ознакомиться на рисунке 2.

**

*Рисунок 2* — *интерфейс PyCharm*

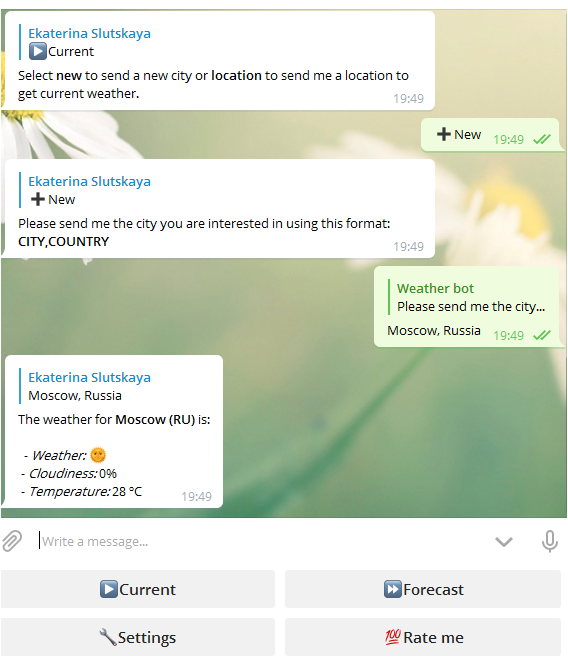
## **1.4. Обзор существующих решений**

Как уже говорилось выше, все функции представленные в данном программном агенте существуют по-отдельности на разных сайтах и приложениях, но они на данном этапе не представляют интереса, так как разрабатывается именно программный агент с интеграцией этих функций. Поэтому был произведен обзор решений непосредственно в Telegram, с целью проверки уникальности разрабатываемого продукта, поиска идей реализации функций, которые используются в данном боте. По запросу «турист бот» в поиске Telegram обнаружен единственный работающий туристический бот, ознакомиться с ним можно на рисунке 3.

**

*Рисунок 3* — *туристический бот*

Как можно видеть из представленного скриншота, данный бот актуален только для путешественников, которые находятся на Бали. Бот для общих целей в процессе исследования не был обнаружен, что значит, что разрабатываемый продукт является уникальным. В этом боте используются встроенные кнопки, что делает процесс общения с ним еще более простым, поэтому было решено использовать эту технологию в разрабатываемом программном агенте. Встроенные кнопки — это кнопки выводимые в диалоговом окне, они привязаны к сообщению и являются его частью. Все найденные боты для проверки курса валют в основной валютой рубль на момент написания этой работы не работают, поэтому эта функция является уникальной. Аналогично и с достопримечательностями. Для прогноза погоды был найден и апробирован работающий бот. Принцип его работы представлен на рисунке 4.

**

*Рисунок 4* — *бот с погодой*

Нареканий к работе этого бота не возникло, единственное, что делает его не совсем подходящим для целевой аудитории разрабатываемого продукта, это отсутствие в нем русского языка. В этом боте привлекательной идей является то, что сообщений бота посылаются в ответ на сообщения пользователя, захватывая и пересылая его, это удобно для тех случаев, когда надо будет вернуться к какому-либо сообщению.

Таким образом, выходит, что разрабатываемый продукт является уникальным, за счет интеграции туда нескольких функций, что делает его привлекательным для пользователя, потому что благодаря ему он сможет подключив одного бота и использовать 3 разные функции, которые ранее не были реализованы в одном программном агенте.

## **1.5. Вывод к первой главе**

На основе вышеописанного материала было принято решение разработать туристического программного веб-агента на базе популярного мессенджера Telegram. Языком программирования после анализа трех популярных языков был выбран Python. Для упрощения разработки было принято решение использовать библиотеку pyTelegramBotApi. Средой разработки для данного продукта станет pycharm. Программный агент будет представлять пользователю информацию о текущем курсе валют, погоде и достопримечательностях в виде текстовых сообщений. Была проведена проверка на наличие аналогичных решений, однако, они не были обнаружены, что говорит об уникальности данного бота.

# 

# 

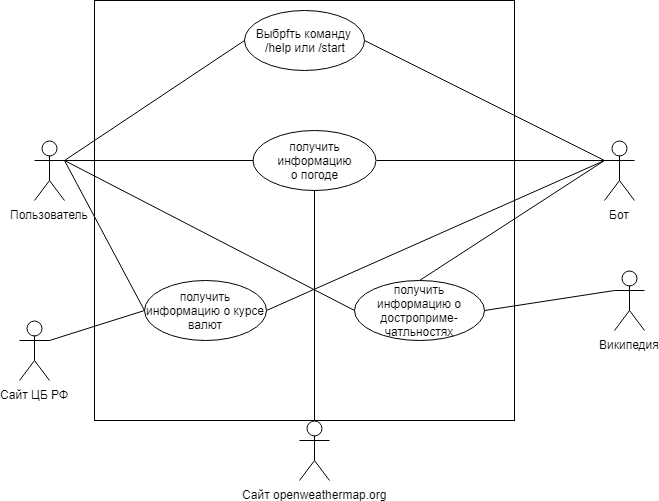
# 

# 

# **ГЛАВА 2**

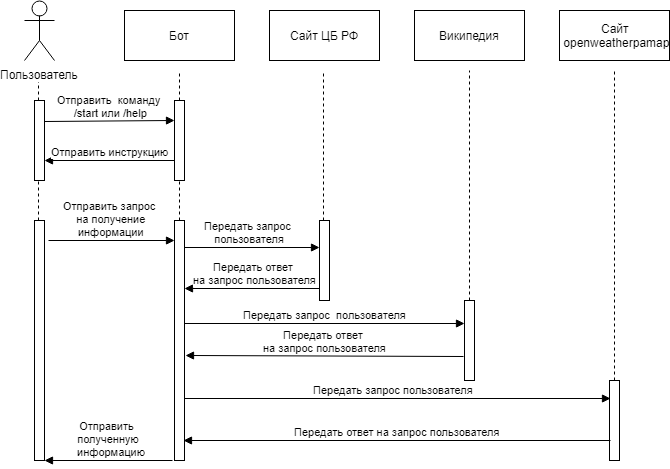
## **2.1. Модель**

Для моделирования разрабатываемого Telegram бота было решено использовать диаграммы UML, так как они наиболее точно и лаконично отображают требования к продукту. Самыми подходящими для этой цели являются диаграмма прецедентов и диаграмма последовательности. Именно они очень точно отражают суть работы данного веб-агента. Для диаграммы прецедентов действующими лицами (актерами) было решено взять пользователя, бота и веб-ресурсы с запрашиваемой информации, а действиями были взяты запросы пользователя в общем виде. С полученной диаграммой можно ознакомиться на рисунке 5.

**

*Рисунок 5* — *диаграмма прецедентов*

На диаграмме последовательности представлены те же действующие лица, но подробнее отображено то, как они взаимодействуют между собой. На рисунке 6 можно увидеть, что бот является связующим звеном между пользователем и веб-ресурсы, выполняя задачу интеграции различных функций в одно место и упрощая работу пользователя с этими ресурсами, именно эти задачи являются основными в данной работе.

**

*Рисунок 6* — *диаграмма последовательности*

После того, как был успешно смоделирован программный агент, можно приступать к его разработке, ориентируясь на полученные диаграммы, на которых отображены все требования к разрабатываемому продукту.

## **2.2. Этапы разработки**

Обозначим основные этапы разработки программного веб-агента, которые были выполнены. Эти этапы послужат планом для дальнейшей работы. Для данного продукта были выделены следующие этапы:

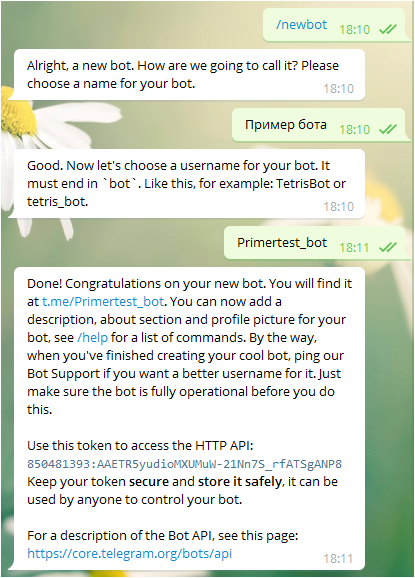
1. Подготовка среды программирования. На этом этапе необходимо установить среду программирования и подготовить ее к работе, скачав все необходимые библиотеки. Помимо этого, надо настроить ее для дальнейшей работы.
2. Получение идентификаторов. Каждый бот в Telegram имеет свой уникальный идентификатор, благодаря которому осуществляются все запросы, поэтому, перед написанием кода программы, необходимо получить этот идентификатор. Сайт openweathermap.org тоже имеет уникальный идентификатор для каждого пользователя, этот идентификатор также надо получить перед началом работы.
3. Написание кода. Самый важный этап разработки, на котором реализуется весь функционал программного агента.
4. Апробация. Проверка выполнения всех требований к разрабатываемому продукту и корректности его работы. Апробация будет выполняться параллельно с написанием кода, так как на основе выявленных неточностей будет улучшаться код программы.

## **2.3. Подготовка к разработке**

Как уже было сказано выше, первым делом необходимо установить и настроить среду разработки. Как было сказано в первой главе, средой разработки для данной работы будет служить PyCharm. Скачать его можно с официального сайта компании JetBrains, где есть бесплатная демо-версия и платная версия, которая имеет бесплатную лицензию для студентов РГПУ им. А. И. Герцена. Установка PyCharm проста и последовательна, достаточно лишь следовать инструкции. После установки PyCharm требуется установить все необходимые библиотеки. Первая и основанная библиотека — это pyTelegrambotapi, именно благодаря ей будущий бот будет получать и отправлять запросы. Для ее установки необходимо набрать в командной строке pip install pyTelegrambotapi. Помимо нее для разработки данного веб-агента потребуются следующие библиотеки: wikipedia — официальная библиотека для работы с Википедией, requests — библиотека, для отправления HTTP-запросов и urllib3 — библиотека для работы с url. Для установки этих библиотек надо выполнить в командной строке команды pip install wikipedia, pip install requests и pip install urllib3 соответственно.

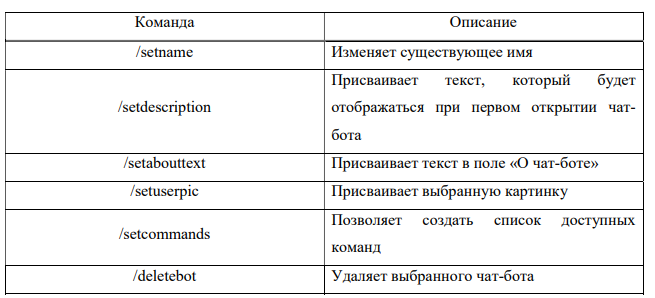
Теперь, когда среда разработки готова, и все необходимые библиотеки установлены, можно приступать непосредственно к разработке самого бота. Первым делом, для его создания, надо зарегистрировать бота в Telegram и получить уникальный токен, который будет использоваться для отправления запросов.

Для регистрации бота в Telegram надо найти в контактах Telegram BotFather. BotFather является официальном ботом Telegram, предназначенным для создания и управления пользовательскими ботами. Для создания нового бота требуется отправить BotFather команду /newbot, а затем, следуя инструкции, дать боту название и логин, по которому его могу найти пользователи. Логин, который является частью ссылки на бота, обязательно должен заканчиваться на Bot. После прохождения регистрации BotFather вышлет уникальный идентификатор для бота. На рисунке 7 показан пример того, как происходит регистрация нового бота в BotFather.

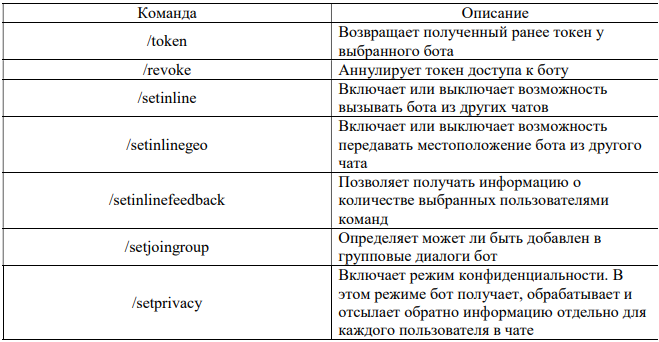
**

*Рисунок 7* — *пример регистрации чат-бота*

В BotFather можно не только создавать ботов, но и управлять уже существующими. Например, добавлять описание, иконку, приветственное сообщение, в случае утечки токена можно обнулить его и сгенерировать новый. Список основных команд, которые доступны в BotFather можно увидеть на рисунке 8, а список расширенных на рисунке 9.

**

*Рисунок 8* — *основные команды BotFather*

**

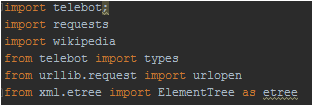
*Рисунок 9* — *дополнительные команды BotFather*

После регистрации бота, для разработки туристического программного веб-агента, необходимо зарегистрироваться на сайте openweathermap.org, чтобы и там получить уникальный идентификатор. Регистрация там бесплатная, хоть и имеется платная версия для расширенного функционала, но она в данном программного агенте не потребуется. Сразу после завершения стандартной регистрации в личном кабинете станет доступен уникальный идентификатор.

Теперь можно считать, что подготовительный этап к созданию бота завершен. И можно приступать к написанию кода.

## **2.4. Написание кода**

Первым делом необходимо импортировать все библиотеки, которые будут использоваться. Все подключенные библиотеки можно увидеть на рисунке 10.

**

*Рисунок 10* — *импортированные библиотеки*

После импорта всех библиотек настало время для подключения токена, полученного от BotFather, к разрабатываемому боту. Именно благодаря ему будут отправляться и обрабатываться все запросы. В коде программы это будет выглядеть так, как показано на рисунке 11, где TOKEN — это переменная, содержащая уникальный идентификатор. Отдельная переменная сделана в целях безопасности, и удобства смены идентификатора в случае его утечки.

**

*Рисунок 11* — *подключение токена*

Затем можно приступить к написанию основной (логической) части кода программы и реализации необходимых функций.

Сначала необходимо объявить метод получения сообщений, реализуется это так, как отображено на рисунке 12.

**

*Рисунок 12* — *объявление метода получения сообщения*

В данном фрагменте кода был объявлен слушатель для текстовых сообщений и метод их обработки. Поле content\_types может принимать не только текстовые значения, а еще, например, документы, аудио, изображение. В разрабатываемом программном агенте требуется обработка только текстовой информации и команд, поэтому код будет содержать два метода обработки сообщений, один для текста, другой для команд.

В предыдущем абзаце были упомянуты команды. В данном программном агенте было решено реализовать только две команды, так как большинство запросов будут индивидуальными, и в виде команд их приставить проблематично, как команду можно представить запрос о помощи (/help) и команду, которая начинает работу с ботом, (/start), эти команды универсальны и могут быть использованы любым пользователем вне зависимости от его целей. Несмотря на то, что бот имеет русскоязычный интерфейс, команды должны быть на английском, таковы условия Telegram.

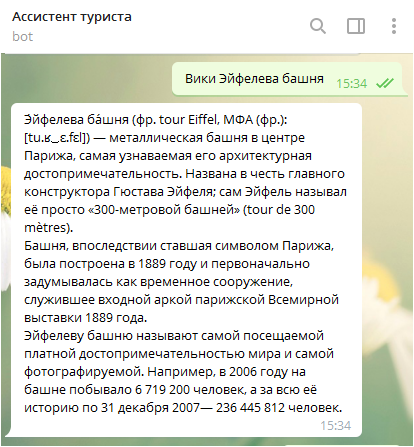
Помимо метода получения сообщений, безусловно, потребуется метод отправки. Библиотека pyTelegrambotapi предлагает для этих целей метод bot.send\_message, который в качестве аргумента использует id пользователя и передаваемое сообщение. Id пользователя бот получает благодаря методу message.from\_user.id, передаваемое сообщение должно быть в виде строки. Строка — это один из основных типов данных в Python.

Для того, чтобы бот непрерывно обращался к серверу, проверяя не пришли ли новые запросы, требуется в самом конце написать строчку bot.polling(). В скобках можно указать частоту обращений, если они регулярны и периодичны, но для данного бота это не требуется.

Теперь осталось реализовать логическую часть, а именно функции, которые будут преобразовывать запрос пользователя в ответ бота. Как было сказано выше, таких функций будет три: текущий курс валют, погода и информация о достопримечательностях. Реализуются эти функции следующим образом:

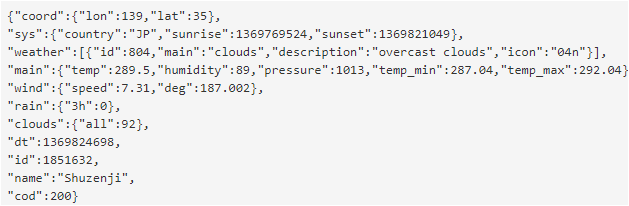
* Информация о достопримечательностях. Официальная библиотека wikipedia для Python позволяет получить информацию из первого абзаца Википедии используя только 1 строку кода. Для краткого ознакомления с достопримечательностью этого абзаца должно хватить. По умолчанию, эта библиотека работает с англоязычной Википедией, однако, можно сменить язык на любой, на котором доступна соответствующая статья. Так как целевая аудитория данного продукта — русскоязычные пользователи, было принято решения сменить язык на русский. Для получения информации о достопримечательностях пользователь должен отправить боту сообщения вида: «Вики название достопримечательности», слово вики в начале помогает боту определить, что от него требуется предоставить информацию из Википедии, а название достопримечательности используется уже для поиска этой информации. Бот передает это название Википедии, а в ответ выдает статью, которая была найдена по нему. Название может быть введено как на русском, так и на языке оригинала, что облегчает пользователю задачу в том случае, когда он не может перевести название достопримечательности на русский язык.

Пример работы данной функции изображен на рисунке 13.

**

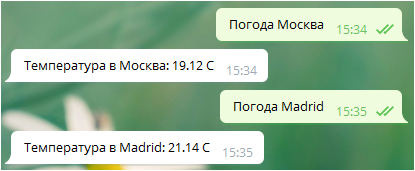
*Рисунок 13* — *пример работы бота*

* Погода. Информация о погоде, как уже было сказано выше, добывается с сайта openweathermap.org. Запрос обрабатывается при помощи библиотеки requests. Запрос состоит из ссылки на сайт с уникальным идентификатором и названия города, которое будет получено от пользователя. После этого запроса, бот получает json в котором содержится информация о погоде в запрашиваемом пользователем городе в виде словаря, словарь — это структура данных в Python, напоминающая массив, но идентификация элементов в нем осуществляется не по порядковому номеру, а по произвольному ключу. С примером полученного словаря можно ознакомиться на рисунке 14.



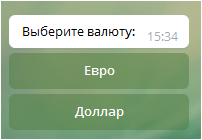
*Рисунок 14* — *полученный json*

Затем из него, при помощи ключей main и temp извлекается информация о текущей температуре воздуха в градусах по Цельсию, так как именно эту единицу измерения использует целевая аудитория продукта, и потом эти данные передаются пользователю в виде текстового сообщения. Пользовательский запрос должен иметь вид «Погода Город», где город — название города на английском языке или языке страны, где этот город находится. Пример работы данной функции можно увидеть на рисунке 15.

**

*Рисунок 15* — *пример работы бота*

* Курс. Было принято решение реализовать получение курса валют для двух самых популярных валют: евро и доллара. Помимо того, что эти две валюты являются официальными в большинстве популярных для туристов странах, в остальных, популярных у путешественников местах несмотря на то, что они не являются официальными валютами страны, их все равно принимают. Курс валют программный агент получает после обработки XML от ЦБ РФ. На сайте Центрального банка Российской Федерации всегда актуальный курс, который находится в открытом доступе. Для реализации этой функции было решено использовать встроенные (inline) кнопки. Пользователь сообщает боту, что хочет получить информацию о курсе отправив сообщение «Курс». После этого он видит меню, которое изображено на рисунке 16.

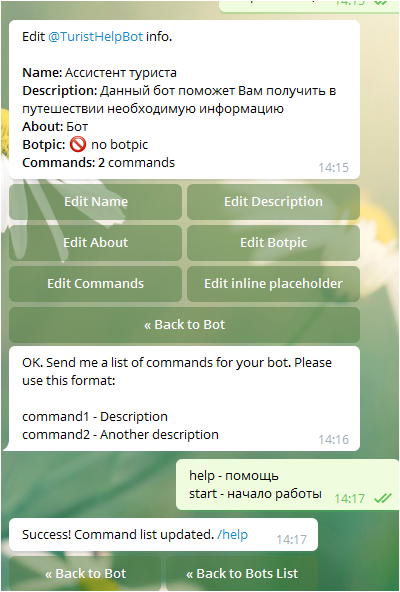


*Рисунок 16 — меню выбора валюты*

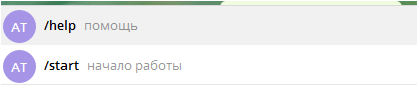
После выбора валюты пользователь получает ее курс. Реализовано это было следующим образом: для принятия запроса пользователя использовался все тот же метод bot.message\_handler, для отправки ответного сообщения был использован метод bot.send\_message, но помимо текста с просьбой выбрать курс, он передает клавиатуру, с двумя callback-кнопками (евро и доллар). Callback-кнопки — это разновидность inline-кнопок, с обратной связью (откликом) от бота. Для создания клавиатуры и кнопок были иcпользованы методы types.InlineKeyboardMarkup и types.InlineKeyboardButton соответственно. Метод types.InlineKeyboardButton передает боту сообщение об отклике, после того, как пользователь выберет кнопку. Для обработки этого отклика был добавлен метод bot.callback\_query\_handler(func=lambda call: True), который уже в своею очередь отправляет пользователю сообщение с курсом валют. Каждая валюта имеет свой идентификатор, по которому она находится в полученном XML.

Так как все эти функции обрабатывают текстовые сообщения, все они занесены в одну функцию, содержащую конструкцию if...elif...else, эта конструкция служит переключателями между различными функциями программного агента.

Теперь, когда вся функциональная часть реализована, надо добавить метод, который будет обрабатывать команды /start и /help, обе эти команды после обработки будут выдавать одинаковый текст, с кратким описанием возможностей бота и тем, как этими возможностями воспользоваться. Первым делом надо добавить эти команды в разрабатываемый бот при помощи вышеупомянутого Telegram-бота BotFather. Для этого необходимо написать ему команду /mybots, выбрать бота, для которого надо добавить команды, выбрать в меню Edit Bot, затем выбрать Edit commands и следовать инструкции так, как показано на рисунке 17. Теперь команды добавлены в меню бота (см рисунок 18).

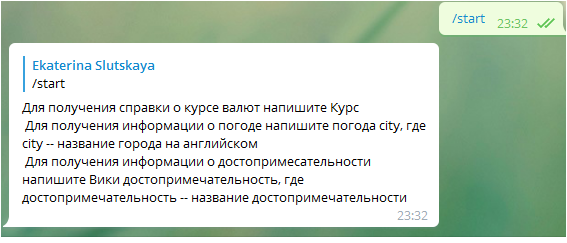
**

*Рисунок 17* — *создание команд в боте*

**

*Рисунок 18* — *меню команд*

После этого надо написать обработчик данных команд с использованием все того же метода bot.message\_handler с указанием в скобках commands=['start', 'help'], текстовое сообщение с инструкцией будет отправлено пользователю при помощи метода bot.reply\_to, этот метод используется для ответа на сообщение, благодаря этому инструкцию будет удобнее найти, чтобы не получать ее второй раз. Достаточно будет ввести в поисковую строку название команды, в ответ на которую была получена инструкция. Выглядеть это будет так, как показано на рисунке 19.

**

*Рисунок 19* — *реагирование бота на команды*

## **2.5. Обработка исключений**

Исключения — это один из типов данных в Python, необходимый для сообщения программисту об ошибках в коде, не прерывая работы программы. Боты в Telegram — это то место, где исключения необходимы, так как одна ошибка в запросе пользователя может привести к сбою всей программы, например, в данном боте, если после слова «погода» ввести название несуществующего города, произойдет сбой, так как запрос не сможет быть выполненным. И программа прервется с ошибкой KeyError (см рисунок 20), так как по неправильному названию города будет получен словарь с ошибкой, в котором нет ключей, по которым ищется интересующая информация.

**

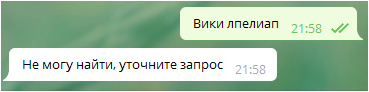
*Рисунок 20* — *ошибка работы программы*

В связи с этим необходимо добавить исключение, обрабатывающие тип ошибки KeyError, так, как это показано рисунке 21.

**

*Рисунок 21* — *обработка исключений*

Теперь при вводе несуществующего города или ошибке в вводе названия города программа не прекращает со работу, а отправляет пользователю сообщение о том, что такого города нет, и после этого пользователь может повторить запрос, введя правильное название, аналогичное исключение, но обрабатывающие другую ошибку необходимо добавить в функцию, которая предоставляет информацию о достопримечательностях. Так как там могут возникнуть ошибки разных типов, например, обнаружение несколько разных страниц в Википедии по запросу или, наоборот, ни одной, и эти ошибки не общего вида, а ошибки в использовании библиотеки, было принято решение добавить исключение без уточнения типа ошибки. При неверном запросе программа не будет прекращать свою работу, а отправит пользователю просьбу уточнить запрос, как это показано на рисунке 22. В функции возвращающую значение курса валют исключения добавлять не надо, так как там всего две кнопки и ошибок возникать не должно, потому что запрос боту посылается по отклику на эти кнопки, а он прописан в программе.

******

*Рисунок 22* — *результат обработки исключений*

Сам бот, так как имеет конструкцию if...elif...else, неправильный запрос обрабатывает в параметре else, поэтому исключение, которое обрабатывает неверный запрос ему не требуется.

Таким образом, были написаны исключения к тем кускам кода, где могут возникать ошибки, приводящие к сбою программы.

## **2.6. Перспективы развития**

В дальнейшем планируется расширять функциональность программного агента, проводя исследования на спрос пользователя. Помимо этого планируется улучшать уже имеющийся функционал, за счет получения отзывов от пользователей, которые будут собираться с определенной периодичностью.

## **2.7. Вывод ко второй главе**

Был смоделирован Telegram бот посредством двух UML диаграмм, на которых были отображены все требования к разрабатываемому продукту. После этого он был разработан в полном соответствии с данными диаграммами. Все требуемые функции были реализованы так, как было заявлено в первой главе данной работы. Апробация показала, что бот работает корректно, нормально реагирует на разные запросы за счет добавления в него исключений.

# **ЗАКЛЮЧЕНИЕ**

В данной выпускной квалификационной работе был разработан туристический программный веб-агент. Программные агенты на сегодняшний день пользуются большим спросом из-за своей минималистичности.

Для его разработки был произведен анализ различных технологий и отобраны те из них, которые являются наиболее оптимальными для поставленных целей.

Платформой для разработки данного программного агента послужил популярный мессенджер Telegram. В настоящее время использование мессенджеров настолько растет, многие операторы отказались от предоставления пакета SMS в своих тарифах, так как на смену им пришли мессенджеры, а теперь, когда в них стали доступны звонки, их популярность выросла еще сильнее. Telegram был выбран по ряду причин описанных в первой главе.

Для разработки данного программного агента был использован язык программирования Рython, который на сегодняшний день является одним из самых популярных языков программирования, и позволяет выполнить все поставленные в данной работе задачи. Для облегчения разработки была использована библиотека pyTelegrambotapi, предназначенная для разработки ботов в Telegram. Разработка производилась в среде программирования РyСharm.

Функционал данного программного агента обусловлен спросом пользователей, которые находятся в путешествии, и возможностями программного агента. Помимо этого, было учтено, что в поездках интернет дорогой, поэтому вся информация предоставляется пользователю в текстовой форме.

Все поставленные задачи были успешно решены. Данный бот корректно выполняет все запросы. Ознакомиться с ботом можно по ссылке: t.me/TuristHelpBot

# **СПИСОК ЛИТЕРАТУРЫ**

1. Назаров Н. Х., Морозов А. Д. Назначение и возможности чат-ботов //МНСК-2018: Школьная секция: Информационные технологии. – 2018. – С. 44-44.
2. Руководство по PHP [Электронный ресурс] / Mehdi Achour, Friedhelm Betz, Antony Dovgal, и др. — Электрон. журн. — Режим доступа: https://www.php.net/manual/ru/index.php, свободный
3. Вязовик Н., Жилин Е. Программирование на Java //М.: Интуит. ру. – 2003.
4. Голошубова О. М., Наумов В. Ю. Python: происхождение, преимущества и перспективы //Инновационные, информационные и коммуникационные технологии. – 2016. – №. 1. – С. 38-40.
5. Боклаг Н., Буйначев С. Основы программирования на языке Python. – Litres, 2017.
6. Степанова А. С., Волова Е. А. КОНКУРЕНТОСПОСОБНОСТЬ ЯЗЫКА ВЫСОКОГО УРОВНЯ PYTHON //WORLD SCIENCE: PROBLEMS AND INNOVATIONS. – 2018. – С. 71-73.
7. Павлов И. Ю., Колосков В. Л., Иванов Е. Б. СОЗДАНИЕ БОТОВ В Telegram: С ИСПОЛЬЗОВАНИЕМ ЯЗЫКА ПРОГРАММИРОВАНИЯ PYTHON //Инновационные, информационные и коммуникационные технологии. – 2016. – №. 1. – С. 107-110.
8. Колосков В. Л., Павлов И. Ю., Иванов Е. Б. Создание ботов для Telegram: с использованием языка программирования Python и облачной операционной системы Corezoid //Системный администратор. – 2016. – №. 5. – С. 90-95
9. Николаев С. В. и др. Пример использования информационных технологий для решения повседневных задач //Постулат. – 2017. – №. 12.
10. Харевич А. А., Кумратова А. М. Telegram BOT API как средство маркетинга и менеджмента //ИНФОРМАЦИОННОЕ ОБЩЕСТВО: СОВРЕМЕННОЕ СОСТОЯНИЕ И ПЕРСПЕКТИВЫ РАЗВИТИЯ. – 2018. – С. 201-202.