

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий
(наименование института полностью)

Кафедра «Прикладная математика и информатика»
(наименование)

02.03.03 Математическое обеспечение и администрирование информационных систем
(код и наименование направления подготовки / специальности)

WEB-дизайн и мультимедиа
(направленность (профиль)/специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему Разработка телеграмм-бота helpdesk для фармацевтической компании «Озон»

Обучающийся

А.В. Кучаев

(Инициалы Фамилия)

(личная подпись)

Руководитель

М.А. Тренина

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Консультант

М.М. Бажутина

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Аннотация

Выпускная квалификационная работа посвящена проектированию и разработке телеграмм-бота «helpdesk» для фармацевтической компании «Озон».

Ключевые слова: разработка, телеграмм-бот, 1С:Предприятие, тестирование.

Объектом исследования в написании ВКР послужил процесс создания телеграмм-бота.

Предметом исследования бакалаврской работы стал разработанный для фармацевтической компании «Озон» телеграмм-бот 1С «helpdesk».

В состав работы входит введение, 3 главы, 3 вывода по написанным главам, итоговое заключение.

Во введении раскрывается актуальность исследования и разработки телеграмм-бота. Описывается цель и задачи выпускной квалификационной работы. Определяются объект и предмет исследования.

В главе первой предложена теоретическая часть разработки бота. Раскрытие понятий «мессенджер» и «чат-бот», а также проведен сравнительный анализ методологий разработки.

В главе второй показан процесс разработки телеграмм-бота «helpdesk» для заказчика. Проведена регистрация бота, подготовка платформы 1С:Предприятие к работе и заполнение конфигурации. Показана реализация основного функционала чат-бота.

В третьей главе описан ход тестирования бота. Показаны его возможности и принцип работы.

Заключение посвящено основным выводам по выпускной квалификационной работе. Были подведены итоги разработки бота.

Объём бакалаврской работы составляет 49 страниц, на которых размещено 33 рисунка. При написании выпускной квалификационной работы использовалось 25 источников.

Abstract

The title of the bachelor's thesis is devoted to the development of the "support service" telegram bot for the pharmaceutical company "Ozon".

Keywords: development, telegram bot, 1C:Enterprise, testing.

The object of the thesis is research in writing the WRC was the process of creating a telegram bot.

The subject of the study of the bachelor's work was the telegram-bot 1C "helpdesk" developed for the pharmaceutical company "Ozon".

The composition of the work includes an introduction, 3 chapters, 3 conclusions on the written chapters, a final conclusion.

The aim of the study is custom telegram bot development for «Ozon» pharmaceutical factory.

The introduction reveals the relevance of research and development of a telegram bot. The purpose and objectives of the final qualifying work are described. The object and subject of research are determined.

In the first part, the theoretical part of the development of the bot is proposed. Disclosure of the concepts of "messenger" and "chat-bot", as well as a comparative analysis of development methodologies.

In the second part shows the process of developing a "helpdesk" telegram bot for a customer. The bot was registered, the «1C» platform was prepared for work and the configuration was filled. The implementation of the main functionality of the chat bot is shown.

In the next part describes the progress of testing the bot. Its capabilities and principle of operation are shown.

The conclusion is devoted to the main conclusions on the final qualifying work. The results of the development of the bot were summed up.

The bachelor's thesis consists of an introduction, three chapters, a conclusion and a list of 25 references. The total volume of the work is 49 pages of a typewritten text, 33 figures.

Содержание

Введение.....	5
1 Описание методов разработки.....	6
1.1 Мессенджеры.....	6
1.2 Чат-боты.....	13
1.3 Обзор аналогов чат-ботов.....	15
1.4 Выбор среды разработки.....	17
1.4.1 Python.....	17
1.4.2 Java.....	19
1.4.3 1С.....	21
1.5 Telegram Bot API.....	24
2 Разработка телеграмм-бота.....	26
2.1 Регистрация чат-бота.....	26
2.2 Подготовка платформы к работе и заполнение конфигурации.....	28
2.3 Реализация основного функционала чат-бота.....	34
3 Тестирование Telegram бота “helpdesk”.....	43
Заключение.....	46
Список используемых источников.....	47

Введение

Интернет в 21 веке – это уникальная среда для всех сфер деятельности. Будь то общение, развлечения или обучение. В настоящее время в мире существует огромное множество средств, типов и способов общения, и огромная часть так или иначе связана с современными техническими возможностями, которые, в частности, представлены использованием глобальной компьютерной сети.

В данной бакалаврской работе мы изучим процесс создания телеграмм-бота и реализуем его на платформе 1С:Предприятие.

Актуальность выпускной квалификационной работы обусловлена высокой популярностью мессенджеров и таких средств автоматизации как чат-боты среди пользователей сети Интернет. Чат-боты легко справляются с задачей облегчения рутинных задач, такие как получение информации о погоде, пробках, последних новостях и другие. Главным достоинством относительно обычных рядовых приложений является возможность совмещения всех возможностей в рамках одного мессенджера.

Объектом исследования в написании ВКР послужил процесс создания телеграмм-бота.

Предметом исследования бакалаврской работы стал разработанный для фармацевтической компании «Озон» телеграмм-бот 1С «helpdesk».

Целью бакалаврской работы является разработка бота для технической поддержки фармацевтической компании «Озон» в мессенджере Telegram.

Задачами выпускной квалификационной работы являются: провести сравнительный анализ IT решений в данной области, спроектировать работу телеграмм-бота helpdesk, разработать работоспособную версию бота, провести тестирование телеграмм-бота.

1 Описание методов разработки

1.1 Мессенджеры

Своё начало общение в сети интернет берет с чатов, потом приоритет отдали мессенджерам, потом на первое место встали социальные сети, но не так давно мессенджеры снова возглавили этот список, став самым перспективным сервисом для общения. Это было подтверждено исследованиями компании «Yota» в 2022 году: количество пользователей, использующих мессенджеры в Российской Федерации, увеличилось на более чем 15%. А также, по средствам опроса пользователей, они определили, как формируется новый цифровой этикет. Не малая часть людей предпочитает голосовые сообщения обычным текстовым сообщениям [7].

Мессенджеры пользуются популярностью в основном у людей среднего возраста, так на долю лиц младше 18 лет будет приходиться всего 8% от всего количества аудитории, пользователи в возрасте до 25 лет составляют 10%, от 25 до 35 лет 32%, от 35 до 45 лет – 26%. При этом 13% людей от 45 до 55 лет, а 8% - от 55 до 64 лет. И всего лишь 3% пользователей находится в возрасте старше 64-х лет (рисунок 1).

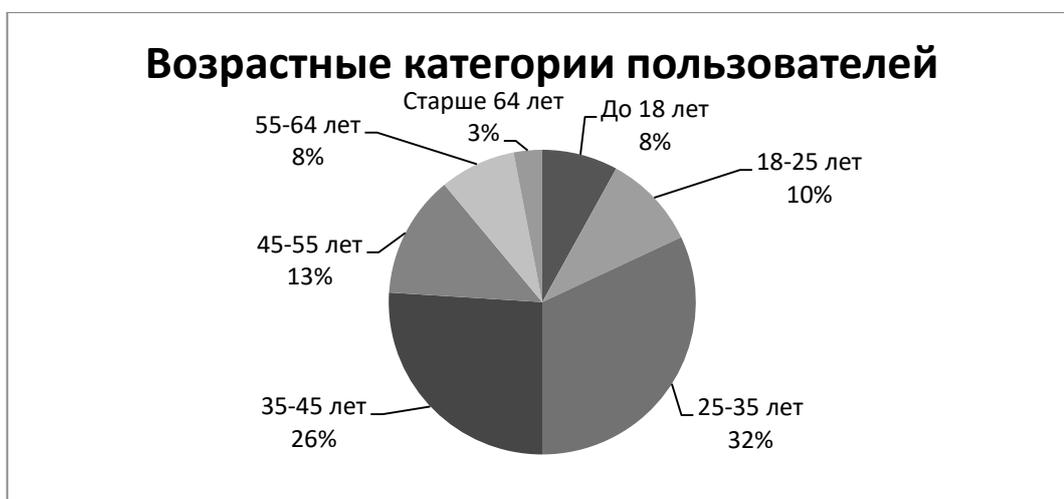


Рисунок 1 – Диаграмма возрастных категорий пользователей

Ещё одной весомой причиной резкой популярности мессенджеров является изменение тарифов мобильного интернета. Например МТС предоставляет пакет с безлимитным трафиком на мессенджеры. Широкое распространение смартфонов любых ценовых категорий, высокие скорости и низкие цены относительно прошлого. Крис Мессина в статье «2016 will be the year of conversational commerce» назвал 2016 год – годом диалоговой или разговорной коммерции. Он понимает под этим понятием мессенджеры, чаты и другие интерфейсы на естественном языке (т.е. голосом), которые дают возможность взаимодействовать людям с брендами и представителями услуг. В конце концов, это приведет к тому, что пользователи будут контактировать с брендами и компаниями через мессенджеры. При этом, по мнению автора статьи, неважно, с кем конкретно будет вестись диалог – с сотрудником компании, с ботом или их гибридом. Все мессенджеры имеют повторяющийся набор инструментов, таких как отправка сообщений, медиафайлов, аудио и видео общение [16].

В данный момент существует большое разнообразие мессенджеров, среди которых нельзя обойти стороной некоторые из них, а также их плюсы и минусы. Особого внимания достоин мессенджер Telegram.

Telegram – это кроссплатформенное приложение, разработанное на языке программирования C++, оно позволяет обмениваться сообщениями и файлами большинства форматов. Мессенджер использует специально разработанную серверную часть с закрытым кодом, работающую на серверах Германии и США.

Telegram имеет ряд преимуществ:

- приватность – все чаты зашифровываются, а если выбрать специальную приватность чата, то сообщения уничтожатся через выбранное время;
- быстрота – скорость отправления сообщений выше, чем у одноклассников;

- распределение – сервера Telegram находятся по всему земному шару, что повышает отказоустойчивость;
- открытость – использование открытого протокола MTProto и API, бесплатных для всех;
- отсутствие подписок и надоедливой рекламы, (рекламу можно встретить непосредственно в чатах от самих пользователей);
- отсутствие ограничений на размер сообщений и вложенных файлов (что очень заметно на фоне социальной сети ВКонтакте).

Особенностями данного мессенджера являются секретные чаты и специально разработанный протокол шифрования MTProto (рисунок 2) [22].

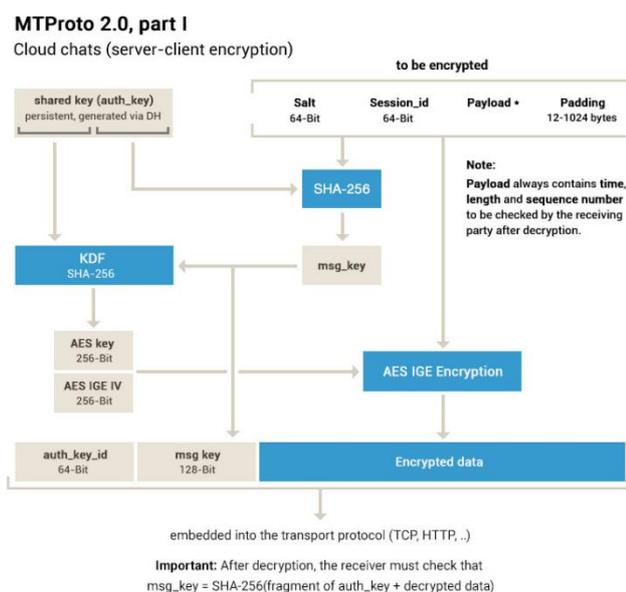


Рисунок 2 – Принцип работы протокола шифрования MTProto

Секретные чаты пригодятся для пользователей, которые хотят обеспечить себя максимально безопасным обменом данными. Помимо использования основного протокола все сообщения шифруются ключами по принципу от устройства до устройства. Никто не в силах перехватить и не сможет расшифровать сообщения, включая даже сотрудников Telegram. Сообщения из таких чатов пересылать невозможно и они не хранятся на

серверах приложения. Но у секретных чатов есть два недостатка – это невозможность открыть чат на устройстве, которое не участвовало в его создании, и никто не защищен от скриншота интерфейса Telegram.

Так же в большинстве версиях для разных операционных систем доступны как минимум английский, русский, польский и немецкий языки. Последние несколько лет Telegram ведет активную разработку своей блокчейн-платформы – «Telegram Open Network» и криптовалюты Gram на ее основе [25].

Отличительные качества Telegram.

Кросс-платформенность, надежность доставки сообщений и скорость.

Факт того, что приложение работает на всех платформах, сейчас не удивляет ни кого, но вопрос в том, насколько связаны разные приложения взаимодействуют друг с другом. Skype, например, просто вездесущ, но при этом постоянно показывает проблемы с синхронизацией уведомлений и звонков.

Уверен, что владельцам таких устройств как Windows Phone знакома такая ситуация, когда жмешь на появившееся уведомление Skype, а потом ждешь минуту или две, пока оно загрузится в приложении. На Android с этим не так плохо, так как приложение гораздо быстрее работает, но там свои моменты бывают [20].

WhatsApp в общем и целом хороший мессенджер, но иногда доставка сообщений тормозит, особенно если они были отправлены при отсутствии связи или ее низком качестве (например, в метро или подвальном помещении). К тому же, хоть и у него появилось десктопное приложение, но из-за ситуации в стране скачать его нельзя, а веб-версия – не совсем то, что хотелось бы видеть на своем устройстве в плане удобства.

Telegram выглядит гораздо лучше во всех этих аспектах, и это не беря во внимание ключевой аспект безопасности. Клиент для Windows и приложение Android работают крайне быстро, а прочитанные сообщения синхронизируются буквально моментально. В отличие от Skype, Telegram

хорошо кэширует сообщения и картинки (настраивается), что значительно ускоряет просмотр истории, в том числе групп и каналов.

Чаты (группы и супергруппы).

Наряду с индивидуальными чатами в Telegram есть группы (как в Viber или WhatsApp), которые хорошо подойдут для частного чата с друзьями или коллегами. Супергруппы обладают возможностью закрепления сообщений и средствами модерирования, необходимыми в публичном чате, а также предоставляет возможность расширить круг пользователей до 5000 участников.

Когда вы создаете группу, вы можете выбрать пользователей из своего списка контактов, если у них есть Telegram. Чтобы пригласить других людей, создайте ссылку-приглашение в настройках группы и пришлите её людям.

По имени пользователя также можно найти любого человека или бота в глобальном поиске Telegram.

Тематические каналы.

В отличие от групп, каналы ведут связь в одном направлении. Сообщения могут отправляться только администраторами канала, но принимаются всеми подписчиками. У каждого канала есть ссылка для просмотра, которой можно поделиться с другими.

Вы можете автоматически передавать контент с вашего сайта или другой социальной сети в свой Telegram-канал и пополнять его другими сообщениями и постами. Вы также можете создать личный блог и наполнить его новостями, фотографиями или своими яркими идеями.

Боты в мессенджерах сейчас везде, и даже Microsoft включилась в игру на Build 2016. Telegram разрабатывает свой API для ботов, и с каждым днем их становится все больше. У каждого бота также есть имя пользователя, поэтому его можно найти в глобальном поиске. Чат-боты делятся на две категории – обычные и встраиваемые (inline).

С обыкновенными ботами пользователь работает напрямую, отправляя им текст (например, свой вопрос или запрос в поисковик) или команды,

которые учёл разработчик.

У большого количества чат-ботов команды продублированы в графическом интерфейсе (меню), поэтому вовсе необязательно вводить их с клавиатуры. Хотя на смартфонах зачастую в поле ввода встраивают кнопку для удобства использования.

Если вы попробуйте пообщаться с парочкой ботов, то вы поймете, как это работает: уа (бот сервисов Яндекса), Meduza (новостной сайт), @truemafiabot (бот для игры в мафию), vkmusic_bot (бот для скачивания музыки из ВКонтакте). Если вы не уверены, с чего начать в работе с чат-ботом, отправьте команду /help и вы увидите список доступных команд.

Некоторые боты, естественно, не будут поддерживать взаимодействие, так как они предназначены под узкую задачу вроде автосообщений в новостном канале.

Встраиваемые (inline) боты умеют принимать команды из других чатов. Например, gif или pic можно вызвать из любого чата для поиска подходящей к теме картинки.

Бесплатные стикеры.

Стикеры – это мини картинки, которые можно загрузить в рамках тематического пакета и вставлять в чат. Кроме того, когда вы начинаете сообщение со смайлика или так называемого emoji, появится список ассоциированных с ним стикеров (так же как и во ВКонтакте например).

В Telegram все стикеры бесплатные, что является большим отличием от всех мессенджеров. Так, популярный в Азии LINE в 2017 году продал пользователям стикеров на \$212 миллионов, а также заработал \$318 миллионов на брэндах, выпускающих свои бесплатные наборы стикеров в рекламных целях. Кстати говоря, Telegram обходится Павлу Дурову в более чем миллион долларов ежемесячно.

Telegram устанавливается с базовым набором стикеров, но скачать любые другие проще простого – нажмите на понравившийся стикер в чате и добавьте пакет. В платформе 2.0 появился встроенный бот @Sticker, с

глобальным поиском стикеров по эмодзи.

А также, если вы не сможете вдруг найти подходящих стикеров, вы со спокойной душой можете создать их сами. Нужны картинки PNG с одной стороной в 512px (другая не должна превышать это значение), а такие можно и в Google найти. Со своими картинками идем в бота @Stickers и следуем его инструкциям.

Актуальность выбора Telegram.

На сегодняшний день, приложение Telegram является одним из самых востребованных мессенджеров во всем мире. Простота использования, легкость установки, доступность на множестве платформ и операционных систем и бесплатная модель распространения позволили мессенджеру уверенно закрепиться на рынке. Одной из главных и уникальных функций данного приложения является возможность создания пользователем чат-бота, которого можно спроектировать практически под любые нужды.

Все компании активно начинают внедрять в свою работу чат-ботов и использование мессенджеров. Это происходит во всех отраслях. Все это из-за того, что там сконцентрирована самая большая и живая аудитория. Они же и являются потенциальными потребителями. Это неоднократно подтвердили различные статистические статьи.

Главной причиной активного введения в свой бизнес работу с мессенджерами и непосредственно чат-ботов является то, что у людей на сегодняшний день, в среднем, на мобильном устройстве стоит до 4 различных мессенджеров. Этот факт, как никакой другой, дает понять, что данная сфера популярна и этим фактом обязательно нужно пользоваться в целях продвижения, рекламы и улучшения своего бизнеса в целом. Так же удастся автоматизировать большое множество процессов, которые раньше выполнялись людьми. А вышеупомянутые чат-боты послужат нам средством для достижения этой цели.

1.2 Чат-боты

Чат-ботов можно разделить на некоторые категории, в зависимости от форматов взаимодействия с ним: Кнопочный чат-бот. Такой бот взаимодействует с пользователем посредством кнопок (уже готовых запросов), другими словами, связь весьма ограничена группой созданных предварительно фраз, нет возможности текстового ввода. Текстовый чат-бот. Данный вид уже более прогрессивный и функциональный: возникает возможность текстового ввода, происходит анализ запроса и выдается на его базу ответ. Следовательно, мы можем иметь диалог между ботом и пользователем [12].

Американский предприниматель Майкл Молдин еще в 1994 г. придумал такой термин как «чат-бот». Если углубиться в статистику использования чат-ботов в наше время, то можно сделать ряд выводов, что боты это некие универсальные средства решения большого множества проблем и жизненных задач, таких как: общение с людьми, развлечение (игры, музыка, фильмы и сериалы), здоровье (например, бот для записи в платную клинику), заказ продуктов или товаров.

Будь то ватс-ап, вайбер или телеграмм, неважно, чат-бот является прикладной программой, которая логически преобразует информацию, полученную от пользователя, в нужный ему ответ. Предоставляет человеку возможность без участия живого человека по ту сторону, получить информацию, которая ему нужна. Сейчас уже создано огромное кол-во ботов для различных идей [13].

Ниже описаны несколько интересных представителей:

- чат-боты с различными играми;
- новостные чат-боты, которые оповещают о последних событиях;
- чат-боты для доставки продуктов, одежды или только что приготовленной пиццы;
- чат-боты для бронирования отелей или билетов на самолет;

– чат-боты юристы, такие могут помочь в консультации по банковским вопросам и т.п.

Очень крепко боты укоренились в жизни человека 21-го века, ведь во всех сферах жизни есть такой легкий в использовании помощник. С каждым днем их становится все больше и их возможности тоже растут. Опять же объяснено это только тем, что люди проводят свое время в смартфоне в большинстве в приложениях мессенджера или e-mail [14].

Выглядит наш бот как обычный чат, вот только разговор ведется не с человеком, а с запрограммированным на определённые действия роботом. Бот готов выполнить любую задачу, которая ему под силу, а вернее ту, которую ему заложил программист при создании. Главное обратится в нужного бота по своему запросу, и ответ не заставит себя долго ждать.

Совсем недавно боты были только развлечением и ничем более серьезным. Но сейчас они превращаются в действительно серьезную вещь, такую, которая способна решить масштабные сложные бизнес задачи. И все это находится у вас в смартфоне. В век технологий и автоматизации это абсолютно нормальное явление. Общество давно уже перешло к новому, одновременно деловому и неделовому, виду общения. Для начала чат-бот – это сервис, для решения любой проблемы будь-то бизнес или запись к врачу.

Программа имеет сформированную базу, которая содержит набор предполагаемых запросов от клиента и соответствующие ответы на данные запросы. Самым типичным примером получения интересующего нас варианта ответа являются ключевые слова, фразы, которые так или иначе похожи на первоначальный запрос, похожий контекст сообщения.

В нашей жизни полно рутинной работы, на которую всегда не хочется тратить своё время. Зачастую это не интересно, не приносит выгоды, но сделать нужно. Тут нам и поможет технология чат-ботов. Старый век машинок для письма, походов в читательские залы и нескончаемых очередей за продуктами питания ушел от нас и больше не вернется. Сейчас для получения информации люди могут использовать бота. Например для

удобства оповещения большой группы лиц, можно организовать новостного бота для отправки им уведомлений о предстоящих событиях [11].

1.3 Обзор аналогов чат-ботов

Мы знаем, что чат-боты бывают абсолютно для любых целей, например: заказ такси, заказ еды на дом, выбор фильма, поиск музыки, ответы на любые вопросы в любой сфере.

Бот для игры в Mafia. @truemafiabot – бот выполняет роль ведущего в популярной стратегической игре «мафия». Вы можете участвовать в игре с тысячами людей из других городов и стран. Это хороший выбор для тех, кто ищет возможность отдохнуть и хорошо провести время, получив великолепные эмоции (рисунок 3).

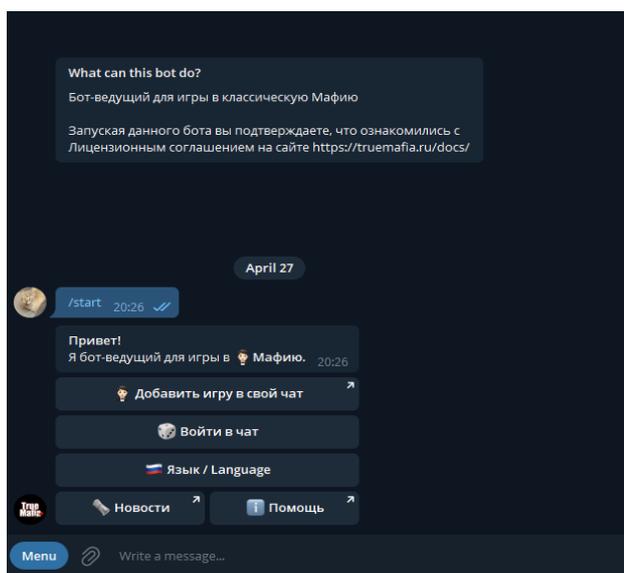


Рисунок 3 – @truemafiabot

Новостной чат-бот. @pdanews_bot – если начать диалог с этим ботом, то он начнет присылать вам сводку новостей по России каждый день или же свежие новости, если вы его попросите. Вы получаете свежие новости из проверенных источников и всегда находите в курсе событий (рисунок 4).

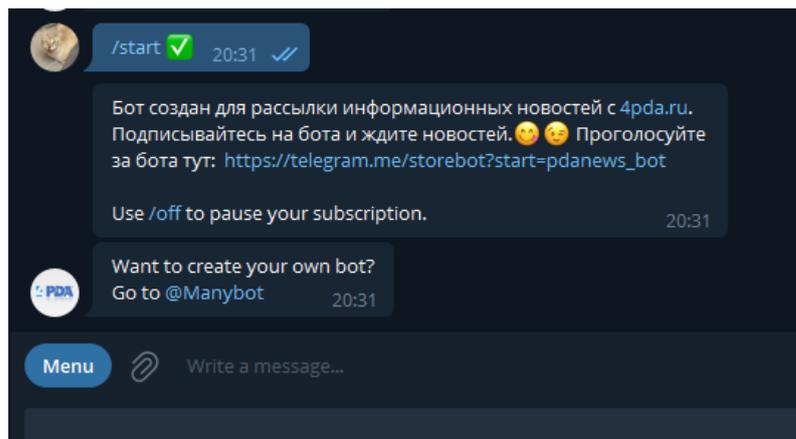


Рисунок 4 – @pdanews_bot

Чат-бот с авиабилетами. @metametabot – представьте, что вы надумали отправиться в отпуск. В какую страну вы бы смогли полететь? Достаточно просто вписать свой город и город, который хотите посетить, например: «Самара-Челябинск». Тут же вам выдаст возможность увидеть ближайшие авиа-рейсы по указанному вами запросу (рисунок 5).

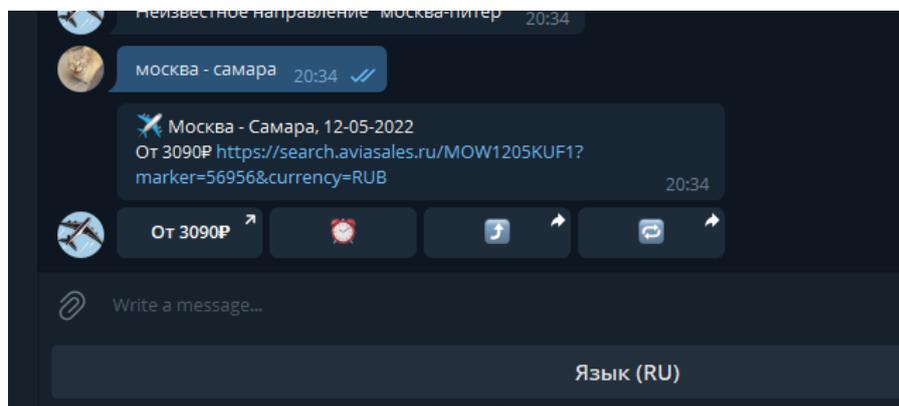


Рисунок 5 – @metametabot

1.4 Выбор среды разработки

1.4.1 Python

В полученном задании от заказчика сказано, что разработка бота ведется на 1С, но действительно ли это самый подходящий язык программирования для нашей цели?

Разберем в сравнении с языком программирования Python.

Python или же «пайтон» стоит на высоком уровне среди языков программирования. Он является полноценным скриптовым языком программирования. Его уникальность в том, что он универсален и подходит для решения различных сложных задач. Пайтон применяется во многих IT сферах, например: машинное обучение, разработка различных приложений, так же в вебе и т.д. [8].

Плюсы Python. Python это интерпретируемый язык. Он не компилируется как обычный язык. Это значит, что пайтон до запуска кода является просто текстом. Следовательно, на нем можно кодить почти везде, на всех платформах. Сам язык отлично спроектирован и логичен.

Примечателен язык также в том, что строчки кода сокращены до минимума. Простота языка это очень хорошее качество, так как разработка программы ведется гораздо быстрее. Вот в пример отрывок кода, написанный на языке Java. Здесь показан вывод в консоль «Hello, Otus!» (рисунок 6).

```
class Main {  
    public static void main(String[] args) {  
        System.out.println("Hello, Otus!");  
    }  
}
```

Рисунок 6 – Пример Java-кода

В Python для выполнения того же самого кода придется написать всего одну строчку (рисунок 7).

```
print("Hello, Otus!")
```

Рисунок 7 – Пример Python-кода

Именно поэтому и считается, что данный язык программирования хорошо подходит для начинающих IT-специалистов.

Как применяют Python? Как уже было сказано выше, Пайтон универсален, его используют практически везде. Как в web разработке, так и на мобильных программах, а также при разработке ИИ (искусственного интеллекта).

Минусы языка программирования Python. Несмотря на множество плюсов (простота, лаконичность, универсальность), у языка есть и минусы.

Скорость слабая сторона языка. Например язык Swift, который используется, например, для разработки программ на iPhone ОС, может выполнить работу в 8 раз быстрее, чем конкурент. Пайтон не очень хороший вариант для выполнения задач, требующих больших объёмов памяти. Такого типа задачи лучше выполнять при помощи языка Си или Си++.

Язык слишком сильно зависит от системных библиотек. Из чего следует то, что возникают сложности при переходе на другие системы. Конечно, данное затруднение можно исправить, используя Virtualenv, но у этого способа свои недочеты есть: костыли, повторение системных библиотек, переизбыток полных методов изоляции.

Глобальный блок интерпретатора (GIL) не позволяет одновременно выполнить несколько потоков Пайтон в реализации CPython, Но Global Interpreter lock можно отключить на время, как например это сделано в пакете NumPy для математики.

Вывод: можно уверенно сказать, Пайтон – популярный и универсальный язык программирования. Он востребован во многих сферах. Высокоуровневый язык программирования со своими достоинствами и недостатками.

1.4.2 Java

Java является языком ООП подхода, то есть следует объектно-ориентированной парадигме. Это делает его языком общего назначения. Java придерживается принципа: «Написал один раз, а потом используешь где хочешь». Язык используется в различных сферах, будь то мобильные приложения или десктопные, сетевые или корпоративные.

Плюсы языка.

ООП (Объектно-ориентированное программирование). Данный язык программирования имеет концепцию ООП. Таким образом мы можем не просто указать тип данных и структуру, а также определить некий набор функций присущий его типу. Тем самым наша структура данных стала объектом, с которым можно взаимодействовать для создания связей с другими объектами.

Java – высокоуровневый язык с довольно простым синтаксисом и хорошим подходом к освоению. Язык похож на наш привычный разговорный. Такие языки преобразуются с помощью компиляторов и интерпретаторов, что делает JAVA языком высокого уровня. Это значительно позволяет упростить разработку, делая ее простой для писания кода, а также его чтения и модерирования.

Стандарт для корпоративных вычислительных систем. Отличительной характеристикой Java еще с 90х годов являются корпоративные приложения. В те года компании начали поиск безопасного и надежного инструмента программирования, отличное от Си. Java имеет очень много различных библиотек в свое арсенале. Они помогают в разработке функций, которые, несомненно, помогут организации. Язык широко известен по миру. Его

использует и преподают в большинстве университетов и даже школ. В ТГУ тоже преподают данный язык. Его возможности удивляют, многие провайдеры предпочитают Java, так как язык легок в обслуживании, работать с ним можно с любой системы и он дешевый в администрировании.

Стабильность и сообщество. Oracle – компания, которой принадлежат права на язык программирования. Большое количество времени корпорация обильно поддерживает развитие языка и развивает сообщество вокруг него. Этому способствует изобилие различных приложений на JVM. Постоянно выходят обновления и новые версии с новыми возможностями.

Многопоточность. Самой малой единицей обработки данных в программировании является поток. Для того чтобы максимизировать КПД процессора, люди придумали запускать несколько потоков одновременно. Их количество будет зависеть от мощности процессора и кол-ва его ядер. Java это умеет, а значит ещё один плюс в копилку.

Автоматическое управление памятью (АММ). Чтобы писать код для управления памятью на Java людям не нужно делать это руками, как например в Си. АММ (Automatic Memory Management) также используется в языке Swift и при очистке памяти приложениями, которые этим занимаются.

Минусы языка.

Платное коммерческое использование. Компания Oracle не так давно объявляла, что начнет брать деньги за свой продукт Java 8 в целях улучшения программного обеспечения. За новые обновления и фикс багов придется платить. Сумма будет разной для разных групп лиц. Заявление было еще в 2019г. и с того времени ни что не мешало мне пользоваться языком. Так что возможно в будущем меня это коснется.

Многословный и сложный код. Код очень большой, это так. Длинные и чрезвычайно сложные операторы затрудняют изучение. Многие считают, что так проще будет разобраться в коде, но для меня это не так. Как и C++, Java высокоуровневый язык программирования. Им свойственно иметь большой развернутый код. Так его лучше будет понять неспециалистом, ведь они

смогут детальней разобрать операторы. Но это делает код громоздким и менее компактным.

Низкая производительность. Как и у своих сверстников Java имеет довольно низкую производительность. Все языки высокого уровня имеют низкую производительность. Это связано с компиляцией при помощи виртуальной машины. Но это не единственная проблема. Если плохо настроить процедуру хеширования это приведет к чрезмерному использованию памяти. Может произойти взаимная блокировка потоков. Виной этому послужит попытка потока получить доступ к одному ресурсу. Здесь появится ошибка из-за нехватки памяти, но это можно решить если умело планировать работу.

1.4.3 1С

Данный язык – это язык программирования, который интегрирован в программу «1С:Предприятие» и используется в ряду приложений от компании 1С. Начиная с версии 7, 1С стал интерпретируемым языком. Стал языком сверхвысокого уровня. Тогда интерпретировалась только часть программного кода, ведь сама интерпретация выполнялась непосредственно по мере обращения к нужному модулю во время работы.

Сами модули и код в них находятся в своей среде разработки. В конфигураторе 1С:Предприятие. Он является одной из основных частей всей программы.

Синтаксис платформы написан полностью на русском языке. От версии к версии он немного меняется, но по-прежнему совместим с новыми версиями. Есть нюанс при переносе кода с 7 версии на 8. В этом по большому счету нет смысла, так как работа с прикладными объектами данных значительно отличается, хотя весь синтаксис не изменен. Язык имеет кучу схожестей с другими. Например, с Pascal или Basic.

Пример вывода сообщения в специальное окно для версии 8.3 (рисунок 8):

```
Процедура ПриветМир()  
    Сообщить("Здравствуй, Мир!");  
КонецПроцедуры
```

Рисунок 8 – Пример 1С-кода

Плюсы программирования на 1С.

Легко начать. Чтобы работать на этой платформе, нужно знать: язык программирования, основанный на Basic, базы данных, принципы работы самой платформы.

Для обучения этим навыкам не нужно быть продвинутым программистом. В интернете много бесплатных курсов, после которых можно работать в франчайзи. Ещё выпускаются бумажные методички разных уровней сложности.

Часто сюда входят студенты или школьники старших классов. Если они смогли это сделать, то также может и любой.

Востребована в России и СНГ. Эта профессия – вторая по необходимости в России после системных администраторов. Платформа 1С зацепилась за отечественный бизнес за счёт качества и средней стоимости программы в 18 тысяч рублей. Эта цена намного меньше, чем зарубежные аналоги.

По официальной статистике 87% рынка пользуется программами 1С. Их продукты есть в каждом городе, которые нужно обслуживать.

Минусы языка.

Платформа востребована только в России и СНГ. 1С разработана отечественными разработчиками и начала внедряться именно в РФ и странах Союза. Компания пытается выйти на европейские рынки, но её там не

пропускают конкуренты.

По своему качеству 1С уступает платформам Европы. Их функционал шире, чем у отечественного. Ещё они изначально заточены на законодательство европейских стран, а 1С – нет.

Для комфортной работы нужно разбираться в других отраслях. Быть программистом — мало для 1С. Писать код без понимания цели – значит тратить время зря. Чтобы разрабатывать эффективные программы на платформе, нужно разбираться в, никак не связанных с программированием, областях.

Помимо языков программирования, специалисту нужно изучить: бухучёт, кадровый и складской учёт. Это занимает то время, которое можно потратить на свои дела. Например, на практику.

Подведём итог. Языки программирования Python и Java выигрывают у конкурента по нескольким параметрам:

- языки легче в освоение (исключительно моё личное мнение);
- удобнее искать информацию в справочниках и в интернете;
- более функционален и удобен;
- Python больше подходит для создания Telegram-бота.

1С:Предприятие:

- разработан отечественными разработчиками;
- не требует знания даже начального английского;
- платформа востребована для бизнеса.

Опираясь на факты и личный опыт, для разработки бота я бы выбрал язык программирования Python. Но есть пару весомых нюансов. Первое и не мало важное это условия заказчика. Компания указала язык разработки в методическом пособии, а от этого мы уйти не можем.

Второе заказчиком является фирма по производству фармацевтики, а как мы знаем 87% компаний в России используют в своей деятельности 1С:Предприятие и другие продукты компании 1С. Что значит, что от этого отходить нельзя.

Получаем, что для реализации Телеграмм-бота я бы выбрал язык Python, но для нашей конкретно цели точно подходит встроенный в 1С:Предприятие язык программирования.

1.5 Telegram Bot API

Это обычный HTTP-interface для взаимодействия с чат-ботами в мессенджере Telegram. Каждый созданный бот должен уметь автоматически отправлять сообщения и принимать входящие. Для этого нам и нужен сервер, принимающий http запросы и выдающий ответы.[24]

Есть всего 2 способа получать обновления с сервера:

– long pulling – телеграмм сам будет раз в какое-то время, например в 1 секунду, спрашивать сервер о новых уведомлениях;

– webhook – или сервера сами будут присылать оповещение, сразу как появится новое уведомление [4].

Все оповещения будут сохранены на сервере без обработки, но не долго, максимум до 24 часов. В ответ будет отправлен объект «Апдейт», заранее сериализованный в JSON (об этом подробнее в главе 2).

Все запросы к Telegram Bot API должны осуществляться через HTTPS в следующем виде: https://api.telegram.org/bot<token>/НАЗВАНИЕ_МЕТОДА. Принцип работы взаимодействия чат-бота и пользователя изображен на рисунке 9 [18].

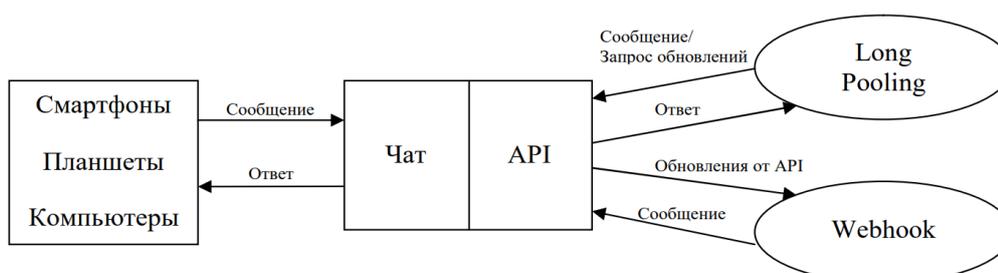


Рисунок 9 – Принцип работы чат-бота на платформе Telegram

Чтобы получить для нашего бота ключ доступа, он же токен, необходимо написать botfather боту. Он создан для того, чтобы создавать ботов. Ниже приведены некоторые примеры методов:

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

– `sendMessage` – метод осуществляет обычную отправку сообщения пользователю.

Вывод к разделу 1: В данном разделе были рассмотрены методы разработки. Данное исследование позволило сделать ряд выводов. Были изучены мессенджеры и их актуальность. Был проведен анализ полезности чат-ботов. Был подобран самый подходящий язык программирования.

2 Разработка телеграмм-бота

2.1 Регистрация чат-бота

Для начала нам потребуется зарегистрировать нашего бота в телеграм, то есть нам нужно получить токен бота, он же ID уникальный идентификатор. В этом может помочь специальный бот @BotFather [10].

Как это сделать:

1. Пишем в бота команду /newbot.
2. Далее пишем команду /newbot (рисунок 10).

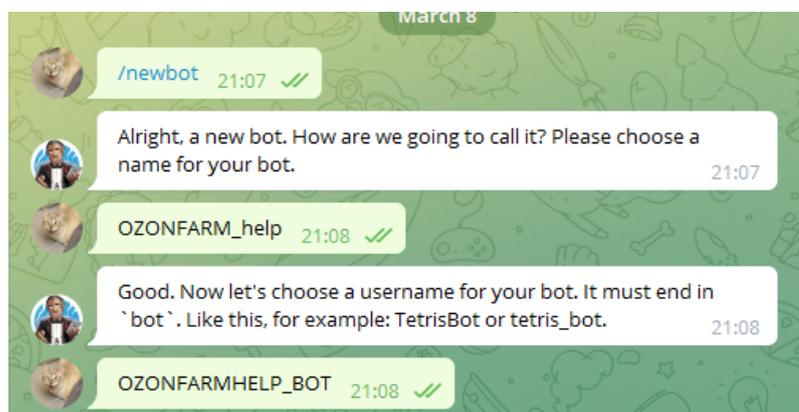


Рисунок 10 – Регистрация бота

3. Придумываем имя для нашего бота. Отмечу что имя бота должно оканчиваться на «bot», иначе бота создать не удастся.
4. Помощник пришлет нам поздравления и выдаст токен (рисунок 11), с помощью которого мы сможем взаимодействовать с «внутренностями» нашего бота.

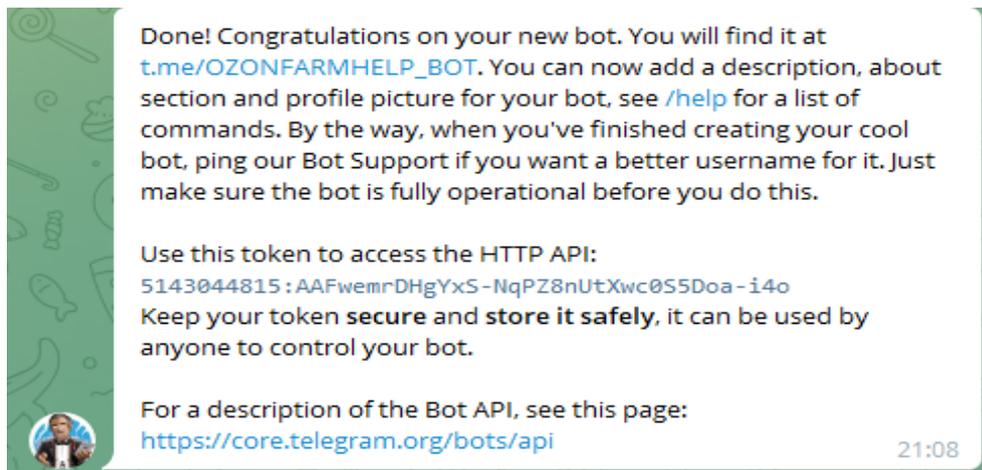


Рисунок 11 – Токен бота

Полученные данные можно проверить, для этого воспользуемся методом `getMe`.

`getMe` – это простой метод для проверки токена аутентификации бота. Он возвращает основную информацию о нём в виде объекта `User`.

Введя команду: `<https://api.telegram.org/bot5167619762:AAHPWTY3YvYujfJSUEedwKfOcsFiln3sbO0/getMe>` с использованием токена бота в строку браузера мы получим следующее:

```
{"ok":true,"result":{"id":5167619762,"is_bot":true,"first_name":"testozonfarm","username":"testozonfarmbot","can_join_groups":true,"can_read_all_group_messages":false,"supports_inline_queries":false}}
```

Можем узнать имя бота, `username` (по которому люди смогут найти бота в поиске мессенджера), параметры бота для присоединения к группам и чатам и само состояние бота.

Готово, теперь у нас есть токен для дальнейшей работы.

2.2 Подготовка платформы к работе и заполнение конфигурации

Для реализации бота будем использовать учебную версию платформы 1С:Предприятие. Для это необходимо на официальном сайте 1С получить ссылку на скачивание, заполнив небольшую анкету. После установки необходимо создать информационную базу (рисунок 12).

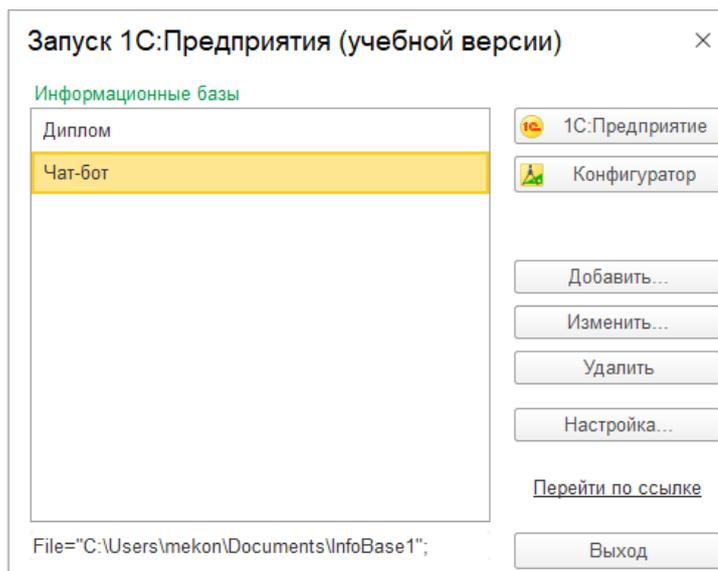


Рисунок 12 – Информационные базы 1С

После создания открываем нашу базу в режиме конфигуратора. Сейчас наша конфигурация пуста. Следующим шагом будет добавление в нашу конфигурацию справочников, регистров сведений и других элементов конфигурации.

В полученном техническом задании от заказчика присутствует минимальный набор метаданных. Требуется задать справочники, регистры сведений, константы которые потребуются для нашего бота [21].

Справочники – это хранилища информации в 1С. Эти прикладные объекты способны, например: записать в себя какой-то список товаров или заполнить себя именами людей, обучающихся в университете. Они имеют списочный характер.

Справочники:

Обращения.

Реквизиты:

НомерОбращения Строка(150)

Пользователь СправочникСсылка(Пользователи)

Текст Строка(0)

Пользователи

Ревизиты:

ИмяПользователя Строка(150)

ИдентификаторПользователя (идентификатор из телеграмма) строка(150)

Добавляем новый справочник Обращения с реквизитами НомерОбращения, Пользователь и Текст (рисунок 13).

Этот справочник в будущем пригодится для записи обращений в историю. Позволит видеть, кто и что прислал.

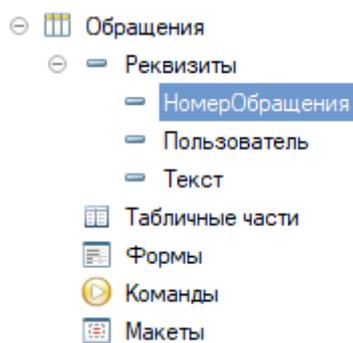


Рисунок 13 – Справочник Обращения

Далее создаем второй справочник Пользователи. В будущем его функцией будет хранение информации о пользователях телеграм. Справочник содержит реквизиты ИмяПользователя и ИдентификаторПользователя, он же user_ID (рисунок 14).

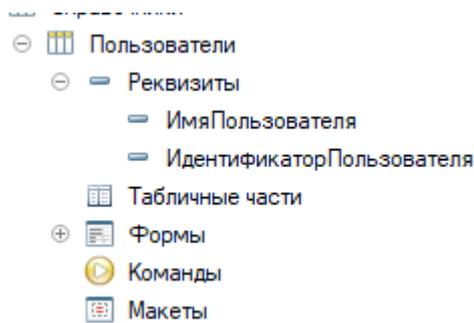


Рисунок 14 – Справочник Пользователи

Создадим Регистры сведений. Регистры сведений – это почти те же самые справочники, но они позволяют хранить в прикладном решении произвольные данные в разрезе нескольких измерений. Например, в регистрах можно хранить курс валют в разрезе валют или цены предприятия в разрезе номенклатуры и типы цен.

РегистрыСведений:

1.СообщенияТелеграмм (периодический независимый)

Хэш: УникальныйИдентификатор

Пользователь: СправочникСсылка(Пользователи)

ИдентификаторПользователя строка(150)

Ресурсы: ТекстСообщения Строка(0)

2. ОчередьУведомленийТелеграмм

Обращение СправочникСсылка Обращения

Хэш: УникальныйИдентификатор

Первый регистр сведений будет СообщенияТелеграмм. Он послужит для записи информации о сообщении, и её периодическом хранении. Также этот регистр будет использоваться для записи истории сообщений. Он будет содержать такие параметры как: хеш, пользователь, текст сообщения (рисунок 15).

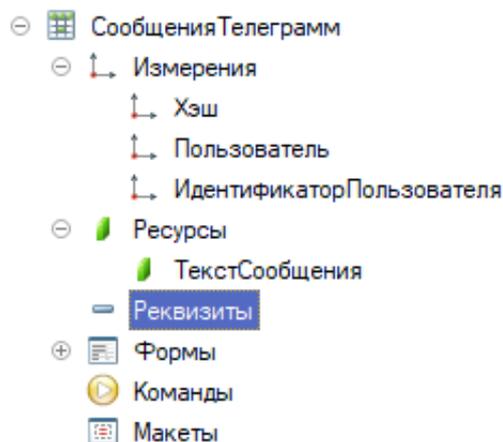


Рисунок 15 – Регистр сведений Сообщения Телеграмм

Второй регистр сведений, который нам понадобится это `Очередь Уведомлений Телеграмм`. Он пригодится нам для создания очереди уведомлений и запросов в наш бот. Он будет содержать такие параметры как: обращение, хеш, и параметр `Уведомление Отправлено` (Рисунок 16) [6].

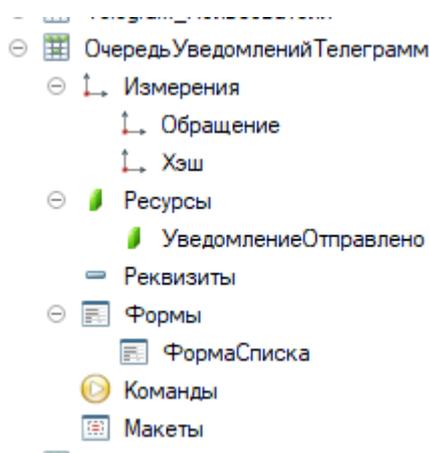


Рисунок 16 – Регистр сведений Очередь Уведомлений Телеграмм

Обработки – это прикладные объекты конфигурации. Они предназначены для выполнения различных действий над информацией.

Можно, например, импортировать данные из других систем. Или удалить из системы старую информацию. Отсюда и название объекта –

Обработка, потому что после ее выполнения данные меняются, добавляются, переносятся или удаляются.

Форм для обработки может быть несколько, с помощью них мы обычно, при необходимости, делаем запись параметров, которые влияют на работы алгоритма. А вывод на экран информации осуществляется из конструктора запроса с обработкой результата.

Главным различием отчета и обработки в том, что первый может использовать схему компоновки данных, в отличие от второго. Ну а больше различий в них и нет.

Создадим обработку Телеграмм бот. Данная обработка будет отвечать за обработку ожиданий, которая будет проверять наличие обновлений от сервера методом `getUpdate`.

Этот метод используется для получения обновлений через `long polling`. Ответ возвращается в виде массива объектов `Update` [3].

В модули обработчика пропишем процедуры. Процедура запуск обработчика ожиданий отвечает за получение обновлений от сервера с параметром времени между обновлениями. Также создадим кнопку в форме для удобной активации обработчика (рисунок 17) [5].

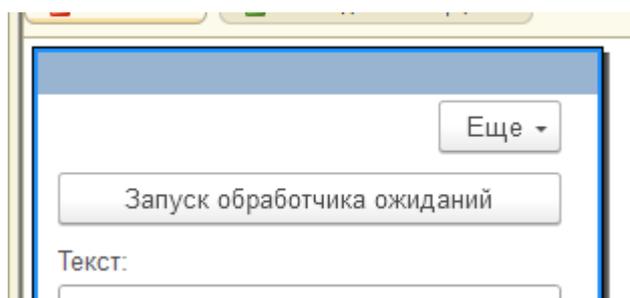


Рисунок 17 – Кнопка запуска обработчика ожиданий

Создаем процедуры которые будут отвечать за читку сообщения от пользователя телеграмм и читку сообщений на сервере (рисунок 18).

```

&НаСервере
[ Процедура ПрочитатьСообщениеНаСервере ()
  Telegram_Сервер.ПрочитатьСообщениеПользователя ();
КонецПроцедуры

&НаКлиенте
[ Процедура ПрочитатьСообщение (Команда)
  ПрочитатьСообщениеНаСервере ();
КонецПроцедуры

&НаКлиенте
[ Процедура ЗапускОбработчикаОжиданий (Команда)
  ПодключитьОбработчикОжидания ("ОбработчикОжиданияПроцедура", 5);
КонецПроцедуры

&НаКлиенте
[ Процедура ОбработчикОжиданияПроцедура ()
  Telegram_Сервер.ПрочитатьСообщениеПользователя ();
КонецПроцедуры

```

Рисунок 18 – Часть модуля обработчика

Также добавляем процедуру, которая имеет возможность разослать пользователям сообщение. В форму вводится текст сообщения и рассылается всем действующим пользователям в телеграмм (рисунок 19).

```

&НаСервере
[ Процедура ОтправитьСообщениеПользователюНаСервере ()
  Запрос = Новый Запрос;
  Запрос.Текст = "ВЫБРАТЬ
  |   Telegram_Пользователи.ИмяПользователяТелеграм КАК ИмяПользователяТелеграм,
  |   Telegram_Пользователи.ID_Пользователя КАК ID_Пользователя,
  |   Telegram_Пользователи.Пользователь КАК Пользователь
  |ИЗ
  |   РегистрСведений.Telegram_Пользователи КАК Telegram_Пользователи";

  ВыборкаЗапроса = Запрос.Выполнить().Выбрать();
  Пока ВыборкаЗапроса.Следующий() Цикл
    Telegram_Сервер.ОтправитьСообщениеПользователю (ВыборкаЗапроса.ID_Пользователя, Текст);

  КонецЦикла;
КонецПроцедуры

```

Рисунок 19 – Процедура для отправки сообщений

Константы – это базовые объекты любого языка программирования, в том числе 1С. Они хранят в себе определённый тип данных и никогда не изменяются, только если мы их изменим конечно. Каждая хранит одно значение. Нам таких понадобится три.

Добавим 3 новые константы ИдентификаторБота, АдресСервера и ИмяБота (рисунок 20).

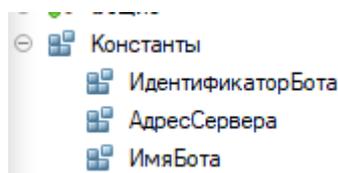


Рисунок 20 – Константы

Эти константы понадобятся нам для хранения таких параметров, которые не изменятся. Это будет имя нашего бота, его токен и адрес сервера, он же `api.telegram.org`.

На этом мы заканчиваем с первичным заполнением конфигурации и приступаем к реализации основного функционала бота. Для этого нам понадобится общий модуль, где мы и будем писать весь основной код.

2.3 Реализация основного функционала чат-бота

Начнем с сериализации и десериализации JSON файлов. Мы должны понимать, что обмен данными с сервером происходит при помощи JSON.

JSON – это один из способов как мы можем представить какие-либо данные. Мы можем хранить данные в таблице создавать столбцы и так далее, а JSON является альтернативой хранения данных. Пример структуры JSON на рисунке 21 [19].

```
{
  "firstName": "Иван",
  "lastName": "Иванов",
  "address": {
    "streetAddress": "Московское ш., 101, кв.101",
    "city": "Ленинград",
    "postalCode": "101101"
  },
  "phoneNumbers": [
    "812 123-1234",
    "916 123-4567"
  ]
}
```

Рисунок 21 – Формат JSON

JSON – это аббревиатура и расшифровывается как JavaScript Object Notation. Он не является единственным способом представления данных. Есть аналоги, такие как XML, YAML и другие. Нужно понимать, что этот формат хранения данных работает не только с JavaScript, он стал настолько популярным, что его начали поддерживать и другие языки программирования, такие как PHP. На сегодняшний день JSON является одним из лучших методов обмена данными между сервером и клиентом в Интернет [23].

Чтобы работать с JSON объектами нужно научиться их сериализовать и десериализовать. Для этого напишем процедуры [9].

Чтобы читать полученные данные от сервера создадим функцию для десериализации JSON (рисунок 22) [17].

```

//Десериализуем JSON
Функция ДесериализоватьJSON(СтрокаJSON)

    Чтение = Новый ЧтениеJSON;
    Чтение.УстановитьСтроку(СтрокаJSON);
    Данные = ПрочитатьJSON(Чтение, Ложь);
    Чтение.Закрыть();

    Возврат Данные;

Конецфункции

```

Рисунок 22 – Десериализация JSON

Также создадим функцию для записи JSON (рисунок 23).

```

// ЗаписатьJS
Функция ЗаписатьJS(СтруктураJS)

    ЗаписьJSON = Новый ЗаписьJSON;
    ЗаписьJSON.УстановитьСтроку();
    ЗаписатьJSON(ЗаписьJSON, СтруктураJS);
    Возврат ЗаписьJSON.Закрыть();

Конецфункции

```

Рисунок 23 – Записать JSON

Создадим процедуру ПрочитатьСообщениеПользователя(). Нам нужно будет установить HTTP соединение и связать это с методом getUpdates и получением JSON объекта [15].

Установим соединение.

Источник = "bot" + СтруктураНастроек.token + "/getUpdates";

HTTPСоединение = Новый

HTTPСоединение(СтруктураНастроек.api, 443, ,, , Новый

ЗащищенноеСоединениеOpenSSL());

HTTPЗапрос = Новый HTTPЗапрос(Источник);

HTTPЗапрос.Заголовки.Вставить("Content-type", "application/json");

```
HTTPОтвет = HTTPСоединение.Получить(HTTPЗапрос);
Далее переходим к десериализации JSON объекта.
Если HTTPОтвет.КодСостояния = 200 Тогда
Данные = ДесериализоватьJSON(HTTPОтвет.ПолучитьТелоКакСтроку());
Если Данные.ok И Данные.result.Количество()>0 Тогда
//Запишем историю сообщений
Для каждого СтрокаМассива Из Данные.result Цикл
Хэш = СтрокаМассива.update_id;
Если ПроверитьНаличиеСообщенияПоХэш(Хэш) Тогда
Продолжить;
КонецЕсли;
Если СтрокаМассива.Свойство("message") Тогда
ЗаписатьИсториюСообщений(СтрокаМассива.message,Хэш);
Иначе КонецЕсли; КонецЦикла; КонецЕсли; КонецЕсли;
```

Создадим процедуру для отправки сообщений из 1С. Пользователи телеграм бота получают сообщение, написанное нами. Для этого создадим форму в обработчике, а точнее кнопку в уже существующей форме. Назовём её «Отправить сообщение пользователю» (рисунок 24).

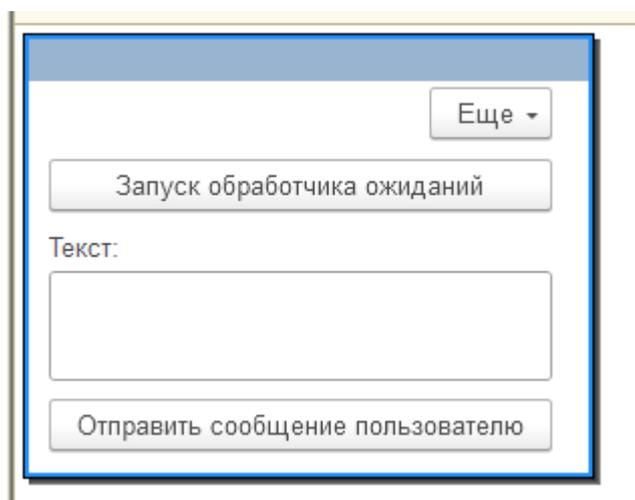


Рисунок 24 – Кнопка в форме обработчика

```

МассивКнопок = Новый Массив;
Кнопки = Новый Массив;
Для каждого кнопка ИЗ МассивКнопок Цикл
Кнопки.Добавить(Новый Структура("text, callback_data", кнопка,
СтрЗаменить(кнопка, " ", "")+"/"+"Истина"+" "/"+"Ложь"));
КонецЦикла;
Далее используем нашу функцию для серелизации JSON.
Строки = Новый Массив;
Строки.Добавить(Кнопки);
КнопкиJs = ЗаписатьJS(Новый Структура("inline_keyboard", Строки));

```

Далее передаем данные на сервер с помощью HTTP запросов.

```

Приемник = "bot" + СтруктураНастроек.token + "/sendMessage?chat_id=" +
СтрЗаменить(Формат(ИдентификаторПользователя, "ЧДЦ=; ЧС=; ЧРГ=."),
".", "") + "&text=" + Текст + "&reply_markup="+КнопкиJs;
HTTPСоединение = Новый
HTTPСоединение(СтруктураНастроек.api,443,,,,,Новый
ЗащищенноеСоединениеOpenSSL());
HTTPЗапрос = Новый HTTPЗапрос(Приемник);
Ответ = HTTPСоединение.Получить(HTTPЗапрос);
ЗаписатьИсториюОтветов(Текст,"");

```

Для создания истории сообщений и ее отображения сделаем 2 процедуры. Проясним отличие процедур от функций в 1С. Функция должна что-то возвращать, а процедура просто выполняет операторы заложенные внутри нее.

Итак, создаем первую процедуру «ЗаписатьИсториюСообщений». Данная процедура будет отвечать за фиксацию истории сообщений отправленных нами. Она будет фиксировать Дату отправки, время, хеш,

текст, ID пользователь и имя пользователя. Всё это функция запишет в регистр сведений.

Для записи данных в регистр воспользуемся менеджером записи.

```
МенеджерЗаписи =  
РегистрыСведений.СообщенияТелеграмм.СоздатьМенеджерЗаписи();  
МенеджерЗаписи.Период = ТекущаяДата();  
МенеджерЗаписи.Пользователь =  
СокрЛП(СтруктураСообщения.chat.first_name);  
МенеджерЗаписи.Хэш = Хэш;  
МенеджерЗаписи.ТекстСообщения = СокрЛП(СтруктураСообщения.text);[1]
```

Подобно первой процедуре создадим вторую для записывания входящих сообщений от пользователей. По аналогии будет записываться текст, пользователь, ID, хэш, дата и время (рисунок 25).

```
Процедура ЗаписатьИсториюОтветов (ОтветСистемы, Хэш)  
  
    МенеджерЗаписи = РегистрыСведений.СообщенияТелеграмм.СоздатьМенеджерЗаписи();  
    МенеджерЗаписи.Период = ТекущаяДата();  
    МенеджерЗаписи.ТекстСообщения = ОтветСистемы;  
    МенеджерЗаписи.Пользователь = Справочники.Пользователи.ПустаяСсылка();  
    МенеджерЗаписи.Хэш = Хэш;  
  
    МенеджерЗаписи.Записать (Истина);  
КонецПроцедуры
```

Рисунок 25 – Процедура записи истории сообщений

Создадим функцию для получения настроек подключения. Она будет получать необходимые данные для связи с сервером из наших констант. То есть токен бота, адрес сервера и имя.

```
СтруктураНастроек.Вставить("ИмяБота","");
СтруктураНастроек.Вставить("token","");
СтруктураНастроек.Вставить("api","");
Запрос = Новый Запрос;
Запрос.Текст = "ВЫБРАТЬ
|     ИмяБота КАК bot,
|     ИдентификаторБота КАК token,
|     АдресСервера КАК api
|ИЗ
|     Константы КАК Константы";
```

Для того чтобы регистрировать новых пользователей телеграм в нашу конфигурацию нам понадобится функция, которую назовем «ВернутьСоздатьПользователяTelegram». При включенном обработчике ожиданий она будет фиксировать новые обращения в бота. Если увидит новый ID то внесет его в регистр.

```
Запрос = Новый Запрос;
Запрос.Текст = "ВЫБРАТЬ
|Telegram_Пользователи.Пользователь КАК Пользователь
|ИЗ
|РегистрСведений.Telegram_Пользователи КАК Telegram_Пользователи
|ГДЕ
|Telegram_Пользователи.ID_Пользователя = &ID_Пользователя";
Запрос.УстановитьПараметр("ID_Пользователя",ID_Пользователя);
ВыборкаЗапроса = Запрос.Выполнить().Выбрать();
```

Используем менеджер записи для добавления в регистр новых данных о пользователях.

```
МенеджерЗаписи =  
РегистрыСведений.Telegram_Пользователи.СоздатьМенеджерЗаписи();  
МенеджерЗаписи.ИмяПользователяТелеграм = ИмяПользователяТелеграм;  
МенеджерЗаписи.ID_Пользователя = ID_Пользователя;  
МенеджерЗаписи.Пользователь = Пользователь;  
МенеджерЗаписи.Записать(Истина);[2]
```

В ТЗ сказано установить обновление данных раз в 30 секунд, для этого в модуле обработчика добавляем процедуру с временем ожидания.

```
Процедура ЗапускОбработчикаОжиданий(Команда)  
ПодключитьОбработчикОжидания("ОбработчикОжиданияПроцедура",30);  
КонецПроцедуры
```

Также нужно чтобы конфигурация выдавала оповещение о поступлении новых обращений. В общем модуле создаем процедуру с подпиской на событие.

```
Процедура ПодпискаНаСобытие1ПередЗаписью(Источник, Отказ,  
Замещение) Экспорт  
Сообщить ("Пришло новое сообщение!!!")  
КонецПроцедуры
```

Для этого дополнительно создаем в конфигурации подписку на событие с видом события «ПередЗаписью» (рисунок 26).

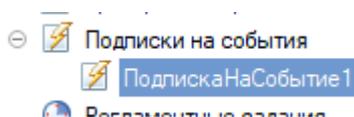


Рисунок 26 – Подписка на событие

Вывод к разделу 2: В данном разделе было реализовано проектирование и создание чат-бота. Телеграмм-бот был зарегистрирован и был получен его токен. Конфигуратор был подготовлен к работе по техническому заданию. Был спрограммирован рабочий чат-бот, выполняющий заложенный в нем функционал. Все рекомендации заказчика и требования к функционалу были учтены и реализованы.

3 Тестирование Telegram бота “helpdesk”

Заполним константы и добавим нашего бота в базу. Внесем токен, имя бота и адрес сервера (рисунок 27).

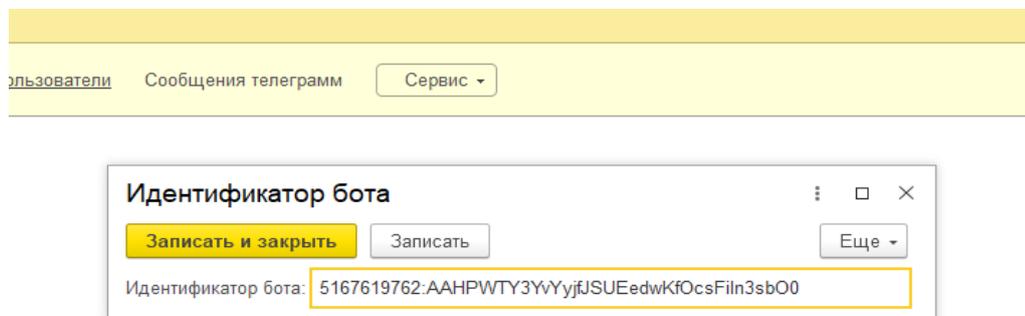


Рисунок 27 – Запись констант

После того как записали константы и данные бота добавлены в конфигурацию, нужно добавить нас как пользователя. Помним что для этого нужно запустить обработчик ожиданий и осуществить активность в телеграмме, то есть написать что-то нашему боту. У меня это будет команда «/start».

После запуска обработчика отправляем сообщение и ждем 30 секунд, потому что мы выставили 30 секунд ожидания по ТЗ. Далее перезапустим конфигурацию и проверим регистр с пользователями и историей сообщений (рисунок 28).

Имя пользователя телеграм	ID пользователя	Пользователь
Calin	291 849 950	Wyoker

Рисунок 28 – Пользователи телеграмм

Мой телеграмм аккаунт записался в регистр конфигурации.

Так же проверим нашу подписку на событие. Напишем сообщение боту и проверим, придёт ли уведомление об обращении.

Уведомление пришло, значит, подписка на событие работает (рисунок 29).

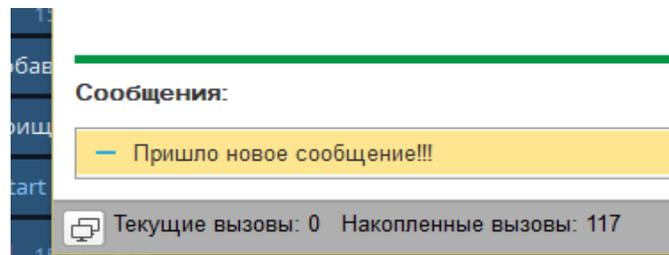


Рисунок 29 – Подписка на событие

Попробуем отправить сообщение пользователям телеграмм. Для этого в форму напишем текст сообщения, например «Привет, друг!».

Всё работает, сообщение пришло (рисунок 30).

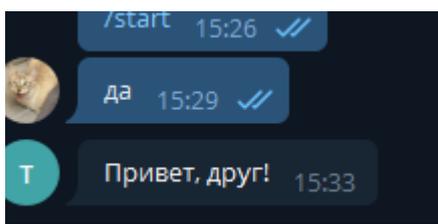


Рисунок 30 – Отправка сообщений

Что касается истории сообщений. Всё фиксируется в регистре (рисунок 31).

29.04.2022 15:36:53	790 247 486	291 849 950	/start
29.04.2022 15:36:58	790 247 486 Wyoker	291 849 950	хай
02.05.2022 15:25:38	790 247 487 Wyoker	291 849 950	еу
02.05.2022 15:25:38	790 247 488 Wyoker	291 849 950	добавь меня
02.05.2022 15:25:43	790 247 489 Wyoker	291 849 950	позиция
02.05.2022 15:26:03	790 247 490 Wyoker	291 849 950	/start

Рисунок 31 – История сообщений

Вывод к разделу 3: В третьем разделе было проведено тестирование программного продукта. Спроектированный телеграмм-бот «helpdesk» успешно прошел этапы тестирования. Он успешно справляется со всеми задачами, которые ему ставятся.

Заключение

В наше время мессенджеры находятся на своем пике популярности среди всех возрастных категорий людей. Ежедневно физические и юридические лица используют их в своих целях. От отправки смешных картинок до учетных записей крупных складов. Обмен сообщениями на огромном расстоянии далеко не новая технология, но именно мессенджеры прямо подошли к задаче, без излишеств. Это и цепляет людей. Скорость, надежность и приватность это самое главное для такого вида работы.

Чат-боты стали неотъемлемой частью этой системы. Теперь без них нам тяжело жить. Если потребуется скачать видео с интернет ресурса, то для этого найдется чат-бот. Отпала необходимость скачивания сотни приложений для каких-либо нужд. Достаточно всего лишь скачать мессенджер и начать работу с нужным ботом.

В рамках выпускной квалификационной работы были выполнены поставленные задачи. Были изучены мессенджеры и их устройство. Было проведено сравнение мессенджеров друг с другом. При сравнении был выявлен лучший в плане удобства в работе и наличия документации Telegram Bot API.

Во-вторых, были рассмотрены аналоги чат-ботов в мессенджере телеграмм. Выявлены их плюсы и минусы, а также сделаны некоторые выводы при разработке «Telegram bot helpdesk».

Было проведено сравнение средств разработки, в следствии которого была выбрана платформа 1С.

Таким образом результатом ВКР является реализованный чат-бот для платформы telegram для связи технической поддержки с персоналом фармацевтического завода ООО «ОзонФарм».

Список используемых источников

1. 1С:Предприятие 8.3. Практическое пособие разработчика. Примеры и типовые приемы [Электронный ресурс]: URL: <https://www.academia.edu/31880297> (дата обращения: 15.03.2022)
2. 1С:Предприятие 8 Конфигурирование и администрирование [Электронный ресурс]: URL: <https://docplayer.com/26266218-Soderzhanie-1s-konfigurirovanie-i-administrirovanie-chast-1-soderzhanie-1.html> (дата обращения: 15.03.2022)
3. Архитектура платформы 1С:Предприятие [Электронный ресурс]: URL: <https://v8.1c.ru/platforma/formy/> (дата обращения: 20.03.2022)
4. Бот Telegram и HTTP сервис в 1С [Электронный ресурс]: URL: <https://infostart.ru/1c/articles/1368626/> (дата обращения: 25.03.2022)
5. Процедуры в языке 1С 8.3, 8.2 [Электронный ресурс]: URL: <https://helpme1s.ru/procedury-v-yazyke-1s-8-v-primerax> (дата обращения: 08.04.2022)
6. Работа в конфигураторе 1С Предприятие 8.3 [Электронный ресурс]: URL: <https://wiseadvice-it.ru/o-kompanii/blog/articles/rabota-v-konfiguratore-1s-predpriyatie-8-3/> (дата обращения: 18.03.2022)
7. Резкий рост популярности российских мессенджеров: анализ интернет-трафика пользователей Yota 2022 год [Электронный ресурс]: URL: https://www.cnews.ru/news/line/2022-0317_rezkij_rost_populyarnosti (дата обращения: 09.04.2022)
8. Рейтинг языков программирования 2020 [Электронный ресурс]: URL: <https://techrocks.ru/2020/02/08/programming-languages-rank-2020> (дата обращения: 11.04.2022)
9. Сериализация в 1С [Электронный ресурс]: URL: <https://www.kodonline.ru/expert/programming/article-serializatsiya-v-1s/> (дата обращения: 02.04.2022)

10. Телеграм бот 1С [Электронный ресурс]: URL: <https://blagin.ru/telegram-bot-1s/> (дата обращения: 20.03.2022)
11. Телеграм-бот как простой и удобный способ получения информации [Электронный ресурс]: URL: <https://cyberleninka.ru/article/v/telegram-bot-kak-prostoy-i-udobnyy-sposob-polucheniya-informatsii> (дата обращения: 02.04.2022)
12. Чат боты и разговорный искусственный интеллект [Электронный ресурс]: URL: <https://elibrary.ru/item.asp?id=46292121> (дата обращения: 08.04.2022)
13. Чат-боты – кто они и что умеют [Электронный ресурс]: URL: <https://efsol.ru/articles/messendzhery-i-chat-boty-dlya-biznesadostavki.html> (дата обращения: 12.04.2022)
14. Чат-боты: история, современность и перспективы [Электронный ресурс]: URL: <http://w7phone.ru/chat-boty-istoriya-sovremennost-i-perspektivy-132460/> (дата обращения: 14.04.2022)
15. Язык запросов 1С:Предприятия 8 [Электронный ресурс]: URL: <https://studfile.net/preview/7285515/> (дата обращения: 18.04.2022)
16. 2016 will be the year of conversational commerce [Электронный ресурс]: URL: <https://medium.com/chris-messina/2016-will-be-the-year-of-conversational-commerce-1586e85e3991> (дата обращения: 20.04.2022)
17. Data Serialization [Электронный ресурс]: URL: https://link.springer.com/chapter/10.1007/978-1-4842-2692-6_4 (дата обращения: 22.04.2022)
18. HTTP [Электронный ресурс]: URL: https://link.springer.com/chapter/10.1007/978-1-4302-5861-2_11 (дата обращения: 04.04.2022)
19. JSON в 1С [Электронный ресурс]: URL: <https://v8.1c.ru/platforma/json/> (дата обращения: 26.04.2022)

20. Katarzyna Musiał & Przemysław Kazienko «Social networks on the Internet» [Электронный ресурс]: URL: <https://link.springer.com/article/10.1007/s11280-011-0155-z#Bib1> (дата обращения: 1.05.2022)
21. Make a Difference in a Different Way: Twitter Bot Creators and Wikipedia Transparency [Электронный ресурс]: Режим доступа: <https://link.springer.com/article/10.1007/s10606-021-09411-6> (дата обращения: 04.05.2022)
22. MTPProto [Электронный ресурс]: URL: <https://ru.wikipedia.org/wiki/MTPProto> (дата обращения: 02.05.2022)
23. Parametric schema inference for massive JSON datasets [Электронный ресурс]: URL: <https://link.springer.com/article/10.1007/s00778-018-0532-7> (дата обращения: 02.05.2022)
24. Telegram Bot API [Электронный ресурс] : URL: <https://core.telegram.org/bots/api> (дата обращения: 15.03.2022)
25. TON's history and roadmap [Электронный ресурс]: URL: <https://ton.org/roadmap> (дата обращения: 15.03.2022)