Муниципальное бюджетное общеобразовательное учреждение

«Вурнарская средняя общеобразовательная школа №1

им. И.Н.Никифорова»

Проектная работа

«Telegram бот»

по информатике

Выполнил ученик 11 класса

Александров Дмитрий Сергеевич

Руководитель:

Чарков Дмитрий Евгеньевич

Вурнары, 2023 года.

**Оглавление**

Введение 3

Глава 1. Теоретические аспекты о чат-ботах 5

1.1. История создания чат-ботов 5

1.2. Принцип работы чат-ботов 9

1.3. Виды и функционал чат-ботов 11

1.4. Современные возможности чат-ботов 13

Глава 2. Создание чат-бота в мессенджере «Telegram» 14

2.1. Этапы разработки чат-бота 14

2.2. Создания чат-бота 15

Заключение 17

Список используемой литературы 18

Приложение 19

Приложение 1 19

Приложение 2 24

# **Введение**

Чат-боты Telegram имеют высокую актуальность на сегодняшний день. С ростом использования мессенджера Telegram, многие компании используют чат-боты для общения с клиентами, выполнения заказов, уведомлений о новых продуктах и т.д. Чат-боты могут помочь автоматизировать рутинные задачи, такие как ответы на часто задаваемые вопросы, и улучшить качество обслуживания клиентов. К тому же, Telegram имеет множество инструментов для разработки чат-ботов, что делает их создание и поддержку проще и более доступными для бизнеса и отдельных пользователей. Telegram-боты становятся все более популярными, потому что они предоставляют пользователю мгновенный доступ к информации и услугам без необходимости устанавливать дополнительные приложения или взаимодействовать с человеком в режиме реального времени. Некоторые из причин, по которым Telegram-боты становятся все более актуальными:

1. Удобство: Боты работают в Telegram, одном из самых популярных мессенджеров, который используется миллионами пользователей по всему миру. Это делает их очень удобными для общения с клиентами.
2. Возможность автоматизации: Боты способны автоматизировать многие процессы, которые ранее выполнялись вручную администраторами. Это позволяет сократить затраты на персонал и ускорить время оказания услуг.
3. Низкие издержки: Создание и развертывание Telegram-ботов обходится значительно дешевле, чем разработка и поддержка мобильных приложений и веб-сайтов.
4. Разнообразие функций: Telegram-боты могут предоставлять пользователю широкий спектр услуг, таких как покупка товаров, бронирование билетов, получение новостей и др.

В целом, Telegram-боты становятся все более актуальными благодаря своей удобности, эффективности и разнообразным функциям, которые они предоставляют. Они позволяют пользователям быстро получать нужную информацию и оказывать услуги, их предпочтительней использовать для связи с клиентами.

**Цель проектной работы:** разработка и создание чат-бота для образовательной организации.

**Задачи:**

1. Исследовать возможности и функциональное предназначение интернет-мессенджера Telegram;
2. Изучить теоретические аспекты по работе чат-ботов в сети Интернет;
3. Исследовать практические инструменты и методы разработки функционала для чат-бота;
4. Создать собственного чат-бота со своим функционалом;

# **Глава 1. Теоретические аспекты о чат-ботах**

* 1. **История создания чат-ботов**

Чат-боты в том виде, в котором мы знаем их сегодня, существуют с середины 1960-х годов. Алан Тьюринг, британский ученый-компьютерщик, в 1950 году написал научную работу под названием "Компьютеры и интеллект". В этой работе ученый предположил, что компьютерные программы могут думать и говорить, как люди. Чтобы доказать это, Тьюринг предложил эксперимент, известный сегодня как тест Тьюринга, "Имитационная игра".

В эксперименте Тьюринга человек, назначенный судьей, разговаривает через компьютер с невидимым человеком и машиной. Задача судьи - отличить компьютер от реального человека. Тьюринг утверждает, что если судья не может определить, какие ответы исходят от компьютера, то это доказывает, что компьютер может имитировать человеческий язык. Он считает, что к 2020 году машины смогут легко пройти его тест.

В 1996 году Джозеф Вайзенбаум из лаборатории искусственного интеллекта Массачусетского технологического института создал ELIZA, первую в мире программу чат-бот в виде психотерапевта ELIZA ищет ключевые слова, введенные пользователем, и предварительно программирует их в соответствии с определенными правилами ELIZA предназначена для работы, изучая ключевые слова, введенные пользователем. взаимодействие с людьми путем активации запрограммированных выходов в соответствии с определенными правилами. ELIZA была разработана для изучения ключевых слов, введенных в качестве пользовательского ввода, и активации запрограммированных выходов в соответствии с определенными правилами; она не понимала структуру разговорного контекста и реагировала только путем анализа пользовательского ввода.

В 1971 году Кеннет Марк Колби, психиатр из лаборатории искусственного интеллекта Стэнфордского университета, задался вопросом, могут ли компьютеры способствовать пониманию работы мозга.

Он считал, что компьютеры могут помочь в лечении пациентов с психическими заболеваниями. Эти идеи привели Колби к разработке PARRY, чат-бота, который играл роль пациента, страдающего параноидальной шизофренией. Он обладал собственной "индивидуальностью", а также определял свои ответы на основе системы "эмоциональных реакций", вызванных изменением предположений и выражений лица пользователя. Интересно, что только 48% психиатров смогли отличить PARRY от настоящей паранойи, и в январе 1973 года американский ученый Винтон Серф решил поставить двух ботов друг против друга. В результате чат-бот ELIZA проводил психотерапию с PARRY. Этот разговор стал одной из первых вех в истории чат-ботов.

Разработанный в 1988 году программистом-самоучкой Лор Карпентер, чат-бот Jabberwacky был нацелен на поощрение естественного человеческого общения в игровой форме. В отличие от предыдущих чатботов со статичными базами данных, он учился на прошлом опыте и развивался с течением времени.

В 1992 году был создан чат-бот MS-Dos "Dr Sbaitso". Он был создан для демонстрации способности генерировать синтетическую речь с помощью звуковой карты. Разговоры с ним были похожи на беседы с психологом, а вопросы в основном были на уровне "почему вы так себя чувствуете". Бот избегал более сложных обменов идеями.

Вдохновленный работой Джозефа Вейзенбаума, другой ведущий компьютерный ученый, Ричард Уоллес, в 1995 году разработал чат-бота ALICE. Бот имеет открытый исходный код и может быть использован другими разработчиками для создания собственных чат-ботов. С этой целью более 500 добровольцев внесли свой вклад в бота и создали 100 000 строк AIML для воспроизведения ALICE в чате. ALICE также послужил вдохновением для американского научно-фантастического романтического сериала «Она».

В 2001 году были заложены основы популярной Siri, а появление бота Smarter Child показало, что люди предпочитают общаться с интеллектуальной машиной, которая дает персонализированные и полезные советы. Более 30 000 пользователей AIM и MSN общались с этим ботом, и он также использовался пользователями AIM и MSN, чтобы помочь им найти информацию о любимых сервисах.

Появление больших данных потребовало создания систем, способных обрабатывать, анализировать и извлекать необходимую информацию, и в 2006 году компания IBM начала разработку бота Watson. Этот бот мог обрабатывать естественный язык и обучаться в процессе общения. Watson мог мгновенно отвечать на вопросы, например, как открыть счет в банке или как купить подарок маме.

Количество виртуальных помощников продолжает расти с 2010 года, когда Apple запустила Siri, первого в мире персонального помощника; Google запустил GoogleNow в 2012 году; MicrosoftCortana и AmazonAlexa появились в 2014 году; а в 2017 году Яндекс представил Алису, русскоязычного голосового помощника.

В 2016 году был запущен Messenger, платформа для чат-ботов; в 2018 году LiveChat запустил ChatBot, фреймворк для создания чат-ботов без программирования. В настоящее время в Messenger активно работает более 300 000 ботов.

Осенью 2020 года Jivo стал поддерживать чат-боты. С тех пор клиенты Jivo стали полагаться на чат-ботов, чтобы снизить нагрузку на службу поддержки, обрабатывать входящие сообщения в любое время и помогать отделу продаж находить клиентов, которые их больше всего интересуют.

В результате сегодня чат-боты используются в самых разных областях - от развлечений до привлечения новых клиентов. И для достижения этих результатов технология прошла долгий путь.

* 1. **Принцип работы чат-ботов**

Чат-боты на основе искусственного интеллекта, правил автоматизации, обработки естественного языка (NLP) и машинного обучения (ML) обрабатывают данные и отвечают на запросы любого типа. Чат-боты могут быть созданы на любом языке программирования с функцией WebAPI: Node.js и PHP являются распространенными, но многие другие библиотеки поддерживают Java и Python.

Двумя распространенными типами ботов являются

**Декларативные чат-боты**, ориентированные на выполнение задач, - это программы, основная цель которых заключается в выполнении одной функции. Используя правила, обработку естественного языка, машинное обучение и другие технологии, эти чат-боты автоматически отвечают на запросы пользователей, но делают это в интерактивном режиме. Поскольку общение с чат-ботами должно быть высокоструктурированным, они в основном используются для поддержки и обслуживания, например, интерактивные, полуфункциональные сервисы ответов на вопросы.

Чат-боты, ориентированные на решение задач, могут отвечать на стандартные вопросы, например, о часах работы учреждения, или выполнять простые операции без большого количества различных переменных; они используют принципы NLP , чтобы дать пользователям возможность общаться в интерактивном режиме, но их функциональность весьма ограничена. В настоящее время эти типы чатботов являются наиболее распространенными.

**Чат-боты с предиктивными данными**, работающие в интерактивном режиме - эти чат-боты часто называют виртуальными помощниками или цифровыми ассистентами. Они более сложные и персонализированные, чем чатботы, ориентированные на решение конкретных задач. Эти чатботы чувствительны к контексту и обучаются, используя принципы понимания естественного языка (NLU), NLP и машинного обучения.

Кроме того, они используют прогностические и аналитические возможности для персонализации на основе профиля пользователя и его поведения в прошлом. Цифровые помощники также могут со временем изучать предпочтения пользователей, давать советы и прогнозировать их потребности. Они могут инициировать диалог, отслеживать данные и намерения. Примерами чат-ботов, основанных на прогнозирующих данных и ориентированных на потребителя, являются Siri от Apple и Alexa от Amazon.

Современные цифровые помощники также могут объединять данные от нескольких узкоспециализированных чат-ботов, собирать различные типы информации от каждого из них и комбинировать эту информацию для выполнения ситуационных задач, чтобы чат-бот не терял нить диалога.

* 1. **Виды и функционал чат-ботов**

Существует несколько типов чат-ботов:

* Консультанты – от простых менеджеров в интернет-магазинах до медицинских и юридических служб. В их обязанности входит общение с потенциальными и реальными клиентами, ответы на всевозможные вопросы.
* Продавцы – основная задача такого чат-бота - увеличить количество заказов. Они могут предложить покупателю приобрести товар или воспользоваться услугой. Он также может предложить дополнительные опции к заказу. При настройке интеграции с платежной системой помощник немедленно выставит счет покупателю или переведет на страницу онлайн-оплаты.
* Помощники – таких чат-ботов можно найти в различных мессенджерах и поисковых системах. Они помогают в поиске информации, а также проводят ее первичный анализ.
* Психолог – в состоянии стресса или депрессии людям может быть трудно поговорить с кем-то о своих проблемах. Чат-боты помогают многим людям в трудных жизненных обстоятельствах. Такая программа запрашивает пользователя о его настроении, самочувствии, планах на день и других аспектах психологического здоровья. Одним из самых успешных проектов в этой области является виртуальный помощник Woebot, разработанный учеными из Стэнфорда.
* Развлечением часто является виртуальный собеседник, который отвечает на вопросы. Программы обучаются на оставленных сообщениях, и со временем они могут заменить реального собеседника.
* Интерфейс – основной задачей такого чат-бота является предоставление пользователям доступа к различным бизнес-сервисам. В качестве таковых могут выступать веб-виджеты или популярные мессенджеры.
* Бизнес–чат-боты - задача этих программ - оптимизировать работу, сделав ее более эффективной.

Диапазон возможностей чат-ботов ограничен только тем, способен ли человек конкретизировать операцию и создать на ее основе пошаговый алгоритм действий. Чат-боты помогают:

Спектр возможностей чат-ботов ограничивается лишь тем, способен ли человек конкретизировать операцию и создать на ее основе пошаговый алгоритм действий. Чат-боты помогают:

* Сократить временные и трудовые затраты. Одна программа, которая решает большинство проблем, позволяет свести к минимуму количество сотрудников, получающих звонки;
* Организовать круглосуточное взаимодействие с клиентами. Чат-бот, который был разработан для конкретного бизнеса, может осуществлять транзакции в любое время дня и ночи без присмотра специалиста;
* Повысить уровень продаж. Чат-боты способны отвечать на вопросы посетителей и предлагают возможность безопасно оплатить заказ, не выходя из диалогового окна. Это способствует увеличению количества транзакций после первого посещения веб-сайта компании;
* Персонализировать общение с клиентом. Чат-бот способен анализировать персональные данные посетителя и его запросы в сети;
* Исключить человеческий фактор. Чат-бот не страдает от "плохого" настроения или отсутствия опыта в разрешении конфликтов. Его ответы соответствуют заданной тональности и содержат конкретные грамотные выражения.
* Проанализировать и систематизировать полученные результаты. По запросу система предоставит пользователю аналитическую информацию и цифры по результатам выполненной работы, систематизирует процесс продаж Диапазон возможностей чат-ботов ограничен только тем, способен ли человек конкретизировать операцию и создать на ее основе пошаговый алгоритм действий. Чат-боты помогают:
* Оптимизировать работу. Бот отвечает на распространенные вопросы посетителей сайта, переводя их специалистам только для решения нестандартных задач;
  1. **Современные возможности чат-ботов**

Чат-боты широко используются благодаря своим качествам. Они могут консультировать по широкому кругу вопросов в таких областях, как медицина, право и юриспруденция, страхование, покупки и продажи, инвестиции.

Фактически, современные боты могут ответить на любой вопрос, и все, что вам нужно сделать, - это загрузить информацию. Самой важной особенностью таких программ является то, что они работают быстро и практически безошибочно.

Одной из самых популярных функций в наши дни является органайзер, когда бот анализирует полученный код и создает собственное расписание, предлагая различные варианты распределения времени. Боты действуют как секретари, помогая экономить время и деньги.

Чат-боты есть также в поисковых системах и помощниках по обмену сообщениями. Яндекс и Telegram, например, уже давно активно предоставляют пользователям дополнительные программы. В их обязанности входит как простой поиск, так и аналитика.

В индустрии развлечений чат-боты используются в качестве интерактивных собеседников или образовательных программ, которые особенно интересны детям.

В онлайн-играх можно встретить ботов, автоматизирующих игровой процесс, где компьютер фактически играет от имени пользователя.

Современные чат-боты постоянно совершенствуются и расширяются. Мы видим их в социальных сетях и в полноценной бизнес-среде. Искусственный интеллект экономит много времени и не совершает ошибок. В эту технологию вкладывается много денег, и ее распространение очень широко.

**Глава 2. Создание чат-бота в мессенджере «Telegram»**

**2.1. Этапы разработки чат-бота**

В качестве практической части проекта я решил создать собственного чат-бота. Процесс разработки был разделен на три основных этапа: анализ, планирование и реализация.

В процессе аналитического этапа я расширял свои знания по теме исследования. Сначала я ознакомился с различными мессенджерами, представленными на российском рынке, такими как ВКонтакте, Viber, Telegram и WhatsApp, затем оценили их на простоту использования и функциональность. Нас привлек Telegram, потому что он наиболее привычен и удобен и имеет множество полезных функций, которые не предлагают другие компании.

На втором шаге я отдал предпочтение мессенджеру Telegram, потому что знал технические возможности выбранной платформы, а также она предлагает открытый и удобный API для создания ботов.

Второй этап состоял из трех основных шагов. На первом этапе я искал основную идею для бота. Я придумал несколько идей для своего чат-бота; на втором этапе я выбрал единственную идею для реализации. Я остановился на боте для образовательной организации.

На третьем этапе я посмотрели на существующих ботов и решил, что бот будет создан с использованием языка программирования.

**2.2. Создания чат-бота**

Мне необходимо было определиться на каком языке программирования удобнее и быстрее всего можно написать бота. Для этого я провел анализ, по результатам которого я определились, что мне подходит язык программирования Python.

Основные причины выбора языка программирования Python:

* Имеет широкий выбор библиотек и фреймворков, которые обеспечивают различные возможности
* Это язык высокого уровня, что означает, что он позволяет программистам писать программы более быстро и эффективнее, чем языки низкого уровня, такие как C или C++
* Имеет активное сообщество разработчиков, которые создают и поддерживают множество полезных инструментов и библиотек. Это позволяет легко найти помощь и поддержку во время разработки.

В качестве среды разработки я выбрал PyCharm.

Основные причины выбора среды разработки PyCharm:

* Среда PyCharm предоставляет удобный интерфейс отладчика, который позволяет разработчикам легко находить и исправлять ошибки.
* PyCharm имеет встроенную поддержку для тестирования кода, что позволяет создавать и запускать тесты внутри среды.
* PyCharm легко интегрируется с другими инструментами и службами, такими как GitHub, что значительно упрощает работу с проектами.

В файле settings.py буду хранить токен и ID администратора в переменных TG\_TOKEN, ID\_ADMIN

Сначала я создам файл menu\_send.py с функцией menu\_send для облегчения дальнейшего написания кода в server.py.

В 7 строке файла settings.py объявлю бота. Затем создам функции:

* file\_open – открывает файлы по переданному названию
* admin\_message – функция позволяет изменять данные о организации, которая запускается по команде «/admin».
* start\_message – функция запускается при старте бота. Она показывает пользователю главное меню.
* callback – обрабатывает нажатие кнопок.
* get\_text\_messages – принимает сообщение. Если администратор редактирует данные, то записывает их в текстовый файл, а иначе пишет «Я вас не понимаю».
* handle\_docs\_photo – принимает документ, и если этот текстовый документ отправил администратор, который включил режим редактирования.

В 140 строке запускается бот.

# **Заключение**

Итак, чтобы создать бота, который будет полезен и интересен пользователям, необходимо иметь четкую концепцию, понимание потребностей и интересов ваших целевых аудиторий, а также умение работать с API и реализовывать необходимую функциональность.

Кроме того, важно помнить о безопасности данных пользователей и соблюдении правил использования API, чтобы избежать блокировки или негативного опыта для пользователей.

Все это требует времени, усилий и опыта, но результат может быть крайне удовлетворительный, если вы посвятите достаточно усилий в создание качественного бота.

Как видно, процесс разработки игры оказался далеко не таким сложным,  
каким его часто описывают. Вся документация находится в открытом  
доступе, и найти её и ознакомится не составит труда при должном желании.  
Таким образом, игру может создать любой желающий.

# **Список используемой литературы**

1. Кузнецов В.В. Перспективы развития чат-ботов
2. Срини Джанарсанам. Разработка чат-ботов и разговорных интерфейсов
3. Тугушева Н.А. Использование чат-ботов в различных сферах повседневной жизни
4. Документация python - <https://www.python.org/doc/>
5. Документация pyTelegramBotAPI - https://pypi.org/project/pyTelegramBotAPI/

# **Приложение**

## **Приложение 1**

Содержание файла server.py

1. import telebot
2. from telebot import types
3. from menu\_send import menu\_send
4. from settings import TG\_TOKEN, ID\_ADMIN
5. bot = telebot.TeleBot(TG\_TOKEN)
6. flag = ''
7. def file\_open(name):
8. f = open('data/' + name, encoding="utf-8")
9. f = f.read()
10. return f
11. @bot.message\_handler(commands=["admin"])
12. def admin\_message(message):
13. if str(message.from\_user.id) == ID\_ADMIN:
14. buttons = types.InlineKeyboardMarkup(row\_width=1)
15. button1 = types.InlineKeyboardButton('Рассказать об организации', callback\_data='but6')
16. button2 = types.InlineKeyboardButton('Предоставить расписание', callback\_data='but7')
17. button3 = types.InlineKeyboardButton('Предоставить информацию о преподавателях', callback\_data='but8')
18. button4 = types.InlineKeyboardButton('Предоставить список направлений', callback\_data='but9')
19. button5 = types.InlineKeyboardButton('Предоставить контактные данные', callback\_data='but10')
20. buttons.add(button1, button2, button3, button4, button5)
21. bot.send\_message(message.chat.id, text='Какой пунк хотие редактировать?', reply\_markup=buttons)
22. else:
23. bot.send\_message(message.from\_user.id, 'Вы не администратор')
24. @bot.message\_handler(commands=["start"])
25. def start\_message(message):
26. flag = ''
27. bot.send\_message(ID\_ADMIN, message.from\_user.id)
28. bot.send\_message(message.chat.id, text='Здравствуйте!\nВы запустили бота')
29. menu\_send(message.chat.id, bot)
30. @bot.callback\_query\_handler(func=lambda call: True)
31. def callback(call):
32. global flag
33. # bot.delete\_message(call.message.chat.id, call.message.message\_id)
34. if call.data == 'but1':
35. name = 'information.txt'
36. text = file\_open(name)
37. buttons = types.InlineKeyboardMarkup(row\_width=1)
38. button = types.InlineKeyboardButton('Вернутся в меню', callback\_data='but11')
39. buttons.add(button)
40. bot.send\_message(call.from\_user.id, text=text, reply\_markup=buttons)
41. elif call.data == 'but2':
42. name = 'schedule.txt'
43. text = file\_open(name)
44. buttons = types.InlineKeyboardMarkup(row\_width=1)
45. button = types.InlineKeyboardButton('Вернутся в меню', callback\_data='but11')
46. buttons.add(button)
47. bot.send\_message(call.from\_user.id, text=text, reply\_markup=buttons)
48. elif call.data == 'but3':
49. name = 'teachers.txt'
50. text = file\_open(name)
51. buttons = types.InlineKeyboardMarkup(row\_width=1)
52. button = types.InlineKeyboardButton('Вернутся в меню', callback\_data='but11')
53. buttons.add(button)
54. bot.send\_message(call.from\_user.id, text=text, reply\_markup=buttons)
55. elif call.data == 'but4':
56. name = 'directions.txt'
57. text = file\_open(name)
58. buttons = types.InlineKeyboardMarkup(row\_width=1)
59. button = types.InlineKeyboardButton('Вернутся в меню', callback\_data='but11')
60. buttons.add(button)
61. bot.send\_message(call.from\_user.id, text=text, reply\_markup=buttons)
62. elif call.data == 'but5':
63. name = 'contacts.txt'
64. text = file\_open(name)
65. buttons = types.InlineKeyboardMarkup(row\_width=1)
66. button = types.InlineKeyboardButton('Вернутся в меню', callback\_data='but11')
67. buttons.add(button)
68. bot.send\_message(call.from\_user.id, text=text, reply\_markup=buttons)
69. elif call.data == 'but6' and str(call.from\_user.id) == ID\_ADMIN:
70. flag = 'information.txt'
71. text = 'Введите текст или отправьте файл в формате txt'
72. bot.send\_message(call.from\_user.id, text)
73. elif call.data == 'but7' and str(call.from\_user.id) == ID\_ADMIN:
74. flag = 'schedule.txt'
75. text = 'Введите текст или отправьте файл в формате txt'
76. bot.send\_message(call.from\_user.id, text)
77. elif call.data == 'but8' and str(call.from\_user.id) == ID\_ADMIN:
78. flag = 'teachers.txt'
79. text = 'Введите текст или отправьте файл в формате txt'
80. bot.send\_message(call.from\_user.id, text)
81. elif call.data == 'but9' and str(call.from\_user.id) == ID\_ADMIN:
82. flag = 'directions.txt'
83. text = 'Введите текст или отправьте файл в формате txt'
84. bot.send\_message(call.from\_user.id, text)
85. elif call.data == 'but10' and str(call.from\_user.id) == ID\_ADMIN:
86. flag = 'contacts.txt'
87. text = 'Введите текст или отправьте файл в формате txt'
88. bot.send\_message(call.from\_user.id, text)
89. elif call.data == 'but11':
90. menu\_send(call.from\_user.id, bot)
91. @bot.message\_handler(content\_types=['text'])
92. def get\_text\_messages(message):
93. global flag
94. if flag:
95. try:
96. src = 'data/' + flag
97. with open(src, 'w', encoding="utf-8") as new\_file:
98. new\_file.write(message.text)
99. bot.reply\_to(message, "Пожалуй, я сохраню это")
100. except Exception as e:
101. bot.reply\_to(message, 'Ошибка')
102. flag = ''
103. else:
104. bot.reply\_to(message, 'Я вас не понимаю')
105. @bot.message\_handler(content\_types=['document'])
106. def handle\_docs\_photo(message):
107. global flag
108. if flag:
109. try:
110. chat\_id = message.chat.id
111. file\_info = bot.get\_file(message.document.file\_id)
112. downloaded\_file = bot.download\_file(file\_info.file\_path)
113. src = 'data/' + flag
114. with open(src, 'wb') as new\_file:
115. new\_file.write(downloaded\_file)
116. bot.reply\_to(message, "Пожалуй, я сохраню это")
117. except Exception as e:
118. bot.reply\_to(message, 'Ошибка')
119. flag = ''
120. bot.polling(none\_stop=True)

## **Приложение 2**

Содержание файла menu\_send.py

1. from telebot import types
2. def menu\_send(id, bot):
3. buttons = types.InlineKeyboardMarkup(row\_width=1)
4. button1 = types.InlineKeyboardButton('Рассказать об организации', callback\_data='but1')
5. button2 = types.InlineKeyboardButton('Предоставить расписание', callback\_data='but2')
6. button3 = types.InlineKeyboardButton('Предоставить информацию о преподавателях', callback\_data='but3')
7. button4 = types.InlineKeyboardButton('Предоставить список направлений', callback\_data='but4')
8. button5 = types.InlineKeyboardButton('Предоставить контактные данные', callback\_data='but5')
9. buttons.add(button1, button2, button3, button4, button5)
10. bot.send\_message(id, text='Выберите действие', reply\_markup=buttons)