

Королев Серафим, 7 класс

МБОУ СОШ №2, г. Ступино

Разработка чат – бота для адаптации новых сотрудников в крупной компании на примере международного аэропорта Шереметьево

Руководитель: Принев Мечислав Александрович

Оглавление

Введение.	Стр. 1
Средства интернет – коммуникаций.	Стр. 1
Международный аэропорт Шереметьево.	Стр. 4
Чат – боты в социальных сетях.	Стр. 5
Проектирование чат-бота.	Стр. 6
Разработка чат – бота.	Стр. 8
Результат работы.	Стр. 10
Заключение.	Стр. 13
Список источников	Стр. 14
Приложения	Стр. 15 - стр.23

Введение.

В рамках проведенной исследовательской работы была поставлена цель разработать чат – бот для адаптации новых сотрудников в крупной компании на примере международного аэропорта Шереметьево.

Каждый новый сотрудник, приходя работать в крупную компанию, задается рядом организационных вопросов, ответы на которые часто приходится искать самостоятельно: о своем рабочем месте, оформлении пропуска, режиме работы, маршрутах корпоративного транспорта, территориальном расположении отделов, обучении и развитии. Это повышает уровень стресса новичка в новой для себя обстановке.

В работе исследовались современные средства интернет-коммуникации. Были проведены анкетирование и опрос сотрудников аэропорта, анализ вариантов сообщений для создания интеллектуального алгоритма обработки информации, а также построены схемы с помощью онлайн – редактора блок схем `programforyou[1]`

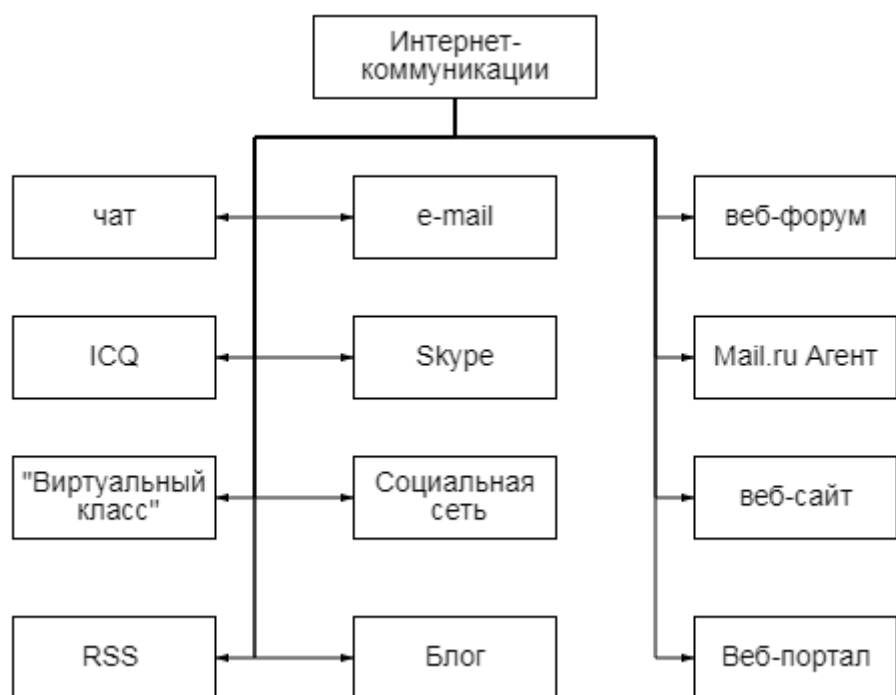
Результатом данной работы является чат-бот в мессенджере Telegram.

Средства реализации – язык программирования Python, библиотека `aiogram`.

Актуальность работы – использование чат-бота в период испытательного срока новых сотрудников позволит избежать потерь времени, которые неизбежно возникают при их введении в штат компании.

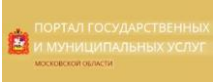
Средства интернет-коммуникаций.

Интернет-коммуникация - это такие методы общения, при которых передача информации происходит по каналам Интернет с использованием стандартных протоколов обмена и представления информации. Информация может передаваться в различной форме - голос, видео, документы, мгновенные сообщения, файлы. Использование традиционных механизмов коммуникаций, таких как факс или проводной телефон сегодня быстро замещаются новыми технологиями - обмен мгновенными сообщениями, электронная почта, передача голоса и видео через Интернет. Средства интернет-коммуникаций представлены на рисунке ниже.[2]



Описание		Пример
Чат	Чат, чаттер (англ. chatter — болтать) — средство обмена сообщениями по компьютерной сети в режиме реального времени, а также программное обеспечение, позволяющее организовывать такое общение. Характерной особенностью является коммуникация именно в реальном времени или близкая к этому, что отличает чат от форумов и других «медленных» средств.	
E-mail	Электронная почта — технология и служба по пересылке и получению электронных сообщений (называемых «письма», «электронные письма» или «сообщения») между пользователями компьютерной сети (в том числе — Интернета)[3]	Яндекс.Почта  https://mail.yandex.ru
Веб-форум	Это разновидность коммуникации среди пользователей в интернете. На интернет-ресурсе такого плана создаются топики для обсуждения, они же темы для общения. Таким образом, веб-форум представляет собой совокупность разделов, в которых между посетителями ведется дискуссия. [4]	Форум веб-программистов - Киберфорум  https://www.cyberforum.ru/web/
ICQ	ICQ (от англ. I seek you — «я ищу тебя») — бесплатная система мгновенного обмена текстовыми сообщениями для мобильных и иных платформ с поддержкой голосовой и видеосвязи. [5]	ICQ (онлайн)  https://icq.online/

Skype	Скайп предназначен для личного и делового общения в любое время, в любых обстоятельствах, на любых устройствах, таких как мобильные телефоны, компьютеры. Уникальные возможности голосовых и видео-звонков, а также текстовых сообщений в Скайпе помогают общаться и работать вместе. Он подходит для совершения звонков один на один или в группе. [6]	 https://www.skype.com/ru/get-skype/
Mail.ru Агент	Программа для мгновенного обмена сообщениями через интернет, развиваемая компанией Mail.ru с 2003 года. Существуют версии клиента для РС, а также для смартфонов и планшетов на операционных системах.[7]	Mail.ru Агент  https://agent.mail.ru/windows
Виртуальный класс	Суть виртуальных классов - в совместном обучении посредством видео и с использованием инструментов, которые позволяют вовлекать учащихся в образовательный процесс, сделать урок интерактивным и своевременно получать обратную связь. Обучение проходит под руководством преподавателя лицом к лицу со студентами.	maibax beta  https://maibax.com/
Социальная сеть	Онлайн-платформа, которая используется для общения, знакомств, создания социальных отношений между людьми, которые имеют схожие интересы или офлайн-связи, а также для развлечения (музыка, фильмы) и работы. .[8]	ВКонтакте  https://vk.com
Веб-сайт	Сайт, или веб-сайт, также веб-узел, — одна или несколько логически связанных между собой веб-страниц; также место расположения контента сервера. Обычно сайт в Интернете представляет собой массив связанных данных, имеющий уникальный адрес и воспринимаемый пользователями как единое целое. .[9]	Автономная некоммерческая общеобразовательная организация "Физтех-лицей" имени П.Л. Капицы  https://anoo.ftl.name/
RSS	Генерируемый текстовый файл с данными сайта или блога в формате XML. RSS-лента (канал) содержит список статей с описанием и ссылкой на них, который блогеры отправляют своим подписчикам. С английского есть несколько версий перевода аббревиатуры: Really Simple Syndication – простое получение информации и Rich Site Summary – полноценная сводка сайта.[10]	RSS-канал  http://all-rss.ru/

блог	Интернет-журнал событий, интернет-дневник, онлайн-дневник — веб-сайт, основное содержимое которого регулярно добавляемые человеком записи, содержащие текст, изображения или мультимедиа.[11]	
Веб - портал	Это сайт, который дает возможность пользователю использовать всевозможные интерактивные сервисы, которые функционируют на одном ресурсе. Иными словами, это доступ в информационное хранилище. .[12]	портал государственных и муниципальных услуг московской области  https://uslugi.mosreg.ru/

Международный аэропорт Шереметьево.

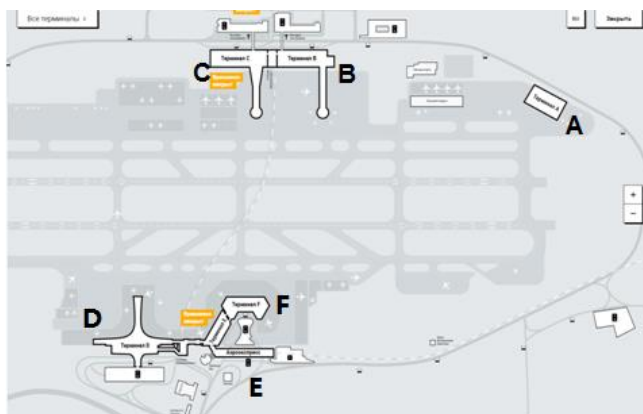
Международный аэропорт Шереметьево имени А. С. Пушкина — российский международный аэропорт федерального значения, первый в России по объёму пассажиропотока [13]. Включает шесть пассажирских терминалов:

1. А (терминал бизнес-авиации),
2. В (ранее «Шереметьево-1»),
3. С (открыт 17 января 2020 года) ,
4. D, E и F (ранее «Шереметьево-2»)
5. два грузовых («Москва-Карго» и «Шереметьево-Карго»).

«Шереметьево» принимает все типы воздушных судов как отечественного, так и иностранного производства Маршрутная сеть составляет более 230 направлений. По итогам 2019 года аэропорт обслужил 49 млн 933 тыс. пассажиров, что на 8,9% превысило итоги 2018 года.

Пассажирские терминалы D, E и F, образуют единый Южный терминальный комплекс. Терминалы соединены между собой пешеходными галереями, что позволяет пассажирам оперативно и беспрепятственно передвигаться по всему комплексу.

Северный терминальный комплекс составляют терминалы А, В и С. Терминал А обслуживает пассажиров деловой авиации, Терминал В — регулярные внутренние рейсы, Терминал С — международные рейсы. Терминалы В и С объединены с терминалами Южного комплекса межтерминальным переходом, который включает в себя два тоннеля для перевозки пассажиров на беспилотных поездах и транспортировки багажа. Время в пути между терминальными комплексами с учётом ожидания — 4 минуты. [14]



Из рисунка видно, что одна из проблем в крупных компаниях – это огромная территория и, попадая в новую для себя среду на испытательном сроке, у каждого сотрудника возникает сложность в оперативном поиске ответов на ряд организационных вопросов. Что повышает уровень стресса на новом рабочем месте.

Опрос и анкетирование сотрудников аэропорта позволили выявить наиболее часто встречаемые вопросы:

1. Как оформить пропуск на территорию аэропорта?
2. Где и когда пройти обучение по охране труда?
3. Информация по маршрутах корпоративного транспорта.
4. Информация о парковочных местах.
5. Информация о получении карточки добровольного медицинского страхования.
6. Информация о скидках для сотрудников в магазинах, расположенных в терминалах.
7. Узнать ФИО своего наставника и как с ним связаться.
8. Узнать телефоны справочных служб.
9. Получить напоминание о важных событиях на испытательном сроке.

Как видно из списка выше, что если каждому новичку при приеме на работу давать ответы на все организационные вопросы, то это займет много времени. Чтобы не отвлекаться от основной работы, новичкам предлагается найти ответы на часто задаваемые вопросы в чат-боте. Чат – боты обладают рядом достоинств, которые позволили выбрать именно это средство коммуникации: дешево, доступность круглосуточная, получение мгновенного ответа.

Чат – боты в социальных сетях.

Чат-боты разделяют на простые, способные выполнять заданные команды, и самообучающиеся с элементами искусственного интеллекта. Также их делят на интерфейсных ботов, взаимодействующих с пользователем через кнопки и картинки, и ботов, понимающих текстовые команды на естественном языке. В соцсетях работают простые боты, понимающие текст, и интерфейсные боты. Рассмотрим некоторые чат-боты в социальных сетях:

Вид	Описание	Пример
Чат боты для продаж	Могут создаваться для продаж билетов (в кино, театр, на концерт, на транспорт), одежды, косметики и других всевозможных товаров	Бот сайта OneTwoTrip – @OneTwoTrip_Bot продает авиабилеты
Служба поддержки	Автоматизация процессов общения с клиентами помогает сэкономить множество времени и сил «живых» работников компании и выделить им больше времени на решение сложных задач	Бот «Почты России» @PochtaBot позволяет отслеживать посылки, задавать возникшие вопросы и оставлять отзывы
Служба доставки	Боты по доставке (цветов, подарков и т.д.) пользуются у клиентов очень большой популярностью. И наиболее популярна служба доставки еды.	Пиццерия Domino имеет свой чатбот – DOM The Pizza Bot . Чтобы заказать пиццу, нужно ввести в чате слово pizza, и бот на основе истории сообщений автоматически подберет предпочитаемый пользователем тип этого итальянского блюда.
Бот собеседник	Эти боты выполняют роль друга или психолога, – им можно излить	В Китае компания Microsoft создала девушку-бота под именем Xiaoice, с

	душу и получить актуальные жизненные советы.	которой делятся своими переживаниями более 20 млн. человек. Этот чат-бот постоянно учится на своих диалогах, а его речь и поведение в чате очень похожи на человеческие.
Бот учитель	Предоставляют учебный материал по преподаваемой дисциплине (небольшие по объему тексты), «общаются» с учениками и оценивают их знания с помощью тестирований.	Чатбот Leslie помогает выучить английский язык, – он беседует с пользователем на определенные темы, объясняет правила грамматики, приводит синонимы/антонимы, спрягает слова и т.д.
Органайзеры	Напоминать о важных встречах или делах.	В «Телеграме» есть чатбот, работа которого основана на технике Pomodoro, – @pomodoro_timer_bot
Заказ такси	Делают процесс заказа и ожидания машины более удобным для клиентов	В «Телеграме» есть чатбот для вызова такси . – @taxikbot

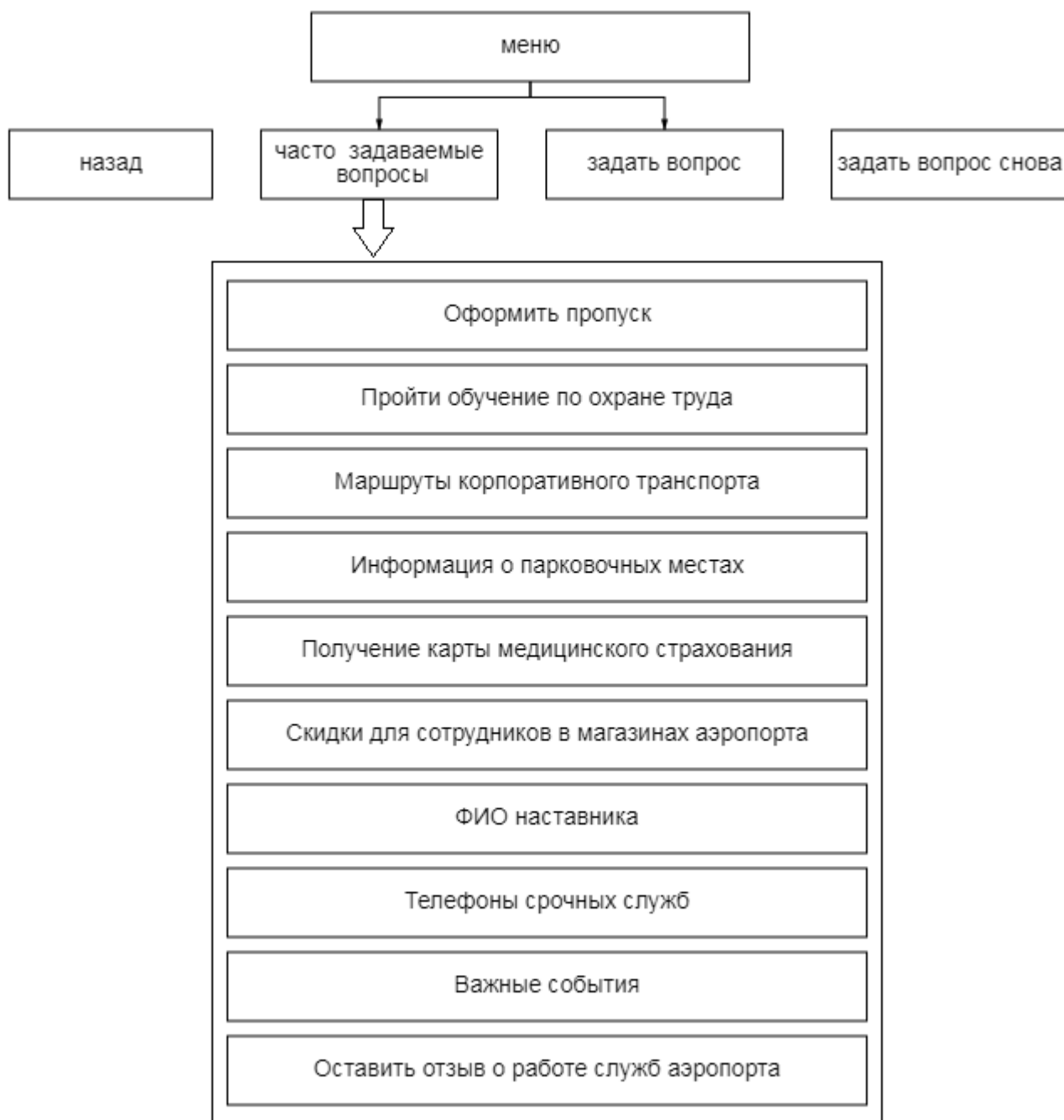
Проектирование чат – бота (ЧБ).

Логика работы проектируемого ЧБ представлена в приложении 1. Открывая ЧБ, пользователь стоит перед выбором либо задать свой собственный вопрос, либо выбрать одну из уже предложенных тем вопросов. Если он задает свой вопрос, то проверяется, корректно ли сформулирован вопрос и после этого дается на него ответ из предложенного списка ответов, сохраненных заранее сотрудниками аэропорта в ЧБ. Если вопрос не понятен ЧБ, то он об этом информирует пользователя и предлагает задать новый вопрос.

Меню ЧБ будет состоять из следующих кнопок, представленных на рисунке ниже. После выбора кнопки главного меню, появляются две кнопки «часто задаваемые вопросы» и «задать вопрос». При нажатии кнопки «часто задаваемые вопросы», появляются кнопки, с помощью которых пользователь может выбрать для себя одну из интересующих тем. При нажатии кнопки «задать вопрос», ЧБ ждет вопрос от пользователя. Так же у пользователя есть возможность выбора кнопки «назад», которая возвращает в главное меню. После получения ответа на задаваемый вопрос, у пользователя есть возможность выбора кнопки «задать вопрос снова».

ЧБ будет вести диалог с пользователем, используя заранее сохраненные сообщения:

1. Привет! Я чат бот аэропорта Шереметьево. Нажми на кнопку, чтобы перейти в главное меню.
2. Ты в главном меню.
3. Чтобы получить ответ выбери тему об интересующей информации.
4. Я тебя не понимаю.
5. Введи мне свой вопрос снова, хочешь вернуться назад, нажми на кнопку.
6. Введи мне свой вопрос, хочешь вернуться назад, нажми на кнопку.



Взаимосвязь между сообщениями ЧБ и кнопок представлена в приложениях 2 и 3. ЧБ диалоговыми сообщениями помогает пользователю делать выбор по кнопкам.

Если пользователь задает вопрос самостоятельно, а не выбирает из встроенных кнопок, то ЧБ ищет в тексте вопроса части слов, по которым можно вывести тот или иной ответ из сохраненного списка от сотрудников компании. Логика поиска следующая. Берутся корни часто встречаемых слов (без окончаний) по той или иной интересующей теме и происходит поиск их в введенном вопросе пользователя. Все корни часто встречаемых слов сгруппированы по интересующим темам и объединены по сохраненным ответам сотрудников аэропорта (приложение 4). И когда ЧБ находит подходящий корень слова в вопросе пользователя, то

выводит тот ответ, что возможно интересовал пользователя. Более подробно алгоритм обработки введенной информации пользователями представлен в приложении 5.

При разработке ЧБ использовался язык программирования Python. Он более прост по своему синтаксису, чем, к примеру, C++ и содержит библиотеки, позволяющие создавать чат – боты в социальной сети Telegram.

Общение с программистами, имеющими опыт по созданию чат – ботов, показал, что они выбирают библиотеку aiogram. Свой выбор в пользу aiogram объясняют тем, что он является довольно простым и полностью асинхронным кодом. Это позволяет делать ботов, работающих быстрее и проще.

Как правило, программный код выполняется последовательно, только одна конкретная операция происходит в данный момент времени. Если функция зависит от результата выполнения другой функции, то она должна дожидаться, пока нужная ей функция не завершит свою работу и не вернет результат, и до тех пор, пока это не произойдет, выполнение программы, по сути, будет остановлено с точки зрения пользователя. Для пользователей такая работа программы не удобна, она зависает. Чтобы избежать этого, используют асинхронный код. Он убирает блокирующую операцию из основного потока программы, и она продолжает работать - для пользователя такая программа не зависает и работает быстрее.

Разработка чат – бота.

1.1 Создание проекта.

Анкетирование сотрудников показало, что они чаще всего пользуются мессенджером Telegram. При разработке ЧБ я просматривал разные статьи в интернете с рекомендациями о том, как создать чат-бот в Telegram с помощью Python. Пользовался информацией об описании библиотеки aiogram на сайте «Добро пожаловать в документацию aiogram» [15], о языке программирования Python на сайте PythonRu [16] и об описании необходимых мне методов для работы Telegram с ботом на сайте Telegram Bot Api [17]. Из статей по созданию чат-ботов мне понравилась «Асинхронный Telegram бот на языке Python 3 с использованием библиотеки aiogram». В ней автор подробно описывает весь процесс написания ЧБ, автор не профессионал, а любитель, поэтому, язык изложенного материала прост и доступен для начинающих разработчиков. [18]

Начало разработки. Открываем Telegram, находим @BotFather, отправляем команду /newbot, пишем свое имя бота и получаем TOKEN Telegram-бота на Python.

BotFather → /newbot → имя бота → api TOKEN

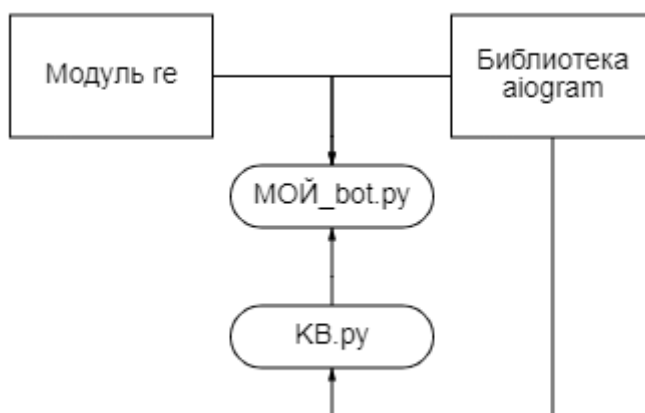
BotFather – это Telegram бот, который правит всеми ботами. При помощи него создаются новые боты.

Api TOKEN – уникальный ключ для управления ботом. API (расшифровывается как Application Programming Interface – программный интерфейс приложения) – это набор команд, функций, классов и других сведений, который предоставляет та или иная программа для взаимодействия с другой программой. TOKEN нужен для взаимодействия создаваемого ЧБ с сервером Telegram, потому что бот – это стороннее приложение, которое должно работать внутри Telegram. Используя уникальный ключ token, осуществляется взаимосвязь создаваемого бота с его прародителем BotFather, который создан внутри Telegram.

Далее создаем новый проект. Для удобства работы с кодом были созданы следующие файлы:

МОЙ bot.py	Основной файл, где прописан код ЧБ
KB.py	Файл, где описаны кнопки.

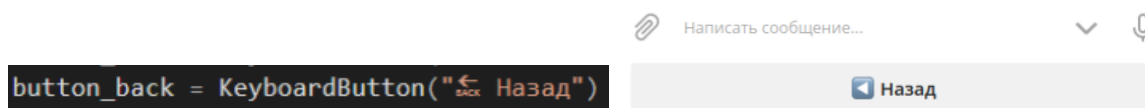
Ниже представлена схема взаимосвязи созданных файлов и библиотеки aiogram.



Модуль re — предоставляет полную поддержку регулярным выражениям. Регулярные выражения — специальная последовательность символов, которая помогает сопоставлять или находить строки Python с использованием специализированного синтаксиса, содержащегося в шаблоне. В создаваемом ЧБ будем использовать функцию `findall` модуля re для поиска всех совпадений корней слов в строке вводимого сообщения пользователем, чтобы определить к какой теме отнести вопрос и какой вывести ответ.

В файле KB.py описаны кнопки. Из библиотеки Aiogram импортировались `ReplyKeyboardMarkup` и `KeyboardButton`.

`KeyboardButton` нужен для описания отдельных кнопок клавиатуры. Имеет один параметр — это текст, который будет названием этой кнопки в интерфейсе чат-бота. Например, ниже описана кнопка «назад» в коде и эта же кнопка в интерфейсе пользователя.



`ReplyKeyboardMarkup` необходим для создания клавиатур. Использовался метод `add` для добавления кнопок.

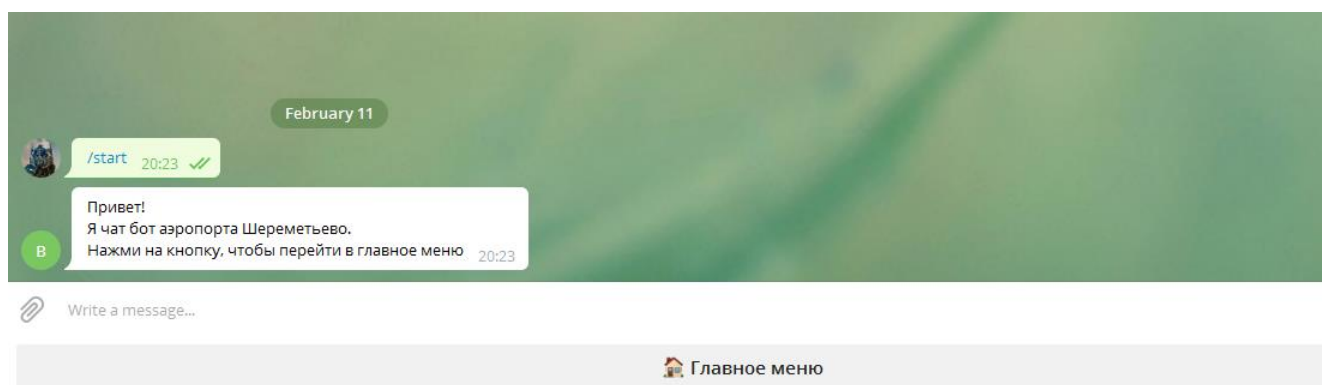
Библиотека Aiogram → МОЙ_bot.py		
класс Dispatcher	С его помощью создается объект dp. Он нужен будет, например, для работы обработчика <code>MessageHandler</code> .	<pre> bot = Bot(token=TOKEN) dp = Dispatcher(bot) </pre>
класс Bot	Импортируем из Aiogram класс Bot, хранящий переменную с токеном.	

модуль executor	Нужен, чтобы ЧБ не выключался и проверял наличие текстовых вводов от пользователей в Telegram. Строка в примере запускает непрерывное отслеживание ботом новых сообщений. Чтобы получать сообщения от серверов Telegram используется polling (polling. to poll - опрашивать), т.е. постоянный опрос сервера на наличие новых сообщений.	<pre>if __name__ == '__main__': executor.start_polling(dp)</pre>
-----------------	---	--

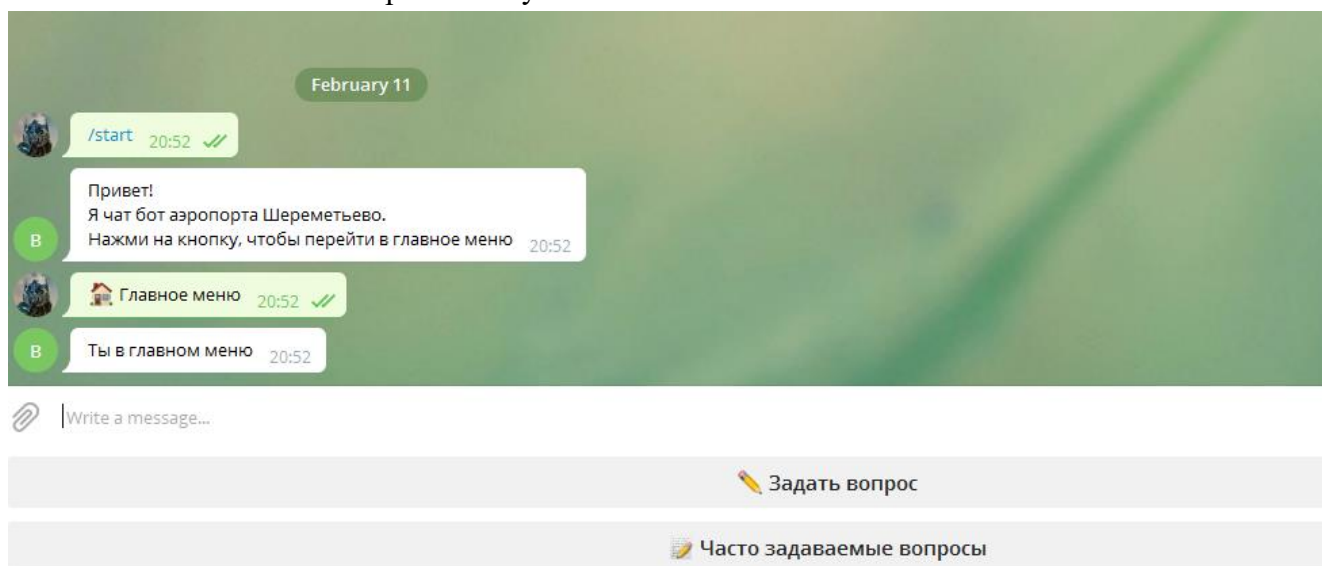
Код файлов представлен в приложениях 6 и 7.

Результат работы.

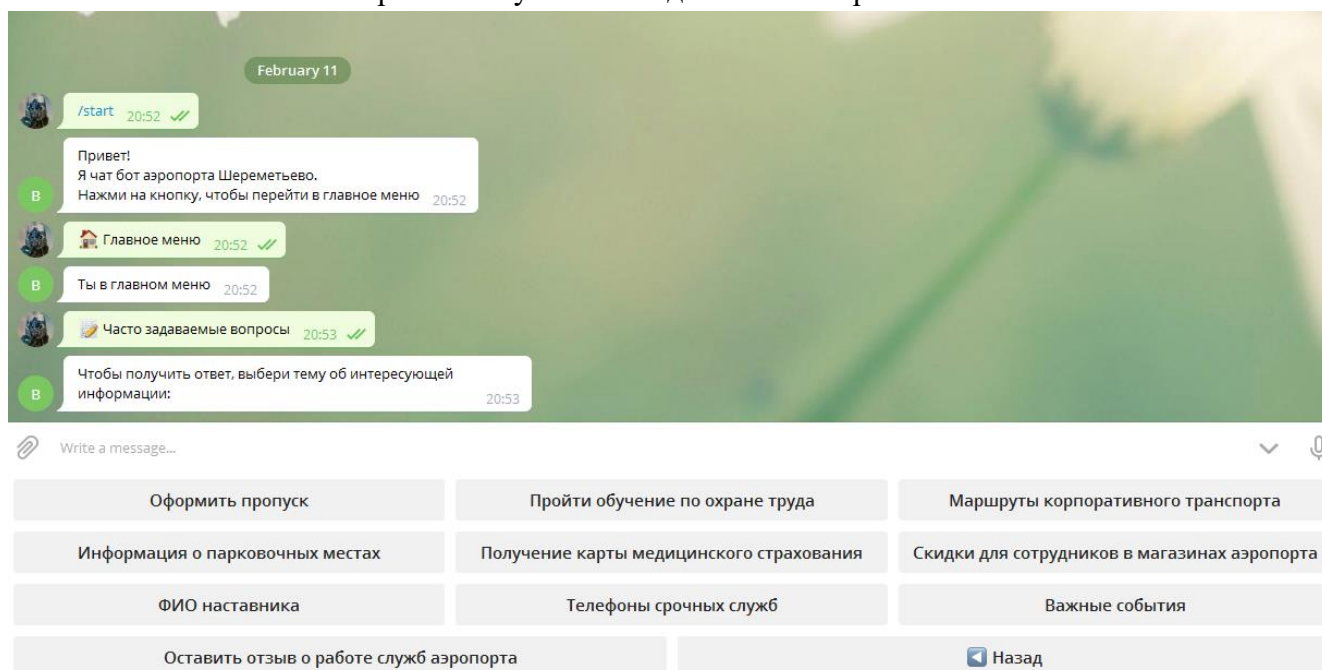
1. Пользователь ввел команду /start.



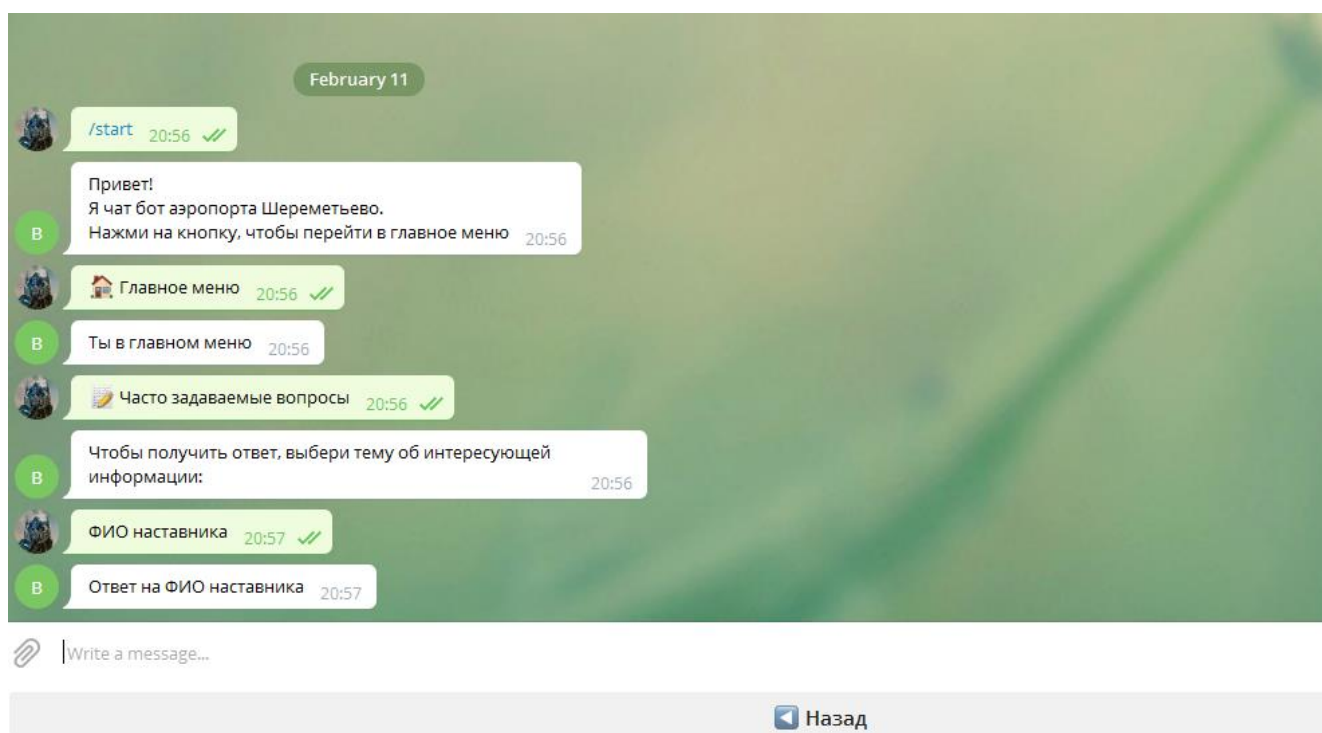
2. Пользователь выбрал кнопку «Главное меню».



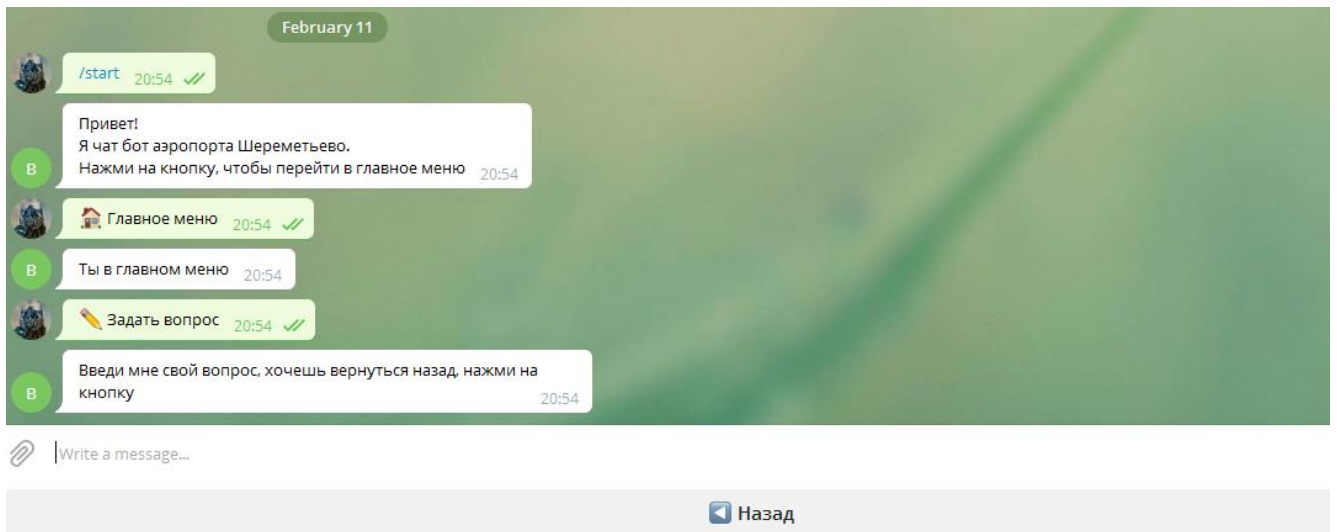
3. Пользователь выбрал кнопку «Часто задаваемые вопросы».



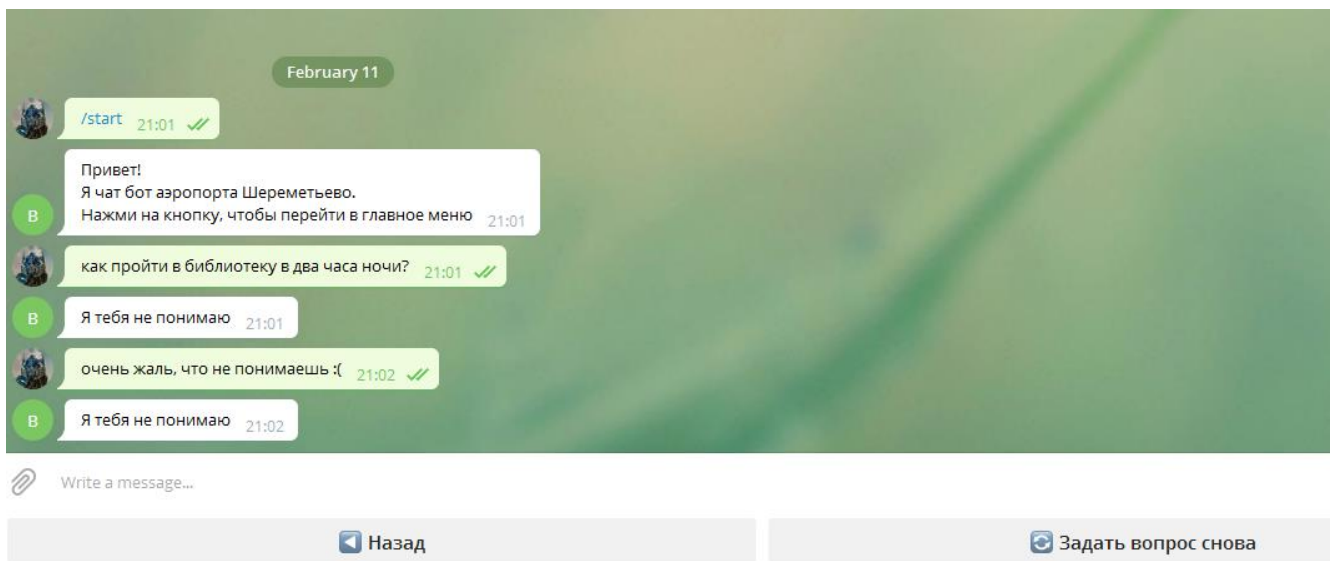
4. Пользователь выбрал интересующую тему «ФИО наставника».



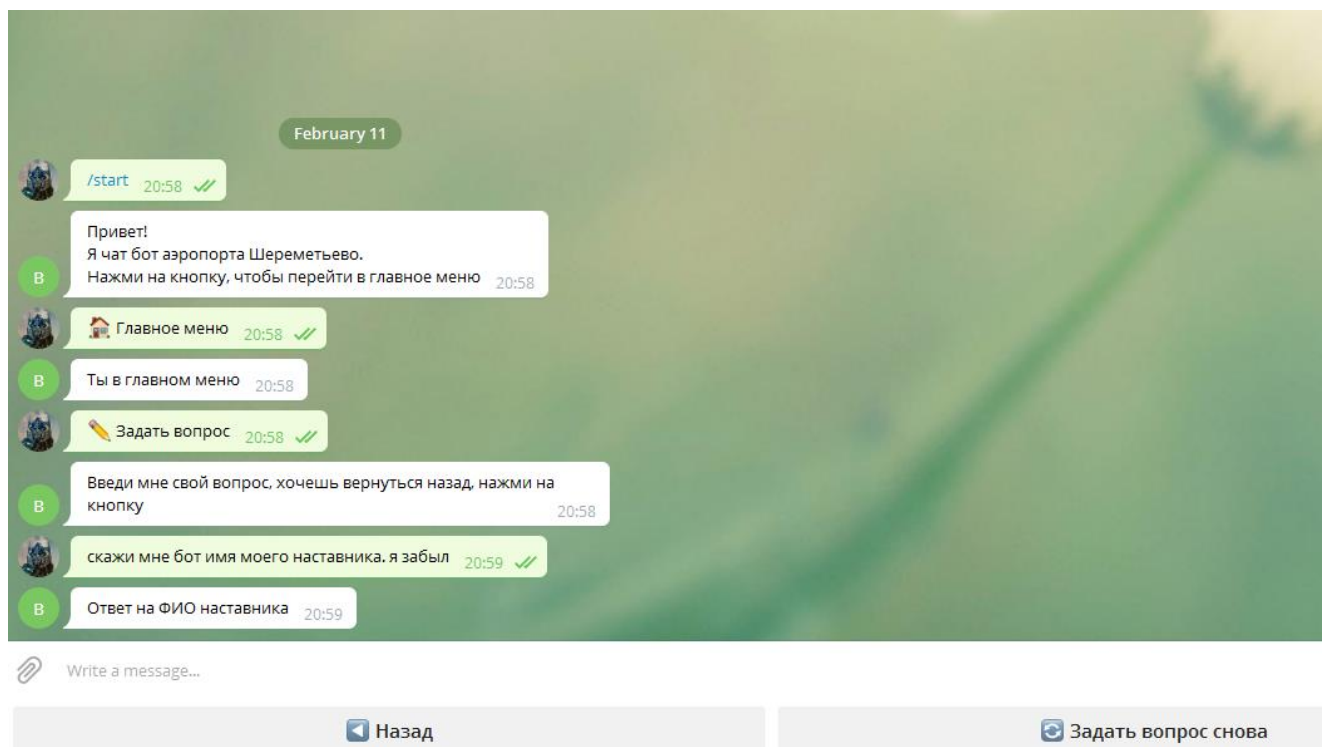
5. Пользователь выбрал кнопку «Задать вопрос».



6. Пользователь ввел не корректно сформулированный вопрос.



7. Пользователь ввел вопрос про имя наставника.



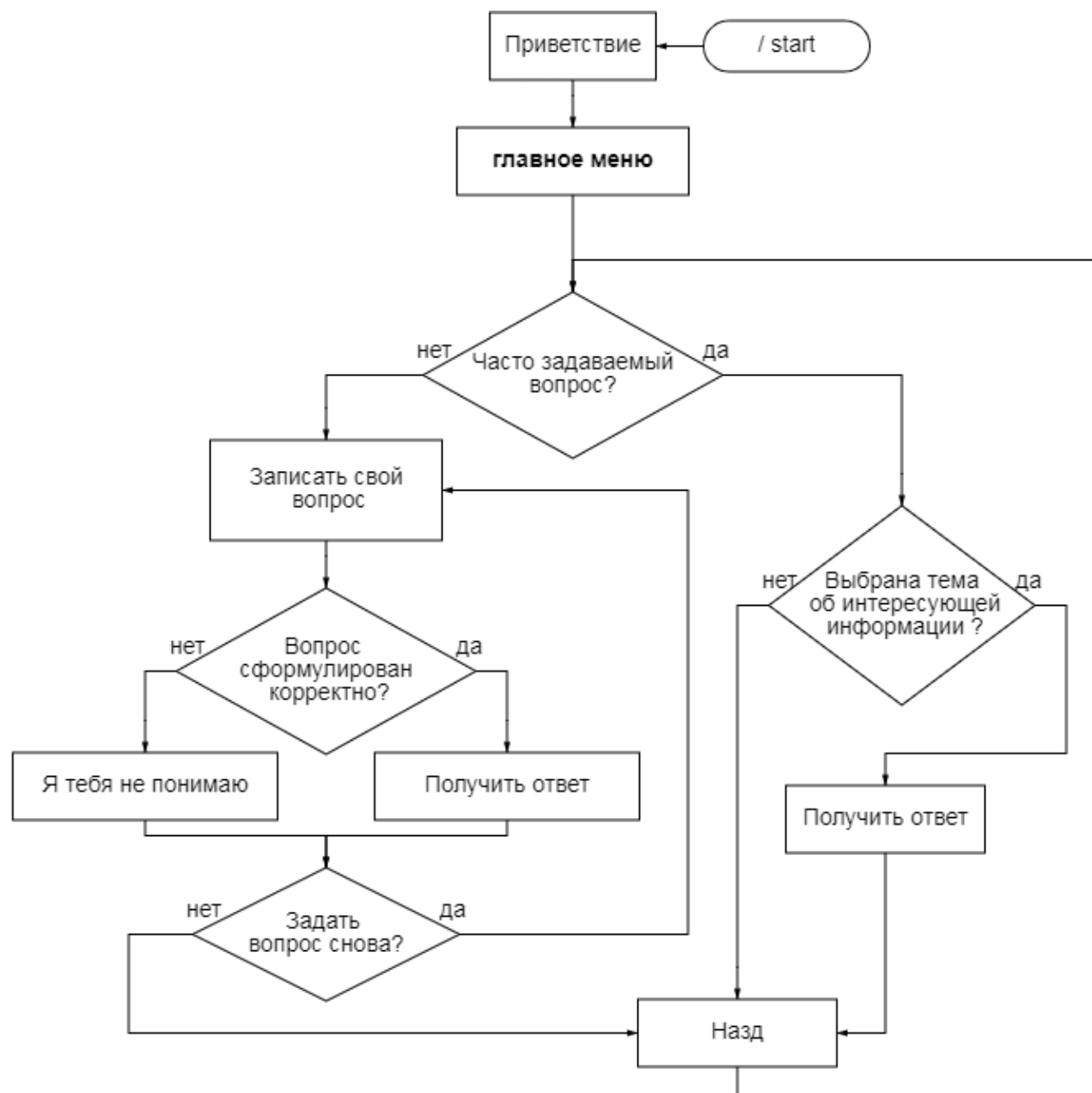
Заключение.

Использование чат-бота в период испытательного срока новых сотрудников позволит избежать потерь времени, которые неизбежно возникают при их введении в штат компании.

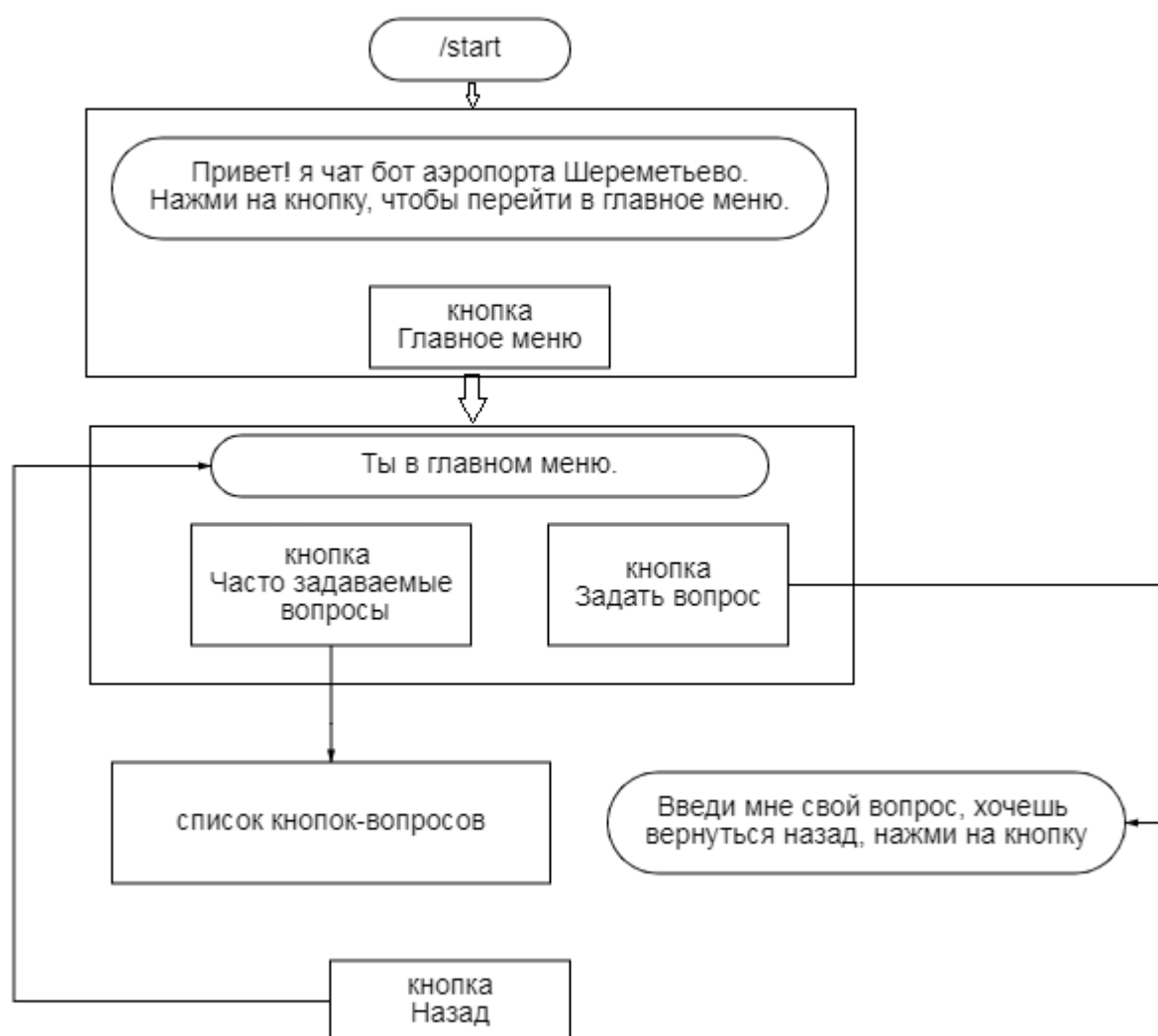
Список источников:

- [1] Онлайн – редактор блок схем Programforyou <https://programforyou.ru/block-diagram-redactor> открытый источник, 05.02.2021
- [2] Вики для программы. Компьютерные сети. <http://sbmtwiki.wikidot.com/wiki:sredstva-internet-kommunikacij:internet-uslugi> , открытый источник, 04.02.2021
- [3] Википедия. Электронная почта. https://ru.wikipedia.org/wiki/Электронная_почта , открытый источник, 04.02.2021
- [4] Линкбилдер. Веб – форум. <https://linkbuilder.su/glossary/veb-forum> открытый источник, 04.02.2021
- [5] Википедия. ICQ <https://ru.wikipedia.org/wiki/ICQ> открытый источник, 04.02.2021
- [6] Microsoft. Скайп <https://www.skype.com/ru/about/> открытый источник, 04.02.2021
- [7] Википедия. Агент Mail.ru https://ru.wikipedia.org/wiki/Агент_Mail.ru открытый источник, 04.02.2021
- [8] Википедия. Социальная сеть. https://ru.wikipedia.org/wiki/Социальная_сеть открытый источник, 04.02.2021
- [9] Википедия. Сайт. <https://ru.wikipedia.org/wiki/Сайт> открытый источник, 04.02.2021
- [10] Calltouch. RSS сата. <https://www.calltouch.ru/glossary/rss-sayta/> открытый источник, 04.02.2021
- [11] Википедия. Блог. <https://ru.wikipedia.org/wiki/Блог> открытый источник, 04.02.2021
- [12] Веб портал <https://zen.yandex.ru/media/id/5b64be5b505f6500a95d481c/webportal-vidy-otlichiia-ot-saita-5dc42cb9d5bbc300b1f52625> открытый источник, 04.02.2021
- [13] Википедия. Шереметьево <https://ru.wikipedia.org/wiki/Шереметьево> , открытый доступ, 03.02.2021
- [14] Международный аэропорт Шереметьево <https://www.svo.aero/ru/map?terminal=all> , открытый источник, 03.02.2021
- [15] Добро пожаловать в документацию aiogram <https://docs.aiogram.dev/en/latest/> открытый доступ, 08.02.2021
- [16] PythonRu <https://pythonru.com/osnovy/modul-re-dlja-reguljarnyh-vyrazhenij-v-python> открытый доступ, 08.02.2021
- [17] Telegram Bot Api <https://core.telegram.org/bots/api#using-a-local-bot-api-server> открытый доступ, 08.02.2021
- [18] Асинхронный Telegram бот на языке Python 3 с использованием библиотеки aiogram <https://surik00.gitbooks.io/aiogram-lessons/content/> открытый доступ, 08.02.2021

Приложение 1. Логика работы ЧБ.



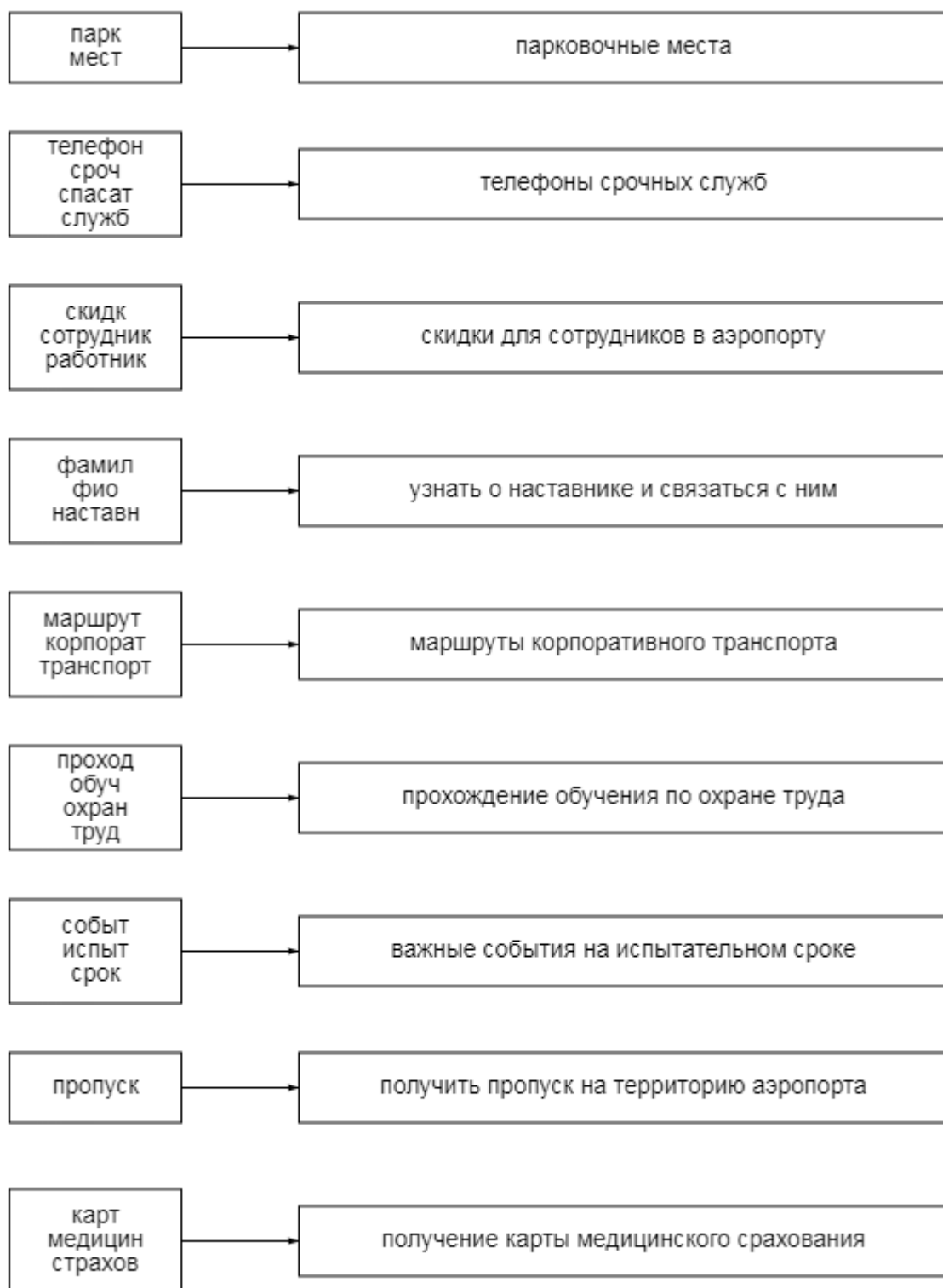
Приложение 2. Взаимосвязь сообщений ЧБ и кнопок.



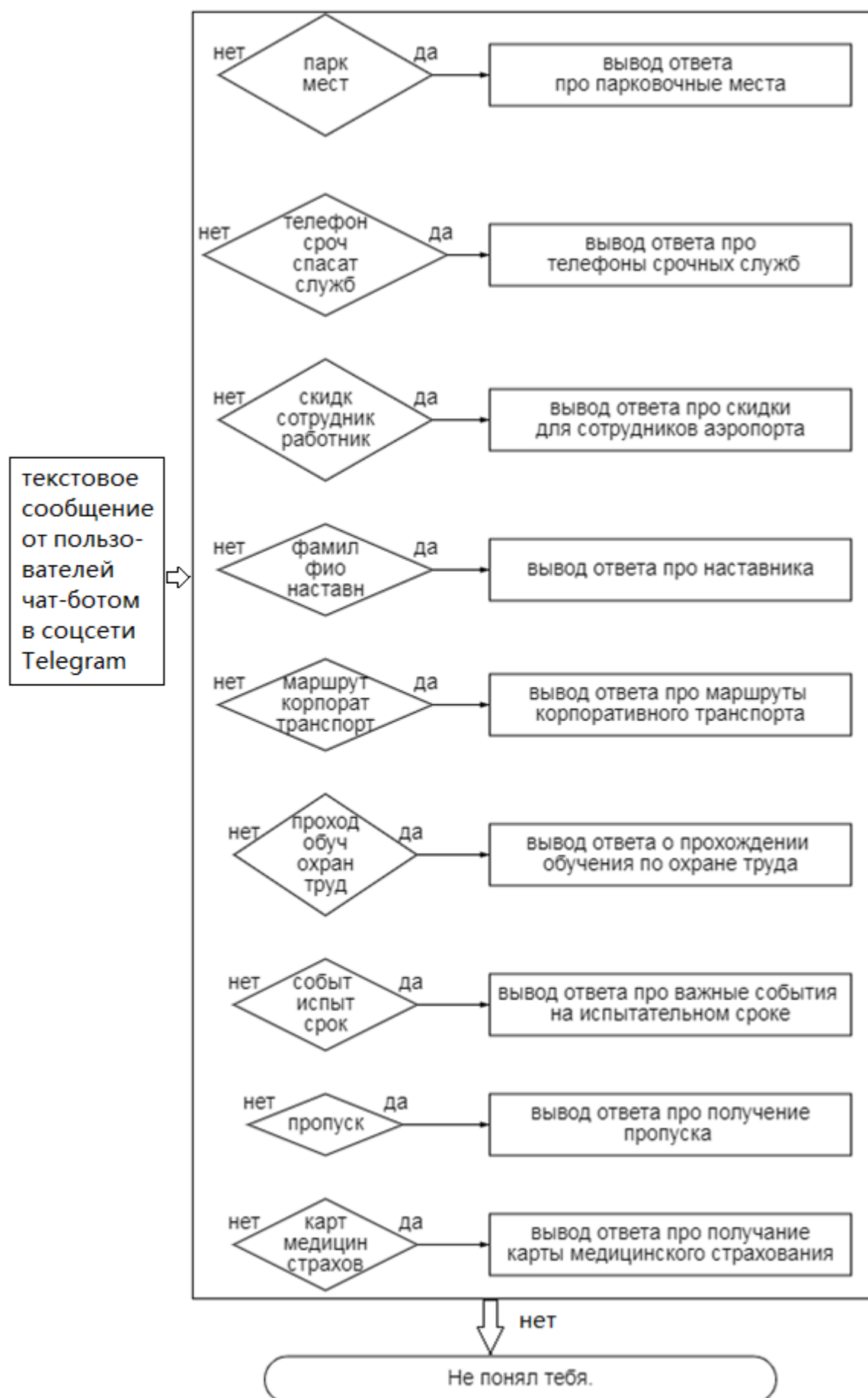
Приложение 3. Взаимосвязь сообщений ЧБ и кнопок из тем часто задаваемых вопросов.



Приложение 4. Группировка корней слов и тем.



Приложение 5. Алгоритм обработки введенных сообщений пользователями.



Приложение 6. Код файла МОЙ_bot.py

```
from aiogram import Bot, types
from aiogram.dispatcher import Dispatcher
from aiogram.utils import executor
import KB
import re
```

Импортируемые классы
и модули

```
#токен
TOKEN = '....здесь токен.....'
```

```
bot = Bot(token=TOKEN)
```

```
dp = Dispatcher(bot)
```

Создание dp

```
#Команда start
```

```
@dp.message_handler(commands=['start'])
```

```
async def start (message: types.Message):
```

```
    await bot.send_message(message.from_user.id, 'Привет!\n'
        'Я чат бот аэропорта Шереметьево.\n'
        'Нажми на кнопку, чтобы перейти в главное
```

```
меню', reply_markup=KB.main_kb)
```

Обработка команды
/ start. В итоге
выведется
приветственное
сообщение и кнопка
главного меню.

```
#Обработка текстовых сообщений
```

```
@dp.message_handler()
```

```
async def text (message: types.Message):
```

```
    answer=message.text
```

```
    park = re.findall('парк', answer, flags=re.I)
```

```
    places = re.findall('мест', answer, flags=re.I)
```

```
    phone = re.findall('телефон', answer, flags=re.I)
```

```
    urgent = re.findall('сроч', answer, flags=re.I)
```

```
    rescue = re.findall('спасат', answer, flags=re.I)
```

```
    service = re.findall('служб', answer, flags=re.I)
```

```
    passage = re.findall('проход', answer, flags=re.I)
```

```
    train = re.findall('обуче', answer, flags=re.I)
```

```
    security = re.findall('охран', answer, flags=re.I)
```

```
    labor = re.findall('труда', answer, flags=re.I)
```

```
    route = re.findall('маршрут', answer, flags=re.I)
```

```
    corporate = re.findall('корпорат', answer, flags=re.I)
```

```
    transport = re.findall('транспорт', answer, flags=re.I)
```

```
    events = re.findall('событ', answer, flags=re.I)
```

```
    tester = re.findall('испыт', answer, flags=re.I)
```

```
    term = re.findall('срок', answer, flags=re.I)
```

```
    sale = re.findall('скидк', answer, flags=re.I)
```

```
    employee = re.findall('сотрудник', answer, flags=re.I)
```

```
    worker = re.findall('работник', answer, flags=re.I)
```

```
    surname = re.findall('фамил', answer, flags=re.I)
```

```
    fcs = re.findall('фио', answer, flags=re.I)
```

```
    mentor = re.findall('настав', answer, flags=re.I)
```

Обработка
текстового
сообщения от
пользователя.
Метод findall
модуля re находит в
тексте сообщений
корни слов по
шаблонам из этих
корней.

```
card = re.findall('карт', answer, flags=re.I)
medicine = re.findall('медицин', answer, flags=re.I)
insurance = re.findall('страхов', answer, flags=re.I)

admission = re.findall('пропуск', answer, flags=re.I)
```

Обработка
текстового
сообщения от
пользователя.

```
if answer == '👉 Главное меню':
    await bot.send_message(message.from_user.id, 'Ты в главном меню', reply_markup=KB.menu_kb)

elif answer == '👉 Назад':
    await bot.send_message(message.from_user.id, 'Ты в главном меню', reply_markup=KB.menu_kb)

elif answer == '👉 Задать вопрос':
    await bot.send_message(message.from_user.id, 'Введи мне свой вопрос, хочешь вернуться назад, нажми на кнопку', reply_markup=KB.back_kb)

elif answer == '👉 Часто задаваемые вопросы':
    await bot.send_message(message.from_user.id, 'Чтобы получить ответ, выбери тему об интересующей информации:', reply_markup=KB.faq_kb)

elif answer == '👉 Задать вопрос снова':
    await bot.send_message(message.from_user.id, 'Введи мне свой вопрос снова, хочешь вернуться назад, нажми на кнопку', reply_markup=KB.back_kb)

elif answer == 'Оформить пропуск':
    await bot.send_message(message.from_user.id, 'Ответ на Оформить пропуск', reply_markup=KB.back_kb)

elif answer == 'Пройти обучение по охране труда':
    await bot.send_message(message.from_user.id, 'Ответ на Пройти обучение по охране труда', reply_markup=KB.back_kb)

elif answer == 'Маршруты корпоративного транспорта':
    await bot.send_message(message.from_user.id, 'Ответ на Маршруты корпоративного транспорта', reply_markup=KB.back_kb)

elif answer == 'Информация о парковочных местах':
    await bot.send_message(message.from_user.id, 'Ответ на Информацию о парковочных местах', reply_markup=KB.back_kb)

elif answer == 'Получение карты медицинского страхования':
    await bot.send_message(message.from_user.id, 'Ответ на Получение карты медицинского страхования', reply_markup=KB.back_kb)

elif answer == 'Скидки для сотрудников в магазинах аэропорта':
    await bot.send_message(message.from_user.id, 'Ответ на Скидки для сотрудников в магазинах аэропорта', reply_markup=KB.back_kb)

elif answer == 'ФИО наставника':
    await bot.send_message(message.from_user.id, 'Ответ на ФИО наставника', reply_markup=KB.back_kb)

elif answer == 'Телефоны срочных служб':
```

Обработка
кнопок меню

Обработка
кнопок с
темами об
интересующей
информации

```

    await bot.send_message(message.from_user.id, 'Ответ на
Телефоны срочных служб',reply_markup=KB.back_kb)

    elif answer == 'Важные события':
        await bot.send_message(message.from_user.id, 'Ответ на
Важные события',reply_markup=KB.back_kb)

    elif answer == 'Оставить отзыв о работе служб аэропорта':
        await bot.send_message(message.from_user.id, 'Ответ на
Оставить отзыв о работе служб аэропорта',reply_markup=KB.back_kb)

```

Обработка
кнопок с
темами об
интересующей
информации

```

    elif park or places:
        await bot.send_message(message.from_user.id, 'Ответ на
Информацию о парковочных местах',reply_markup=KB.gibr_kb)

    elif phone or urgent or rescue or service:
        await bot.send_message(message.from_user.id, 'Ответ на
Телефоны срочных служб',reply_markup=KB.gibr_kb)

    elif passage or train or security or labor:
        await bot.send_message(message.from_user.id, 'Ответ на Пойти
обучение по охране труда',reply_markup=KB.gibr_kb)

    elif route or corporate or transport:
        await bot.send_message(message.from_user.id, 'Ответ на
Маршруты корпоративного транспорта',reply_markup=KB.gibr_kb)

    elif events or tester or term:
        await bot.send_message(message.from_user.id, 'Ответ на Важные
события',reply_markup=KB.gibr_kb)

    elif sale or employee or worker:
        await bot.send_message(message.from_user.id, 'Ответ на Скидки
для сотрудников в магазинах аэропорта',reply_markup=KB.gibr_kb)

    elif surname or fcs or mentor:
        await bot.send_message(message.from_user.id, 'Ответ на ФИО
наставника',reply_markup=KB.gibr_kb)

    elif card or medicine or insurance:
        await bot.send_message(message.from_user.id, 'Ответ на
Получение карты медицинского страхования',reply_markup=KB.gibr_kb)

    elif admission:
        await bot.send_message(message.from_user.id, 'Ответ на
Оформить пропуск',reply_markup=KB.gibr_kb)

    else:
        await bot.send_message(message.from_user.id, 'Я тебя не
понимаю',reply_markup=KB.gibr_kb)

```

Обработка от
результата
работы findall

Иначе вывод
сообщения «Я тебя
не понимаю» и
кнопок Назад и
Задать вопрос
снова

```

if __name__ == '__main__':
    executor.start_polling(dp)

```

Непрерывность работы чат-бота

Приложение 7. Код файла KB.py

```
from aiogram.types import ReplyKeyboardRemove, \
    ReplyKeyboardMarkup, KeyboardButton
```

```
#создание кнопок
```

```
button_main = KeyboardButton('🏠 Главное меню')
button_back = KeyboardButton('◀ Назад')
button_que = KeyboardButton('❓ Задать вопрос')
button_faq = KeyboardButton('📖 Часто задаваемые вопросы')
button_again = KeyboardButton('🔄 Задать вопрос снова')
```

Создание
кнопок
меню

```
button_1 = KeyboardButton('Оформить пропуск')
button_2 = KeyboardButton('Пройти обучение по охране труда')
button_3 = KeyboardButton('Маршруты корпоративного транспорта')
button_4 = KeyboardButton('Информация о парковочных местах')
button_5 = KeyboardButton('Получение карты медицинского страхования')
button_6 = KeyboardButton('Скидки для сотрудников в магазинах аэропорта')
button_7 = KeyboardButton('ФИО наставника')
button_8 = KeyboardButton('Телефоны срочных служб')
button_9 = KeyboardButton('Важные события')
button_10 = KeyboardButton('Оставить отзыв о работе служб аэропорта')
```

Создание
кнопок
тем об
интересующей
информации

```
#создание клавиатур
```

```
main_kb = ReplyKeyboardMarkup(resize_keyboard=True, one_time_keyboard=True).add(button_main)
menu_kb = ReplyKeyboardMarkup(resize_keyboard=True,
    one_time_keyboard=True).add(button_que).add(button_faq)
back_kb = ReplyKeyboardMarkup(resize_keyboard=True, one_time_keyboard=True).add(button_back)
agaun_kb = ReplyKeyboardMarkup(resize_keyboard=True, one_time_keyboard=True).add(button_again)
#faq_kb = ReplyKeyboardMarkup(resize_keyboard=True,
    one_time_keyboard=True).insert(button_1).insert(button_2).insert(button_3).insert(button_4).in
sert(button_5).insert(button_6).insert(button_7).insert(button_8).insert(button_9).insert(butt
on_10).insert(button_back)
faq_kb = ReplyKeyboardMarkup(resize_keyboard=True,
    one_time_keyboard=True).row(button_1,button_2).row(button_3,button_4).row(button_5,button_6).r
ow(button_7,button_8).row(button_9,button_10).row(button_back)
gibr_kb = ReplyKeyboardMarkup(resize_keyboard=True,
    one_time_keyboard=True).add(button_back,button_again)
```