



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования

«Московский государственный технический университет  
имени Н.Э. Баумана (национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления» (ИУ)  
КАФЕДРА «Информационная безопасность» (ИУ8)

### ОТЧЕТ ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ

Тип практики: эксплуатационная

Название предприятия: ФГАНУ ЦИТиС

Студент:

Ананикян Александр Александрович, группа ИУ8-103 (5 курс)

Руководитель от предприятия:

руководитель проектов Мухунов Станислав Анатольевич

Руководитель от кафедры:

доцент кафедры ИУ8 Зайцева Анастасия Владленовна

Оценка: \_\_\_\_\_

*Ананикян* 16.08.2021  
(подпись, дата)  
*Мухунов* 16.08.2021  
(подпись, дата)  
*Зайцева*  
(подпись, дата)

ФГАНУ «Центр информационных систем органов исполнительной власти и космических исследований»  
И КОМПЕТЕНТНО-ТЕХНОЛОГИЧЕСКАЯ ДЕЯТЕЛЬНОСТЬ

Москва, 2021



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования

«Московский государственный технический университет  
имени Н.Э. Баумана (национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

Факультет «Информатика и системы управления» (ИУ)

Кафедра «Информационная безопасность» (ИУ8)

### ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ НА ПРАКТИКУ

Тип практики: эксплуатационная

Название предприятия: ФГАНУ ЦИТиС

Сроки практики: с 02 августа 2021 г. по 15 августа 2021 г.

Специальность: 10.05.03 «Информационная безопасность автоматизированных систем»

За время прохождения практики студенту надлежит согласно программе практики:

- Разработка проекта по распознаванию речи в текст на Python
- Приобрести практический опыт работы с сервисом Wit.ai и Telegram API
- Получение результата обработки текста с помощью искусственного интеллекта

Студент:

Ананьян Александр Александрович, группа ИУ8-103 (5 курс)

*Ананьян* 16.08.2021  
(подпись, дата)

Руководитель от предприятия:

руководитель проектов Мухунов Станислав Анатольевич

*Мухунов* 16.08.2021  
(подпись, дата)

Руководитель от кафедры:

доцент кафедры ИУ8 Зайцева Анастасия Владленовна

\_\_\_\_\_  
(подпись, дата)

## ОГЛАВЛЕНИЕ

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ .....	4
ВВЕДЕНИЕ .....	6
ОСНОВНАЯ ЧАСТЬ .....	7
1 Характеристика организации.....	7
2 Исследовательская часть.....	9
2.1 Общие сведения.....	9
2.2 Инструменты.....	10
3 Практическая часть.....	11
3.1 Создание ключа API.....	11
3.2 Взаимодействие с Wit Speech API.....	12
3.3 Работа с конфигурационным файлом.....	14
3.4 Взаимодействие с Telegram API.....	15
3.5 Создание исполняемого файла.....	17
3.6 Обработка текста с помощью ИИ.....	18
ЗАКЛЮЧЕНИЕ.....	19
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	20

## ГЕРМИНЫ И ОПРЕДЕЛЕНИЯ

- API – описание способов (набор классов, процедур, функций, структур или констант), которыми одна компьютерная программа может взаимодействовать с другой программой.
- Access-токен (токен доступа) — это токен, который предоставляет доступ его владельцу к защищенным ресурсам сервера.
- HTTP - протокол прикладного уровня передачи данных, изначально — в виде гипертекстовых документов в формате HTML, в настоящее время используется для передачи произвольных данных.
- INI-файл - это файл конфигурации, который содержит данные настроек для Microsoft Windows, Windows NT и некоторых приложений.
- Python – высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода, синтаксис ядра Python минималистичен, в то же время стандартная библиотека включает большой набор полезных функций.
- SDK (software development kit) — набор средств разработки, позволяющий специалистам по программному обеспечению создавать приложения для определённого пакета программ, программного обеспечения базовых средств разработки, аппаратной платформы, компьютерной системы, игровых консолей, операционных систем и прочих платформ.
- Telegram - кроссплатформенный мессенджер с функциями VoIP, позволяющий обмениваться текстовыми, голосовыми и видеосообщениями, стикерами и фотографиями, файлами многих форматов. Также можно совершать видео- и аудиозвонки, организовывать конференции, многопользовательские группы и каналы.
- Wit.ai - это интерфейс на естественном языке для приложений, способных преобразовывать предложения в структурированные данные. Он также довольно точен для распознавания речи и транскрипции звука.



- Авторизация - это назначение субъекту полномочий (прав доступа) и присвоение аутентифицированному субъекту (principal) прав и привилегий на доступ.
- Аутентификация - это проверка соответствия субъекта и того, за кого он пытается себя выдать, с помощью некой уникальной информации (имени входа и пароля, QR-кода).
- Конфигурационный файл (конфиг) - текстовый файл, содержащий информацию, которая определяет поведение приложения, компьютера или сетевого устройства.
- Искусственный интеллект (ИИ; англ. artificial intelligence, AI) — свойство интеллектуальных систем выполнять творческие функции, которые традиционно считаются прерогативой человека; наука и технология создания интеллектуальных машин, особенно интеллектуальных компьютерных программ.
- Мессенджер (Система мгновенного обмена сообщениями) – это программа, мобильное приложение или веб-сервис для мгновенного обмена сообщениями.
- Скрипт - это последовательность действий, описанных с помощью скриптового языка программирования для автоматического выполнения определенных задач.
- Распознавание речи - это часть обработки естественного языка, которая является частью искусственного интеллекта . В распознавании речи произнесенные слова / предложения переводятся в текст с помощью компьютера. Он также известен как преобразование речи в текст (STT).
- Хеш-функция (хэш, функция свёртки) — функция, осуществляющая преобразование массива входных данных произвольной длины в выходную битовую строку установленной длины, выполняемое определённым алгоритмом.

## ВВЕДЕНИЕ

Распознавание речи является одной из важнейших задач в области взаимодействия человека с компьютером. Распознавание речи имеет различные приложения - от автоматической транскрипции речевых данных (например, голосовой почты) до взаимодействия с роботами посредством речи. Успех продуктов с поддержкой речи доказал, что некоторая степень поддержки речи станет важным аспектом бытовой техники в обозримом будущем. В этой работе будет рассмотрено:

- как передать аудиофайл на распознавание речи по Wit.Ai API Speech
- как обойти ограничение длительности аудио-файла
- как передать результат в Telegram бот, чтобы узнать характер человека по его речи, словам в мессенджерах или письмах.

**Задача данной работы** — это приобрести навыки работы с сервисом по распознаванию речи wit.ai и получение результата обработки текста с помощью ИИ в Telegram боте PersonGuideRU.

## **ОСНОВНАЯ ЧАСТЬ**

### **1 Характеристика организации**

Практика проходит на основе Центра информационных технологий и систем органов исполнительной власти (ЦИТиС), который создан постановлением Совета Министров – Правительства Российской Федерации от 20 июня 1993 года № 563 по предложению Администрации Президента Российской Федерации. Одновременно ЦИТиС был определен в качестве российского участника Международного центра по информатике и электронике (ИнтерЭВМ).[5]

Миссия учреждения: «Участие в реализации государственной научно-технической и инновационной политики в части создания, развития и использования эффективной инфокоммуникационной инфраструктуры государственного и муниципального управления на основе защищенных отечественных информационных технологий, обеспечивающих кардинальное повышение уровня интеллектуальной поддержки процессов подготовки принятия решений и повышения эффективности государственного управления, а также удовлетворяющих актуальные потребности общества в современных наукоемких, высокотехнологичных инфокоммуникационных услугах».

Основные результаты научно-технической деятельности:

- разработан инструментальный технологический комплекс создания автоматизированных информационных систем различного назначения;
- разработаны новые принципы защиты информации в распределенных информационно-вычислительных средах коллективного пользования национальной нанотехнологической системы;
- созданы криптографические средства защиты информации нового поколения в высокоскоростных (10 Гбит/с) каналах связи с использованием протоколов IPv4/IPv6;
- разработан удостоверяющий центр нового поколения с повышенными технико-эксплуатационными характеристиками;



- создан ряд автоматизированных информационных систем специального назначения, существенно повышающих результативность служебной деятельности;
- разработаны архитектура и экспериментальные образцы персональных суперЭВМ на базе графических плат nVidia;
- создан цифровой репозиторий диссертаций и научно-технической документации НИОКР.

Парадигма научно-технической деятельности ФГАНУ ЦИТиС:

Обеспечение высокого научно-технического уровня разработок на основе соединения теоретико-экспериментальных результатов академической науки с передовыми российскими инфокоммуникационными технологиями.

Ключевой элемент технологической компетенции ФГАНУ ЦИТиС - адаптивная технология автоматизированного создания программно-информационных комплексов с использованием механизмов предметно-лингвистического анализа слабоструктурированных данных и синтеза новых знаний на базе онтологического подхода.

Программная научно-техническая деятельность:

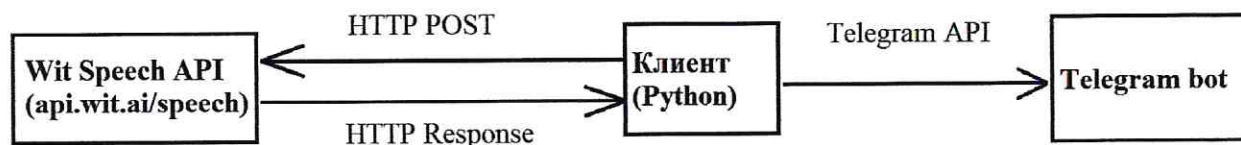
- Участие в ФЦП «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2007-2012 годы»
- Участие в ФЦП «Развитие инфраструктуры наноиндустрии в Российской Федерации на 2008 - 2010 годы»
- Участие в ФЦП «Мировой океан»
- Участие в ФЦП «Информационное общество»



## 2 Исследовательская часть

### 2.1 Общие сведения

Во введении была поставлена основная задача: работа с сервисом wit.ai и отправка текста в отдельный Telegram бот. Для решения задачи использовался язык программирования Python и вспомогательные библиотеки: модуль PyWit и Telethon. На рисунке 1 представлен процесс взаимосвязи компонентов между собой.



*Рисунок 1. Взаимосвязь компонентов*

PyWit - это пакет SDK Python для Wit.ai и предоставляет все необходимые методы для работы с API-интерфейсом сервиса [1].

Telegram Bot API, также известный как HTTP Bot API и отныне именуемый просто «Bot API» - это официальный способ Telegram для разработчиков контролировать своих собственных Telegram-ботов.

## 2.2 Инструменты

*Anaconda* — дистрибутив языков программирования Python и R, включающий набор популярных свободных библиотек, объединённых проблематиками науки о данных и машинного обучения. Основная цель — поставка единым согласованным комплектом наиболее востребованных соответствующим кругом пользователей тематических модулей (таких как NumPy, SciPy, Astropy и других) с разрешением возникающих зависимостей и конфликтов, которые неизбежны при одиночной установке. По состоянию на 2019 год содержит более 1,5 тыс. модулей.

*Pycharm* – интегрированная среда разработки для языка программирования Python [2]. Предоставляет средства для анализа кода, графический отладчик, инструмент для запуска юнит-тестов и поддерживает веб-разработку на Django. PyCharm разработана компанией JetBrains.

### 3 Практическая часть

#### 3.1 Создание ключа API

Чтобы использовать Wit.ai API необходимо создать приложение Wit.ai. Каждое приложение Wit.ai имеет токен доступа к серверу, который можно использовать в качестве ключа API.

Выполнив следующие действия можно создать приложение Wit.ai и сгенерировать ключ API:

- Перейти на домашнюю страницу Wit.ai и войти в свою учетную запись GitHub или Facebook.
- Нажать на значок « + » в строке меню вверху и создать новое приложение.
- Теперь необходимо открыть панель управления приложения и перейти в «Настройки» приложения.
- В разделе «Настройки» в разделе «Сведения об API» следует копировать токен доступа к серверу и теперь его можно использовать в качестве ключа API.

The screenshot displays the 'Settings' page for a Wit.ai application. At the top, it shows the 'App ID' as 1195105444. Below this, the 'App Name' is set to 'test1'. There are two dropdown menus: 'Language' is set to 'Russian (RU) (BETA)' and 'Timezone' is set to 'Europe/Moscow'. The 'Server Access Token' section is highlighted with a red rectangle; it shows the token 'IFDJVAH456ZSNO2KDPP' and a 'Generate new token' button. Below this is the 'Client Access Token' section, which shows the token 'U7032CYR2ZCB7K32337' and another 'Generate new token' button. At the bottom, there is an 'HTTP API' section with a text input field labeled 'Type in an utterance to...'. The entire interface is in Russian.

Рисунок 1. Получение Server Access Token



### 3.2 Взаимодействие с Wit Speech API

Wit.ai — это бесплатный движок распознавания речи [6]. Он поддерживает лишь аудио-записи длиной менее 15 секунд, но в Питоне существует библиотека «pydub», которая позволяет автоматически разделять аудио-файл на отрезки длиной в 15 секунд и распознавать их раздельно.

```
def split(filepath):  
    myaudio = AudioSegment.from_mp3(filepath)  
    chunk_length_ms = 15000 # pydub calculates in millisec  
    chunks = make_chunks(myaudio, chunk_length_ms) # Make chunks of one sec  
    return chunks
```

Рисунок 2. Функция для разделения длинного аудио-файла

Для передачи аудио-файла в формате MP3 на сервис следует использовать официальную библиотеку для работы с wit.ai - модуль PyWit. Ниже приведён фрагмент кода скрипта Python для взаимодействия с Wit Speech API.

```
def start_wit(filepath):  
    client_wit = Wit(config.get("Settings", "WIT_ACCESS_TOKEN"))  
    # client_wit= Wit("IFDJVAH456ZSN")  
    text = []  
    for i, chunk in enumerate(split(filepath)):  
        chunk_name = "{0}.mp3".format(i)  
        # print("exporting", chunk_name)  
        chunk.export(chunk_name, format="mp3")  
        with open(chunk_name, 'rb') as f:  
            resp = client_wit.speech(f, {'Content-Type': 'audio/mpeg3'})  
            # time.sleep(2)  
            text.append(resp['text'])  
            # print(resp['text'])  
    # print(text)  
    # print('OK')  
    return text
```

Рисунок 3. Функция для загрузки аудио-файла на wit.ai и получение результата

Wit Speech API принимает запросы HTTP POST. Запрос POST должен содержать:

- заголовки

headers = {'authorization': 'Bearer ' + wit\_access\_token, 'Content-Type': 'audio/mpeg3'} где wit\_access\_token - это API-ключ, который был сгенерирован на шаге 1 .

- данные

data = audio Передаваемые данные - это аудиопоток в формате MP3. Как можно заметить, файл длиннее 15 секунд сохраняется по шаблону <числовой ряд>.mp3

Данный HTTP-запрос имеет следующую конечную точку: <https://api.wit.ai/speech>

Пример ответа на HTTP-запрос выглядит так:

```
{u'_text': u'привет как ты',  
u'entities': {},  
u'msg_id': u'1ca8f790-4e83-443c-915c-914bc1a42100'}
```

### 3.3 Работа с конфигурационным файлом

Для работы с конфигурационными файлами изобретен формат - .ini, который позволяет содержать данные настроек программы Python. Библиотека в ядре Python включает в себя модуль, под названием configparser, который можно использовать для создания и работы с файлами конфигурации. На рисунке 4 продемонстрирована функция для чтения значения, записанных в .INI-файл

```
def Config(path):  
    """  
    Чтение конфига  
    :param path: путь к файлу конфигурации  
    :return: параметры конфига  
    """  
    import configparser # импортируем библиотеку  
    config = configparser.ConfigParser() # создаём объекта парсера  
    config.read(path, encoding='utf-8') # читаем конфиг  
    return config
```

*Рисунок 4. Функция чтения конфигурационного файла*

На рисунке 5 показано содержимое INI-файла, в котором записаны значения параметров – строка вида «ключ-значение».

```
# main.py  
[Settings]  
;path to the MP3 file  
PATH_MP3_FILE = 50_ready.mp3  
  
WIT_ACCESS_TOKEN = IFDJVAH456ZSN02KDPP0U  
  
; Developer Tools of Telegram  
api_id = 798  
api_hash = 913a0373a8b49fbd6712  
channel_id = 1459
```

*Рисунок 5. Файл config.ini*



### 3.4 Взаимодействие с Telegram API

Программа создана с помощью библиотеки «telethon» [3]. Эта библиотека предназначена для того, чтобы было легко писать программы на Python, которые могут взаимодействовать с Telegram. Когда программа запускается первый раз, то потребуется ввести телефон и код который приходит в Telegram, а потом это сохраняется как сессия файл и вводить уже не нужно.

```
def send_text_to_telegram():
    # +79854544883
    name = 'session_name'
    api_id = int(config.get("Settings", "api_id"))
    api_hash = config.get("Settings", "api_hash")
    channel_id = int(config.get("Settings", "channel_id"))

    client = TelegramClient(name, api_id, api_hash)
    client.start()
    for i, val in enumerate(start_with(config.get("Settings", "PATH_MP3_FILE"))):
        client.send_message(channel_id, val)
```

*Рисунок 7. Функция для отправки сообщений в определённый Telegram-канал*

#### Инструкция по отправке сообщения в Telegram из Python:

1. Первый шаг - создать бота с помощью @BotFather и получить token.
  - Открыть приложение Telegram и найти @BotFather.
  - Нажать кнопку «Пуск» или отправить «/start».
  - Затем отправить сообщение «/newbot», чтобы указать имя и имя пользователя.
  - После установки имени и имени пользователя, @BotFather предоставит токен API (HTTP API), который является токеном бота.
2. Затем создайте приложение в Telegram. Выполните следующие шаги:
  - a. Войти в ядро Telegram: <https://my.telegram.org>
  - b. Перейти в «Инструменты разработки API» и заполнить форму (в URL можно написать <http://localhost.ru>)

- с. Будут получены параметры `api_id` и `api_hash`, необходимые для авторизации пользователя.

### 3. Третий шаг - получить «chat\_id»

Необходимо создать тестовый канал и узнать id своего канала для того, чтобы программно слать туда сообщение.

- а. Нужно написать что-то в тестовом канале -> потом переотправить сообщение с тестового канала боту `@getmyid_bot` и далее будет сказано то, какой id тестового канала. Ответ будет с минусом, без кавычек.

- б. Второй способ — отправка в «Избранное»:

Посетите <https://api.telegram.org/bot<YourBOTToken>/getUpdates>

и получите chat\_id под `"my_chat_member"{"chat":{"id":`

### 3.5 Создание исполняемого файла

Существует большое количество библиотек, позволяющих сделать исполняемый файл формата .exe, среди которых самые популярные: cx\_Freeze, py2exe, nuitka, PyInstaller и другие.

Основные преимущества PyInstaller перед аналогичными инструментами заключаются в следующем, PyInstaller работает с Python 3.5-3.8, он строит меньше исполняемые файлы благодаря прозрачному сжатию, он полностью мультиплатформенный и использует поддержку ОС для загрузки динамических библиотек, таким образом, обеспечивается полная совместимость. PyInstaller анализирует файл .py и делает следующее:

1. Пишет файл .spec в той же папке, где находится скрипт.
2. Создает папку build в той же папке, где находится скрипт.
3. Записывает некоторые логи и рабочие файлы в папку build.
4. Создает папку dist в той же папке, где находится скрипт.
5. Пишет исполняемый файл в папку dist.

В итоге для сборки .exe файла с иконкой приложения использовалась следующая команда: `pyinstaller -F --icon=speech.ico main.py`

Результат команды продемонстрирован на рисунке 19.

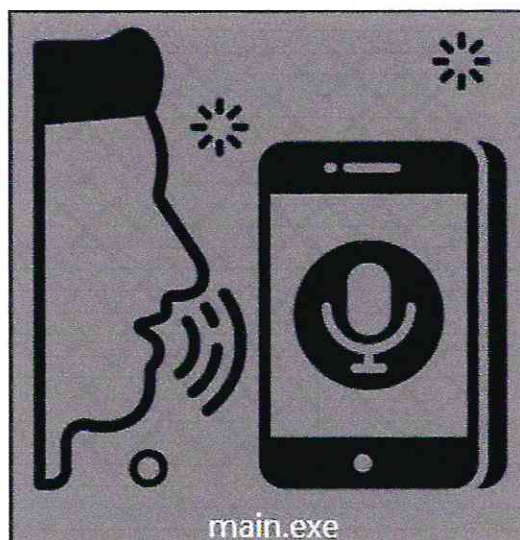


Рисунок 8. Исполняемый файл

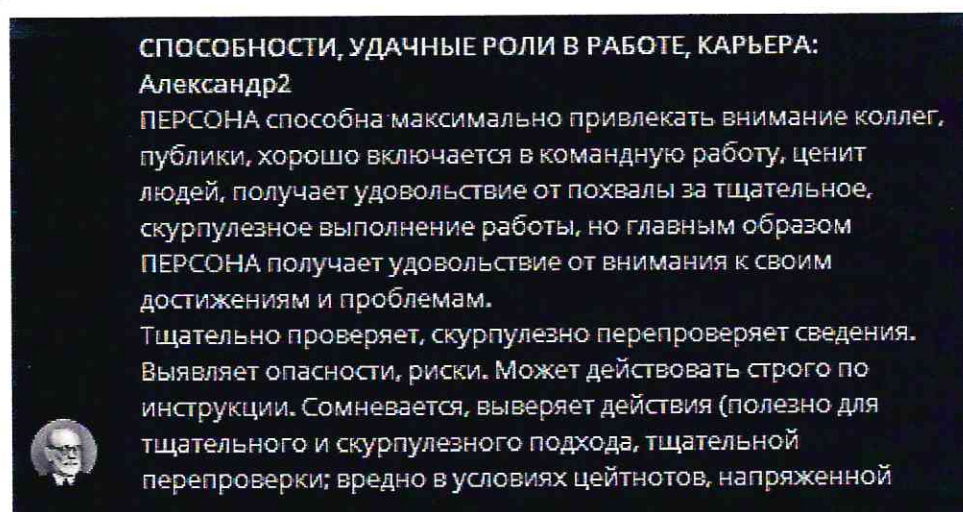


### 3.6 Обработка текста с помощью ИИ

Telegram бот PersonGuideRU помогает узнать характер человека по его речи, словам в мессенджерах или письмах [4]. Для этого следует передать минимум 1000 слов. Получение результата обработки текста с помощью искусственного интеллекта позволяет получить следующие характеристики:

- Прогноз способностей и сфер успеха
- Что можно доверять
- Привычки человека
- Прогноз семейных отношений
- Как вызвать интерес
- Типичные намерения. Прогноз будущих отношений

Ниже приведён фрагмент использования бота.



*Рисунок 9. Выдача характеристики после обработки переданного текста*

## ЗАКЛЮЧЕНИЕ

На практике были получены навыки работы с сервисом по распознаванию речи в текст Wit.ai, навык работы с Telegram API, работа с конфигурационными файлами и создание исполняемого файла на Windows. Таким образом, в результате прохождения практики было создано и получено 3 файла:

- INI-файл конфигурации – config.ini
- Python-скрипт – main.py
- Исполняемый файл для Windows – main.exe

Результат работы выложен по следующему URL:

[https://github.com/poljkee2010/Wit\\_speech\\_to\\_telegram\\_channel](https://github.com/poljkee2010/Wit_speech_to_telegram_channel)

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1) Документация PyWit [Электронный ресурс]

URL: <https://github.com/wit-ai/pywit>

(дата обращения: 14.08.2021)

2) Документация Python3 [Электронный ресурс]

URL: <https://docs.python.org/3/>

(дата обращения: 15.08.2021)

3) Документация Telethon [Электронный ресурс]

URL: <https://docs.telethon.dev/en/latest/index.html>

(дата обращения: 17.08.2021)

4) Официальный сайт Telegram бота PersonalGuide [Электронный ресурс]

URL: <http://personguide.mozello.ru/>

(дата обращения: 19.08.2021)

5) Официальный сайт ФГАУ ЦИТиС [Электронный ресурс]

URL: [https://www.citis.ru/citis\\_about.html](https://www.citis.ru/citis_about.html)

(дата обращения: 20.08.2021)

6) Официальный сайт Wit.ai [Электронный ресурс]

URL: <https://wit.ai/>

(дата обращения: 21.08.2021)