ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Факультет информационных технологий
Кафедра Информатики и информационных технологий

направление подготовки 09.04.02 «Информационные системы и технологии»,
профиль «Мобильные приложения»

Практическая работа №4

«Разработка программы распознавания и генерации речи»

Дисциплина: Искусственный интеллект в мобильных системах

Выполнил: студент группы 234-332

Киселев С.А.

Дата, подпись 23.03.2025 _____

(Дата) (Подпись)

Проверила: Дагаев А.Е. _____

(Оценка)

Дата, подпись _____ _____

(Дата) (Подпись)

Замечания:

_____

_____

Москва

2025

Оглавление

Цель

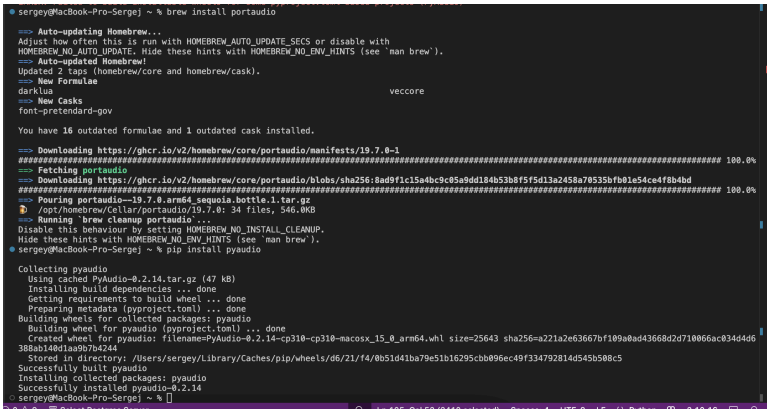
Целью работы является написание программы на Python, которая бы воспринимала 7-8 голосовых команд и реагировала на голосовую команду stop - окончить работу, используя материалы лекции "Распознавание и генерация речи в мобильных системах". Ответы программа должна выдавать в виде голосовых сообщений (желательно на русском языке, но можно и на английском).

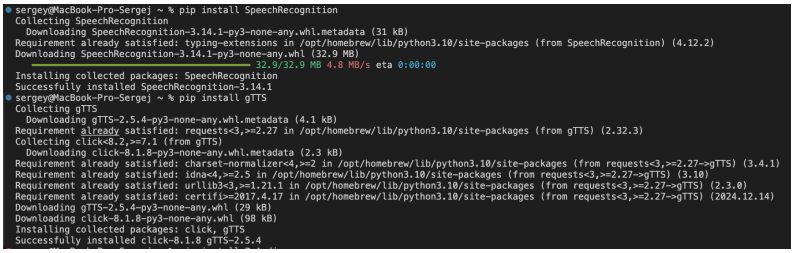
Реализация

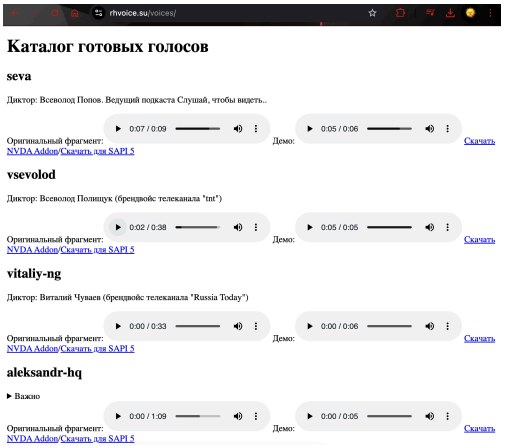
В ходе данной практической работы была разработана программа на Python, которая воспринимает 9 голосовых команд:

- "открой почту" -> открывает сайт почты https://mail.google.com/
- "сколько времени" / "который час" / "сколько время" -> произносит текущее время
- "как тебя зовут” -> сообщает, что это голосовой ассистент
- "какие новости" / "что нового" -> читает последние новости
- "какая погода" -> произносит погоду в городе Москва
- "найди в гугле" / "загугли" -> выполняет поиск
- "расскажи анекдот" -> рассказывает анекдот
- "случайное число" -> называет случайное число от 0 до 10
- "стоп" -> завершает работу программы

Сначала устанавливаем необходимые для работы библиотеки:







Установка дополнительных библиотек

Теперь необходимо написать код программы. Программа представляет собой голосового ассистента, который воспринимает голосовые команды пользователя, обрабатывает их и отвечает голосом.

speak(text) используется для озвучивания текста. Преобразует переданный текст в речь с помощью библиотеки pyttsx3 и воспроизводит его через динамики.

```
def speak(text):
    """Функция озвучивания текста"""
    engine.say(text)
    engine.runAndWait()
```

recognize_speech() - функция для распознавания речи. Слушает звук с микрофона и преобразует речь в текст с помощью Google Speech Recognition. Если речь не распознана, возвращает None. Возвращает текст команды, которую распознал ассистент.

```
def recognize_speech():
    """Функция распознавания речи"""
    recognizer = sr.Recognizer()
    with sr.Microphone() as source:
        print("Слушаю...")
        recognizer.adjust_for_ambient_noise(source)
        try:
            audio = recognizer.listen(source, timeout=5)
            command = recognizer.recognize_google(audio, language="ru-RU").lower()

            print(f"Вы сказали: {command}")
            return command
        except sr.UnknownValueError:
            print("Не удалось распознать речь")
            return None
        except sr.RequestError:
            print("Ошибка сервиса распознавания")
            return None
```

time_to_text() - функция, которая преобразует текущее время в текстовый формат. Получает текущее время и преобразует его в строку, используя склонения для "часов" и "минут".

```python
def time_to_text():
    dict_hours = {1: 'час', 2: 'часа', 3: 'часа', 4: 'часа', 5: 'часов', 6: 'часов',
                  7: 'часов', 8: 'часов', 9: 'часов', 10: 'часов', 11: 'часов', 12: 'часов',
                  13: 'часов', 14: 'часов', 15: 'часов', 16: 'часов', 17: 'часов', 18: 'часов',
                  19: 'часов', 20: 'часов', 21: 'час', 22: 'часа', 23: 'часа', 0: 'часов'}
    dict_minutes = {
        'минута': [1, 21, 31, 41, 51],
        'минуты': [2, 3, 4, 22, 23, 24, 32, 33, 34, 42, 43, 44, 52, 53, 54],
        'минут': [5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20,
                  25, 26, 27, 28, 29, 30,
                  35, 36, 37, 38, 39, 40,
                  45, 46, 47, 48, 49, 50,
                  55, 56, 57, 58, 59]}
    now = datetime.datetime.now()
    h = now.hour
    m = now.minute

    str_time = str(h) + dict_hours[h] + ''
    for minutes in dict_minutes:
        if m in dict_minutes[minutes]:
            str_time += str(m) + ' ' + minutes
            break

    return str_time
```

get_weather() - функция для получения текущей погоды в городе (по умолчанию в Москве). Использует OpenWeather API для получения информации о текущей погоде в городе, указанном в переменной CITY. Возвращает описание погоды.

```python
def get_weather():
    """Получение текущей погоды"""
    url = f"https://api.openweathermap.org/data/2.5/weather?q={CITY}&appid={API_KEY}&units=metric&lang=ru"
    try:
        response = requests.get(url)
        data = response.json()
        if data["cod"] == 200:
            temp = data["main"]["temp"]
            description = data["weather"][0]["description"]
            return f"Сейчас в {CITY} {description}, температура {temp} градусов."
        else:
            return "Не удалось получить данные о погоде."
    except:
        return "Ошибка при получении погоды."
```

get_news() - функция для получения последних новостей из Интернета. Использует News API для получения новостей. В данный момент это новости по ключевому запросу "technology". Получает заголовки 3 последних новостей и возвращает их в виде текста.

```python
def get_news():
    """Получение новостей"""
    url = f"https://newsapi.org/v2/everything?q=technology&apiKey={NEWS_API_KEY}"
    try:
        response = requests.get(url)
        data = response.json()
        if data["status"] == "ok":
            articles = data["articles"][:3]  # Берем 3 главные новости
            news_list = [article["title"] for article in articles]
            return "Вот главные новости: " + ". ".join(news_list)
        else:
            return "Не удалось получить новости."
    except:
        return "Ошибка при получении новостей."
```

main() - основная функция программы, которая слушает голосовые команды и выполняет соответствующие действия. Запускает цикл, в котором программа распознает голосовые команды. Для каждой команды выполняет соответствующие действия (озвучивание времени, открытие почты, получение погоды, новостей и другие действия).

```python
def main():
    speak("Привет! Я голосовой ассистент. Скажите команду.")
    while True:
        command = recognize_speech()

        if command == "который час" or  command == "сколько времени" or  command == "сколько время":
            speak(time_to_text())
            print(time_to_text())
            continue
        if command == "открой почту":
            URL = "https://mail.google.com/"
            webbrowser.open(URL)
        if command == "какая погода":
            speak(get_weather())
            print(get_weather())
            continue
        if command == "что нового" or command == "какие новости":
            speak(get_news())
            print(get_news())
            continue
        if command == "найди в гугле" or command == "загугли":
            speak("открываю гугл")
            search_query = command
            webbrowser.open(f"https://www.google.com/")
        if command == "случайное число":
            number = random.randint(0, 10)
            print(f"случайное число: {number}")
            speak(f"случайное число: {number}")
            continue
        if command:
            response = commands.get(command, "Я не знаю такой команды.")
            speak(response)
            if command == "стоп":
                break

if __name__ == "__main__":
    main()
```

Результат работы программы представлен ниже, некоторые программы продублированы выводом ответа в консоль для понимания того, что ответила программа.

```
sergey@MacBook-Pro-Sergej II % python speech.py
Слушаю...
Вы сказали: как тебя зовут
Слушаю...
Вы сказали: сколько время
11часов58 минут
Слушаю...
Вы сказали: какая погода
Сейчас в Москва ясно, температура 5.65 градусов.
Слушаю...
Вы сказали: расскажи анекдот
Слушаю...
Вы сказали: какие новости
Вот главные новости: Under Trump, AI Scientists Are Told to Remove ‘Ideological Bias’ From Powerful Models. Google Is Developing Technology to Delive
r Internet Via Light Bridges. LG’s new air conditioner directs cool air towards people it detects.
Слушаю...
Вы сказали: случайное число
случайное число: 9
Слушаю...
Вы сказали: открой почту
Слушаю...
Вы сказали: стоп
sergey@MacBook-Pro-Sergej II %
```

Вывод

В ходе выполнения практического задания была успешно программа на Python, которая бы воспринимает 9 голосовых команд: "открой почту", "сколько времени" / "который час" / "сколько время", "как тебя зовут”, "какие новости" / "что нового", "найди в гугле" / "загугли" , "расскажи анекдот", "случайное число", "открой почту", "стоп".

Ответы на эти команды программа выдает в виде голосовых сообщений на русском языке.