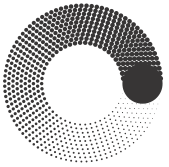
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

****

**МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

***Факультет информационных технологий***

***Кафедра Информатики и информационных технологий***

**направление подготовки**

**09.04.02 «Информационные системы и технологии»,**

**ПРАКТИЧЕСКАЯ РАБОТА № 4**

**Дисциплина: Искусственный интеллект в мобильных системах**

**Тема: Распознавание и генерация речи в мобильных системах**

**Выполнил: студент группы 224-371** Неретин В.М.

(Фамилия И.О.)

**Проверил:** Попов Д.И.

(Фамилия И.О. )

**Дата, подпись** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  ***\_\_\_\_\_\_\_\_\_\_\_***

(Дата) (Подпись)

ОГЛАВЛЕНИЕ

[ЗАДАНИЕ 4](#_Toc159676870)

[ВАРИАНТ 4](#_Toc159676871)

[ХОД РАБОТЫ 5](#_Toc159676872)

[ВЫВОД 12](#_Toc159676873)

# **ЗАДАНИЕ**

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

# **ХОД РАБОТЫ**

Для распознавания речи была использована библиотека SpeechRecognition. Библиотека SpeechRecognition в Python предоставляет возможность программного распознавания речи. Она использует API распознавания Google для перевода аудиофайлов в текст. Для использования нужно иметь установленный Python 3.6 или выше и микрофон, подключенный к компьютеру или устройству.

Чтобы научить ИИ разговаривать использовалась библиотека gTTS. Библиотека gTTS (Google Text-to-Speech) используется для синтеза речи в Google Cloud. Она позволяет генерировать аудиофайл из текста на разных языках и с разными голосами.

PyAudio – ещё одна библиотека для генерации звука. PyAudio — это небольшая переносная Python-библиотека, предоставляющая Python-интерфейс к аудиоустройствам, которые поддерживаются в операционной системе. PyAudio может быть использована для записи и воспроизведения звука, а также для управления настройками аудиоустройства.

PyAudio является кросс-платформенной библиотекой и работает на различных операционных системах, включая Windows, macOS и Linux. PyAudio предоставляет простой и удобный интерфейс для работы со звуковыми устройствами, что делает его популярным выбором для разработчиков, работающих в области обработки звука и музыки.

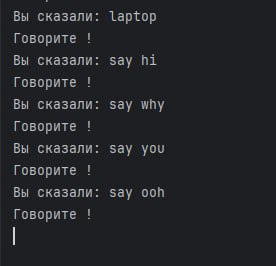
В данном коде приводится пример настройки голоса ИИ:

engine = pyttsx3.init()  
voices = engine.getProperty('voices')  
for voice in voices:  
 print('------')  
 print(f'Имя: {voice.name}')  
 print(f'ID: {voice.id}')  
 print(f'Язык(и): {voice.languages}')  
 print(f'Пол: {voice.gender}')  
 print(f'Возраст: {voice.age}')  
engine.setProperty('voice', 0)  
engine.setProperty('rate', 150)  
engine.setProperty('volume', 0.5)  
engine.say(words)  
engine.runAndWait()

Обработчик команд пользователя выглядит следующим образом:

def command():  
 r = sr.Recognizer()  
 with sr.Microphone() as source:  
 print("Говорите !")  
 talk('Говорите !')  
 r.pause\_threshold = 1  
 r.adjust\_for\_ambient\_noise(source, duration=1)  
 audio = r.listen(source)  
 try:  
 zadanie = r.recognize\_google(audio).lower()  
 print("Вы сказали: " + zadanie)  
 talk("Вы сказали: " + zadanie)  
 r.pause\_thresholds=1  
 except sr.UnknownValueError:  
 talk("I don't understand you")  
 zadanie = command()  
 return zadanie

Примеры взаимодействия с ИИ:



Исходный код представлен в приложении А

# **ВЫВОД**

В результате выполнения практической работы была написана программа на Python, воспринимающая более 5 голосовых команд и реагирующая на голосовую команду stop. Ответа программа выдает на английском языке.

Приложение А — исходный код практической работы

import speech\_recognition as sr

import os

import sys

import webbrowser

import pyttsx3

import datetime

def talk(words):

microphone\_list = sr.Microphone.list\_microphone\_names()

print(microphone\_list)

engine = pyttsx3.init()

voices = engine.getProperty('voices')

for voice in voices:

print('------')

print(f'Имя: {voice.name}')

print(f'ID: {voice.id}')

print(f'Язык(и): {voice.languages}')

print(f'Пол: {voice.gender}')

print(f'Возраст: {voice.age}')

engine.setProperty('voice', 0)

engine.setProperty('rate', 150)

engine.setProperty('volume', 0.5)

engine.say(words)

engine.runAndWait()

def command():

r = sr.Recognizer()

with sr.Microphone() as source:

print("Говорите !")

talk('Говорите !')

r.pause\_threshold = 1

r.adjust\_for\_ambient\_noise(source, duration=1)

audio = r.listen(source)

try:

zadanie = r.recognize\_google(audio).lower()

print("Вы сказали: " + zadanie)

talk("Вы сказали: " + zadanie)

r.pause\_thresholds = 1

except sr.UnknownValueError:

talk("I don't understand you")

zadanie = command()

return zadanie

def ParseZadanie(zadanie):

if 'open mail' in zadanie:

talk('Open right now')

URL = "https://mail.ru"

webbrowser.open(URL)

elif 'stop' in zadanie:

talk('OK stopping now')

sys.exit()

elif 'laptop' in zadanie:

talk('Laptop is?')

url = 'https://en.wikipedia.org/wiki/Laptop'

webbrowser.open(url)

elif 'say hi' in zadanie:

talk('Haaaaaaaaaaaaay!')

elif 'say why' in zadanie:

talk('WOOOOOOOOOOOOAH!')

elif 'say you' in zadanie:

talk('Ooooooooooh!')

location = command()

elif 'play music' in zadanie:

talk('papaaa ba baaaaaaa!')

elif 'keep counting' in zadanie:

talk('one, two, three, four!')

elif 'my name is' in zadanie:

talk('nice to meet you! my name is ...')

while True:

ParseZadanie(command())

talk("Again?")