Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по программе «**Умный ассистент**»

Выполнил: Калинин Алексей

Группа: ПР-32

Преподаватель: Шевцова В.Д.

2023

**Задание:**

Создать умного ассистента на python, который будет распознавать и

синтезировать речь, «Подбрасывать монетку», здороваться и прощаться, менять настройки языка распознавания и синтеза речи на ходу.

**Входные данные**

Пользователь с помощью микрофона говорит фразы

**Выходные данные**

Программа говорит что сказал пользователь и выполняет команды, если таковые были услышаны

**Листинг программы**

Файл main.py

from flask import Flask, render\_template, request, jsonify  
import threading  
import assistent  
app = Flask(\_\_name\_\_)  
conversation = []  
@app.route('/')  
def index():  
 return render\_template('index.html')  
@app.route('/start\_assistant', methods=['POST'])  
def start\_assistant():  
 global conversation  
 conversation = []  
 threading.Thread(target=run\_assistant).start()  
 return jsonify(success=True)  
@app.route('/get\_conversation', methods=['GET'])  
def get\_conversation():  
 global conversation  
 return jsonify(conversation=conversation)  
def run\_assistant():  
 global conversation  
 speech\_synthesizer = assistent.initialize\_speech\_synthesizer()  
 online\_recognizer = assistent.initialize\_speech\_recognizer()  
 offline\_recognizer = assistent.initialize\_vosk\_recognizer()  
 while True:  
 conversation.append("Скажите что-нибудь:")  
 user\_input = assistent.recognize\_speech\_online(online\_recognizer)  
 if user\_input:  
 conversation.append(f"Вы сказали: {user\_input}")  
 print(f"Вы сказали: {user\_input}")  
 if "подбрось монетку" in user\_input.lower():  
 assistent.flip\_coin(speech\_synthesizer)  
 elif assistent.greet\_or\_farewell(speech\_synthesizer, user\_input):  
 break  
 elif "английский" in user\_input.lower():  
 assistent.change\_language\_settings(online\_recognizer, speech\_synthesizer, "en-US")  
 elif "русский" in user\_input.lower():  
 assistent.change\_language\_settings(online\_recognizer, speech\_synthesizer, "ru-RU")  
 else:  
 assistent.synthesize\_speech(speech\_synthesizer, f"Вы сказали: {user\_input}")  
if \_\_name\_\_ == '\_\_main\_\_':  
 app.run(debug=True)

файл assistant.py

import speech\_recognition as sr  
import pyttsx3  
import vosk  
import os  
import pyaudio  
import random  
def initialize\_speech\_synthesizer():  
 engine = pyttsx3.init()  
 return engine  
def initialize\_speech\_recognizer():  
 recognizer = sr.Recognizer()  
 return recognizer  
def initialize\_vosk\_recognizer():  
 model\_path = r"C:\Users\ASUS\PycharmProjects\voice\model\vosk-model-small-ru-0.22"  
 if not os.path.exists(model\_path):  
 print("Пожалуйста, скачайте модель с https://alphacephei.com/vosk/models и распакуйте её в папку 'model' в текущем каталоге.")  
 exit(1)  
 model = vosk.Model(model\_path)  
 recognizer = vosk.KaldiRecognizer(model, 16000)  
 return recognizer  
def recognize\_speech\_online(recognizer):  
 with sr.Microphone() as source:  
 print("Скажите что-нибудь:")  
 audio = recognizer.listen(source)  
 try:  
 text = recognizer.recognize\_google(audio, language="ru-RU")  
 print(f"Вы сказали: {text}")  
 return text  
 except sr.UnknownValueError:  
 print("Не удалось распознать речь")  
 except sr.RequestError as e:  
 print(f"Ошибка сервиса распознавания речи; {e}")  
 return None  
def recognize\_speech\_offline(recognizer):  
 p = pyaudio.PyAudio()  
 stream = p.open(format=pyaudio.paInt16, channels=1, rate=16000, input=True, frames\_per\_buffer=8000)  
 stream.start\_stream()  
  
 print("Скажите что-нибудь:")  
 while True:  
 data = stream.read(4000)  
 if len(data) == 0:  
 break  
 if recognizer.AcceptWaveform(data):  
 result = recognizer.Result()  
 result\_dict = eval(result)  
 print(f"Вы сказали: {result\_dict['text']}")  
 return result\_dict['text']  
 return None  
def synthesize\_speech(engine, text):  
 engine.say(text)  
 engine.runAndWait()  
def flip\_coin(engine):  
 result = random.choice(["орел", "решка"])  
 synthesize\_speech(engine, f"Выпал {result}")  
def greet\_or\_farewell(engine, text):  
 if "привет" in text.lower():  
 synthesize\_speech(engine, "Здравствуйте!")  
 elif "до свидания" in text.lower():  
 synthesize\_speech(engine, "До свидания!")  
 return True  
 return False  
def change\_language\_settings(recognizer, engine, language\_code):  
 recognizer.language = language\_code  
 voices = engine.getProperty('voices')  
 for voice in voices:  
 if language\_code in voice.languages:  
 engine.setProperty('voice', voice.id)  
 break  
 synthesize\_speech(engine, f"Язык изменен на {language\_code}")  
def main():  
 speech\_synthesizer = initialize\_speech\_synthesizer()  
 online\_recognizer = initialize\_speech\_recognizer()  
 offline\_recognizer = initialize\_vosk\_recognizer()  
 while True:  
 user\_input = recognize\_speech\_online(online\_recognizer)  
 if user\_input:  
 print(f"Вы сказали: {user\_input}")  
 if "подбрось монетку" in user\_input.lower():  
 flip\_coin(speech\_synthesizer)  
 elif greet\_or\_farewell(speech\_synthesizer, user\_input):  
 break  
 elif "английский" in user\_input.lower():  
 change\_language\_settings(online\_recognizer, speech\_synthesizer, "en-US")  
 elif "русский" in user\_input.lower():  
 change\_language\_settings(online\_recognizer, speech\_synthesizer, "ru-RU")  
 else:  
 synthesize\_speech(speech\_synthesizer, f"Вы сказали: {user\_input}")  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

**Скрин-шот экранов (если есть)**

**Изображение выглядит как снимок экрана, текст, мультфильм

Автоматически созданное описание**