# импорт необходимых модулей  
  
import pyttsx3  
import os  
import random  
import webbrowser  
import time  
import speech\_recognition as sr  
import pandas as pd  
from tkinter import \*  
from fuzzywuzzy import fuzz  
from colorama import \*  
import datetime  
  
import Speak  
  
# раздел глобальных переменных  
  
text = ''  
r = sr.Recognizer()  
engine = pyttsx3.init()  
adress = ''  
j = 0  
task\_number = 0  
  
ndel = ['джарвис', 'jarvis', 'джар', 'ладно', 'не могла бы ты', 'пожалусйта']  
  
commands = ['привет', 'открой файл', 'выключи комп', 'выруби компьютер', 'пока', 'покажи файл', 'покажи список команд',  
 'открой vk', 'открой браузер', 'включи vk', 'открой интернет', 'открой youtube', 'включи музон', 'кто ты'  
 'вруби музыку', 'очисти файл', 'найди', 'время', 'покажи погоду', 'ринх', 'ргэу', 'открой почту',  
 'открой стату', 'покажи cтатистику','открой музыку', 'переведи', 'планы', 'на будущее',  
 'что планируется', 'как тебя зовут','кто']  
  
  
# раздел описания функций комманд  
  
def pri\_com(): # выводит на экран историю запросов, также использован модуль pandas  
 z = {}  
 mas = []  
 mas2 = []  
 mas3 = []  
 mas4 = []  
 file = open('commands.txt', 'r', encoding='UTF-8')  
 k = file.readlines()  
 for i in range(len(k)):  
 line = str(k[i].replace('\n', '').strip())  
 mas.append(line)  
 file.close()  
 for i in range(len(mas)):  
 x = mas[i]  
 if x in z:  
 z[x] += 1  
 if not (x in z):  
 b = {x: 1}  
 z.update(b)  
 if not (x in mas2):  
 mas2.append(x)  
 for i in mas2:  
 mas3.append(z[i])  
 for i in range(1, len(mas3) + 1):  
 mas4.append(str(i) + ') ')  
 list = pd.DataFrame({  
 'command': mas2,  
 'count': mas3  
 }, index=mas4)  
 list.index.name = '№'  
 print(list)  
  
  
def plans():  
 global engine  
 plans = '''   
 Моя задача заключается четко выполнять требования ТамирлАна, ведь он меня сделал.  
 Также в дальнейшем я буду работать как умный дом и я думаю, что ТамирлАн сделает оптимизацию моего кода,  
 ведь он хорший кодер. В дальнейшем планируется распрделиться по всему дому с помощью  
 микроконтроллера Ардуино. Планируется добавить к нему ИИ и сделать тело, а также разместить  
 на таких площадках как Play Market.   
 '''  
 engine.say(plans)  
  
  
def clear\_analis(): # очистка файла с историей запросов  
 global engine  
 file = open('C:\\Users\\tchin\\Desktop\\commands.txt', 'w', encoding='UTF-8')  
 file.close()  
 engine.say('Файл аналитики очищен!')  
  
  
def add\_file(x):  
 file = open('C:\\Users\\tchin\\Desktop\\commands.txt', 'a', encoding='UTF-8') # открыть факл с кодировкой ютф-8  
 if x != '':  
 file.write(x + '\n')  
 file.close()  
  
  
  
  
  
  
def comparison(x): # осуществляет поиск самой подходящей под запрос функции  
 global commands, j, add\_file  
 ans = ''  
 for i in range(len(commands)): # Цикл до длины команды  
 k = fuzz.ratio(x, commands[i])  
 if (k > 50) & (k > j): # Схожесть должна быть больше 50 процентов  
 ans = commands[i]  
 j = k  
 if (ans != 'пока') & (ans != 'привет'):  
 add\_file(ans)  
 return (str(ans))  
  
  
def web\_search(): # осуществляет поиск в интернете по запросу (adress)  
 global adress  
 webbrowser.open('https://yandex.ru/yandsearch?clid=2028026&text={}&lr=11373'.format(  
 adress)) # Поиск то, что находится в перем текст  
  
  
def check\_searching(): # проверяет нужно-ли искать в интернете  
 global text, wifi\_name, add\_file  
 global adress  
 global web\_search  
 if 'найди' in text:  
 add\_file('найди')  
 adress = text.replace('найди', '').strip()  
 text = text.replace(adress, '').strip()  
 web\_search()  
 text = ''  
 elif 'найти' in text:  
 add\_file('найди')  
 adress = text.replace('найти', '').strip()  
 text = text.replace(adress, '').strip()  
 web\_search()  
 text = ''  
 adress = ''  
  
  
def clear\_task(): # удаляет ключевые слова  
 global text, ndel  
 for z in ndel:  
 text = text.replace(z, '').strip()  
 text = text.replace(' ', ' ').strip()  
  
  
def hello(): # функция приветствия  
 global engine  
 z = ["Jarvis здесь, Рад снова вас слышать!", 'Что вам угодно?', 'Привет. Чем-нибудь помочь?'] # Несколько вариаций  
 x = random.choice(z) # Рандомом выбираем вариацию  
 engine.say(x) # Проговариваем переменную x  
  
  
def quit(): # функция выхода из программы  
 global engine  
 x = ['Тамирлан, жаль что ты уходишь', 'рада была помочь', 'всегда к вашим услугам'] # Несколько вариаций  
 engine.say(random.choice(x)) # Проговорить переменную x с помощью биб pyttsx3  
 engine.runAndWait()  
 engine.stop()  
 os.system('cls') # очистка консоли  
 exit(0) # завершение  
  
  
def show\_cmds(): # выводит на экран список доступных комманд  
 my\_com = ['привет', 'открой файл', 'выключи компьютер', 'пока', 'покажи список команд',  
 'открой vk', 'открой интернет', 'открой youtube', 'включи музыку', 'очисти файл', 'покажи cтатистику',  
 'перевод', 'показывает время', 'ринх', 'открой почту']  
 # Сверху были представлены доступные команды  
 for i in my\_com: # их вывод через цикл  
 print(i) # вывод  
 time.sleep(2)  
  
  
def brows(): # открывает браузер  
 webbrowser.open('https://google.ru') # Открывает гугл  
  
  
def ovk(): # открывает вк  
 webbrowser.open('https://vk.com/feed') # Открывает вк  
  
  
def music():  
 webbrowser.open('https://vk.com/audios265260473') # открывает музыку  
  
  
def youtube(): # открывает ютюб  
 webbrowser.open('https://www.youtube.com') # Открывает ютюб  
  
  
def shut(): # выключает компьютер  
 global quit  
 os.system('shutdown /s /f /t 10') # Стандартная команда для выключения ПК через cmd  
 quit() # функция выхода из программы  
  
  
def weather():  
 webbrowser.open(  
 'https://yandex.ru/search/?text=%D0%BF%D0%BE%D0%B3%D0%BE%D0%B4%D0%B0%20%D0%B2%20%D1%80%D0%BE%D1%81%D1%82%D0%BE%D0%B2%D0%B5%20%D0%BD%D0%B0%20%D0%B4%D0%BE%D0%BD%D1%83&clid=2270455&banerid=6301000000%3A5f5bdc4eaac798001b7c8db4&win=454&&lr=39') # Открывает погоду  
  
  
def rsue():  
 engine.say("Ростовский государственный университет находится на улице Большая Садовая ул., 69, Ростов-на-Дону")  
 engine.say("Открываю официальный сайт")  
 webbrowser.open('https://rsue.ru')  
  
  
def mail():  
 webbrowser.open('https://e.mail.ru/inbox/?back=1')  
  
  
def who():  
 engine.say("я Jarvis, симуляция искусственого интелекта который использовал железный человек начиная с фильма "  
 "мстители эра альтрона")  
  
  
def hours():  
 now = datetime.datetime.now()  
 engine.say("Сейчас " + str(now.hour) + ":" + str(now.minute))  
  
  
def check\_translate():  
 global text, tr  
 tr = 0  
 variants = ['переведи', 'перевести', 'переводить', 'перевод']  
 for i in variants:  
 if (i in text) & (tr == 0):  
 word = text  
 word = word.replace('переведи', '').strip()  
 word = word.replace('перевести', '').strip()  
 word = word.replace('переводить', '').strip()  
 word = word.replace('перевод', '').strip()  
 word = word.replace('слово', '').strip()  
 word = word.replace('слова', '').strip()  
 webbrowser.open('https://translate.google.ru/#view=home&op=translate&sl=auto&tl=ru&text={}'.format(word))  
 tr = 1  
 text = ''  
  
  
cmds = {  
 'привет': hello, 'выключи компьютер': shut,  
 'пока': quit, 'покажи cтатистику': pri\_com, 'покажи список команд': show\_cmds,  
 'открой браузер': brows, 'включи vk': ovk,  
 'открой youtube': youtube, 'включи музыку': music, 'открой vk': ovk,  
 'включи музон': music, 'очисти файл': clear\_analis, 'покажи файл': pri\_com,  
 'открой файл': pri\_com, 'открой музыку': music,  
 'планы': plans, 'на будущее': plans, 'что планируется': plans,  
 'переведи': check\_translate, 'найти': check\_searching, 'найди': check\_searching,  
 'текущее время': hours, 'сейчас времени': hours, 'который час': hours, 'время': hours,  
 'покажи погоду': weather, 'ринх': rsue, 'ргэу': rsue, 'открой почту': mail, 'кто ты': who,  
 'как тебя зовут': who, 'кто': who,  
}  
  
  
# распознавание  
  
def talk():  
 global text, clear\_task # Глобальная переменная текст и очищение текста  
 text = ''  
 with sr.Microphone() as sourse: # Использовать микрофон  
 print('Я вас слушаю: ')  
 r.adjust\_for\_ambient\_noise(sourse, duration=0.5) # настройка посторонних шумов  
 audio = r.listen(sourse, phrase\_time\_limit=3) # Задае речим лимит 3 секунды, на обработку  
 try:  
 text = (r.recognize\_google(audio, language="ru-RU")).lower() # Распознование русской речи через либ  
 except(sr.UnknownValueError):  
 pass  
 except(TypeError):  
 pass  
 os.system('cls')  
 lb['text'] = text  
 clear\_task()  
  
  
# выполнение команд  
  
def cmd\_exe():  
 global cmds, engine, comparison, check\_searching, task\_number, text, lb  
 check\_translate()  
 text = comparison(text)  
 print(text)  
 check\_searching()  
 if (text in cmds):  
 if (text != 'привет') & (text != 'пока') & (text != 'покажи список команд'):  
 k = ['Секундочку', 'Сейчас сделаю', 'уже выполняю']  
 engine.say(random.choice(k))  
 cmds[text]()  
 elif text == '':  
 pass  
 else:  
 print('Команда не найдена!')  
 task\_number += 1  
 if (task\_number % 10 == 0):  
 engine.say('У вас будут еще задания?')  
 engine.runAndWait()  
 engine.stop()  
  
  
# исправляет цвет  
  
print(Fore.GREEN + '', end='')  
os.system('cls')  
  
  
# основной бесконечный цикл  
  
def main():  
 global text, talk, cmd\_exe, j  
 try:  
 talk()  
 if text != '':  
 cmd\_exe()  
 j = 0  
 except(UnboundLocalError):  
 pass  
 except(TypeError):  
 pass  
  
  
# раздел создания интерфейса  
  
root = Tk()  
root.geometry('250x300')  
root.configure(bg='gray23')  
root.title('Jarvis')  
root.resizable(False, False)  
  
lb = Label(root, text=text)  
lb.configure(bg='gray')  
lb.place(x=25, y=25, height=25, width=200)  
  
but1 = Button(root, text='Сказать', command=main)  
but1.configure(bd=1, font=('Castellar', 25), bg='gray')  
but1.place(x=50, y=160, height=50, width=150)  
  
but2 = Button(root, text='Выход', command=quit)  
but2.configure(bd=1, font=('Castellar', 25), bg='gray')  
but2.place(x=50, y=220, height=50, width=150)  
  
root.mainloop()  
  
while True:  
 main()