****

Министерство образования и науки 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)

**Домашнее задание по курсу**

**"Технологии мультимедиа"**

**(ИУ5)**

**«Переводчик с русского на французский»**

**Овчинников С.С.**

**ИУ5-61Б**

**Москва, МГТУ - 2021 год**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Задание:** Голосовой переводчик с русского на французский язык.

# Пример реализации

eSpeak это компактный свободный программный синтезатор речи, поддерживающий Speech Synthesis Markup Language (SSML). В настоящее время исходный проект находится в неактивном состоянии в связи с исчезновением его автора Джонатана Даддингтона. Сообществом разрабатывается его форк eSpeakNG.

Слова входного текста для синтезирования проходят два этапа обработки:

* слово в голосовом представлении преобразуется в слова;
* генерируется звуковой сигнал на основе полученной последовательности.

Правила для получения последовательности фонем хранятся в виде «A, B, C = D». Где B — это рассматриваемая буква, A и C — контекст окружения этой буквы в слове, и D — это фонема, в которую эта буква может быть преобразована. Контекст окружения может задаваться как конкретными буквами, так и специальными символами, обозначающими группы букв. Правила синтезатора допускают неоднозначное определение таких цепочек. Для разрешения этой неоднозначности синтезатор назначает приоритет каждому правилу, который вычисляется на основе количества букв, задействованных в правиле, и степени конкретности определения контекста окружения. В правилах также можно указывать различия в трансляции в зависимости от ударения.

В eSpeak гласные звуки всегда синтезируются, звонкие согласные получаются путём смешивания синтезируемых звуков с предварительно записанными шумами голоса, а все остальные звуки просто записаны.

Каждый звук, кроме глухих согласных, представлен последовательностью формант. Помимо информации о формантах, каждая фонема имеет информацию о своей амплитуде, длительности звучания и задержке перед последующей фонемой. На основе этих параметров синтезируется звук гласной буквы при помощи алгоритмов, реализованных в синтезаторе. Информация о фонемах и формантах хранится в отдельных файлах, также впоследствии компилируемых в бинарный формат.

Вместе с синтезатором поставляется утилита eSpeak Edit. Это GUI-приложение, написанное при помощи библиотеки WXLib. Оно позволяет визуально редактировать готовые фонемы. Фонема представляется в виде графика кривой, где последовательно можно выбирать форманты и изменять их значения, такие как частота, высота и ширина. Благодаря этим возможностям, на основе уже готовых фонем можно получать новые, более верные звуки для некоторого языка. В то же время, некоторые из фонем получить модификацией существующих не получается. Например, при разработке русскоязычной части eSpeak был специально записан звук, так как для него не нашлось достойного аналога в других языках

# Код программы

#from translate import Translator

from deep\_translator import GoogleTranslator

import speech\_recognition as sr

import os

import sys

import webbrowser

import pyttsx3

flag = 1

def command():

# Создаем объект на основе библиотеки

# speech\_recognition и вызываем метод для определения данных

r = sr.Recognizer()

# Начинаем прослушивать микрофон и записываем данные в source

with sr.Microphone() as source:

# Просто вывод, чтобы мы знали когда говорить

print("Говорите:")

print("-----------------------------------------------")

# Устанавливаем паузу, чтобы прослушивание

# началось лишь по прошествию 1 секунды

r.pause\_threshold = 1

# используем adjust\_for\_ambient\_noise для удаления

# посторонних шумов из аудио дорожки

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

# Полученные данные записываем в переменную audio

# пока мы получили лишь mp3 звук

audio = r.listen(source)

try: # Обрабатываем все при помощи исключений

zadanie = r.recognize\_google(audio, language="ru-RU").lower()

# Просто отображаем текст что сказал пользователь

print("Русский:")

print(zadanie)

# Если не смогли распознать текст, то будет вызвана эта ошибка

except sr.UnknownValueError:

# Здесь просто проговариваем слова "Я вас не поняла"

# и вызываем снова функцию command() для

# получения текста от пользователя

zadanie = command()

# В конце функции возвращаем текст задания

# или же повторный вызов функции

return zadanie

print("---Перевод с русского на французский---")

print(" ")

while(flag == 1):

zadanie = command()

#translator= Translator(from\_lang="russian",to\_lang="french")

#translation = translator.translate(zadanie)

translator = GoogleTranslator(source='russian', target='french')

translation = translator.translate(zadanie)

print("-----------------------------------------------")

print("Французский:")

print(translation)

print("-----------------------------------------------")

engine = pyttsx3.init() # инициализация движка

# зададим свойства

engine.setProperty('rate', 150) # скорость речи

engine.setProperty('volume', 0.9) # громкость (0-1)

es\_voice\_id = "HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Speech\Voices\Tokens\TTS\_MS\_FR-FR\_HORTENSE\_11.0"

engine.setProperty('voice', es\_voice\_id)

engine.say(translation)

engine.runAndWait()

print("Продолжить? (y/n)")

k = input()

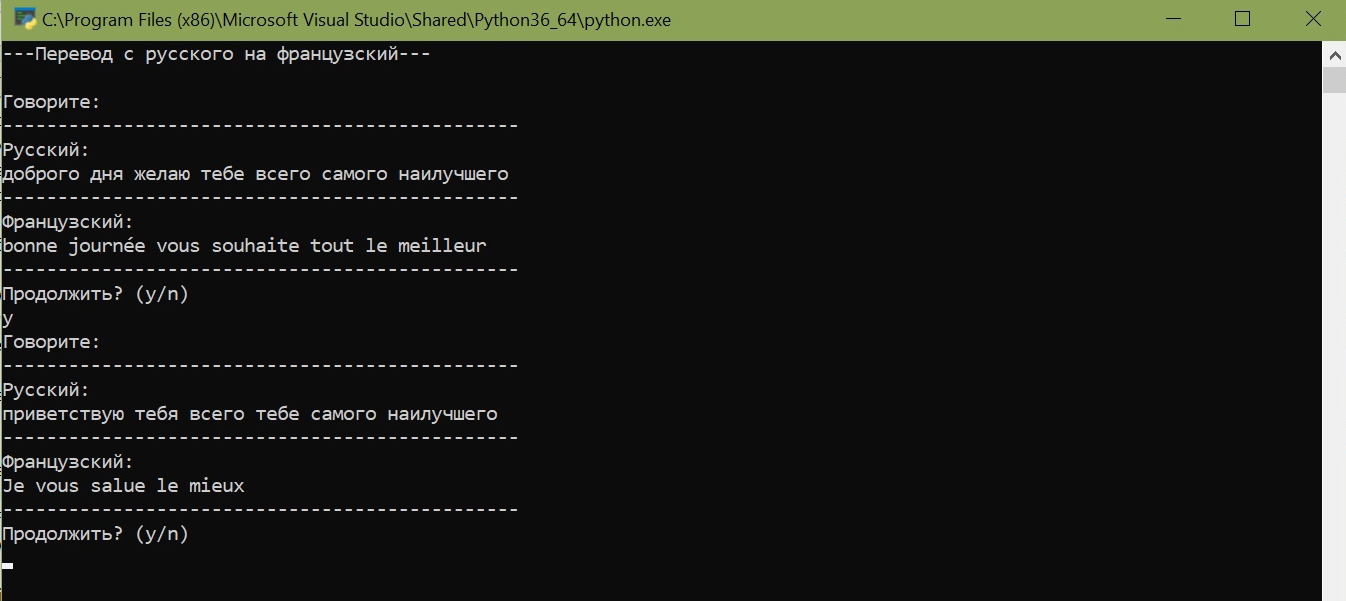
if (k == 'y'):

flag = 1

else:

flag = 0

Скрин выполнения программы



# Выводы

В процессе выполнения домашнего задания мной были усвоены навыки перевода русской речи во французскую и так далее.