МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
 ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

«НОВОСИБИРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

(НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ, НГУ)

Гуманитарный факультет

*Кафедра фундаментальной и прикладной лингвистики*

**Отчет по учебной практике**

ТЕМА ПРОЕКТА

SoundBot

**ФАМИЛИЯ ИМЯ ОТЧЕСТВО**

 Захарова Полина Ивановна

Русина Влада Александровна

Новосибирск

2017

Оглавление

[Введение 3](#_Toc414371980)

[1 Описание задачи 4](#_Toc414371981)

[2 Разработка приложения 5](#_Toc414371982)

[2.1 Средства разработки 5](#_Toc414371983)

[2.2 Анализ имеющихся аналогов 5](#_Toc414371984)

[2.3 Описание реализации 5](#_Toc414371985)

[3 Тестирование приложения 7](#_Toc414371986)

[Заключение 8](#_Toc414371987)

[Список литературы 9](#_Toc414371988)

[Приложение А. Код разработанного приложения 10](#_Toc414371989)

# Введение

Выбранная нами тема проекта является актуальной в настоящее время, т.к. существуют люди с различными нарушениями/ дефектами речи. Этим людям необходимо коммуницировать с окружающим социумом.

Данная программа подходит и для здоровых людей. К примеру, людям за рулем эта технология может быть полезна для получения информации с навигатора. Дабы не отвлекаться от вождения и получать информацию с дополнительного оборудования автомобиля.

Также эта технология может быть полезна для real-time перевода: для людей, разговаривающих исключительно на родном языке. Т.к. перевод из текста в текст существует, то можно двигаться дальше: добавлять функции распознавания речи в текст и наоборот.

# 1 Описание задачи

Задача состоит в переводе текстового потока в звуковой.

# 2 Разработка приложения

## 2.1 Средства разработки

Здесь должны быть перечислены языки программирования, IDE и стороннее ПО, использованное при разработке.

Язык программирование – Python.

IDE - Pycharm.

Стороннее ПО - [https://www.pygame.org/](https://www.pygame.org/news)   
 <https://pypi.python.org/pypi/gTTS>

## 2.2 Анализ имеющихся аналогов

Аналоги: <https://translate.google.ru/?hl=ru&tab=wT>   
 [https://alice.yandex.ru/](https://alice.yandex.ru/windows/?from=link_main_search-yb-win_alice3&banerid=0401003155:7623302151088906708&yclid=7623302151088906708)

Данные аналоги используют трансляцию из текста в речь и из речи в текст. При этом google использует ту же технологию, что и наш проект.

## 2.3 Описание реализации

Данная программа обеспечивает озвучивание любого введенного текста, для работы требуется соединение интернет, поскольку для озвучивания используется модуль gtts (google text to speech), который подключается к сервису озвучивания текста от google. Модуль pygame служит для воспроизведения принятого файла на компьютере конечного пользователя. Озвученные текста сохраняются в папку проекта с расширением mp3 при надобности можно их забрать и передать, куда надо.

Программа условно может быть разбита на 4 части:

1. Считывание текстового файла
2. Разбиение текста по предложениям
3. Отправка предложений в google- сервис для перевода в звуковой файл формата mp3
4. Запуск звукового файла на локальной машине

**БЛОК-СХЕМА**

Exit

Инициализация библиотек

Загрузка текстового файла и разбиение на предложения

Отправка предложения в google-сервис, загрузка звукового файла mp3, соответствующего этому предложению

Вывод ошибки.

Error

Error

Error

Да

Нет

Предложения закончились?

# 3 Тестирование приложения

Для тестирования может быть взят произвольный текстовый файл. В результате тестирования было проверено, что текст корректно разбивается на предложения, предложения отправляются в google-сервис, корректно загружаются mp3 файлы озвученных предложений и происходит воспроизведение на компьютере.

# Заключение

# Написана программа с использованием библиотек Pygame и gtts. Показано, что библиотеки обеспечивают достаточно адекватное качество конвертации текста в звуковой облик. Программа успешно работает и совместно с программой предыдущей курсовой работы (Spell checker) позволяет не только озвучивать текст, но и производить предварительную корректировку текста перед озвучиванием.

# Список литературы

https://discordapp.com/developers/docs/resources/channel <https://habrahabr.ru/company/Voximplant/blog/329122/>

<https://habrahabr.ru/post/304284/>

<https://gcup.ru/publ/engines/osnovy_pygame_vvedenie/2-1-0-257>

<https://habrahabr.ru/post/59748/>

<http://open-life.org/blog/python/1017.html>

# 

# Приложение А. Код разработанного приложения

# Чтение вслух

import os

import re

from tkinter import \*

from pygame import mixer

import datetime

import time

from gtts import gTTS

# Для того чтобы не возникало коллизий при удалении mp3 файлов

# заведем переменную mp3\_nameold в которой будем хранить имя предыдущего mp3 файла

mp3\_nameold = '111'

mp3\_name = "1.mp3"

tk = Tk()

f = open("text.txt", "w")

f.close()

# Инициализируем звуковое устройство

mixer.init()

text = StringVar()

text.set('')

tk.title('SpeechChat')

tk.geometry('400x300')

flag = False

log = Text(tk)

msg = Entry(tk, textvariable=text)

msg.pack(side='bottom', fill='x', expand='true')

log.pack(side='top', fill='both', expand='true')

def loopproc():

global flag

global mp3\_name

global mp3\_nameold

log.see(END)

try:

if flag:

# Открываем файл с текстом и по очереди читаем с него строки в ss

f = open("text.txt", "r")

ss = f.readline()

while ss:

# Делим прочитанные строки на отдельные предложения

split\_regex = re.compile(r'[.|!|?|…]')

sentences = filter(lambda t: t, [t.strip() for t in split\_regex.split(ss)])

# Перебираем массив с предложениями

for x in sentences:

if (x != ""):

print(x)

# Эта строка отправляет предложение которое нужно озвучить гуглу

tts = gTTS(text=x, lang='ru')

# Получаем от гугла озвученное предложение в виде mp3 файла

tts.save(mp3\_name)

# Проигрываем полученный mp3 файл

mixer.music.load(mp3\_name)

mixer.music.play()

while mixer.music.get\_busy():

time.sleep(0.1)

# Если предыдущий mp3 файл существует удаляем его

# чтобы не захламлять папку с приложением кучей mp3 файлов

if (os.path.exists(mp3\_nameold) and (mp3\_nameold != "1.mp3")):

os.remove(mp3\_nameold)

mp3\_nameold = mp3\_name

# Формируем имя mp3 файла куда будет сохраняться озвученный текст текущего предложения

# В качестве имени файла используем текущие дату и время

now\_time = datetime.datetime.now()

mp3\_name = now\_time.strftime("%d%m%Y%I%M%S") + ".mp3"

# Читаем следующую порцию текста из файла

ss = f.readline()

# Закрываем файл

f.close

# Устанвливаем текущим файлом 1.mp3 и закрываем звуковое устройство

# Это нужно чтобы мы могли удалить предыдущий mp3 файл без колизий

mixer.music.load('1.mp3')

mixer.stop

mixer.quit

flag = False

# Удаляем последний предыдущий mp3 файл

if (os.path.exists(mp3\_nameold)):

os.remove(mp3\_nameold)

except:

tk.after(1, loopproc)

flag = False

return

tk.after(1, loopproc)

flag = False

return

def sendproc(event):

global flag

if text.get() != '':

f = open("text.txt", "w")

send = text.get()

log.insert(END, send + '\n')

text.set('')

f.write(send + '\n')

flag = True

f.close()

msg.bind('<Return>', sendproc)

msg.focus\_set()

tk.after(1, loopproc)

tk.mainloop()