

Министерство образования Республики Беларусь  
Учреждение образования  
«Брестский государственный технический университет»  
Кафедра ИИТ

Лабораторная работа №7-8

По дисциплине: «Естественно-языковой интерфейс ИС» Темы: «Разработка  
систем анализа речи», «Разработка систем синтеза речи»

Выполнил:

Студент 4 курса

Группы ИИ-21

Пучинский А.А.

Проверил:

Булей Е. В.

**Цель:** освоить на практике основные принципы создания систем анализа и синтеза речи.

### Лабораторная работа №7:

#### Ход работы:

№ варианта	Система	Поддерживаемые языки	Предметная область
9	Анализа	Английский,	Научные статьи по computer science

- ✓ Задание списка операций, на которые система может реагировать.
- ✓ Организация автоматической реакции системы на речевой сигнал с уведомлением пользователя о происходящем. ✓ Настройки (выбор) ЕЯ.

#### Код программы:

```
import tkinter as tk
from tkinter import ttk
import speech_recognition as sr
import threading

class SpeechApp:
    def __init__(self, root):
        self.root = root
        self.root.title("Приложение распознавания речи")
        self.root.configure(bg="#2E2E2E") # Изменение фона

        # Заголовок
        self.label = tk.Label(root, text="Нажмите кнопку и говорите:", font=("Arial", 16),
bg="#2E2E2E", fg="white")
        self.label.pack(pady=20)

        # Поле для вывода результата
        self.result_text = tk.Text(root, height=10, width=50, bg="#3C3C3C", fg="white",
font=("Arial", 12))
        self.result_text.pack(pady=20)

        # Выпадающий список для выбора языка
        self.language_var = tk.StringVar(value='ru-RU') # По умолчанию русский
        self.language_label = tk.Label(root, text="Выберите язык:", font=("Arial", 12),
bg="#2E2E2E", fg="white")
        self.language_label.pack(pady=5)

        self.language_combobox = ttk.Combobox(root, textvariable=self.language_var,
font=("Arial", 12),
values=['ru-RU', 'en-US', 'de-DE', 'fr-FR'])
        self.language_combobox.pack(pady=5)

        # Кнопка для распознавания речи
        self.recognize_button = tk.Button(root, text="Распознать речь",
command=self.start_recognition_thread,
font=("Arial", 12), bg="#4CAF50", fg="white")
```

```

self.recognize_button.pack(pady=20)

def start_recognition_thread(self):
    # Запуск функции распознавания речи в отдельном потоке
    threading.Thread(target=self.recognize_speech).start()

def recognize_speech(self):
    recognizer = sr.Recognizer()
    recognizer.pause_threshold = 1
    with sr.Microphone() as source:
        self.label.config(text="Слушаю...")
        try:
            audio = recognizer.listen(source, timeout=3, phrase_time_limit=12)
        except sr.WaitTimeoutError as e:
            self.label.config(text=f"Не удалось получить результаты; {e}.")
            return

        try:
            language = self.language_var.get() # Получаем выбранный язык
            text = recognizer.recognize_google(audio, language=language)
            self.result_text.delete(1.0, tk.END) # Очистить предыдущий текст
            self.result_text.insert(tk.END, text) # Вставить распознанный текст
            self.label.config(text="Нажмите кнопку и говорите:")

            # Автоматическая реакция на определенные фразы
            self.respond_to_command(text)
        except sr.UnknownValueError:
            self.label.config(text="Не удалось распознать аудио.")
        except sr.RequestError as e:
            self.label.config(text=f"Не удалось получить результаты; {e}.")
        except sr.WaitTimeoutError as e:
            self.label.config(text=f"Не удалось получить результаты; {e}.")

def respond_to_command(self, text):
    # Реакция на команды
    if "привет" in text.lower():
        self.label.config(text="Привет! Как я могу помочь?")
    elif "стоп" in text.lower():
        self.label.config(text="А я что машина?")
    elif "как дела" in text.lower():
        self.label.config(text="У меня все хорошо, спасибо!")

if __name__ == '__main__':
    root = tk.Tk()
    app = SpeechApp(root)
    root.mainloop()

```

Результат:

Исходное главное меню:

Нажмите кнопку и говорите:

Выберите язык:

ru-RU



Распознать речь

Реакция на операцию:

Привет! Как я могу помочь?

Привет

Выберите язык:

ru-RU



Распознать речь

### Анализ текста:

Нажмите кнопку и говорите:

ехал Владик в лес по грибы а нашёл мухомора

Выберите язык:

ru-RU



Распознать речь

## Лабораторная работа №8:

### Код программы:

```
import tkinter as tk
from tkinter import ttk
import pyttsx3

class TTSApp:
    def __init__(self, root):
        self.root = root
        self.root.title("Система синтеза речи")

        self.root.configure(bg="#2E2E2E")

        self.engine = pyttsx3.init()
        voices = self.engine.getProperty('voices')
        for index, voice in enumerate(voices):
            print(f"Voice {index}: {voice.name}")

        # Метка для ввода текста
        self.label = tk.Label(root, text="Введите текст:", font=("Arial", 16), bg="#2E2E2E",
fg="#FFFFFF")
        self.label.pack(pady=10)

        # Поле для ввода текста
        self.text_entry = tk.Text(root, height=10, width=50, bg="#3C3C3C", fg="#FFFFFF",
insertbackground="white", font=("Arial", 12))
        self.text_entry.pack(pady=10)

        # Кнопка для воспроизведения текста
        self.speak_button = tk.Button(root, text="ПРОИЗНЕСТИ", command=self.speak_text,
font=("Arial", 12), bg="#4A90E2", fg="#FFFFFF", relief="flat")
        self.speak_button.pack(pady=10)

        # Метка для выбора голоса
        self.voice_label = tk.Label(root, text="Выберите голос:", font=("Arial", 12),
bg="#2E2E2E", fg="#FFFFFF")
        self.voice_label.pack(pady=5)

        # Выпадающий список для выбора голоса
        self.voices = self.engine.getProperty('voices')
        self.voice_var = tk.StringVar(value=self.voices[0].id) # Установка первого голоса
по умолчанию
        self.voice_combobox = ttk.Combobox(root, textvariable=self.voice_var,
values=[voice.name for voice in self.voices], font=("Arial", 12), state="readonly")
        self.voice_combobox.pack(pady=5)

        # Метка для выбора скорости
        self.speed_label = tk.Label(root, text="Установите скорость:", font=("Arial", 12),
bg="#2E2E2E", fg="#FFFFFF")
        self.speed_label.pack(pady=5)
```

```

        # Ползунок для выбора скорости
        self.speed_scale = tk.Scale(root, from_=50, to=300, orient=tk.HORIZONTAL,
label="Скорость (слов в минуту)", length=400, bg="#2E2E2E", fg="#FFFFFF", sliderlength=20,
font=("Arial", 10))
        self.speed_scale.set(150) # Установка значения по умолчанию
        self.speed_scale.pack(pady=10)

        # Метка для выбора громкости
        self.volume_label = tk.Label(root, text="Установите громкость:", font=("Arial", 12),
bg="#2E2E2E", fg="#FFFFFF")
        self.volume_label.pack(pady=5)

        # Ползунок для выбора громкости
        self.volume_scale = tk.Scale(root, from_=0.0, to=1.0, resolution=0.1,
orient=tk.HORIZONTAL, label="Громкость", length=400, bg="#2E2E2E", fg="#FFFFFF",
sliderlength=20, font=("Arial", 10))
        self.volume_scale.set(1.0) # Установка значения по умолчанию
        self.volume_scale.pack(pady=10)

        # Метод для воспроизведения текста
        def speak_text(self):
            text = self.text_entry.get("1.0", tk.END).strip() # Получаем текст
            self.engine.setProperty('voice', self.get_selected_voice_id()) # Устанавливаем
выбранный голос
            self.engine.setProperty('rate', self.speed_scale.get()) # Устанавливаем скорость
            self.engine.setProperty('volume', self.volume_scale.get()) # Устанавливаем
громкость
            self.engine.say(text) # Произносим текст
            self.engine.runAndWait() # Ждем завершения

        # Метод для получения ID выбранного голоса
        def get_selected_voice_id(self):
            selected_voice_name = self.voice_var.get() # Получаем выбранное имя голоса
            for voice in self.voices:
                if voice.name == selected_voice_name:
                    return voice.id # Возвращаем ID выбранного голоса
            return self.voices[0].id # Если ничего не найдено, возвращаем первый голос

if __name__ == '__main__':
    root = tk.Tk()
    app = TTSAApp(root)
    root.mainloop()

```

**Исходное главное меню:**

Введите текст:

ПРОИЗНЕСИТЬ

Выберите голос:

HKEY\_LOCAL\_MACHINE ▾

Установите скорость:

Скорость (слов в минуту)

150

Установите громкость:

Громкость

1.0

Изменение скорости чтения и синтез речи:



Введите текст:

Вася и Петя украли Петуха

ПРОИЗНЕСИТЬ

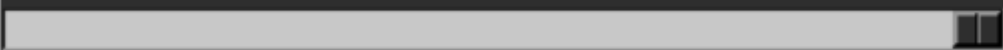
Выберите голос:

Microsoft Zira Desktop - E ▾

Установите скорость:

Скорость (слов в минуту)


300



Установите громкость:

Громкость

0.5



**Вывод:** в ходе выполнения лабораторной работы освоил на практике основные принципы создания систем анализа и синтеза речи.