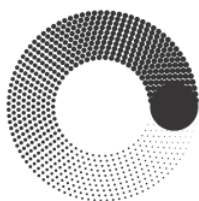


**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**



**МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**Факультет информационных технологий  
Кафедра Информатики и информационных технологий**

**направление подготовки 09.03.02 «Информационные системы и  
технологии»,  
профиль «Цифровая трансформация»**

**ЛАБОРАТОРНАЯ РАБОТА №7**

**Дисциплина:** Технологии прикладного программирования

**Выполнил:** студент группы 231-337

**Сильченко Александр Алексеевич**

**Дата, подпись 28/02/2024** \_\_\_\_\_

(Дата)

(Подпись)

**Проверила:** Полубояринова А.С. \_\_\_\_\_

(Оценка)

**Дата, подпись** \_\_\_\_\_

(Дата)

(Подпись)

**Замечания:**

---

---

**Москва**

**2024**

## Текст задачи

Разработать программу "синтезатор речи" или программу "распознавание речи" на выбор.

Функционал:

- 1.Загрузка файла с исходным материалом (текст или аудио)
- 2.распознавание/синтез речи с последующим сохранением результата
- 3.настройка работы программы через prompt

## Код вёрстки окон и код программной логики

```
<Window x:Class="JP7.MainWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:local="clr-namespace:JP7"
    mc:Ignorable="d"
    Title="Синтезатор речи" Height="400" Width="600">
    <Grid>
        <Button x:Name="Download" Content="Вопросизвести" Height="30" Width="120"
            VerticalAlignment="Center" HorizontalAlignment="Center" Click="Btn_Say_Click" Margin="0,0,250,0" />
        <Button x:Name="Btn_Say" Content="Загрузить текст" Height="30" Width="120"
            VerticalAlignment="Center" HorizontalAlignment="Center" Click="Btn_Say_File_Click" Margin="0,0,0,35" />
        <Button x:Name="Btn_Say_File" Content="Загрузить файл" Height="30" Width="120"
            VerticalAlignment="Center" HorizontalAlignment="Center" Click="downloading" Margin="0,35,0,0" />
        <Button x:Name="Btn_Save" Content="Сохранить" Height="30" Width="120"
            VerticalAlignment="Center" HorizontalAlignment="Center" Click="Btn_Save_Click" Margin="250,0,0,0" />
        <StackPanel Orientation="Horizontal">
            <TextBox x:Name="Box_Input" Text="По умолчанию" Grid.Column="0" Grid.Row="0" TextWrapping="Wrap"
                Width="300" Height="100" VerticalAlignment="Top"/>
            <ComboBox Width="100" Height="20" VerticalAlignment="Top" Text="Скорость"
                IsEditable="True" IsReadOnly="True" SelectionChanged="Speed_change" x:Name="speed">
                <ComboBoxItem Content="Default"/>
                <ComboBoxItem Content="ExtraSlow"/>
                <ComboBoxItem Content="Slow"/>
                <ComboBoxItem Content="Medium"/>
                <ComboBoxItem Content="Fast"/>
                <ComboBoxItem Content="ExtraFast"/>
            </ComboBox>
            <ComboBox Width="100" Height="20" VerticalAlignment="Top" Text="Громкость"
                IsEditable="True" IsReadOnly="True" SelectionChanged="Volume_change" x:Name="volume">
                <ComboBoxItem Content="Default"/>
                <ComboBoxItem Content="Silent"/>
                <ComboBoxItem Content="ExtraSoft"/>
                <ComboBoxItem Content="Soft"/>
                <ComboBoxItem Content="Medium"/>
                <ComboBoxItem Content="Loud"/>
                <ComboBoxItem Content="ExtraLoud"/>
            </ComboBox>
        </StackPanel>
    </Grid>
</Window>
```

```
using System.Windows;
using System.Windows.Media;
using System.Speech.Synthesis;
using Microsoft.Win32;
using System.Windows.Controls;
```

```

namespace JIP7
{
    public partial class MainWindow : Window
    {
        private MediaPlayer mediaPlayer = new MediaPlayer();
        PromptBuilder promptBuilder = new PromptBuilder();

        PromptRate promptRate = PromptRate.Medium;
        PromptVolume promptVolume = PromptVolume.Default;

        public MainWindow()
        {
            InitializeComponent();
        }

        private void downloading(object sender, RoutedEventArgs e)
        {
            OpenFileDialog openFileDialog = new OpenFileDialog();
            openFileDialog.Filter = "WAV files (*.wav)|*.wav|All files (*.*)|*.*";
            if (openFileDialog.ShowDialog() == true)
            {
                mediaPlayer.Open(new System.Uri(openFileDialog.FileName));
            }
            mediaPlayer.Play();
        }

        private void Btn_Say_Click(object sender, RoutedEventArgs e)
        {
            SpeechSynthesizer speechSynthesizer = new SpeechSynthesizer();
            PromptStyle promptStyle = new PromptStyle
            {
                Volume = promptVolume,
                Rate = promptRate
            };

            promptBuilder.StartStyle(promptStyle);
            promptBuilder.AppendText(Box_Input.Text);
            promptBuilder.EndStyle();

            speechSynthesizer.Speak(promptBuilder);

            promptBuilder.ClearContent();
            speechSynthesizer.Dispose();
        }

        private void Btn_Save_Click(object sender, RoutedEventArgs e)
        {
            SpeechSynthesizer speechSynthesizer = new SpeechSynthesizer();

            promptBuilder.AppendText(Box_Input.Text);

            SaveFileDialog saveFileDialog = new SaveFileDialog();
            saveFileDialog.Filter = "WAV files (*.wav)|*.wav|All files (*.*)|*.*";
            if (saveFileDialog.ShowDialog() == true)
            {
                speechSynthesizer.SetOutputToWaveFile(saveFileDialog.FileName);
            }
            speechSynthesizer.Speak(promptBuilder);

            speechSynthesizer.Dispose();
            promptBuilder.ClearContent();
        }

        private void Btn_Say_File_Click(object sender, RoutedEventArgs e)

```

```

{
    OpenFileDialog openFileDialog = new OpenFileDialog();
    openFileDialog.Filter = "txt files (*.txt)|*.txt|All files (*.*)|*.*";
    if (openFileDialog.ShowDialog() == true)
    {
        Box_Input.Text = System.IO.File.ReadAllText(openFileDialog.FileName);
    }
}

private void Speed_change(object sender, SelectionChangedEventArgs e)
{
    switch(speed.SelectedIndex)
    {
        case 0:
            promptRate = PromptRate.NotSet; break;
        case 1:
            promptRate = PromptRate.ExtraSlow; break;
        case 2:
            promptRate = PromptRate.Slow; break;
        case 3:
            promptRate = PromptRate.Medium; break;
        case 4:
            promptRate = PromptRate.Fast; break;
        case 5:
            promptRate = PromptRate.ExtraFast; break;
        default:
            break;
    }
}

private void Volume_change(object sender, SelectionChangedEventArgs e)
{
    switch(volume.SelectedIndex)
    {
        case 0:
            promptVolume = PromptVolume.Default; break;
        case 1:
            promptVolume = PromptVolume.ExtraSoft; break;
        case 2:
            promptVolume = PromptVolume.Soft; break;
        case 3:
            promptVolume = PromptVolume.Medium; break;
        case 4:
            promptVolume = PromptVolume.Loud; break;
        case 5:
            promptVolume = PromptVolume.ExtraLoud; break;
        default:
            break;
    }
}
}

```

