МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

## Факультет информационных технологий и робототехники

Кафедра программного обеспечения информационных систем и технологий

**Отчет по лабораторной работе № 10**

по дисциплине: ”Разработка простейшего приложения в визуальной среде WPF”

на тему: ***”Разработка приложения с сохранением параметров и установок в Ini-файлах”***

Вариант 14

Выполнил**:** студент группы 107012019 Кузьмич И.В.

Принял**:** Гурский Н.Н.

Минск 2020

***Цель лабораторной работы:*** Изучить возможности автоматического сохранения параметров и установок, принятых в программе.

## Постановка задачи: Разработать приложение, поддерживающее чтение и сохранение настроек в Ini – файлах.

***Задание:***

Реализовать чтение и сохранение параметров при запуске и завершении программы. В качестве задания использовать лабораторную работу №5.

***Код программы:***

Файл Commands.cs

﻿using System.Windows.Input;

namespace Laba10.modules

{

public static class CustomCommands

{

public static readonly RoutedUICommand Save = new RoutedUICommand

(

"Save",

"SaveFromComboBox",

typeof(CustomCommands),

new InputGestureCollection()

{

new KeyGesture(Key.Return)

}

);

public static readonly RoutedUICommand Exit = new RoutedUICommand

(

"Exit",

"Exit",

typeof(CustomCommands),

new InputGestureCollection()

{

new KeyGesture(Key.F4, ModifierKeys.Alt)

}

);

}

}

Файл ProjectObject.cs

﻿namespace Laba10.modules

{

public class ProjectObject

{

public string Title { get; set; }

public int Width { get; set; }

public int Height { get; set; }

public string Content { get; set; }

public ProjectObject(string title, int width, int height, string content)

{

Title = title;

Width = width;

Height = height;

Content = content;

}

public ProjectObject() : this("Title", 550, 500, string.Empty)

{

}

}

}

Файл JsonFileReader.cs

﻿using System.IO;

using System.Text.Json;

using System.Threading.Tasks;

namespace Laba10.modules

{

public class JsonFileReader

{

public readonly string \_path;

public JsonFileReader(string path)

{

\_path = path;

}

public async Task<ProjectObject> ReadFile()

{

ProjectObject projectObj;

using (FileStream fin = new FileStream(\_path, FileMode.OpenOrCreate))

{

try

{

projectObj = await JsonSerializer.DeserializeAsync<ProjectObject>(fin);

}

catch

{

projectObj = new ProjectObject();

}

}

return projectObj;

}

public async void WriteFile(ProjectObject projectObj)

{

using (FileStream fout = new FileStream(\_path, FileMode.Create))

{

await JsonSerializer.SerializeAsync<ProjectObject>(fout, projectObj);

}

}

}

}

Файл MainWindow.xaml.cs

﻿using Laba10.modules;

using System;

using System.Collections.ObjectModel;

using System.Windows;

using System.Windows.Input;

namespace Laba10

{

/// <summary>

/// Interaction logic for MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

private static readonly string pathSettings = "settings.json";

private static readonly string pathEntry = "file.json";

private static readonly JsonFileReader \_jsonFileReader = new JsonFileReader(pathSettings);

private async void ReadJsonFile()

{

var projectSettings = await \_jsonFileReader.ReadFile();

Width = projectSettings.Width;

Height = projectSettings.Height;

Title = projectSettings.Title;

}

private void Window\_Loaded(object sender, RoutedEventArgs e)

{

string[] commands = { "read", "write" };

listBox.Items.Add(commands[0]);

listBox.Items.Add(commands[1]);

ReadJsonFile();

}

private void SaveCommand\_CanExecute(object sender, CanExecuteRoutedEventArgs e)

{

e.CanExecute = true;

}

private void SaveCommand\_Executed(object sender, ExecutedRoutedEventArgs e)

{

\_jsonFileReader.WriteFile(new ProjectObject(Title, (int)Width, (int)Height, string.Empty));

}

private void ExitCommand\_CanExecute(object sender, CanExecuteRoutedEventArgs e)

{

e.CanExecute = true;

}

private void ExitCommand\_Executed(object sender, ExecutedRoutedEventArgs e)

{

Application.Current.Shutdown();

}

private async void button\_Execute(object sender, RoutedEventArgs e)

{

var commandType = listBox.SelectedItem.ToString();

var jsonFileReader = new JsonFileReader(pathEntry);

var text = sourceTextBox.Text;

var obj = new ProjectObject(string.Empty, 0, 0, text);

if (commandType == "write")

{

jsonFileReader.WriteFile(obj);

}

else

{

var projectObj = await jsonFileReader.ReadFile();

resultTextBox.Text = projectObj.Content;

}

}

}

}

**MainWindow.xaml**

<Window x:Class="Laba10.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:Laba10.modules"

mc:Ignorable="d"

Loaded="Window\_Loaded"

Style="{DynamicResource MainFontStyle}"

Width="450"

Height="350">

<Window.CommandBindings>

<CommandBinding Command="local:CustomCommands.Save" CanExecute="SaveCommand\_CanExecute" Executed="SaveCommand\_Executed"/>

<CommandBinding Command="local:CustomCommands.Exit" CanExecute="ExitCommand\_CanExecute" Executed="ExitCommand\_Executed"/>

</Window.CommandBindings>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="8\*"></RowDefinition>

<RowDefinition Height="5\*"></RowDefinition>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="100\*"></ColumnDefinition>

<ColumnDefinition Width="100\*"></ColumnDefinition>

<ColumnDefinition Width="100\*"></ColumnDefinition>

</Grid.ColumnDefinitions>

<TextBox x:Name="sourceTextBox" Text="Text" TextWrapping="Wrap" Grid.Row="0" Grid.Column="0"/>

<ListBox x:Name="listBox" Grid.Column="1" Margin="5,0,5,50" HorizontalContentAlignment="Center" SelectedIndex="0" />

<Button Style="{StaticResource StraightCorner}" Background="Ivory" x:Name="button" Content="Execute" Grid.Column="1" Height="30" VerticalAlignment="Bottom" Click="button\_Execute"/>

<TextBox x:Name="resultTextBox" Text="Text" TextWrapping="Wrap" Grid.Column="2" IsReadOnly="True"/>

<Grid Grid.ColumnSpan="3" Grid.Row="1">

<Grid.ColumnDefinitions>

<ColumnDefinition></ColumnDefinition>

<ColumnDefinition></ColumnDefinition>

</Grid.ColumnDefinitions>

<Button Command="local:CustomCommands.Save" Style="{StaticResource RoundCorner}" BorderThickness="1" BorderBrush="Black" Margin="5,20,5,20" Grid.Column="0" Grid.Row="1">

<TextBlock Text="Save" Style="{StaticResource RoundTextBlock}" HorizontalAlignment="Center" VerticalAlignment="Center"/>

</Button>

<Button Command="local:CustomCommands.Exit" Style="{StaticResource RoundCorner}" BorderThickness="1" BorderBrush="Black" Margin="5,20,5,20" Grid.Column="2" Grid.Row="1">

<TextBlock Text="Exit" Style="{StaticResource RoundTextBlock}" HorizontalAlignment="Center" VerticalAlignment="Center"/>

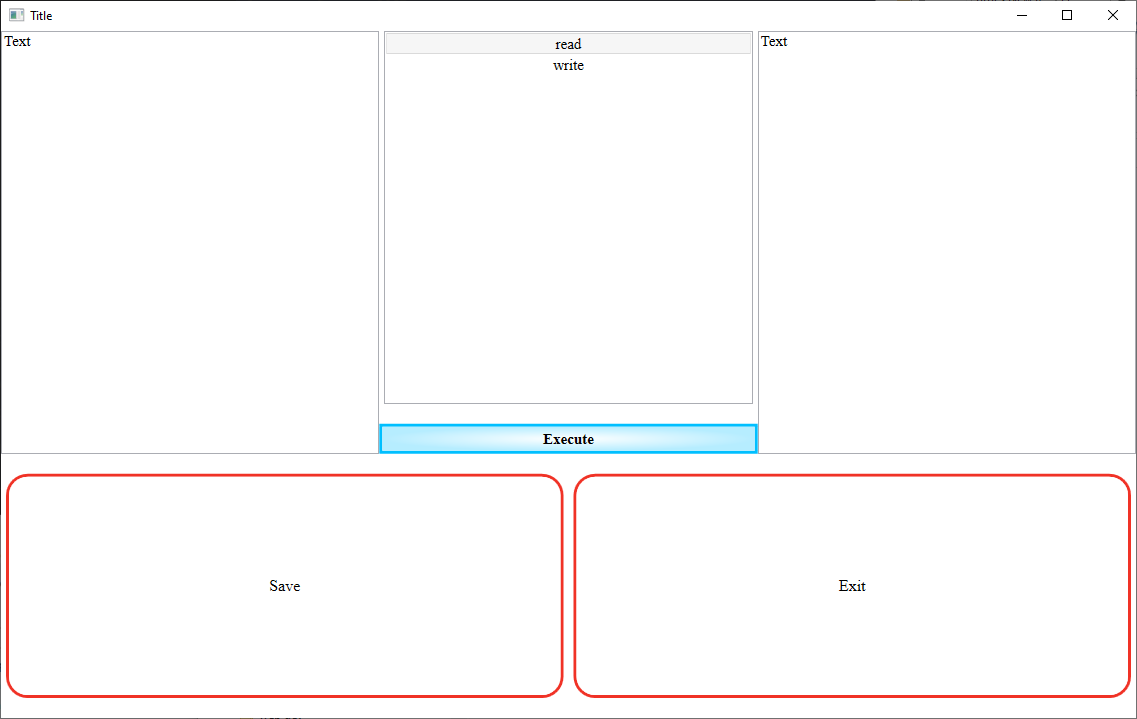
</Button>

</Grid>

</Grid>

</Window>

***Скриншоты результатов:***



**Рисунок 1. Результат работы программы**