**27. Технология Windows Presentation Foundation**

Задание 1. Измените интерфейс проекта WPF – XmlTaskWPF следующим образом:

1. Добавьте возможность скрывать функциональность под ЭУ
2. Предложите выбор вывода списка в качестве Компонеты с использованием RadioButtons:
3. Добавьте вывод списка элементов в TreeView, ListBox.

Решение:

<Window x:Class="XmlTaskWPF.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

Title="Записная книжка" Height="400" Width="600">

<Grid>

<Expander Header="Добавить контакт" VerticalAlignment="Top" HorizontalAlignment="Left" Width="580" Margin="10">

<StackPanel Margin="10">

<Label Content="Фамилия" />

<TextBox x:Name="LastNameTextBox" Width="200"/>

<Label Content="Дата рождения" />

<DatePicker x:Name="BirthDatePicker" Width="200"/>

<Label Content="Номер телефона" />

<TextBox x:Name="PhoneNumberTextBox" Width="200"/>

<Button Content="Сохранить" Width="75" Margin="0,10,0,0" Click="SaveButton\_Click"/>

</StackPanel>

</Expander>

<StackPanel VerticalAlignment="Top" HorizontalAlignment="Left" Margin="10,160,0,0">

<Label Content="Выберите способ вывода списка:" />

<RadioButton x:Name="TextBoxRadioButton" Content="Вывод элементов в TextBox" Checked="RadioButton\_Checked" />

<RadioButton x:Name="TreeViewRadioButton" Content="Вывод элементов в TreeView" Checked="RadioButton\_Checked" />

<RadioButton x:Name="ListBoxRadioButton" Content="Вывод элементов в ListBox" Checked="RadioButton\_Checked" />

</StackPanel>

<StackPanel x:Name="OutputStackPanel" VerticalAlignment="Top" HorizontalAlignment="Left" Margin="10,230,0,0" Width="580">

<!-- Здесь будут динамически добавляться элементы для вывода -->

</StackPanel>

</Grid>

</Window>

using System;

using System.Collections.Generic;

using System.Windows;

using System.Windows.Controls;

using System.Xml.Linq;

namespace XmlTaskWPF

{

public partial class MainWindow : Window

{

private List<Contact> contacts = new List<Contact>();

public MainWindow()

{

InitializeComponent();

}

private void SaveButton\_Click(object sender, RoutedEventArgs e)

{

string lastName = LastNameTextBox.Text;

DateTime? birthDate = BirthDatePicker.SelectedDate;

string phoneNumber = PhoneNumberTextBox.Text;

if (string.IsNullOrWhiteSpace(lastName) || !birthDate.HasValue || string.IsNullOrWhiteSpace(phoneNumber))

{

MessageBox.Show("Пожалуйста, заполните все поля.");

return;

}

Contact contact = new Contact

{

LastName = lastName,

BirthDate = birthDate.Value,

PhoneNumber = phoneNumber

};

contacts.Add(contact);

SaveContactToXml(contact);

UpdateOutput();

}

private void SaveContactToXml(Contact contact)

{

string documentsPath = Environment.GetFolderPath(Environment.SpecialFolder.MyDocuments);

string filePath = System.IO.Path.Combine(documentsPath, "Notebook.xml");

XElement contactElement = new XElement("Contact",

new XElement("LastName", contact.LastName),

new XElement("BirthDate", contact.BirthDate.ToString("yyyy-MM-dd")),

new XElement("PhoneNumber", contact.PhoneNumber)

);

XDocument xDocument;

if (System.IO.File.Exists(filePath))

{

xDocument = XDocument.Load(filePath);

xDocument.Root.Add(contactElement);

}

else

{

xDocument = new XDocument(new XElement("Notebook", contactElement));

}

xDocument.Save(filePath);

MessageBox.Show("Контакт успешно сохранен.");

}

private void RadioButton\_Checked(object sender, RoutedEventArgs e)

{

UpdateOutput();

}

private void UpdateOutput()

{

OutputStackPanel.Children.Clear();

if (TextBoxRadioButton.IsChecked == true)

{

TextBox textBox = new TextBox

{

Text = string.Join(Environment.NewLine, contacts),

IsReadOnly = true,

VerticalScrollBarVisibility = ScrollBarVisibility.Auto,

Height = 150,

Width = 560

};

OutputStackPanel.Children.Add(textBox);

}

else if (TreeViewRadioButton.IsChecked == true)

{

TreeView treeView = new TreeView

{

Height = 150,

Width = 560

};

foreach (var contact in contacts)

{

TreeViewItem item = new TreeViewItem

{

Header = contact.LastName

};

item.Items.Add($"Дата рождения: {contact.BirthDate.ToString("yyyy-MM-dd")}");

item.Items.Add($"Номер телефона: {contact.PhoneNumber}");

treeView.Items.Add(item);

}

OutputStackPanel.Children.Add(treeView);

}

else if (ListBoxRadioButton.IsChecked == true)

{

ListBox listBox = new ListBox

{

Height = 150,

Width = 560

};

foreach (var contact in contacts)

{

listBox.Items.Add($"{contact.LastName}, {contact.BirthDate.ToString("yyyy-MM-dd")}, {contact.PhoneNumber}");

}

OutputStackPanel.Children.Add(listBox);

}

}

}

public class Contact

{

public string LastName { get; set; }

public DateTime BirthDate { get; set; }

public string PhoneNumber { get; set; }

public override string ToString()

{

return $"{LastName}, {BirthDate.ToString("yyyy-MM-dd")}, {PhoneNumber}";

}

}

}

Анализ результатов:

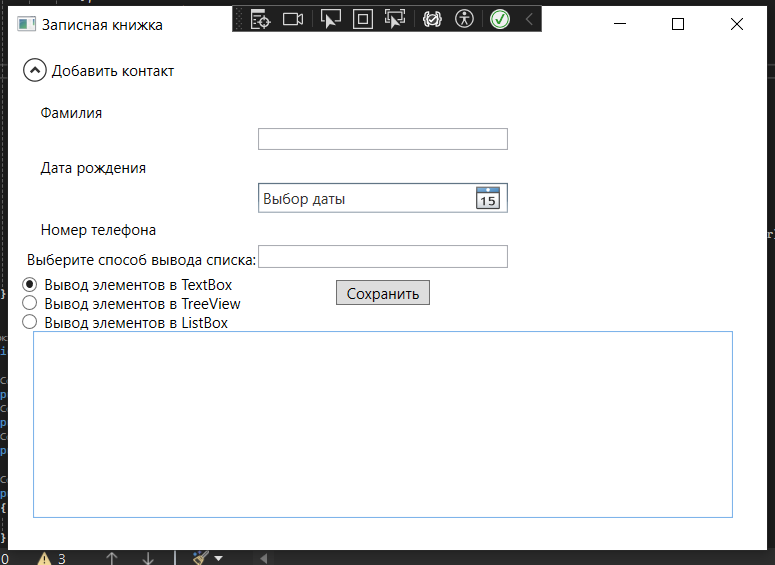


Рисунок 27.1 – Результат задания