**26 ЯЗЫК РАЗМЕТКИ XAML**

Задание №1. Cоздать XML файл Библиотека, состоящий из книг (название книги, автор, год издания), состоящий минимум из 7 элементов. Дополнительно реализовать Форму добавления записи в XML Файл. Добавить прокрутку элемента TextBlock для вывода информации. Проверку доступности кнопок если XML файл не выбран.

Листинг программы:

public partial class MainWindow : Window

{

string path = string.Empty;

static XmlDocument xDoc = new XmlDocument();

public MainWindow()

{

InitializeComponent();

}

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

{

path = textBoxPath.Text;

PrintXmlContent(path);

}

private void PrintXmlContent(string localPath)

{

textBoxContent.Text = string.Empty;

try

{

xDoc.Load(localPath);

XmlElement xRoot = xDoc.DocumentElement;

foreach (XmlNode xnode in xRoot)

{

if (xnode.Attributes.Count > 0)

{

XmlNode attr = xnode.Attributes.GetNamedItem("name");

if (attr != null)

textBoxContent.Text += attr.Value + " | "; }

foreach (XmlNode childnode in xnode.ChildNodes)

{

if (childnode.Name == "author")

{

textBoxContent.Text += "Автор: " + childnode.InnerText + " | ";

}

if (childnode.Name == "publishingYear")

{

textBoxContent.Text += "Год: " + childnode.InnerText + '\n';

}

}

}

}

catch (Exception)

{

path = String.Empty;

MessageBox.Show("Неверный путь к файлу!", "Error!", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

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

{

PritnSomeXmlElement(path);

}

private void PritnSomeXmlElement(string path)

{

if (path != string.Empty)

{

if (path == textBoxPath.Text)

{

int count = 0;

textBoxResult.Text = string.Empty;

xDoc.Load(path);

XmlElement xRoot = xDoc.DocumentElement;

foreach (XmlNode xnode in xRoot)

{

if (xnode.Attributes.Count > 0)

{

XmlNode attr = xnode.Attributes.GetNamedItem("name");

if (attr != null && attr.Value == textBoxSearch.Text)

{

count++;

textBoxResult.Text += attr.Value + " | ";

foreach (XmlNode childnode in xnode.ChildNodes)

{

if (childnode.Name == "author")

{

textBoxResult.Text += "Автор: " + childnode.InnerText + " | ";

}

if (childnode.Name == "publishingYear")

{

textBoxResult.Text += "Год: " + childnode.InnerText + '\n';

}

}

}

}

}

if (count == 0)

{

textBoxResult.Text = "Совпадений не найдено";

}

}

else MessageBox.Show("Путь изменён, сначала откройте файл!", "Error!", MessageBoxButton.OK, MessageBoxImage.Error);

}

else MessageBox.Show("Откройте файл!", "Error!", MessageBoxButton.OK, MessageBoxImage.Error);

}

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

{

if (path != string.Empty)

{

if (path == textBoxPath.Text)

{

xDoc.Load(path);

XmlElement xRoot = xDoc.DocumentElement;

string deleteName = textBoxDelete.Text;

try

{

DeleteNodeByAttName(deleteName, xDoc);

xDoc.Save(path);

textBoxResult.Text = $"Книга {deleteName} успешно удаленa";

PrintXmlContent(path);

}

catch (Exception)

{

if (deleteName == String.Empty)

textBoxResult.Text = $"Книга не найдена";

else textBoxResult.Text = $"Книга {deleteName} не найдена";

}

}

else MessageBox.Show("Путь изменён, сначала откройте файл!", "Error!", MessageBoxButton.OK, MessageBoxImage.Error);

}

else MessageBox.Show("Откройте файл!", "Error!", MessageBoxButton.OK, MessageBoxImage.Error);

}

private void DeleteNodeByAttName(string name, XmlDocument xmlDoc)

{

XmlNode node = xmlDoc.SelectSingleNode(String.Format("/library/book[@name='{0}']", name));

if (node != null)

{

xmlDoc.SelectSingleNode("library").RemoveChild(node);

}

else throw new Exception();

}

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

{

Application.Current.Shutdown();

}

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

{

if (path == textBoxPath.Text)

{

if (path != string.Empty)

{

xmlAddWindows xmlAddWindows = new xmlAddWindows();

xmlAddWindows.path = textBoxPath.Text;

xmlAddWindows.ShowDialog();

}

else MessageBox.Show("Введите путь к файлу!", "Error!", MessageBoxButton.OK, MessageBoxImage.Error);

}

else if (path == "") MessageBox.Show("Откройте файл!", "Error!", MessageBoxButton.OK, MessageBoxImage.Error);

else MessageBox.Show("Путь изменён, сначала откройте файл!", "Error!", MessageBoxButton.OK, MessageBoxImage.Error);

}

private void textBoxContent\_KeyDown(object sender, System.Windows.Input.KeyEventArgs e)

{

e = null; } }

Таблица 26.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| file path, string | strings |

Источник: собственная разработка

Анализ результатов представлен на рисунке 26.1.

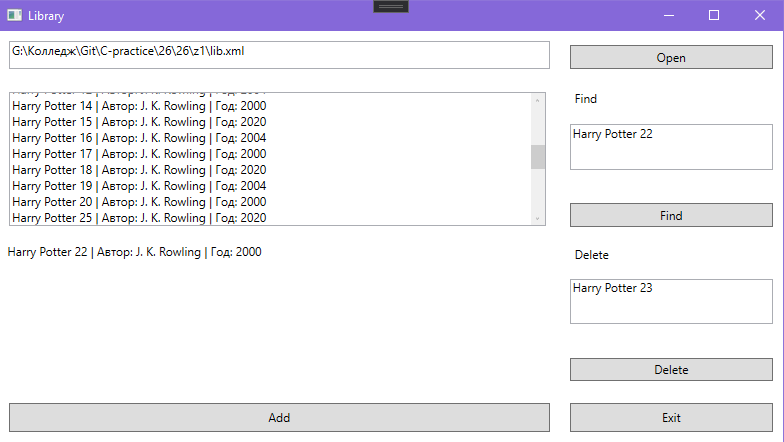


Рисунок 26.1 – Результат работы программы

Источник: собственная разработка