**26 ЯЗЫК РАЗМЕТКИ XML**

Задание 1. Создать XML файл Автостоянка, состоящий из автомобилей (марка, год выпуска, срок аренды стоянки(дни)).

Листинг программы:

using System;

using System.Windows;

using System.Xml;

namespace z1

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

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;

// обход всех узлов в корневом элементе

foreach (XmlNode xnode in xRoot)

{

// получаем атрибут name

if (xnode.Attributes.Count > 0)

{

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

if (attr != null)

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

}

// обходим все дочерние узлы элемента user

foreach (XmlNode childnode in xnode.ChildNodes)

{

// если узел - company

if (childnode.Name == "author")

{

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

}

// если узел age

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)

{

// получаем атрибут name

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)

{

// если узел - company

if (childnode.Name == "author")

{

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

}

// если узел age

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);

}

}

}

/\*

using System.Xml;

using System.Xml.Linq;

XmlDocument xDoc = new XmlDocument();

string xmlPath = @"../../../lib.xml";

xDoc.Load(xmlPath);

// получим корневой элемент

XmlElement xRoot = xDoc.DocumentElement;

Console.WriteLine("Why you wont?");

var swithcer = Console.ReadLine();

while (true)

{

switch (swithcer)

{

case "1":

addXMLelement(xDoc, xRoot);

xDoc.Save(xmlPath);

break;

case "2":

Console.WriteLine("Enter name: ");

var name = Console.ReadLine();

deleteNodeByAttName(name, xDoc);

xDoc.Save(xmlPath);

break;

case "3":

printAllNodes(xRoot);

break;

case "4":

editNodeByAttName(xDoc);

xDoc.Save(xmlPath);

break;

default: return;

}

Console.WriteLine("Why you wont?");

swithcer = Console.ReadLine();

}

static void editNodeByAttName(XmlDocument doc)

{

Console.WriteLine("Enter book name: ");

var bookName = Console.ReadLine();

foreach (XmlNode xn in doc.GetElementsByTagName("book"))

{

if (xn.Attributes["name"].Value == bookName)

{

xn.RemoveChild(xn.SelectSingleNode("author"));

XmlNode addNode = doc.CreateElement("author");

Console.WriteLine("Enter author: ");

var author = Console.ReadLine();

addNode.InnerText = author;

xn.AppendChild(addNode);

xn.RemoveChild(xn.SelectSingleNode("publishingYear"));

XmlNode addNodeYear = doc.CreateElement("publishingYear");

Console.WriteLine("Enter publishing Year: ");

var year = Console.ReadLine();

addNodeYear.InnerText = year;

xn.AppendChild(addNodeYear);

break;

}

}

}

static void deleteNodeByAttName(string name, XmlDocument xmlDoc)

{

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

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

}

static void printAllNodes(XmlElement xRoot)

{

// обход всех узлов в корневом элементе

foreach (XmlNode xnode in xRoot)

{

// получаем атрибут name

if (xnode.Attributes.Count > 0)

{

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

if (attr != null)

Console.WriteLine(attr.Value);

}

// обходим все дочерние узлы элемента user

foreach (XmlNode childnode in xnode.ChildNodes)

{

// если узел - company

if (childnode.Name == "author")

{

Console.WriteLine("Компания: {0}", childnode.InnerText);

}

// если узел age

if (childnode.Name == "publishingYear")

{

Console.WriteLine("Возраст: {0}", childnode.InnerText);

}

}

Console.WriteLine();

}

}

static void addXMLelement(XmlDocument xDoc, XmlElement xRoot)

{

XmlElement userElem = xDoc.CreateElement($"book");

// создаем атрибут name

XmlAttribute nameAttr = xDoc.CreateAttribute($"name");

XmlElement companyElem = xDoc.CreateElement($"author");

XmlElement ageElem = xDoc.CreateElement("publishingYear");

Console.WriteLine("Enter book name, author, year: ");

var nameTextName = Console.ReadLine();

var companyTextName = Console.ReadLine();

var ageTextName = Console.ReadLine();

// создаем текстовые значения для элементов и атрибута

XmlText nameText = xDoc.CreateTextNode($"{nameTextName}");

XmlText authorText = xDoc.CreateTextNode($"{companyTextName}");

XmlText ageText = xDoc.CreateTextNode($"{ageTextName}");

//добавляем узлы

nameAttr.AppendChild(nameText);

companyElem.AppendChild(authorText);

ageElem.AppendChild(ageText);

userElem.Attributes.Append(nameAttr);

userElem.AppendChild(companyElem);

userElem.AppendChild(ageElem);

xRoot.AppendChild(userElem);

xDoc.Save("lib.xml");

}

Таблица 26.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Представлены на рисунке 26.1 | |

Анализ результатов:

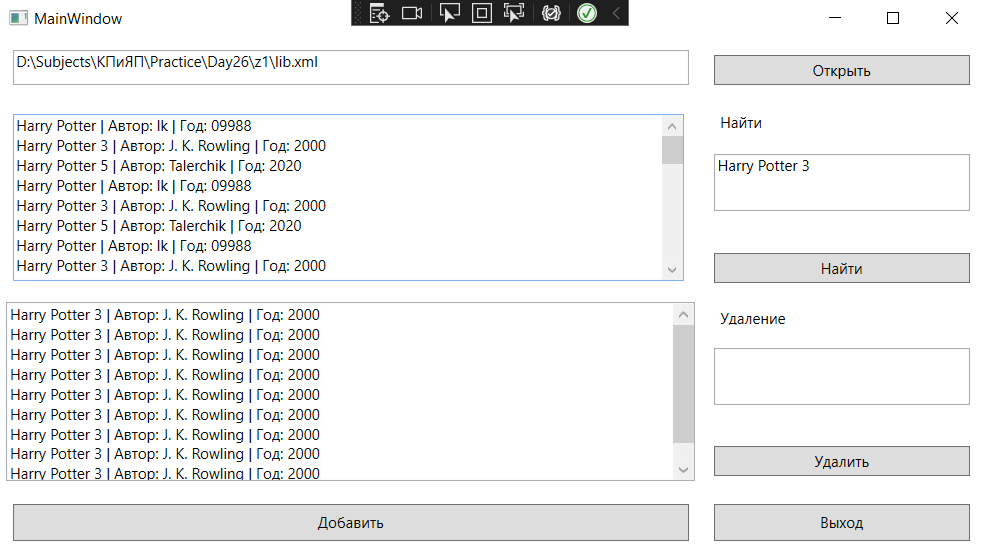


Рисунок 26.1 – результат работы программы

Источник: собственная разработка