**ПРАКТИЧЕСКОЕ ЗАДАНИЕ 27**

**КОМПОНОВКА И ЭЛЕМЕНТЫ УПРАВЛЕНИЯ WPF**

Задание 1. В работе 24 «Язык разметки XAML». Измените интерфейс проекта WPF – XmlTaskWPF

Листинг программы

using Microsoft.Extensions.Logging;

using Practice\_26.Task\_1.InterFaces;

using Practice\_26.Task\_1.models;

using System;

using System.Collections.Generic;

using System.Xml;

namespace Practice\_26.Task\_1.Share

{

public class XmlDocumentWorker:IXmlWorker

{

private readonly XmlDocument \_document;

private readonly ILogger \_logger;

private string \_xmlFilePath;

public XmlDocumentWorker(ILogger logger)

{

\_logger = logger;

\_document = new XmlDocument();

}

public void Add(Auto auto)

{

var xRoot = \_document.DocumentElement;

XmlElement parkingElem = \_document.CreateElement("Auto");

XmlAttribute markAttribute = \_document.CreateAttribute("mark");

XmlText markText = \_document.CreateTextNode(auto.Mark);

markAttribute.AppendChild(markText);

XmlElement yearElem = \_document.CreateElement("year");

XmlText yearInnerText = \_document.CreateTextNode(auto.Year.ToString());

yearElem.AppendChild(yearElem);

parkingElem.AppendChild(yearElem);

XmlElement dateElem = \_document.CreateElement("RentDate");

XmlText dateInnerText = \_document.CreateTextNode(auto.RentDays.ToString());

dateElem.AppendChild(dateInnerText);

parkingElem.AppendChild(dateElem);

xRoot.AppendChild(parkingElem);

\_document.Save(\_xmlFilePath);

}

public void Delete(string mark)

{

var xRoot = \_document.DocumentElement;

foreach (XmlNode xNode in xRoot)

{

if (xNode.Attributes.Count > 0)

{

var attributeMark = xNode.Attributes.GetNamedItem("mark");

try

{

var atrributeMarkText = attributeMark?.InnerText;

if (atrributeMarkText.Equals(mark))

{

xRoot.RemoveChild(xNode);

}

}

catch (Exception ex) when (ex is XmlException || ex is NullReferenceException)

{

\_logger.LogWarning(ex.Message, nameof(attributeMark));

}

}

}

}

public Auto FindBy(string mark)

{

Auto parking = null;

var xRoot = \_document.DocumentElement;

foreach (XmlNode xmlNode in xRoot)

{

parking = GetParking(xmlNode);

if (parking.Mark.Equals(mark))

{

return parking;

}

}

return parking;

}

public List<Auto> GetAll()

{

List<Auto> parkingg = new List<Auto>();

var xRoot = \_document.DocumentElement;

foreach (XmlNode node in xRoot)

{

var parking = GetParking(node);

parkingg.Add(parking);

}

return parkingg;

}

public void Load(string xmlDocPath)

{

\_xmlFilePath = xmlDocPath;

\_document.Load(xmlDocPath);

}

private Auto GetParking(XmlNode node)

{

var parking = new Auto();

if (node.Attributes.Count > 0)

{

var attributeMark = node.Attributes.GetNamedItem("mark");

parking.Mark = attributeMark?.Value;

}

foreach (XmlNode childNode in node.ChildNodes)

{

try

{

if (childNode.Name.Equals("year"))

{

parking.Year = (childNode.InnerText);

}

if (childNode.Name.Equals("RentDate"))

{

parking.RentDays = int.Parse(childNode.InnerText);

}

}

catch (Exception ex) when (ex is FormatException || ex is NullReferenceException)

{

\_logger.LogError(ex.Message, ex.StackTrace, nameof(childNode.InnerText));

}

}

return parking;

}

}

Анализ результатов:

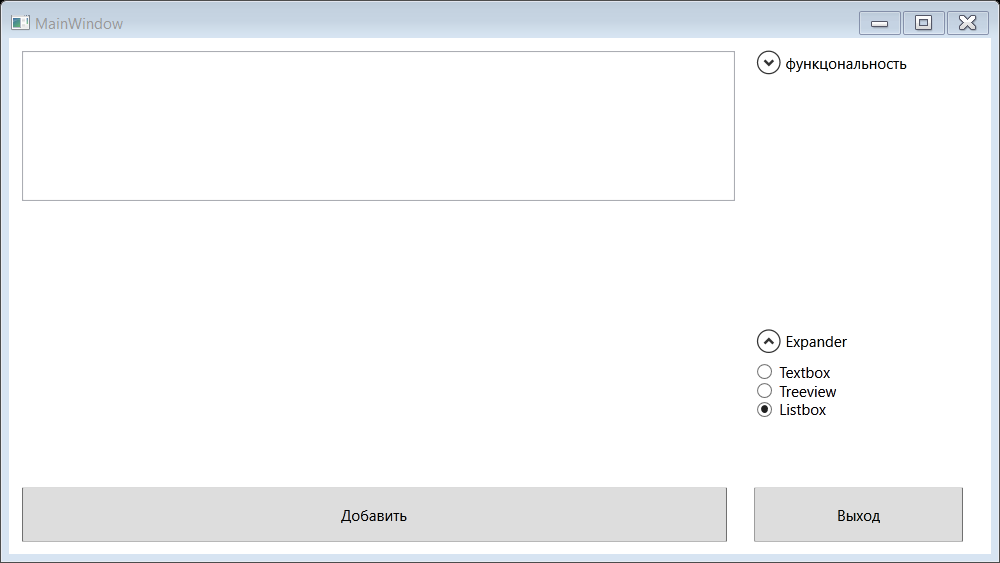


Рисунок 1.1 – Результат работы программы