**27** **Компоновка и элементы управления в WPF**

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

Добавьте возможность скрывать функциональность под ЭУ Expander;

Предложите выбор вывода списка в качестве Компонеты с использованием RadioButtons;

Добавьте вывод списка элементов в TreeView, ListBox.

Листинг программы:

private void Button\_Click\_3(object sender, RoutedEventArgs e)

{

// Получить выбранный элемент

string selectedItem = null;

if (ListBoxCh.IsChecked == true)

{

selectedItem = XmlDocView.SelectedItem as string;

}

else if (TreeViewCh.IsChecked == true)

{

TreeViewItem selectedTreeViewItem = XmlDocViewTV.SelectedItem as TreeViewItem;

if (selectedTreeViewItem != null)

{

selectedItem = selectedTreeViewItem.Header.ToString();

}

}

// Проверить, что элемент выбран

if (selectedItem != null)

{

try

{

// Создать экземпляр XmlDocument

XmlDocument xmlDoc = new XmlDocument();

// Загрузить XML-документ из файла

xmlDoc.Load(XmlFilePath);

// Найти элемент для удаления

XmlNodeList products = xmlDoc.GetElementsByTagName("Товар");

foreach (XmlNode productNode in products)

{

string productName = productNode.SelectSingleNode("Название").InnerText;

string expirationDate = productNode.SelectSingleNode("СрокГодности").InnerText;

string price = productNode.SelectSingleNode("Стоимость").InnerText;

string productInfo = $"Название товара: {productName}, Срок годности: {expirationDate}, Стоимость: {price}";

if (productInfo == selectedItem)

{

// Удалить найденный элемент

productNode.ParentNode.RemoveChild(productNode);

break;

}

}

// Сохранить изменения обратно в файл

xmlDoc.Save(XmlFilePath);

// Обновить содержимое соответствующего элемента управления

if (ListBoxCh.IsChecked == true)

{

UpdateXmlDocView(XmlDocView);

}

else if (TextBlockCh.IsChecked == true)

{

UpdateXmlDocViewTB(XmlDocViewTB);

}

else if (TreeViewCh.IsChecked == true)

{

UpdateXmlDocViewTV(XmlDocViewTV);

}

}

catch (Exception ex)

{

// Обработка ошибок при удалении элемента

MessageBox.Show("Ошибка удаления элемента: " + ex.Message);

}

}

}

private void AddItemWindow\_ItemAdded(object sender, EventArgs e)

{

if (ListBoxCh.IsChecked == true)

{

UpdateXmlDocView(XmlDocView);

}

else if (TextBlockCh.IsChecked == true)

{

UpdateXmlDocViewTB(XmlDocViewTB);

}

else if (TreeViewCh.IsChecked == true)

{

UpdateXmlDocViewTV(XmlDocViewTV);

}

}

private void UpdateXmlDocView(ListBox listBox)

{

// Обновление содержимого ListBox после добавления элемента

XmlDocument xmlDocument = new XmlDocument();

xmlDocument.Load(XmlFilePath);

// Получение содержимого XML

XmlNodeList products = xmlDocument.GetElementsByTagName("Товар");

// Формирование списка строк для ListBox

List<string> productList = new List<string>();

foreach (XmlNode productNode in products)

{

string productName = productNode.SelectSingleNode("Название").InnerText;

string expirationDate = productNode.SelectSingleNode("СрокГодности").InnerText;

string price = productNode.SelectSingleNode("Стоимость").InnerText;

string productInfo = $"Название товара: {productName}, Срок годности: {expirationDate}, Стоимость: {price}";

productList.Add(productInfo);

}

// Очистка содержимого ListBox

listBox.Items.Clear();

// Присвоение содержимого ListBox

foreach (string productInfo in productList)

{

listBox.Items.Add(productInfo);

}

}

Анализ результатов:

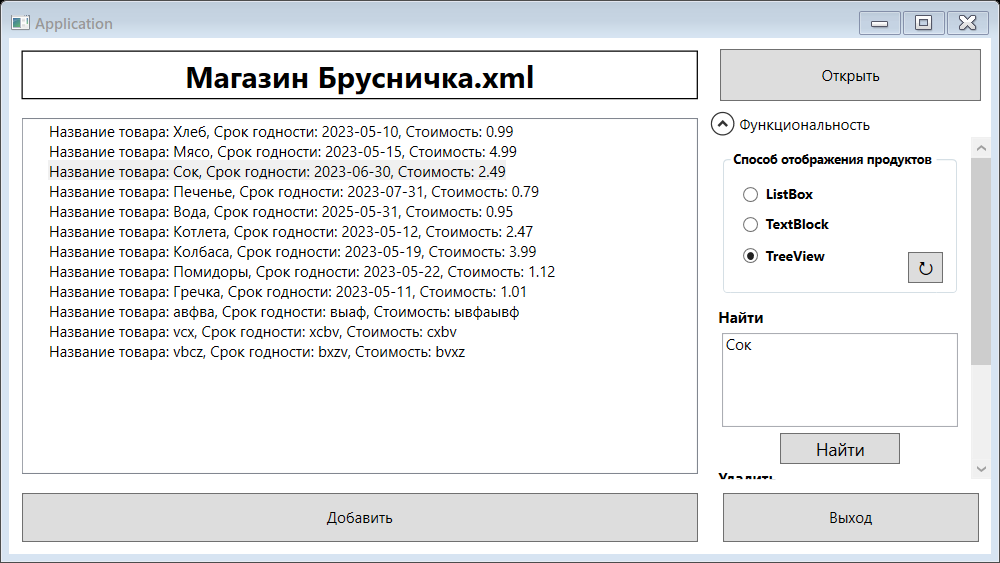


Рисунок 27.1 – Результат работы программы