

## 26 Язык разметки XAML

Задание 1. Создать XML файл состоящий минимум из 7 элементов. WPF по примеру. Дополнительно реализовать Форму добавления записи в XML Файл. Добавить прокрутку элемента TextBlock для вывода информации. Проверку доступности кнопок если XML файл не выбран. Создать XML файл Магазин состоящий из товара (название товара, срок годности, стоимость).

Листинг программы:

```
namespace Task1
{
    /// <summary>
    /// Логика взаимодействия для MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        private string XmlFilePath = "";

        public MainWindow()
        {
            InitializeComponent();
        }

        private void Button_Click_2(object sender, RoutedEventArgs e)
        {
            OpenFileDialog openFileDialog = new OpenFileDialog();
            openFileDialog.Filter = "XML Files (*.xml)|*.xml";

            if (openFileDialog.ShowDialog() == true)
            {
                string filePath = openFileDialog.FileName;
                XmlFilePath = filePath;

                string fileName = System.IO.Path.GetFileName(filePath);
                XmlDocName.Text = fileName;

                // Обработка XML-документа
                XmlDocument xmlDocument = new XmlDocument();
                xmlDocument.Load(filePath);
            }
        }
    }
}
```

					УП 2-40 01 01.37ТП.227.23.26		
Изм.	Лист	№ док	Подпись	Дата			
Разраб.		Сорокина Е.А.			Язык разметки XAML		
Проверил.		Новик А.И.					
Н.контр.							
Утвердил.							
						Лит	Лист
							114
						Листов	
						Гродненский ГКТТид	

```

        // 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);
        }

        // Присвоение содержимого ListBox
        XmlDocView.ItemsSource = productList;
    }
}

private void Button_Click(object sender, RoutedEventArgs e)
{
    // Создание экземпляра AddItemWindow
    AddItemWindow addItemWindow = new AddItemWindow();

    addItemWindow.XmlFilePath = XmlFilePath;
    addItemWindow.ItemAdded += AddItemWindow_ItemAdded;

    // Открытие окна AddItemWindow
    addItemWindow.ShowDialog();
}

private void AddItemWindow_ItemAdded(object sender, EventArgs e)
{
    UpdateXmlDocView();
}

private void Button_Click_3(object sender, RoutedEventArgs e)

```

```

{
    // Получить выбранный элемент ListBox
    var selectedItem = XmlDocView.SelectedItem as string;

    // 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 ListBox
            UpdateXmlDocView();
        }
    }
}

```

```

        catch (Exception ex)
        {
            // Обработка ошибок при удалении элемента
            MessageBox.Show("Ошибка удаления элемента: " + ex.Message);
        }
    }

private void UpdateXmlDocView()
{
    // Обновление содержимого ListBox после удаления 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);
    }

    // Присвоение содержимого ListBox
    XmlDocView.ItemsSource = productList;
}

private void Button_Click_4(object sender, RoutedEventArgs e)
{
    string searchQuery = SearchTextBox.Text.ToLower();

    // Пройти по каждому элементу ListBox и проверить название товара
    foreach (var listBoxItem in XmlDocView.Items)
    {

```

Изм.	Лист	№ док	Подпись	Дата

```

var listBoxItemContainer =
XmlDocView.ItemContainerGenerator.ContainerFromItem(listBoxItem) as
ListBoxItem;

string productInfo = listBoxItem.ToString().ToLower();

// Если название товара содержит искомую фразу
if (productInfo.Contains(searchQuery))
{
    // Пометить элемент как выбранный
    listBoxItemContainer.IsSelected = true;

    // Прокрутить к выбранному элементу
    XmlDocView.ScrollIntoView(listBoxItem);
}
}

private void Button_Click_1(object sender, RoutedEventArgs e)
{
    this.Close();
}
}

```

Анализ результатов:

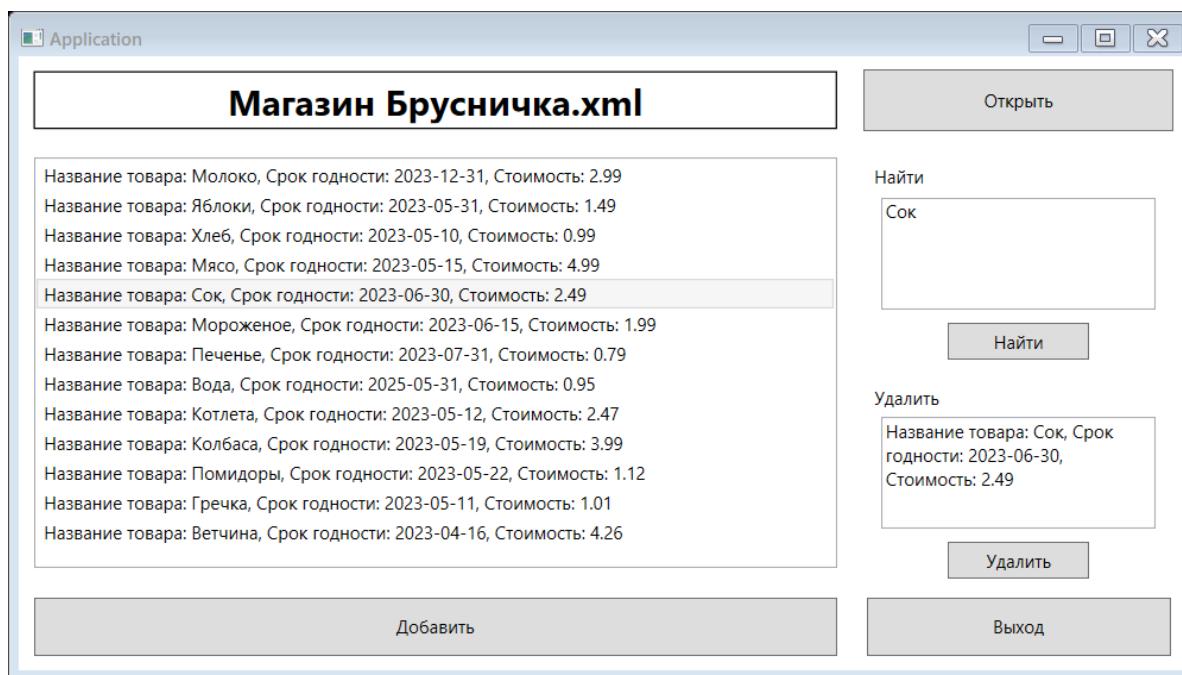


Рисунок 24.1 – Результат работы программы

Изм.	Лист	№ док	Подпись	Дата