**Работа с базами данных. Технологии доступа к данным**

**Задание 1**

**АРМ работника склада**

• **CRUD-операции:** создать ProductRepository с методами GetAllAsync(), AddAsync(Product), UpdateAsync(Product), DeleteAsync(Product)  
• **Привязка:** во WarehouseViewModel свойство public ObservableCollection<Product> Products { get; } и в XAML <DataGrid ItemsSource="{Binding Products}" />  
• **Сохранение:** в командах AddProductCommand и DeleteProductCommand после вызова репозитория вызывать await \_context.SaveChangesAsync()

**Код**

using System.Text;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using WpfApp1.Models;

using WpfApp1.ViewModels;

using Microsoft.EntityFrameworkCore;

namespace WpfApp1

{

/// <summary>

/// Interaction logic for MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

// Создаем базу данных, если она не существует

using (var context = new WarehouseDbContext())

{

context.Database.EnsureCreated();

}

DataContext = new WarehouseViewModel();

}

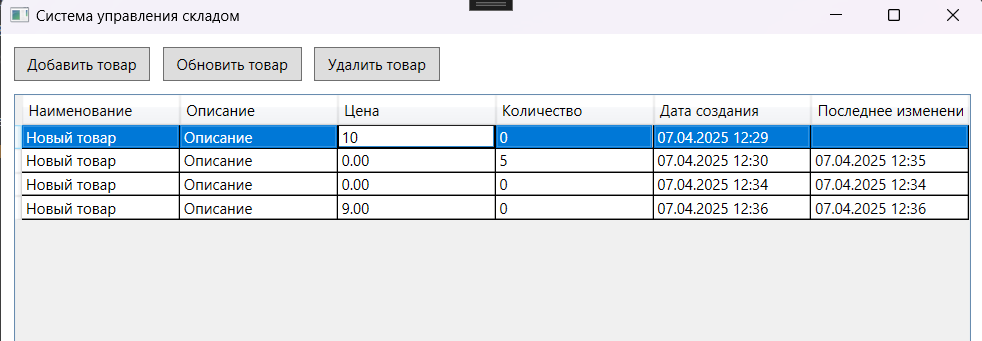
}

}

**Таблица 19 – Входные и выходные данные**

|  |  |
| --- | --- |
| **Входные данные** | **Выходные данные** |
| **Новый товар** | **Новый товар** |

**Анализ результата**

****