**Технологии создания графического пользовательского интерфейса**

**Задание 1**

**АРМ работника склада**

**Функционал:** управление списком товаров, отображение информации о складе.

**1. DataBinding (Привязка данных)**

* Привязать TextBox к свойствам Название товара, Количество и Цена через TwoWay привязку.
* Привязать ComboBox для выбора категории товара (OneWay привязка).

**Код**

using System;

using System.Collections.ObjectModel;

using System.ComponentModel;

using System.Windows;

using System.Windows.Input;

namespace Task1

{

public class MainViewModel : INotifyPropertyChanged

{

public ObservableCollection<Product> Products { get; }

public ICommand AddProductCommand { get; }

public ICommand EditProductCommand { get; }

public ICommand DeleteProductCommand { get; }

public ICommand ExitCommand { get; }

private Product \_selectedProduct;

public Product SelectedProduct

{

get => \_selectedProduct;

set

{

\_selectedProduct = value;

OnPropertyChanged(nameof(SelectedProduct));

((RelayCommand)EditProductCommand).RaiseCanExecuteChanged();

((RelayCommand)DeleteProductCommand).RaiseCanExecuteChanged();

}

}

public MainViewModel()

{

Products = new ObservableCollection<Product>();

AddProductCommand = new RelayCommand(AddProduct);

EditProductCommand = new RelayCommand(EditProduct, CanEditOrDelete);

DeleteProductCommand = new RelayCommand(DeleteProduct, CanEditOrDelete);

ExitCommand = new RelayCommand(ExitApplication);

}

private bool CanEditOrDelete(object parameter) => SelectedProduct != null;

private void AddProduct(object parameter)

{

var addProductWindow = new AddProductWindow();

if (addProductWindow.ShowDialog() == true)

{

Products.Add(new Product

{

Name = addProductWindow.ProductName,

Quantity = addProductWindow.ProductQuantity,

Price = addProductWindow.ProductPrice,

Category = addProductWindow.ProductCategory

});

}

}

private void EditProduct(object parameter)

{

if (SelectedProduct == null) return;

var editProductWindow = new EditProductWindow(SelectedProduct.Name, SelectedProduct.Quantity, SelectedProduct.Price, SelectedProduct.Category);

if (editProductWindow.ShowDialog() == true)

{

SelectedProduct.Name = editProductWindow.ProductName;

SelectedProduct.Quantity = editProductWindow.ProductQuantity;

SelectedProduct.Price = editProductWindow.ProductPrice;

SelectedProduct.Category = editProductWindow.ProductCategory;

}

}

private void DeleteProduct(object parameter)

{

if (SelectedProduct == null) return;

if (MessageBox.Show("Вы уверены, что хотите удалить товар?", "Подтверждение", MessageBoxButton.YesNo) == MessageBoxResult.Yes)

{

Products.Remove(SelectedProduct);

}

}

private void ExitApplication(object parameter) => Application.Current.Shutdown();

public event PropertyChangedEventHandler PropertyChanged;

protected void OnPropertyChanged(string propertyName)

{

PropertyChanged?.Invoke(this, new PropertyChangedEventArgs(propertyName));

}

}

}

**Таблица 14.1 – Входные и выходные данные**

|  |  |
| --- | --- |
| **Входные данные** | **Выходные данные** |
| **товар123**  **123** | **овар123**  **123**  **0.0$** |