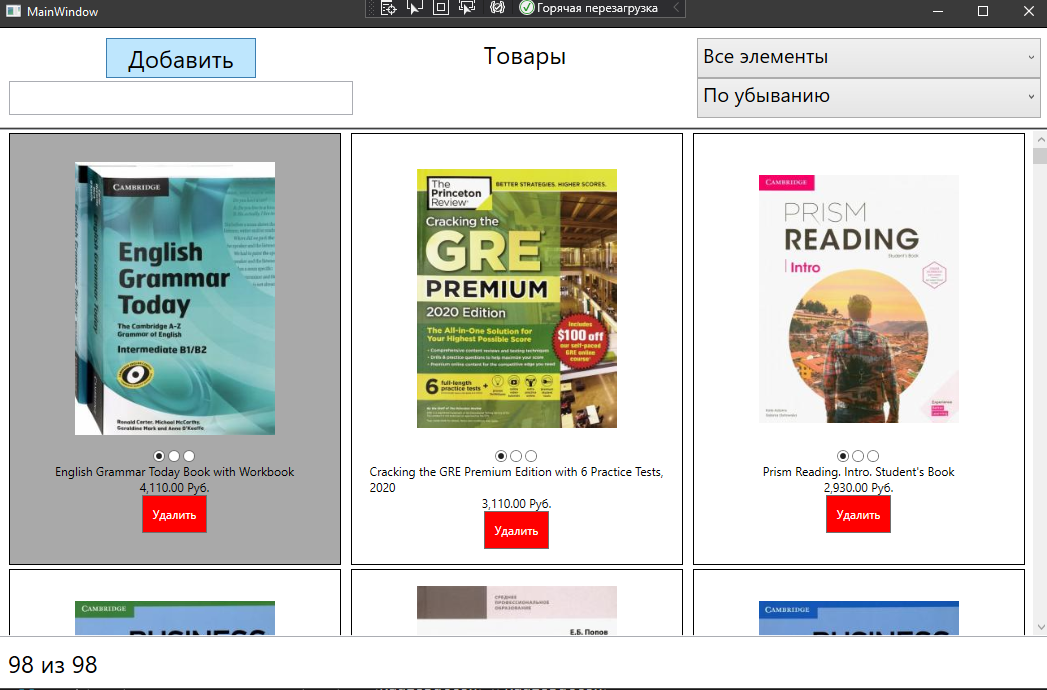
Сессия 2

# Задание на 03.06.2021 Разработка

## Список товаров

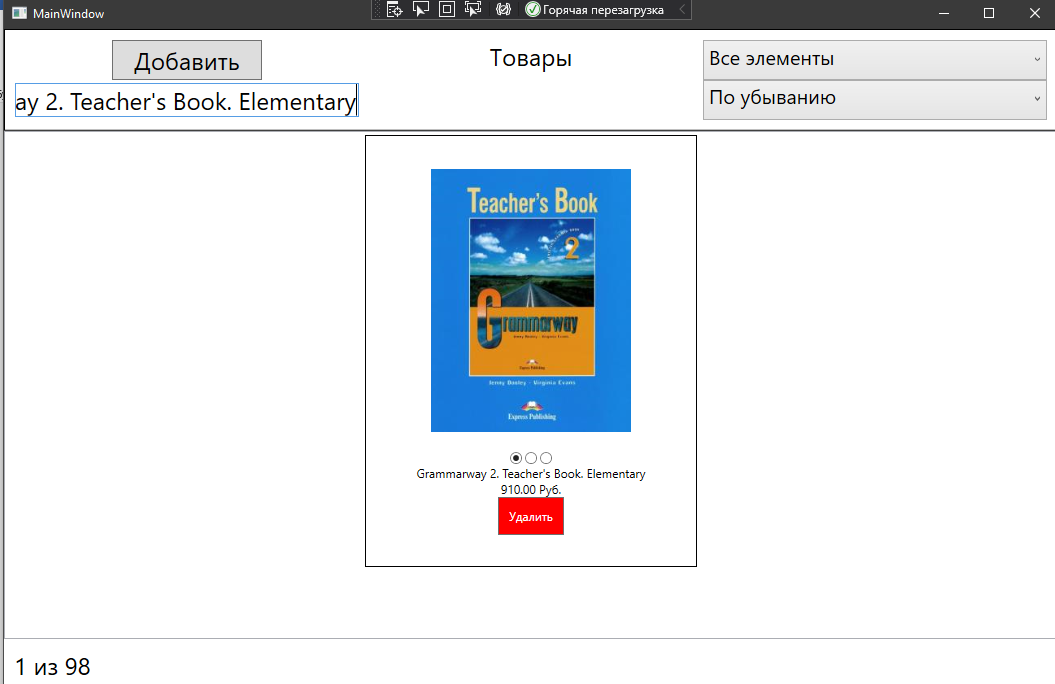
Необходимо отобразить список товаров (в виде плитки), который должен соответствовать макету, предоставленному в ресурсах к заданию (пример в файле products\_grid.png). Однако это не означает, что необходимо следовать макету в точности до пикселя - это всего лишь схема расположения элементов, на которую нужно ориентироваться при разработке пользовательского интерфейса.



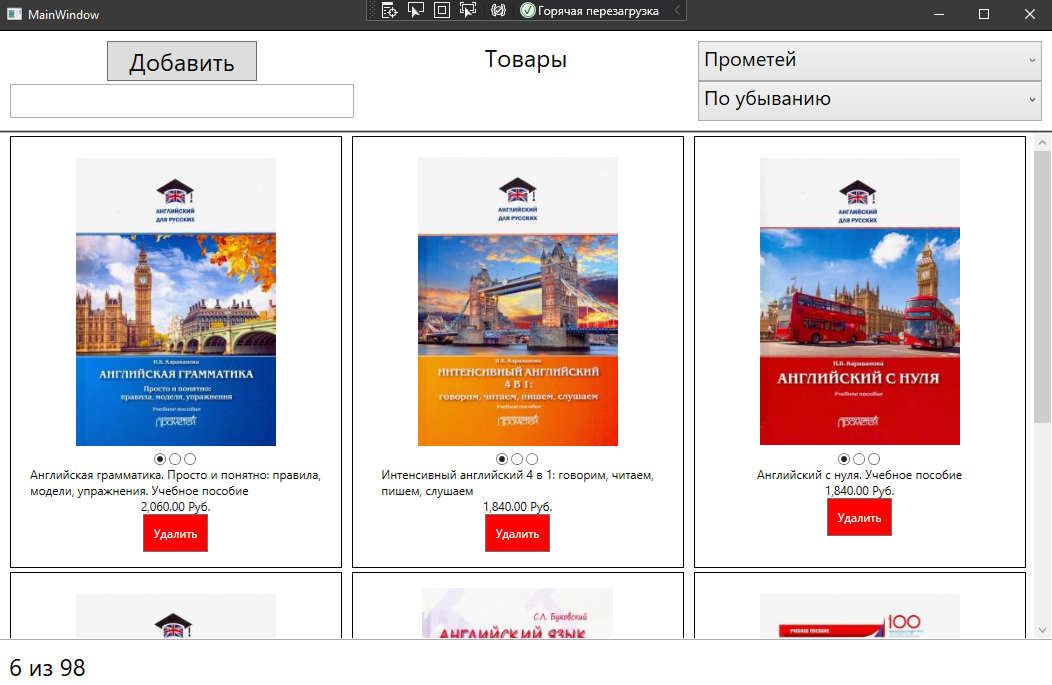
Каждый товар должен содержать следующую информацию: главное изображение, наименование, стоимость и отметку об активности (является товар на данный момент актуальным или нет).

Если товар не активен, то фон элемента должен быть серого цвета, в остальных случаях - белого.

На странице со списком товаров необходимо добавить возможность поиска товаров по названию и описанию. Поиск должен работать в реальном времени (то есть без необходимости нажатия кнопки “найти”).

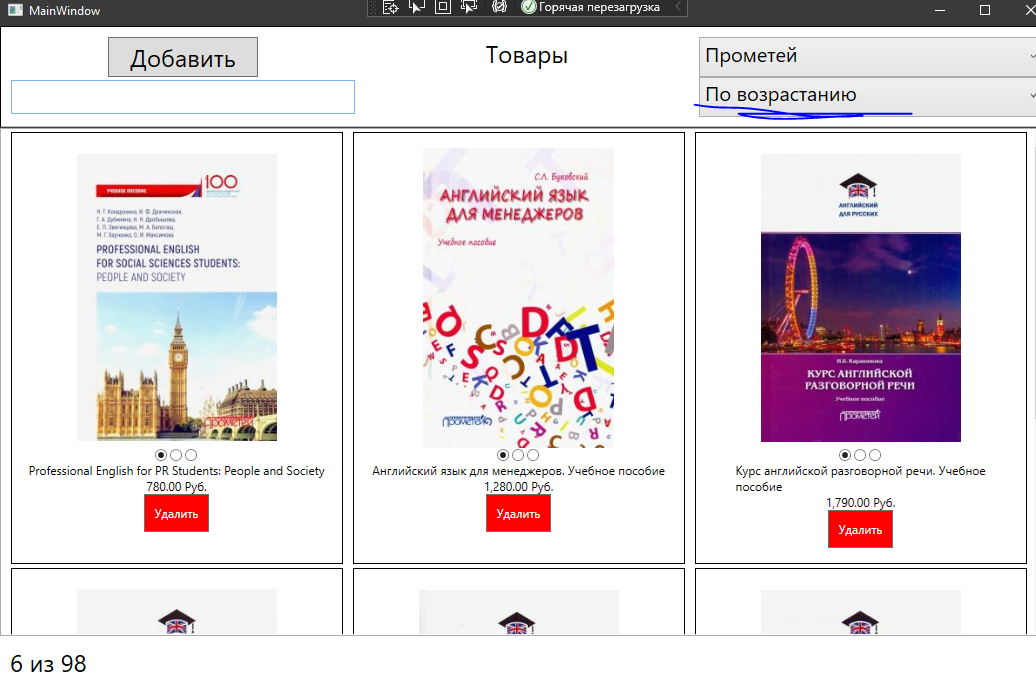


На странице со списком товаров необходимо добавить возможность фильтрации списка по производителю. Все производители из базы данных должны быть выведены в выпадающий список для фильтрации. Первым элементом в выпадающем списке должен быть “Все элементы”, при выборе которого настройки фильтра сбрасываются. Фильтрация должна работать в реальном времени (то есть без необходимости нажатия кнопки “найти”).

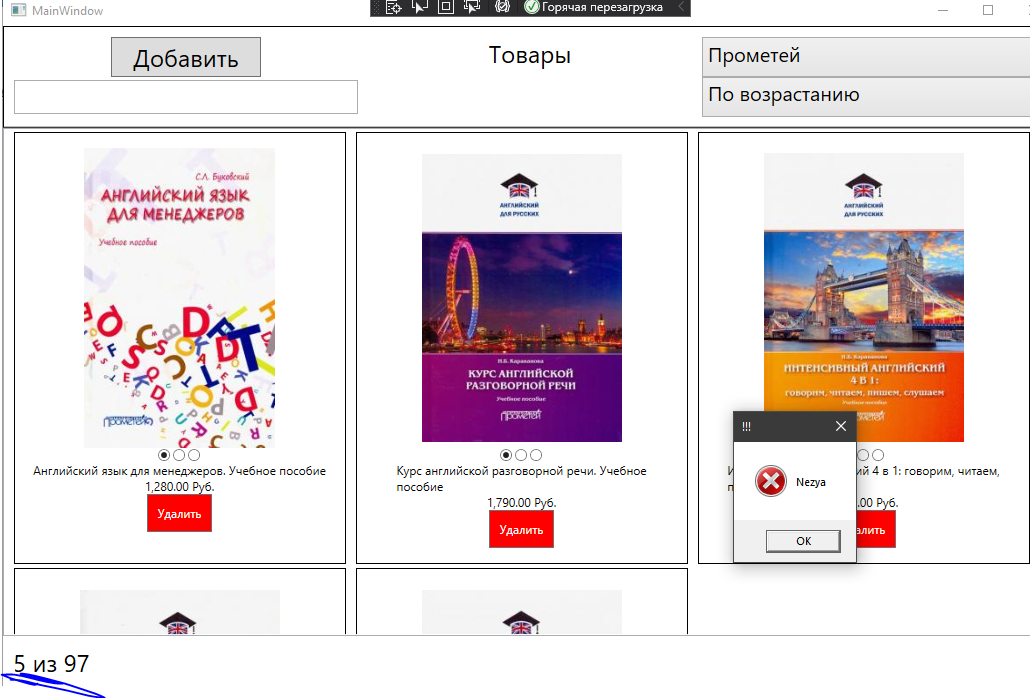


Функции фильтрации и поиска должны применяться совместно к итоговой выборке.

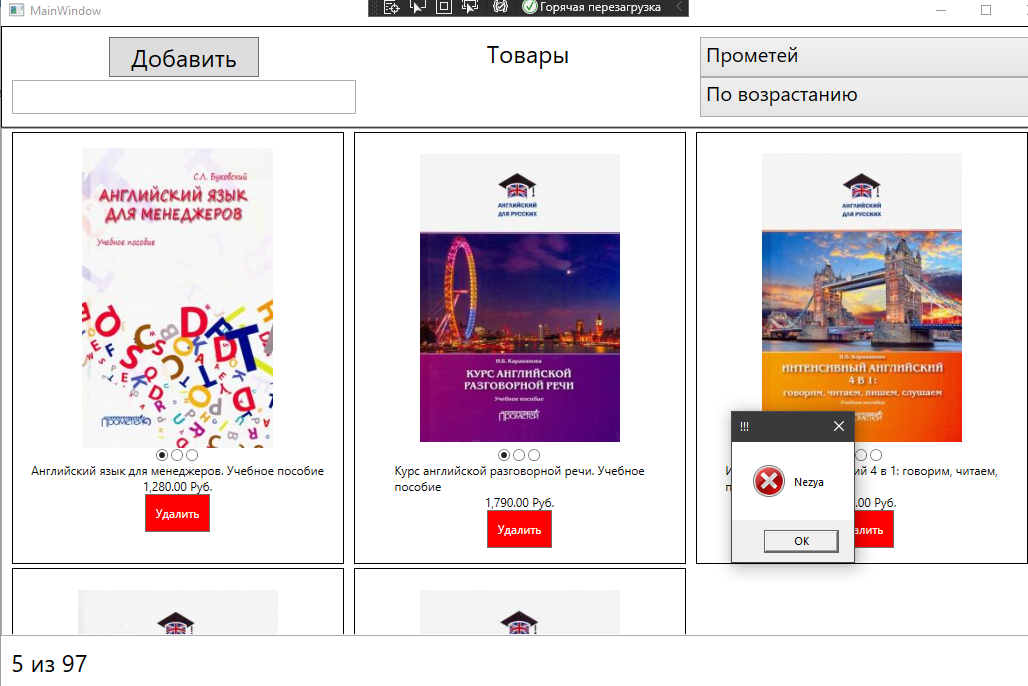
Добавьте возможность отсортировать список по убыванию и возрастанию стоимости товара. Сортировка должна быть реализована отдельно (не с помощью стандартных функций элементов управления), потому что в дальнейшем планируется усложнение этого функционала путем добавления дополнительных условий.



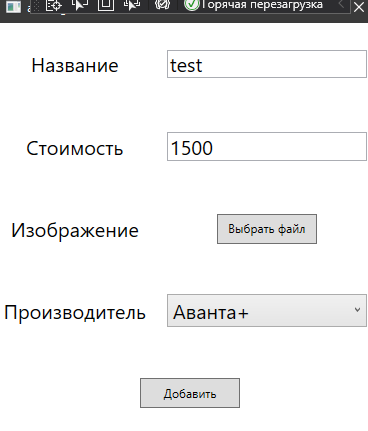
В нижней части окна необходимо показывать количество выведенных данных и общее количество записей в базе. Например, 230 из 450. В случае если данные в таблицу выводятся после фильтрации или поиска, количество выведенных данных необходимо обновить исходя из размера выборки.

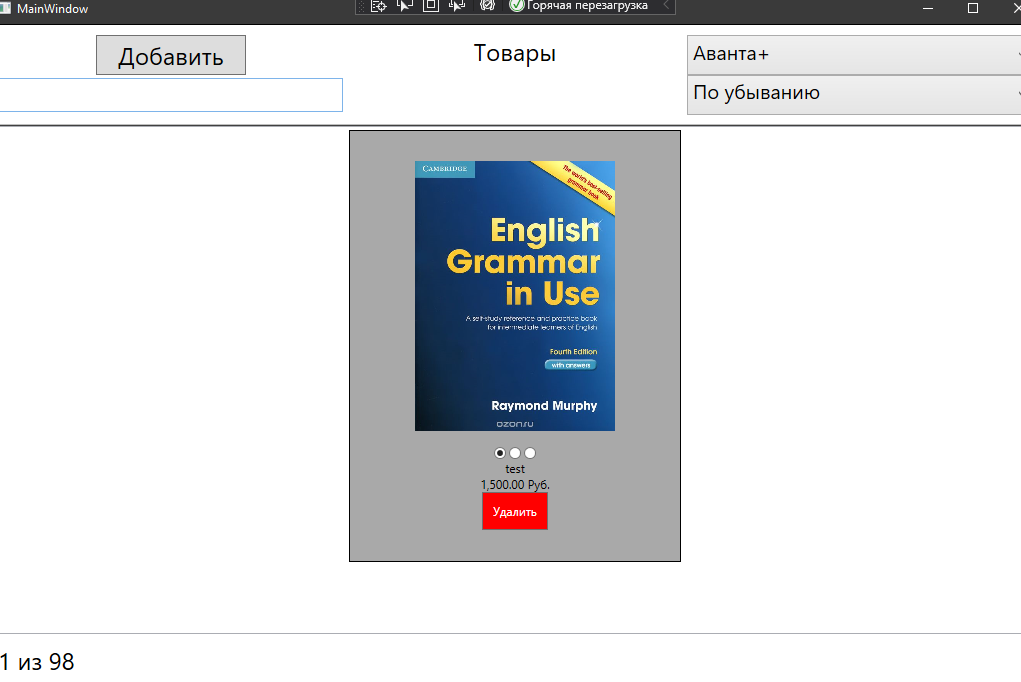


Должна быть реализована возможность удаления выбранного товара. При этом должны соблюдаться следующие условия. Если у товара есть прикрепленные дополнительные товары, то информация по этим прикрепленным товарам может быть удалена вместе с основным. Но если у товара есть информация о его продажах, то удаление товара из базы данных должно быть запрещено.



Добавление:





**Код разметки главной:**

<Window x:Class="session2.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:session2"

mc:Ignorable="d"

Title="MainWindow" Height="700" Width="1070">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="0.2\*"></RowDefinition>

<RowDefinition></RowDefinition>

<RowDefinition Height="0.1\*"></RowDefinition>

</Grid.RowDefinitions>

<Border BorderBrush="Black" BorderThickness="1">

<Grid Margin="10">

<Grid.RowDefinitions>

<RowDefinition></RowDefinition>

<RowDefinition></RowDefinition>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition></ColumnDefinition>

<ColumnDefinition></ColumnDefinition>

<ColumnDefinition></ColumnDefinition>

</Grid.ColumnDefinitions>

<Button Content="Добавить" Width="150" Grid.Column="0" FontSize="24" Click="addButton"></Button>

<TextBlock Grid.Column="1" FontSize="24" HorizontalAlignment="Center" >Товары</TextBlock>

<ComboBox SelectionChanged="manufacturerChange" DisplayMemberPath="Name" FontSize="20" Grid.Column="2" x:Name="combobox1"></ComboBox>

<TextBox TextChanged="nameEnterChange" Grid.Column="0" Grid.Row="1" FontSize="24" VerticalAlignment="Center" ></TextBox>

<ComboBox x:Name="sortBox" SelectionChanged="priceChange" Grid.Column="2" Grid.Row="1" Text="Сортировка по цене" FontSize="20"></ComboBox>

</Grid>

</Border>

<ListView Grid.Row="1" x:Name="listView1" ScrollViewer.HorizontalScrollBarVisibility="Disabled" HorizontalContentAlignment="Center">

<ListView.ItemsPanel>

<ItemsPanelTemplate>

<WrapPanel Orientation="Horizontal" HorizontalAlignment="Center"></WrapPanel>

</ItemsPanelTemplate>

</ListView.ItemsPanel>

<ListView.ItemTemplate>

<DataTemplate>

<Border Background="{Binding colorItem}" BorderBrush="Black" Padding="10" BorderThickness="1">

<StackPanel Width="300" Height="400" Margin="5" Orientation="Vertical">

<Image Width="200" Height="300">

<Image.Source>

<Binding Path="MainImage">

</Binding>

</Image.Source>

</Image>

<StackPanel HorizontalAlignment="Center" Orientation="Horizontal">

<RadioButton IsChecked="True"></RadioButton>

<RadioButton></RadioButton>

<RadioButton></RadioButton>

</StackPanel>

<TextBlock HorizontalAlignment="Center" TextWrapping="Wrap" Text="{Binding Title}"></TextBlock>

<TextBlock HorizontalAlignment="Center" Text="{Binding Cost, StringFormat={}{0:N2} Руб.}"></TextBlock>

<Button Foreground="White" Padding="10" Click="deleteButton" Content="Удалить" HorizontalAlignment="Center" Background="Red"></Button>

</StackPanel>

</Border>

</DataTemplate>

</ListView.ItemTemplate>

</ListView>

<TextBlock FontSize="24" Margin="10" Grid.Row="2" x:Name="countItems"></TextBlock>

</Grid>

</Window>

**Код главной:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

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;

namespace session2

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

Manufacturer toSearch = null;

private string currTitle;

public MainWindow()

{

InitializeComponent();

updateList();

combobox1.Items.Add(new Manufacturer() { Name = "Все элементы" });

foreach (var item in db.GetContext().Manufacturer.ToList())

{

combobox1.Items.Add(item);

}

combobox1.SelectedIndex = 0;

sortBox.Items.Add("По убыванию");

sortBox.Items.Add("По возрастанию");

sortBox.SelectedIndex = 0;

}

private void updateList()

{

IEnumerable<Product> list;

if (toSearch != null)

{

list = db.GetContext().Product.ToList().Where(p => p.Manufacturer == toSearch);

}

else

{

list = db.GetContext().Product;

}

switch (sortBox.SelectedIndex)

{

case 0:

list = from u in list

orderby u.Cost descending

select u;

break;

case 1:

list = from u in list

orderby u.Cost ascending

select u;

break;

default:

break;

}

if(!string.IsNullOrEmpty(currTitle))

list = list.Where(p => p.Title == currTitle);

countItems.Text = $"{list.ToList().Count} из {db.GetContext().Product.ToList().Count}";

listView1.ItemsSource = list.ToList();

}

private void deleteButton(object sender, RoutedEventArgs e)

{

var deleteItem =(((sender) as Button).DataContext as Product);

if (db.GetContext().ProductSale.ToList().FirstOrDefault(p => p.Product == deleteItem) != null)

{

MessageBox.Show("Nezya", "!!!", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

// Если ок удаляем!

db.GetContext().Product.Remove(((sender) as Button).DataContext as Product);

db.GetContext().SaveChanges();

updateList();

}

private void addButton(object sender, RoutedEventArgs e)

{

new addPage().ShowDialog();

updateList();

}

private void manufacturerChange(object sender, SelectionChangedEventArgs e)

{

if (combobox1.SelectedIndex == 0)

toSearch = null;

else

toSearch = combobox1.SelectedItem as Manufacturer;

updateList();

}

private void priceChange(object sender, SelectionChangedEventArgs e)

{

updateList();

}

private void nameEnterChange(object sender, TextChangedEventArgs e)

{

var currTextBox = sender as TextBox;

currTitle = currTextBox.Text;

updateList();

}

}

}

**Код разметки страницы добавления:**

<Window x:Class="session2.addPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:session2"

mc:Ignorable="d"

Title="addPage" Height="450" Width="400">

<Window.Resources>

<Style x:Key="tbstyle" TargetType="{x:Type TextBlock}">

<Setter Property="HorizontalAlignment" Value="center"></Setter>

<Setter Property="VerticalAlignment" Value="center"></Setter>

<Setter Property="FontSize" Value="20"></Setter>

</Style>

<Style x:Key="tboxstyle" TargetType="{x:Type TextBox}">

<Setter Property="HorizontalAlignment" Value="center"></Setter>

<Setter Property="VerticalAlignment" Value="center"></Setter>

<Setter Property="FontSize" Value="20"></Setter>

<Setter Property="Grid.Column" Value="1"></Setter>

<Setter Property="Width" Value="200"></Setter>

</Style>

<Style x:Key="cboxstyle" TargetType="{x:Type ComboBox}">

<Setter Property="HorizontalAlignment" Value="center"></Setter>

<Setter Property="VerticalAlignment" Value="center"></Setter>

<Setter Property="FontSize" Value="20"></Setter>

<Setter Property="Width" Value="200"></Setter>

<Setter Property="Grid.Column" Value="1"></Setter>

</Style>

</Window.Resources>

<Grid>

<Grid.Resources>

<Style BasedOn="{StaticResource tbstyle}" TargetType="{x:Type TextBlock}" />

<Style BasedOn="{StaticResource tboxstyle}" TargetType="{x:Type TextBox}" />

<Style BasedOn="{StaticResource cboxstyle}" TargetType="{x:Type ComboBox}"/>

</Grid.Resources>

<Grid.RowDefinitions>

<RowDefinition></RowDefinition>

<RowDefinition></RowDefinition>

<RowDefinition></RowDefinition>

<RowDefinition></RowDefinition>

<RowDefinition></RowDefinition>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition></ColumnDefinition>

<ColumnDefinition Width="1.5\*"></ColumnDefinition>

</Grid.ColumnDefinitions>

<TextBlock>Название</TextBlock>

<TextBox Grid.Column="1" Text="{Binding Title}"></TextBox>

<TextBlock Grid.Row="1">Стоимость</TextBlock>

<TextBox Grid.Row="1" Grid.Column="1" Text="{Binding Cost}"></TextBox>

<TextBlock Grid.Row="2">Изображение</TextBlock>

<Button Grid.Row="2" Grid.Column="1" Content="Выбрать файл" Width="100" Height="30" Click="addPhoto"></Button>

<TextBlock Grid.Row="3">Производитель</TextBlock>

<ComboBox DisplayMemberPath="Name" Grid.Row="3" SelectedItem="{Binding Manufacturer}" x:Name="combobox1"></ComboBox>

<Button Grid.Row="4" Grid.ColumnSpan="2" Width="100" Height="30" Click="addProduct" Content="Добавить"></Button>

</Grid>

</Window>

**Код страницы добавления:**

using Microsoft.Win32;

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

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.Shapes;

namespace session2

{

/// <summary>

/// Логика взаимодействия для addPage.xaml

/// </summary>

public partial class addPage : Window

{

Product product = new Product() { IsActive = false };

public addPage()

{

InitializeComponent();

DataContext = product;

combobox1.ItemsSource = db.GetContext().Manufacturer.ToList();

}

private void addPhoto(object sender, RoutedEventArgs e)

{

OpenFileDialog openFileDialog = new OpenFileDialog();

openFileDialog.Filter = "Файлы рисунков(\*.bmp, \*.jpg)| \*.bmp; \*.jpg | Все файлы(\*.\*) | \*.\*";

if (openFileDialog.ShowDialog() == true)

product.MainImage = File.ReadAllBytes(openFileDialog.FileName);

}

private void addProduct(object sender, RoutedEventArgs e)

{

if (product.Title != null && product.Manufacturer != null && product.Cost != 0 && product.MainImage != null)

{

db.GetContext().Product.Add(product);

db.GetContext().SaveChanges();

this.Close();

}

else

{

MessageBox.Show("Не все поля заполнены!", "Не все поля заполнены!");

}

}

}

}

# Задание на 17.06.2021 Разработка тестовых сценариев (Test-cases)

Для выполнения процедуры тестирования удаления товаров Вам нужно описать пять сценариев. Удаление может быть выполнимо, а может быть отклонено согласно требованиям предметной области. Необходимо, чтобы варианты тестирования демонстрировали различные исходы работы алгоритма. Для описания тестовых сценариев в ресурсах предоставлен шаблон testing-template.docx.

## Тестовый пример #1:

|  |  |
| --- | --- |
| **Тестовый пример #** | ТС\_ПИ\_1 |
| **Приоритет тестирования** | Низкий |
| **Заголовок/название теста** | Удаление элемента |
| **Краткое изложение теста** | Удаление элемента и связанных с ним |
| **Этапы теста** | 1. Выбор нужного элемента 2. Нажатие кнопки удаления |
| **Тестовые данные** | Элемент Tree or Three? An elementary pronunciation course |
| **Ожидаемый результат** | Удаление Tree or Three? An elementary pronunciation course и New Enterprise A2. Workbook with digibook app |
| **Фактический результат** | Удаление Tree or Three? An elementary pronunciation course и New Enterprise A2. Workbook with digibook app |
| **Статус** | Зачет |
| **Предварительное условие** | Запуск программы |
| **Постусловие** | Стало на 2 элемента меньше |
| **Примечания/комментарии** |  |

## Тестовый пример #2:

|  |  |
| --- | --- |
| **Тестовый пример #** | ТС\_ПИ\_2 |
| **Приоритет тестирования** | Низкий |
| **Заголовок/название теста** | Удаление элемента |
| **Краткое изложение теста** | Сообщение о невозможности удаления элемента |
| **Этапы теста** | 1. Выбор нужного элемента 2. Нажатие кнопки удаления |
| **Тестовые данные** | Удаление элемента Super Minds. Level 4. Workbook with Online Resources |
| **Ожидаемый результат** | Ошибка |
| **Фактический результат** | Ошибка |
| **Статус** | Зачет |
| **Предварительное условие** | Запуск программы |
| **Постусловие** | Без изменений |
| **Примечания/комментарии** |  |

## Тестовый пример #3:

|  |  |
| --- | --- |
| Тестовый пример **#** | ТС\_ПИ\_3 |
| **Приоритет тестирования** | Низкий |
| **Заголовок/название теста** | Удаление элемента |
| **Краткое изложение теста** | Удаление единственного элемента |
| **Этапы теста** | 1. Выбор нужного элемента 2. Нажатие кнопки удаления |
| **Тестовые данные** | Элемента Super Minds. Level 5 |
| **Ожидаемый результат** | Удаление Super Minds. Level 5 |
| **Фактический результат** | Удаление Super Minds. Level 5 |
| **Статус** | Зачет |
| **Предварительное условие** | Запуск программы |
| **Постусловие** | Удален 1 элемент |
| **Примечания/комментарии** |  |

## Тестовый пример #4:

|  |  |
| --- | --- |
| Тестовый пример **#** | ТС\_ПИ\_4 |
| **Приоритет тестирования** | Низкий |
| **Заголовок/название теста** | Удаление элемента |
| **Краткое изложение теста** | Удаление единственного элемента |
| **Этапы теста** | 1. Выбор нужного элемента 2. Нажатие кнопки удаления |
| **Тестовые данные** | Элемент Английский без барьеров. For beginners |
| **Ожидаемый результат** | Удаление Английский без барьеров. For beginners |
| **Фактический результат** | УдалениеАнглийский без барьеров. For beginners и Быстрый английский. Тренажер по чтению |
| **Статус** | Незачет |
| **Предварительное условие** | Запуск программы |
| **Постусловие** | Удалено 2 элемента |
| **Примечания/комментарии** |  |

## Test case #5:

|  |  |  |
| --- | --- | --- |
| **Test Case #** | ТС\_ПИ\_4 |  |
| **Приоритет тестирования** | Низкий |  |
| **Заголовок/название теста** | Удаление элемента |  |
| **Краткое изложение теста** | Удаление единственного элемента |  |
| **Этапы теста** | 1. Выбор нужного элемента 2. Нажатие кнопки удаления |  |
| **Тестовые данные** | Элемент Ship or Sheep? An intermediate pronunciation course |  |
| **Ожидаемый результат** | Удаление Ship or Sheep? An intermediate pronunciation course |  |
| **Фактический результат** | ошибка |  |
| **Статус** | Незачет |  |
| **Предварительное условие** | Запуск программы |  |
| **Постусловие** | Без изменений |  |
| **Примечания/комментарии** |  |  |