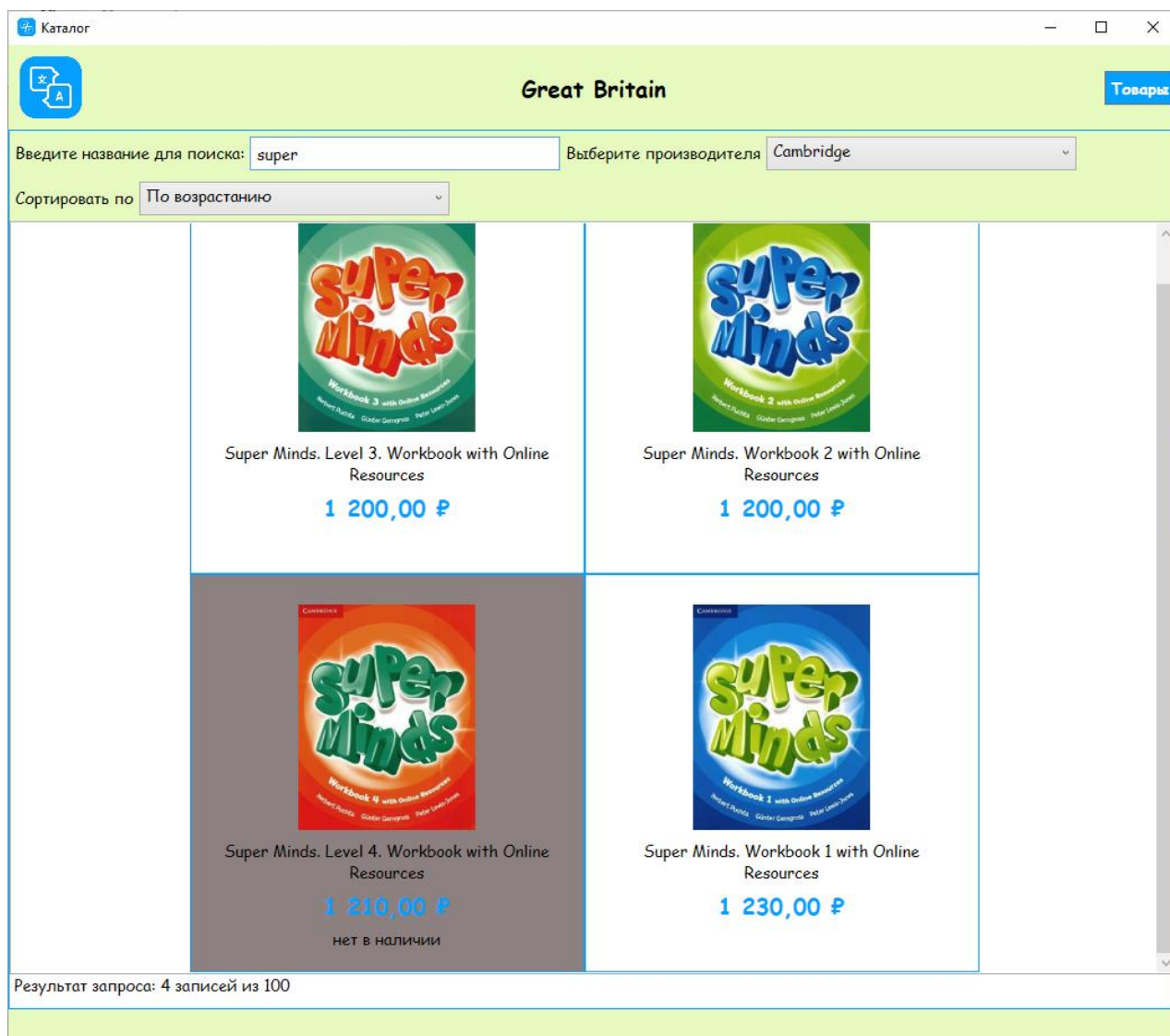


Разработка проекта приложения по технологии WPF на тему Школа Great Britain

Задание 2. Создание приложения

Страница каталога товаров CatalogOfGoodsPage

1. Откройте созданную страницу **CatalogOfGoodsPage.xaml**. Разработайте дизайн данной страницы:



Добавить стили:

```
<!--стиль для меток ComboBoxItem -->
<Style TargetType="ComboBoxItem">
    <Setter Property="Background" Value="{StaticResource main}"/>
    <Setter Property="Height" Value="40"/>
</Style>
<!--стиль для меток ListViewItem -->
<Style TargetType="ListViewItem" x:Key="good_item">
    <Setter Property="Background" Value="{Binding GetColor}" />
    <Setter Property="BorderBrush" Value="{StaticResource accent}"/>
    <Setter Property="BorderThickness" Value="1"/>
</Style>
```

```

Title="Katanor"
IsVisibleChanged="PageIsVisibleChanged"
Style="{StaticResource base_page}">
<Grid>
    <Grid.RowDefinitions>
        <RowDefinition Height="auto" />
        <RowDefinition Height="*" />
        <RowDefinition Height="30" />
    </Grid.RowDefinitions>
    <WrapPanel Grid.Row="0"
        HorizontalAlignment="Stretch"
        Background=#FFD700 "{StaticResource additional}">
        <StackPanel Orientation="Horizontal"
            Margin="0 5">
            <TextBlock Text="Введите название для поиска:"
                Style="{StaticResource base_textblock}" />
            <TextBox Width="275"
                x:Name="TBoxSearch"
                TextChanged="TBoxSearchTextChanged" />
        </StackPanel>
        <StackPanel Orientation="Horizontal"
            Margin="0 5">
            <TextBlock Text="Выберите производителя"
                Style="{StaticResource base_textblock}" />
            <ComboBox Width="275"
                x:Name="ComboDeveloper"
                SelectionChanged="ComboTypeSelectionChanged"
                DisplayMemberPath="Name" />
        </StackPanel>
        <StackPanel Orientation="Horizontal"
            Margin="0 5">
            <TextBlock Text="Сортировать по"
                Style="{StaticResource base_textblock}" />
            <ComboBox Width="275"
                x:Name="ComboSort"
                SelectionChanged="ComboSortSelectionChanged">
                <ComboBoxItem Content="По возрастанию" />
                <ComboBoxItem Content="По убыванию" />
            </ComboBox>
        </StackPanel>
    </WrapPanel>

```

```

<ListView x:Name="LViewGoods"
    Grid.Row="1"
    ScrollViewer.HorizontalScrollBarVisibility="Disabled"
    ItemContainerStyle="{StaticResource good_item}">
    <ListView.ItemsPanel>
        <ItemsPanelTemplate>
            <WrapPanel Orientation="Horizontal"
                HorizontalAlignment="Center" />
        </ItemsPanelTemplate>
    </ListView.ItemsPanel>
    <ListView.ItemTemplate>
        <DataTemplate>
            <Grid Margin="20"
                Width="300">
                <Grid.RowDefinitions>
                    <RowDefinition Height="210" />
                    <RowDefinition Height="auto" />
                    <RowDefinition Height="20" />
                </Grid.RowDefinitions>
                <Image Width="240"
                    Grid.Row="0"
                    Stretch="Uniform"
                    HorizontalAlignment="Center"
                    Margin="5"
                    Source="{Binding Path=GetPhoto}" />
                <StackPanel Grid.Row="1"
                    Height="auto">
                    <TextBlock Padding="3"
                        Height="Auto"
                        Style="{StaticResource item_textblock}">
                        <TextBlock.Text>
                            <MultiBinding StringFormat="{0}{0}">
                                <Binding Path="Name" />
                            </MultiBinding>
                        </TextBlock.Text>
                    </TextBlock>
                    <TextBlock Text="{Binding Price, StringFormat='c', ConverterCulture='ru-RU'}"
                        Height="Auto"
                        Style="{StaticResource item_textblock}"
                        VerticalAlignment="Center"
                        Foreground="■" {StaticResource akcent}"
                        FontWeight="Bold"
                        FontSize="20" />
                </StackPanel>
                <TextBlock Text="{Binding GetStatus}"
                    Height="Auto"
                    Style="{StaticResource item_textblock}"
                    Grid.Row="2" />
            </Grid>
        </DataTemplate>
    </ListView.ItemTemplate>
</ListView>
<TextBlock Grid.Row="2"
    x:Name="TextBlockCount" />

```

2. Реализуйте обработчики событий, используемые при выборе элементов полей:

```

namespace WpfEngSchool_EV.Pages
{
    /// <summary>
    /// Логика взаимодействия для CatalogOfGoodsPage.xaml
    /// </summary>
    Ссылка 4
    public partial class CatalogOfGoodsPage : Page
    {
        int _itemcount = 0;
        Ссылка 1
        public CatalogOfGoodsPage()
        {
            InitializeComponent();
            // загрузка данных в combobox + добавление дополнительной строки
            using (var context = new BD26_EVEntities())
            {
                var developers = context.Manufacturer.OrderBy(p => p.Name).ToList();
                developers.Insert(0, new Manufacturer
                {
                    Name = "Все типы"
                });
                ComboDeveloper.ItemsSource = developers;
                ComboDeveloper.SelectedIndex = 0;
                // загрузка данных в listview сортируем по названию
                LViewGoods.ItemsSource = context.Product.OrderBy(p => p.Name).ToList();
                _itemcount = LViewGoods.Items.Count;
                // отображение количества записей
                TextBlockCount.Text = $" Результат запроса: {_itemcount} записей из {_itemcount}";
            }
        }
        Ссылка 1
        private void PageIsVisibleChanged(object sender, DependencyPropertyChangedEventArgs e)
        {
            using (var context = new BD26_EVEntities())
            //обновление данных после каждой активации окна
            if (Visibility == Visibility.Visible)
            {
                context.ChangeTracker.Entries().ToList().ForEach(p => p.Reload());
                LViewGoods.ItemsSource = context.Product.OrderBy(p => p.Name).ToList();
            }
        }
    }
}

```

```

// Поиск товаров, которые содержат данную поисковую строку
Ссылка: 1
private void TBoxSearchTextChanged(object sender, TextChangedEventArgs e)
{
    UpdateData();
}
// Поиск товаров конкретного производителя
Ссылка: 1
private void ComboTypeSelectionChanged(object sender, SelectionChangedEventArgs e)
{
    UpdateData();
}
/// <summary>
/// Метод для фильтрации и сортировки данных
/// </summary>
Ссылка: 3
private void UpdateData()
{
    using (var context = new BD26_EVEntities())
    {
        // получаем текущие данные из бд
        var currentGoods = context.Product.OrderBy(p => p.Name).ToList();
        // выбор только тех товаров, которые принадлежат данному производителю
        if (ComboDeveloper.SelectedIndex > 0)
            currentGoods = currentGoods.Where(p => p.ManufacturerID == (ComboDeveloper.SelectedItem as Manufacturer).ID).ToList();
        // выбор тех товаров, в названии которых есть поисковая строка
        currentGoods = currentGoods.Where(p => p.Name.ToLower().Contains(TBoxSearch.Text.ToLower())).ToList();
        // сортировка
        if (ComboSort.SelectedIndex >= 0)
        {
            // сортировка по возрастанию цены
            if (ComboSort.SelectedIndex == 0)
                currentGoods = currentGoods.OrderBy(p => p.Price).ToList();
            // сортировка по убыванию цены
            if (ComboSort.SelectedIndex == 1)
                currentGoods = currentGoods.OrderByDescending(p => p.Price).ToList();
        }
        // В качестве источника данных присваиваем список данных
        LViewGoods.ItemsSource = currentGoods;
        // отображение количества записей
        TextBlockCount.Text = $" Результат запроса: {currentGoods.Count} записей из {_itemcount}";
    }
}
// сортировка товаров
Ссылка: 1
private void ComboSortSelectionChanged(object sender, SelectionChangedEventArgs e)
{
    UpdateData();
}

```

- Откройте файл **Product.cs** из папки **Models**. Добавьте внутрь класса следующий программный код для задания цветовых акцентов и пути к изображениям:

Подключить пространство имен:

`using System.IO;`

в класс `public partial class Product` добавить

```
/// <summary>
/// Возвращает абсолютный путь к изображению
/// </summary>
Ссылка 0
public string GetPhoto
{
    get
    {
        if (Image is null)
            return null;
        return Directory.GetCurrentDirectory() + @"\" + Image.Trim();
    }
}
```

```

/// <summary>
/// Задаёт цвет фона товара
/// </summary>
Ссылка 0
public string GetColor
{
    get
    {
        if (IsActive)
            return "#fff";
        else
            return "#8C8181";
    }
}
/// <summary>
/// Текстовое представление активности товара
/// </summary>
Ссылка 0
public string GetStatus
{
    get
    {
        if (IsActive)
            return "";
        else
            return "нет в наличии";
    }
}
Ссылка 0
public string GetInfo
{
    get
    {
        return $"{Name} – {Price.ToString("c")} рублей.";
    }
}

```

4. Проверьте работу приложения