

**Решение "UI-PR-TASK-30" (1 проекта 1)**

**UI-PR-TASK-30**

- Properties
- Зависимости
- AppStyle
- Images
- UserControls
- App.xaml**
- AssemblyInfo.cs
- MainWindow.xaml

```
<Application x:Class="UI_PR_TASK_30.App"  
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"  
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"  
    xmlns:local="clr-namespace:UI_PR_TASK_30"  
    xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"  
    StartupUri="MainWindow.xaml">  
    <Application.Resources>  
        <ResourceDictionary>  
            <ResourceDictionary.MergedDictionaries>  
                <materialDesign:BundledTheme BaseTheme="Light" PrimaryColor="{x:Null}" SecondaryColor="Lime" />  
                <ResourceDictionary Source="pack://application:///MaterialDesignThemes.Wpf;component/Themes/MaterialDesignTheme.Defaults.xaml" />  
                <ResourceDictionary Source="AppStyle/MainStyle.xaml"/>  
            </ResourceDictionary.MergedDictionaries>  
        </ResourceDictionary>  
    </Application.Resources>  
</Application>
```

**Решение "UI-PR-TASK-30" (1 проекта 1)**

**UI-PR-TASK-30**

- Properties
- Зависимости
- AppStyle**
  - MainStyle.xaml**
- Images
- UserControls
- App.xaml
- AssemblyInfo.cs
- MainWindow.xaml

```
<ResourceDictionary xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">
    <Style TargetType="Button" x:Key="menuButton">
        <Setter Property="Background" Value="Transparent" />
        <Setter Property="Margin" Value="15 0 15 20" />
        <Setter Property="Cursor" Value="Hand" />
        <Setter Property="Height" Value="50" />
        <Setter Property="FocusVisualStyle" Value="{x:Null}" />
        <Setter Property="Template">
            <Setter.Value>
                <ControlTemplate TargetType="Button">
                    <Border Background="{TemplateBinding Background}"
                        CornerRadius="10"
                        BorderThickness="1"
                        BorderBrush="Transparent">
                        <ContentPresenter HorizontalAlignment="Center"
                            VerticalAlignment="Center"/>
                    </Border>
                </ControlTemplate>
            </Setter.Value>
        </Setter>
    <Style.Triggers>
        <Trigger Property="IsMouseOver" Value="True">
            <Setter Property="Template">
                <Setter.Value>
                    <ControlTemplate TargetType="Button">
                        <Border Background="{TemplateBinding Background}"
                            CornerRadius="10"
                            BorderThickness="1"
                            BorderBrush="#FFFFFF">
                            <ContentPresenter HorizontalAlignment="Center"
                                VerticalAlignment="Center"/>
                        </Border>
                    </ControlTemplate>
                </Setter.Value>
            </Setter>
        </Trigger>
    </Style.Triggers>
</Style>
```

```
<Style TargetType="Border" x:Key="borderPage">
    <Setter Property="Background" Value="#000000" />
    <Setter Property="CornerRadius" Value="25" />
    <Setter Property="Grid.Column" Value="1" />
    <Setter Property="Margin" Value="30 0 0 0" />
    <Setter Property="Effect">
        <Setter.Value>
            <DropShadowEffect ShadowDepth="1"
                               BlurRadius="20"
                               Opacity="0.5"/>
        </Setter.Value>
    </Setter>
</Style>

<Style TargetType="ToggleButton" x:Key="toggleButton">
    <Setter Property="Cursor" Value="Hand" />
    <Setter Property="Template">
        <Setter.Value>
            <ControlTemplate TargetType="ToggleButton">
                <Grid>
                    <Border x:Name="Border"
                            Background="#475272"
                            CornerRadius="13"
                            Width="50"
                            Height="26"
                            VerticalAlignment="Center"
                            HorizontalAlignment="Center"/>
                    <Ellipse x:Name="Ellipse"
                             Fill="#FBFBFB"
                             Height="24"
                             Width="24"
                             Margin="25 0 0 0"
                             HorizontalAlignment="Left"
                             VerticalAlignment="Center"/>
                </Grid>
            <ControlTemplate.Triggers>
                <Trigger Property="IsChecked" Value="True">
                    <Trigger.EnterActions>
                        <BeginStoryboard>
                            <Storyboard>
```

```

        <ColorAnimation Storyboard.TargetName="Border"
                        Storyboard.TargetProperty="(Border.Background).(SolidColorBrush.Color)"
                        To="#9391AB"
                        Duration="0:0:0:.2"/>
        <ThicknessAnimation Storyboard.TargetName="Ellipse"
                            Storyboard.TargetProperty="Margin"
                            To="1 0 0 0"
                            Duration="0:0:0:.2"/>
    </Storyboard>
</BeginStoryboard>
</Trigger.EnterActions>

<Trigger.ExitActions>
    <BeginStoryboard>
        <Storyboard>
            <ColorAnimation Storyboard.TargetName="Border"
                            Storyboard.TargetProperty="(Border.Background).(SolidColorBrush.Color)"
                            To="#475272"
                            Duration="0:0:0:.2"/>
            <ThicknessAnimation Storyboard.TargetName="Ellipse"
                                Storyboard.TargetProperty="Margin"
                                To="25 0 0 0"
                                Duration="0:0:0:.2"/>
        </Storyboard>
    </BeginStoryboard>
</Trigger.ExitActions>
</Trigger>
</ControlTemplate.Triggers>
</ControlTemplate>
<Setter.Value>
</Setter>
</Style>

<Style TargetType="Button" x:Key="addButton">
    <Setter Property="Background" Value="&#FFFFF" />
    <Setter Property="Foreground" Value="&#00192E" />
    <Setter Property="Padding" Value="0 3" />
    <Setter Property="Margin" Value="0 0 0 5" />
    <Setter Property="Cursor" Value="Hand" />
    <Setter Property="Height" Value="90" />
    <Setter Property="FontSize" Value="16" />
    <Setter Property="FontWeight" Value="Bold" />
    <Setter Property="FocusVisualStyle" Value="{x:Null}" />
    <Setter Property="Template">

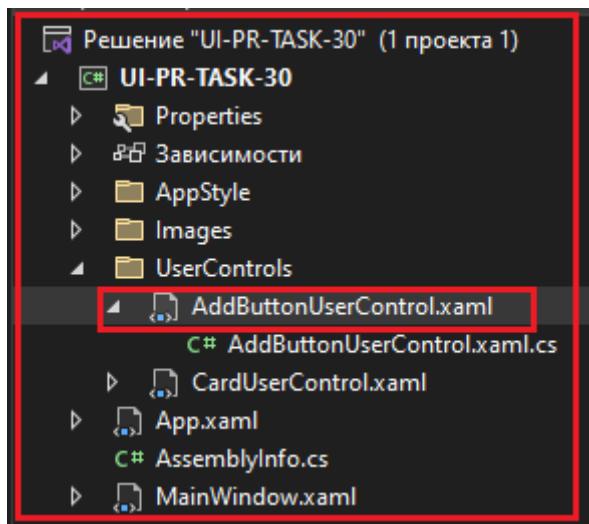
```

```

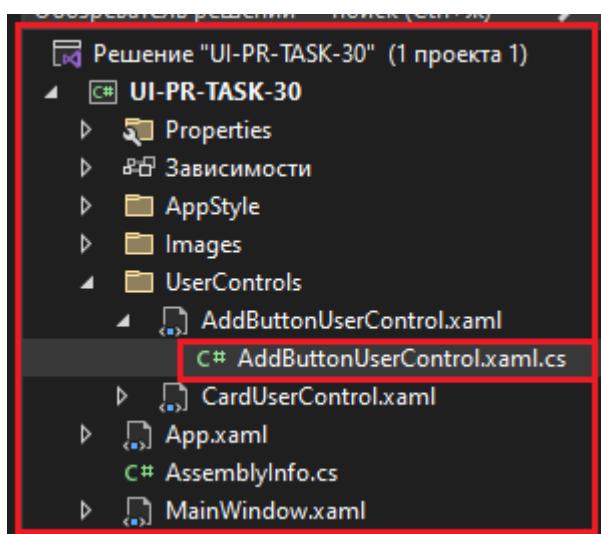
<Setter.Value>
    <ControlTemplate TargetType="Button">
        <Border Background="{TemplateBinding Background}"
                CornerRadius="45">
            <ContentPresenter HorizontalAlignment="Center"
                              VerticalAlignment="Center"/>
        </Border>
    </ControlTemplate>
</Setter.Value>
</Setter>

<Style.Triggers>
    <Trigger Property="IsMouseOver" Value="True">
        <Setter Property="Background" Value="&#E0E0E0" />
    </Trigger>
</Style.Triggers>
</Style>

```



```
<UserControl x:Class="UI_PR_TASK_30.UserControls.AddButtonUserControl"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"
    x:Name="addButtonUC">
    <Button Width="300"
        VerticalAlignment="Top"
        HorizontalAlignment="Right"
        Style="{StaticResource addButton}">
        <StackPanel Orientation="Horizontal">
            <TextBlock Text="{Binding ElementName=addButtonUC, Path=Title}"
                VerticalAlignment="Center"/>
            <Border Margin="20 0 0 0"
                Width="37"
                Padding="10"
                Background="#475272"
                CornerRadius="50">
                <materialDesign:PackIcon Kind="AddBold"
                    Foreground="White"/>
            </Border>
        </StackPanel>
    </Button>
</UserControl>
```

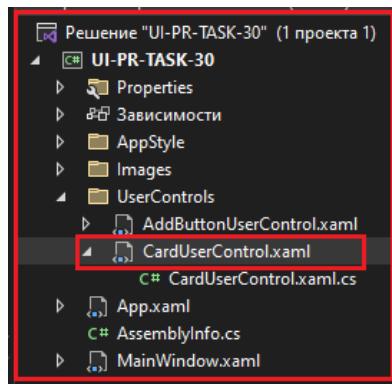


```
using System.Windows;
using System.Windows.Controls;

namespace UI_PR_TASK_30.UserControls
{
    public partial class AddButtonUserControl : UserControl
    {
        public AddButtonUserControl()
        {
            InitializeComponent();
        }

        public string Title
        {
            get => (string)GetValue>TitleProperty;
            set => SetValue>TitleProperty, value;
        }

        public static readonly DependencyProperty TitleProperty =
            DependencyProperty.Register<string>("Title", typeof(string), typeof(AddButtonUserControl));
    }
}
```



```
<UserControl x:Class="UI_PR_TASK_30.UserControls.CardUserControl"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    x:Name="cardUC">
    <Border CornerRadius="25"
        Padding="20"
        Height="150"
        HorizontalAlignment="Stretch">
        <Border.Effect>
            <DropShadowEffect ShadowDepth="1"
                BlurRadius="20"
                Opacity="0.5"/>
        </Border.Effect>
        <Border.Style>
            <Style TargetType="Border">
                <Setter Property="Width" Value="150" />
                <Style.Triggers>
                    <DataTrigger Binding="{Binding ElementName=toggle, Path=IsChecked}"
                        Value="True">
                        <Setter Property="Background" Value="#475272" />
                    </DataTrigger>
                    <DataTrigger Binding="{Binding ElementName=toggle, Path=IsChecked}"
                        Value="False">
                        <Setter Property="Background" Value="#FFFFFF" />
                    </DataTrigger>
                    <DataTrigger Binding="{Binding ElementName=cardUC, Path=IsHorizontal}"
                        Value="True">
                        <Setter Property="Width" Value="190" />
                    </DataTrigger>
                    <DataTrigger Binding="{Binding ElementName=cardUC, Path=IsHorizontal}"
                        Value="False">
                        <Setter Property="Width" Value="150" />
                    </DataTrigger>
                </Style.Triggers>
            </Style>
        </Border.Style>
    </Border>

```

```
<Grid>
    <TextBlock FontWeight="SemiBold"
        VerticalAlignment="Top"
        HorizontalAlignment="Left"
        Margin="0 5 0 0">
        <TextBlock.Style>
            <Style TargetType="TextBlock">
                <Style.Triggers>
                    <DataTrigger Binding="{Binding ElementName=toggle, Path=IsChecked}"
                        Value="True">
                        <Setter Property="Text" Value="On" />
                        <Setter Property="Foreground" Value="#FFFFFF" />
                    </DataTrigger>

                    <DataTrigger Binding="{Binding ElementName=toggle, Path=IsChecked}"
                        Value="False">
                        <Setter Property="Text" Value="Off" />
                        <Setter Property="Foreground" Value="#475272" />
                    </DataTrigger>
                </Style.Triggers>
            </Style>
        </TextBlock.Style>
    </TextBlock>

    <CheckBox x:Name="toggle"
        Background="Black"
        HorizontalAlignment="Right"
        IsChecked="{Binding ElementName=cardUC, Path=.IsChecked}">
        <CheckBox.Style>
            <Style TargetType="CheckBox" BasedOn="{StaticResource toggleButton}">
                <Setter Property="VerticalAlignment" Value="Top" />
                <Style.Triggers>
                    <DataTrigger Binding="{Binding ElementName=cardUC, Path=IsHorizontal}"
                        Value="True">
                        <Setter Property="VerticalAlignment" Value="Center" />
                    </DataTrigger>

                    <DataTrigger Binding="{Binding ElementName=cardUC, Path=IsHorizontal}"
                        Value="False">
                        <Setter Property="VerticalAlignment" Value="Top" />
                    </DataTrigger>
                </Style.Triggers>
            </Style>
        </CheckBox.Style>
    </CheckBox>
```

```

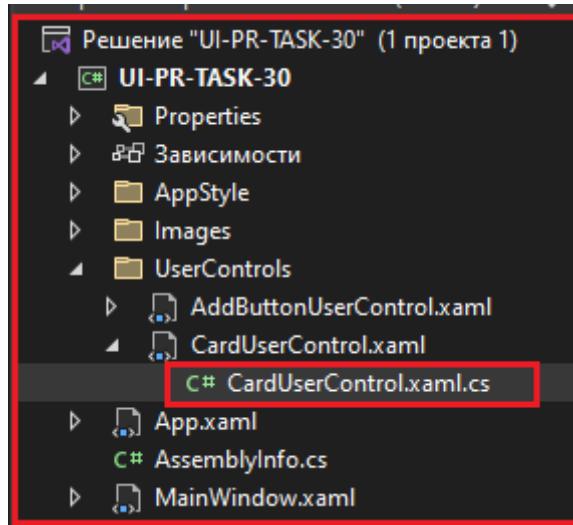
<Image VerticalAlignment="Center"
       Width="40"
       Height="40"
       HorizontalAlignment="Left">
    <Image.Style>
        <Style TargetType="Image">
            <Style.Triggers>
                <DataTrigger Binding="{Binding ElementName=toggle, Path=IsChecked}"
                             Value="True">
                    <Setter Property="Source" Value="{Binding ElementName=cardUC, Path=ImageOn}" />
                </DataTrigger>

                <DataTrigger Binding="{Binding ElementName=toggle, Path=IsChecked}"
                             Value="False">
                    <Setter Property="Source" Value="{Binding ElementName=cardUC, Path=ImageOff}" />
                </DataTrigger>
            </Style.Triggers>
        </Style>
    </Image.Style>
</Image>

<TextBlock VerticalAlignment="Bottom"
           FontWeight="SemiBold"
           Text="{Binding ElementName=cardUC, Path=Title}">
    <TextBlock.Style>
        <Style TargetType="TextBlock">
            <Style.Triggers>
                <DataTrigger Binding="{Binding ElementName=toggle, Path=IsChecked}"
                             Value="True">
                    <Setter Property="Foreground" Value="#FFFFFF" />
                </DataTrigger>

                <DataTrigger Binding="{Binding ElementName=toggle, Path=IsChecked}"
                             Value="False">
                    <Setter Property="Foreground" Value="#2F2C37" />
                </DataTrigger>
            </Style.Triggers>
        </Style>
    </TextBlock.Style>
</TextBlock>
</Grid>
</Border>

```



```
Ссылок: 9
public partial class CardUserControl : UserControl
{
    Ссылок: 0
    public CardUserControl()
    {
        InitializeComponent();
    }

    Ссылок: 0
    public string Title
    {
        get => (string)GetValue>TitleProperty;
        set => SetValue>TitleProperty, value;
    }

    public static readonly DependencyProperty TitleProperty =
        DependencyProperty.Register("Title", typeof(string), typeof(CardUserControl));

    Ссылок: 0
    public bool IsChecked
    {
        get => (bool)GetValue.IsCheckedProperty;
        set => SetValue.IsCheckedProperty, value;
    }

    public static readonly DependencyProperty IsCheckedProperty =
        DependencyProperty.Register("IsChecked", typeof(bool), typeof(CardUserControl));

    Ссылок: 0
    public bool IsHorizontal
    {
        get => (bool)GetValue.IsCheckedProperty;
        set => SetValue.IsCheckedProperty, value;
    }

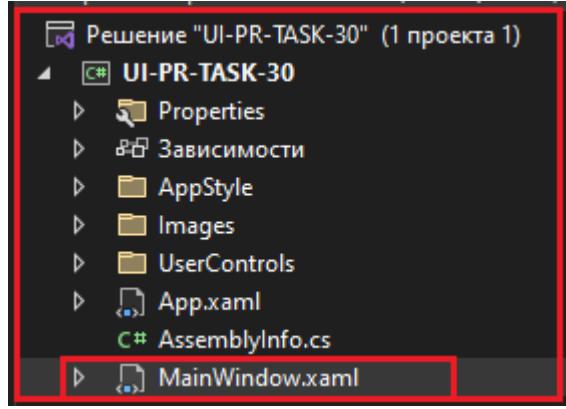
    public static readonly DependencyProperty IsHorizontalProperty =
        DependencyProperty.Register("IsHorizontal", typeof(bool), typeof(CardUserControl));

    Ссылок: 0
    public ImageSource ImageOn
    {
        get => (ImageSource)GetValue>ImageOnProperty;
        set => SetValue>ImageOnProperty, value;
    }

    public static readonly DependencyProperty ImageOnProperty =
        DependencyProperty.Register("ImageOn", typeof(ImageSource), typeof(CardUserControl));
}
```

```
Ссылок: 0
public ImageSource ImageOff
{
    get => (ImageSource)GetValue>ImageOffProperty;
    set => SetValue>ImageOffProperty, value;
}

public static readonly DependencyProperty ImageOffProperty =
    DependencyProperty.Register("ImageOff", typeof(ImageSource), typeof(CardUserControl));
```



```
<Window x:Class="UI_PR_TASK_30.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:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"
    xmlns:local="clr-namespace:UI_PR_TASK_30"
    xmlns:uc ="clr-namespace:UI_PR_TASK_30.UserControls"
    mc:Ignorable="d"
    Width="1080"
    Height="720"
    WindowStyle="None"
    WindowStartupLocation="CenterScreen"
    Background="#FFF"
    AllowsTransparency="True"
    x:Name="mainWindow">

    <Border Background="#FFFFFF"
        Padding="30"
        CornerRadius="25"
        MouseDown="Border_MouseDown"
        MouseLeftButtonDown="Border_MouseLeftButtonDown">

        <Grid>
            <Grid.ColumnDefinitions>
                <ColumnDefinition Width="80"/>
                <ColumnDefinition Width="*"/>
            </Grid.ColumnDefinitions>

            <Border CornerRadius="10"
                Background="#475272">
                <Border.Effect>
                    <DropShadowEffect ShadowDepth="1"
                        BlurRadius="20"
                        Opacity="0.4"/>
                </Border.Effect>

                <Grid>
                    <StackPanel>
                        <Button Style="{StaticResource menuButton}"
                            Margin="15 40 15 20"
                            Background="White">
                            <Button.Triggers>
                                <EventTrigger RoutedEvent="Button.Click">
                                    <BeginStoryboard>
                                        <Storyboard>
```

```

<DoubleAnimation Storyboard.TargetName="page1"
    Storyboard.TargetProperty="(Border.RenderTransform).(TranslateTransformY)"
    To="0"
    Duration="0:0:0.5"/>

<DoubleAnimation Storyboard.TargetName="page2"
    Storyboard.TargetProperty="(Border.RenderTransform).(TranslateTransformY)"
    To="{Binding ElementName=mainWindow, Path=ActualHeight}"
    Duration="0:0:0.3"/>

<DoubleAnimation Storyboard.TargetName="page1"
    Storyboard.TargetProperty="Opacity"
    To="1.0"
    Duration="0:0:0.5"/>

<DoubleAnimation Storyboard.TargetName="page2"
    Storyboard.TargetProperty="Opacity"
    To="0.0"
    Duration="0:0:0.5"/>

        </Storyboard>
    </BeginStoryboard>
</EventTrigger>
</Button.Triggers>
<materialDesign:PackIcon Kind="Home"
    Foreground="#475272"
    Height="25"
    Width="25"/>
</Button>

<Button Style="{StaticResource menuButton}">

    <Button.Triggers>
        <EventTrigger RoutedEvent="Button.Click">
            <BeginStoryboard>
                <Storyboard>
                    <DoubleAnimation Storyboard.TargetName="page1"
                        Storyboard.TargetProperty="(Border.RenderTransform).(TranslateTransformY)"
                        To="{Binding ElementName=mainWindow, Path=ActualHeight}"
                        Duration="0:0:0.3"/>

                    <DoubleAnimation Storyboard.TargetName="page2"
                        Storyboard.TargetProperty="(Border.RenderTransform).(TranslateTransformY)"
                        To="0"
                        Duration="0:0:0.5"/>

                    <DoubleAnimation Storyboard.TargetName="page1"
                        Storyboard.TargetProperty="Opacity"
                        To="0.0"
                        Duration="0:0:0.5"/>

                    <DoubleAnimation Storyboard.TargetName="page2"
                        Storyboard.TargetProperty="Opacity"
                        To="1.0"
                        Duration="0:0:0.5"/>

                </Storyboard>
            </BeginStoryboard>
        </EventTrigger>
    </Button.Triggers>

```

```
        </EventTrigger>
    </Button.Triggers>
    <materialDesign:PackIcon Kind="Bed"
        Foreground="White"
        Height="25"
        Width="25"/>
</Button>

<Button Style="{StaticResource menuButton}">
    <materialDesign:PackIcon Kind="Users"
        Foreground="White"
        Height="25"
        Width="25"/>
</Button>

<Button Style="{StaticResource menuButton}">
    <materialDesign:PackIcon Kind="Shield"
        Foreground="White"
        Height="25"
        Width="25"/>
</Button>
<Button Style="{StaticResource menuButton}">
    <materialDesign:PackIcon Kind="Location"
        Foreground="White"
        Height="25"
        Width="25"/>
</Button>

<Button Style="{StaticResource menuButton}">
    <materialDesign:PackIcon Kind="ChartArc"
        Foreground="White"
        Height="25"
        Width="25"/>
</Button>

<Button Style="{StaticResource menuButton}">
    <materialDesign:PackIcon Kind="Settings"
        Foreground="White"
        Height="25"
        Width="25"/>
</Button>
</StackPanel>

<Button Style="{StaticResource menuButton}"
    VerticalAlignment="Bottom"
    Margin="15 0 15 25">
    <materialDesign:PackIcon Kind="Power"
        Foreground="White"
        Height="25"
        Width="25"/>
</Button>
</Grid>
</Border>
```

```
<Border x:Name="page1"
        Style="{StaticResource borderPage}">
    <Border.RenderTransform>
        <TranslateTransform Y="0"/>
    </Border.RenderTransform>

    <Border CornerRadius="25">
        <Border.Background>
            <ImageBrush Stretch="Fill"
                        Opacity="0.6"
                        ImageSource="Images/rp1.jpg"/>
        </Border.Background>

        <Grid>
            <Grid.ColumnDefinitions>
                <ColumnDefinition Width="*"/>
                <ColumnDefinition Width="Auto"/>
            </Grid.ColumnDefinitions>

            <StackPanel VerticalAlignment="Center"
                        Margin="60 0 0 0">
                <TextBlock Text="Управление"
                           Foreground="#FFFFFF"
                           FontSize="22"/>
                <TextBlock Text="Спальня"
                           Foreground="#FFFFFF"
                           FontSize="36"
                           Margin="0 10"
                           FontWeight="Bold"
                           TextWrapping="Wrap"/>

                <TextBlock Text="Практическая работа № 30"
                           FontSize="32"
                           Foreground="#FFFFFF"/>
            </StackPanel>

            <StackPanel Grid.Column="1"
                        VerticalAlignment="Center"
                        Margin="0 0 50 0">
                <Grid>
                    <Grid.ColumnDefinitions>
                        <ColumnDefinition Width="*"/>
                        <ColumnDefinition Width="*"/>
                    </Grid.ColumnDefinitions>

                    <Grid.RowDefinitions>
                        <RowDefinition Height="*"/>
                        <RowDefinition Height="*"/>
                    </Grid.RowDefinitions>
                </Grid>
            </StackPanel>
        </Grid>
    </Border>

```

```
<uc:CardUserControl Grid.Row="0"
    Grid.Column="0"
    Margin="0 0 8 8"
    Title="Холодильник"
    ImageOff="Images/roff.png"
    ImageOn="Images/ron.png"/>

<uc:CardUserControl Grid.Row="0"
    Grid.Column="1"
    Margin="8 0 0 8"
    Title="Кондиционер"
    ImageOff="Images/koff.png"
    ImageOn="Images/kon.png"
    IsChecked="True"/>

<uc:CardUserControl Grid.Row="1"
    Grid.Column="0"
    Margin="0 8 8 0"
    Title="Плита"
    ImageOff="Images/soff.png"
    ImageOn="Images/son.png"
    IsChecked="True"/>

<uc:CardUserControl Grid.Row="1"
    Grid.Column="1"
    Margin="0 8 8 0"
    Title="Свет"
    ImageOff="Images/loff.png"
    ImageOn="Images/lon.png"/>
</Grid>

<uc:AddButtonUserControl
    Title="Подключить"
    Margin="0 50 0 0"
    Grid.Column="1"
    HorizontalAlignment="Center"/>

</StackPanel>
</Grid>
</Border>
</Border>

<Border x:Name="page2"
    Style="{StaticResource borderPage}">
<Border.RenderTransform>
    <TranslateTransform Y="{Binding ElementName=mainWindow, Path=ActualHeight}"/>
</Border.RenderTransform>

<Border CornerRadius="25">
    <Border.Background>
        <ImageBrush Stretch="Fill"
            Opacity="0.6"
            ImageSource="Images/p1.jpg"/>
    </Border.Background>

```

```
<Grid>
    <Grid.RowDefinitions>
        <RowDefinition Height="*"/>
        <RowDefinition Height="Auto"/>
    </Grid.RowDefinitions>

    <uc:AddButtonUserControl
        Title="Подключить"
        Margin="0 40 40 0"/>

    <StackPanel VerticalAlignment="Center"
                HorizontalAlignment="Center"
                Margin="0 40 0 0">
        <TextBlock Text="Управление"
                   Foreground="#FFFFFF"
                   FontSize="22"/>
        <TextBlock Text="Кухня и Гостиная"
                   Foreground="#FFFFFF"
                   FontSize="36"
                   Margin="0 10"
                   FontWeight="Bold"
                   TextWrapping="Wrap"/>
        <TextBlock Text="Практическая работа № 30"
                   FontSize="32"
                   Foreground="#FFFFFF"/>
    </StackPanel>

    <StackPanel Grid.Row="1"
                HorizontalAlignment="Left"
                Margin="30 0 0 15">
        <Grid>
            <Grid.ColumnDefinitions>
                <ColumnDefinition Width="*"/>
                <ColumnDefinition Width="*"/>
            </Grid.ColumnDefinitions>

            <Grid.RowDefinitions>
                <RowDefinition Height="*"/>
                <RowDefinition Height="*"/>
            </Grid.RowDefinitions>

            <uc:CardUserControl Grid.Row="0"
                                Grid.Column="0"
                                Margin="0 0 8 8"
                                Title="Холодильник"
                                ImageOff="Images/roff.png"
                                ImageOn="Images/ron.png"/>
        </Grid>
    </StackPanel>

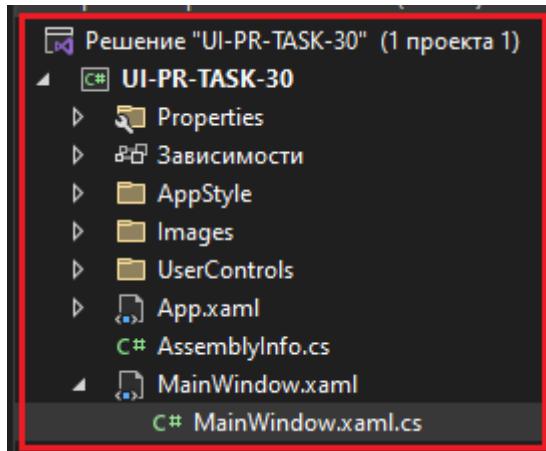
```

```
<uc:CardUserControl Grid.Row="0"
    Grid.Column="1"
    Margin="8 0 0 8"
    Title="Кондиционер"
    ImageOff="Images/koff.png"
    ImageOn="Images/kon.png"
    IsChecked="True"/>

<uc:CardUserControl Grid.Row="1"
    Grid.Column="0"
    Margin="0 8 8 0"
    Title="Плита"
    ImageOff="Images/soff.png"
    ImageOn="Images/son.png"
    IsChecked="True"/>

<uc:CardUserControl Grid.Row="1"
    Grid.Column="1"
    Margin="0 8 8 0"
    Title="Свет"
    ImageOff="Images/loff.png"
    ImageOn="Images/lon.png"/>

</Grid>
</StackPanel>
</Grid>
</Border>
</Border>
</Grid>
</Border>
</Window>
```



```
Ссылок: 4
public partial class MainWindow : Window
{
    private bool _isMaximized = false;
    Ссылок: 0
    public MainWindow()
    {
        InitializeComponent();
    }

    Ссылок: 1
    private void Border_MouseDown(object sender, MouseButtonEventArgs e)
    {
        if (e.ChangedButton == MouseButton.Left)
        {

            DragMove();
        }
    }

    Ссылок: 1
    private void Border_MouseLeftButtonDown(object sender, MouseButtonEventArgs e)
    {
        if (e.ClickCount == 2)
        {
            if (_isMaximized)
            {
                WindowState = WindowState.Normal;
                Width = 1080;
                Height = 720;

                _isMaximized = false;
            }
            else
            {
                WindowState = WindowState.Maximized;

                _isMaximized = true;
            }
        }
    }
}
```