**Занятие № 25**

**Дата выполнения работы:** 19.05.2023

# **Тема работы:** «Разработка отладка и испытание программ с

# пользовательским интерфейсом WPF»

# **Ход работы**

**Задание 1**

Создать приложение WPF, содержащее три кнопки, при нажатии на которые кнопки должны исчезать с анимацией изменения формы.

**Листинг программы:**

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

DoubleAnimation buttonAnimation = new DoubleAnimation();

buttonAnimation.From = btn.ActualWidth;

buttonAnimation.To = 150;

buttonAnimation.Duration = TimeSpan.FromSeconds(1);

btn.BeginAnimation(Button.WidthProperty, buttonAnimation);

if (btn.ActualWidth == 150)

{

grid.Children.Remove(btn);

}

}

private void Button\_Click\_1(object sender, RoutedEventArgs e)

{

DoubleAnimation buttonAnimation = new DoubleAnimation();

buttonAnimation.From = btn1.ActualHeight;

buttonAnimation.To = 100;

buttonAnimation.Duration = TimeSpan.FromSeconds(0.7);

btn1.BeginAnimation(Button.HeightProperty, buttonAnimation);

if (btn1.Height == 100)

{

grid.Children.Remove(btn1);

}

}

private void Button\_Click\_2(object sender, RoutedEventArgs e)

{

DoubleAnimation buttonAnimation = new DoubleAnimation();

buttonAnimation.From = btn2.ActualHeight;

buttonAnimation.To = 100;

buttonAnimation.Duration = TimeSpan.FromSeconds(0.5);

btn2.BeginAnimation(Button.HeightProperty, buttonAnimation);

btn2.BeginAnimation(Button.WidthProperty, buttonAnimation);

if (btn2.ActualHeight == 100)

{

grid.Children.Remove(btn2);

}

}

}

<Window x:Class="zd24.MainWindow"

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

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

Title="MainWindow" Height="350" Width="525">

<Window.Resources>

<Storyboard x:Key="collapseStoryboard">

<DoubleAnimation Storyboard.TargetProperty="RenderTransform.ScaleX"

To="0.5" Duration="0:0:0.1" />

<DoubleAnimation Storyboard.TargetProperty="RenderTransform.ScaleY"

To="0.5" Duration="0:0:0.1" />

</Storyboard>

<Storyboard x:Key="expandStoryboard">

<DoubleAnimation Storyboard.TargetProperty="RenderTransform.ScaleX"

To="1" Duration="0:0:0.1" />

<DoubleAnimation Storyboard.TargetProperty="RenderTransform.ScaleY"

To="1" Duration="0:0:0.1" />

</Storyboard>

</Window.Resources>

<Grid x:Name="grid">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="0\*"/>

<ColumnDefinition Width="211\*"/>

<ColumnDefinition Width="307\*"/>

</Grid.ColumnDefinitions>

<Button Name="btn" Content="Кнопка 1&#xD;&#xA;" HorizontalAlignment="Left" Margin="188,50,0,0" VerticalAlignment="Top" Width="105" Click="Button\_Click" Grid.ColumnSpan="3" Height="46"/>

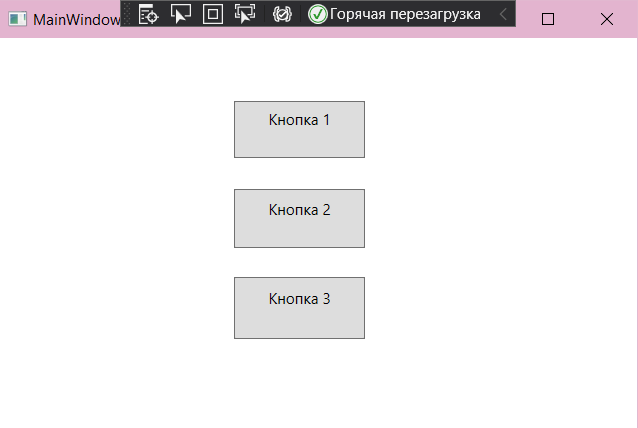
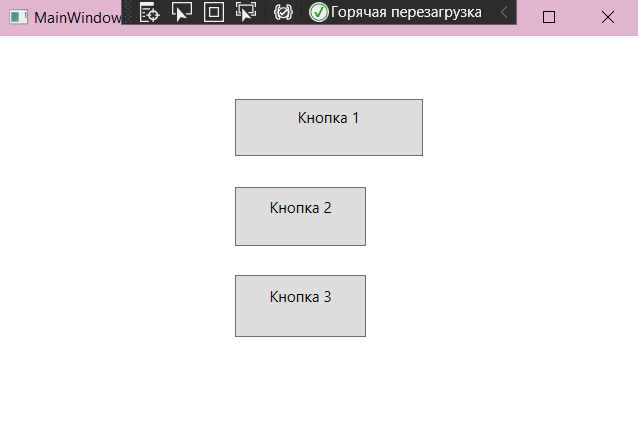
<Button Name="btn1" Content="Кнопка 2&#xD;&#xA;" Grid.Column="1" HorizontalAlignment="Left" Margin="188,121,0,0" VerticalAlignment="Top" Width="105" Grid.ColumnSpan="2" Height="47" Click="Button\_Click\_1"/>

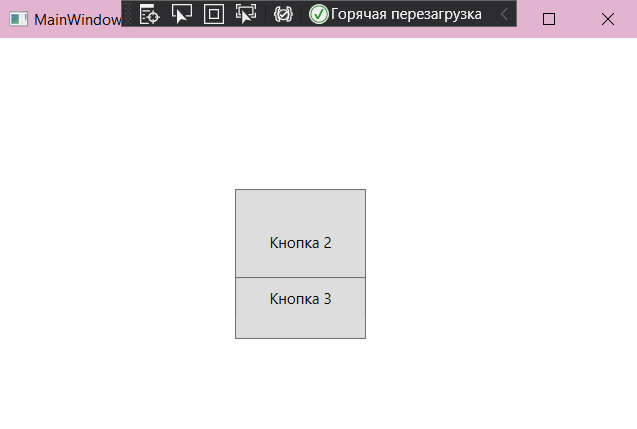
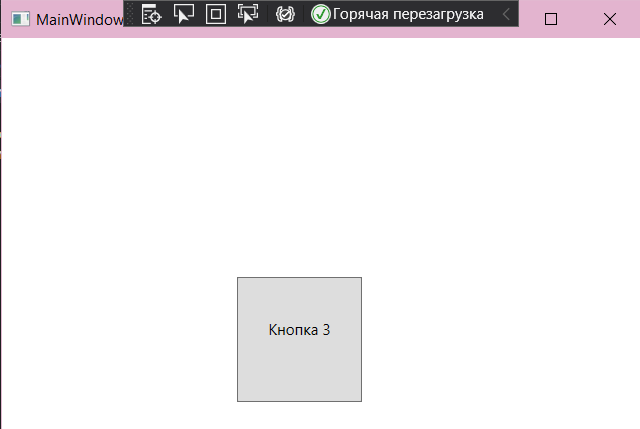
<Button Name="btn2" Content="Кнопка 3&#xD;&#xA;" Grid.Column="1" HorizontalAlignment="Left" Margin="188,191,0,0" VerticalAlignment="Top" Width="105" Grid.ColumnSpan="2" Height="50" Click="Button\_Click\_2"/>

</Grid>

</Window>

**Результат:**



** **

**Задание 2**

Создать приложение WPF, позволяющее отображать анимацию положения солнца в текущий момент времени.

**Листинг программы:**

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

MyImageControl.Source = new BitmapImage(new Uri(@"C:\Users\Батон\Desktop\Практика\images.jpg"));

MyImageControl1.Source = new BitmapImage(new Uri(@"C:\Users\Батон\Desktop\Практика\imagesss.jpg"));

MyImageControl.RenderTransform = transform;

MyImageControl1.RenderTransform = transform;

}

bool b = true;

DoubleAnimation anim = new DoubleAnimation();

TranslateTransform transform = new TranslateTransform();

private void Button\_Click\_1(object sender, RoutedEventArgs e)

{

if (b)

{

anim.To = 500;

b = false;

}

else

{

anim.To = 0;

b = true;

}

transform.BeginAnimation(TranslateTransform.XProperty, anim);

}

}

<Window x:Class="WpfApplication17.MainWindow"

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

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

Title="MainWindow"

Height="366.519"

Width="531.195">

<Grid>

<Button Content="Утро&#xD;&#xA;Ночь"

HorizontalAlignment="Left"

Margin="174,288,0,0"

VerticalAlignment="Top"

Width="166"

Height="39"

Click="Button\_Click\_1" />

<Image x:Name="MyImageControl"

HorizontalAlignment="Left"

Height="223"

Margin="39,47,0,0"

VerticalAlignment="Top"

Width="441" />

<Image x:Name="MyImageControl1"

HorizontalAlignment="Left"

Height="223"

Margin="-364,47,0,0"

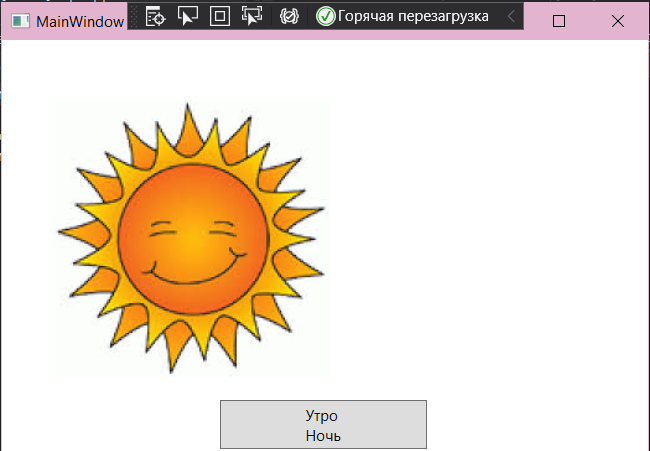
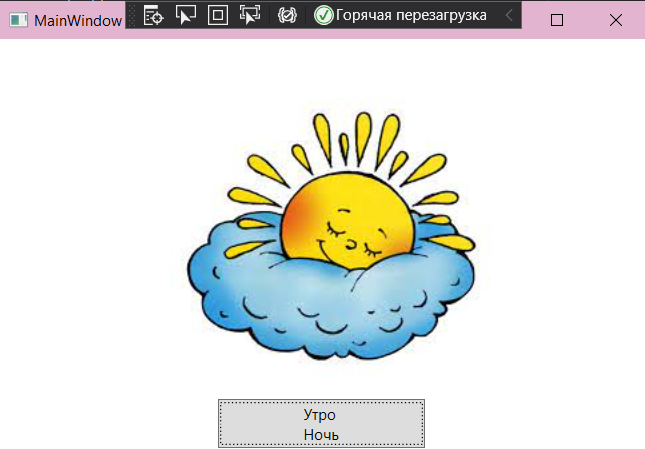
VerticalAlignment="Top"

Width="441" />

</Grid>

</Window>

**Результат:**

****

**Вывод:** Разработка программ с пользовательским интерфейсом WPF требует проектирования интерфейса, разработки логики приложения на C#, отладки и тестирования. Внимание к деталям и систематический подход помогут создать стабильное и интуитивно понятное приложение для Windows.