**Министерство образования и науки Забайкальского края   
Государственное профессиональное образовательное учреждение  
«ЧИТИНСКИЙ ТЕХНИКУМ ОТРАСЛЕВЫХ ТЕХНОЛОГИЙ И БИЗНЕСА»  
(ГПОУ «ЧТОТиБ»)**

УТВЕРЖДАЮ  
Зам. директора по учебной работе  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_А.С. Варфоломеева  
«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2023 г.

**ОТЧЕТ ПО УЧЕБНОЙ ПРАКТИКЕ**

**УП.01 Разработка модулей программного обеспечения для компьютерных систем**

|  |  |
| --- | --- |
|  | Выполнил: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Поляков Р.Р,  студент группы ИСиП 21-1п  «\_\_\_» \_\_\_\_\_\_\_\_\_\_2023 г.  Проверил работу:  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Степанова Э.П.  «\_\_\_» \_\_\_\_\_\_\_\_\_\_2023 г. |

# Теоретическая часть

## 1.1 Введение

Технология WPF (Windows Presentation Foundation) является часть экосистемы платформы .NET и представляет собой подсистему для построения графических интерфейсов.

Если при создании традиционных приложений на основе WinForms за отрисовку элементов управления и графики отвечали такие части ОС Windows, как User32 и GDI+, то приложения WPF основаны на **DirectX**. В этом состоит ключевая особенность рендеринга графики в WPF: используя WPF, значительная часть работы по отрисовке графики, как простейших кнопочек, так и сложных 3D-моделей, ложиться на графический процессор на видеокарте, что также позволяет воспользоваться аппаратным ускорением графики.

Одной из важных особенностей является использование языка декларативной разметки интерфейса XAML, основанного на XML: вы можете создавать насыщенный графический интерфейс, используя или декларативное объявление интерфейса, или код на управляемых языках C#, VB.NET и F#, либо совмещать и то, и другое.

Первая версия - WPF 3.0 вышла вместе с .NET Framework 3.0 и операционной системой Windows Vista в 2006 году. И с тех пор платформа WPF является частью экосистемы .NET и развивается вместе с фреймворком .NET. Например, на сегодняшний день последней версией фреймворка .NET является .NET 8, и WPF полностью поддерживается этой версией фреймворка.

## 1.2 Преимущества WPF

Что вам, как разработчику, предлагает WPF?

* Использование традиционных языков .NET-платформы - C#, F# и VB.NET для создания логики приложения
* Возможность **декларативного определения** графического интерфейса с помощью специального языка разметки XAML, основанном на xml и представляющем альтернативу программному созданию графики и элементов управления, а также возможность комбинировать XAML и C#/VB.NET
* **Независимость от разрешения экрана**: поскольку в WPF все элементы измеряются в независимых от устройства единицах, приложения на WPF легко масштабируются под разные экраны с разным разрешением.
* Новые возможности, которых сложно было достичь в WinForms, например, создание трехмерных моделей, привязка данных, использование таких элементов, как стили, шаблоны, темы и др.
* Хорошее **взаимодействие с WinForms**, благодаря чему, например, в приложениях WPF можно использовать традиционные элементы управления из WinForms.
* **Богатые возможности** по созданию различных приложений: это и мультимедиа, и двухмерная и трехмерная графика, и богатый набор встроенных элементов управления, а также возможность самим создавать новые элементы, создание анимаций, привязка данных, стили, шаблоны, темы и многое другое
* **Аппаратное ускорение графики** - вне зависимости от того, работаете ли вы с 2D или 3D, графикой или текстом, все компоненты приложения транслируются в объекты, понятные Direct3D, и затем визуализируются с помощью процессора на видеокарте, что повышает производительность, делает графику более плавной.
* Создание приложений под множество ОС семейства Windows

В тоже время WPF имеет определенные ограничения. Несмотря на поддержку трехмерной визуализации, для создания приложений с большим количеством трехмерных изображений, прежде всего игр, лучше использовать другие средства - DirectX или специальные фреймворки, такие как Monogame или Unity.

Также стоит учитывать, что по сравнению с приложениями на Windows Forms объем программ на WPF и потребление ими памяти в процессе работы в среднем несколько выше. Но это с лихвой компенсируется более широкими графическими возможностями и повышенной производительностью при отрисовке графики.

Кроме того, несмотря на то, что WPF работает поверх кроссплатформенной среды .NET, но в силу природы WPF и зависимости от компонентов Windows, на данный момент создавать приложения на WPF можно только под ОС Windows.

1.3 XAML

(eXtensible Application Markup Language) - язык разметки, используемый для инициализации объектов в технологиях на платформе .NET. Применительно к WPF (а также к Silverlight) данный язык используется прежде всего для создания пользовательского интерфейса декларативным путем. Хотя функциональность XAML только графическими интерфейсами не ограничивается: данный язык также используется в технологиях WCF и WF, где он никак не связан с графическим интерфейсом. То есть его область шире. Применительно к WPF мы будем говорить о нем чаще всего именно как о языке разметки, который позволяет создавать декларативным путем интерфейс, наподобие HTML в веб-программировании. Однако опять же повторюсь, сводить XAML к одному интерфейсу было бы неправильно, и далее на примерах мы это увидим.

XAML - не является обязательной частью приложения, мы вобще можем обходиться без него, создавая все элементы в файле связанного с ним кода на языке C#. Однако использование XAML все-таки несет некоторые преимущества:

* Возможность отделить графический интерфейс от логики приложения, благодаря чему над разными частями приложения могут относительно автономно работать разные специалисты: над интерфейсом - дизайнеры, над кодом логики - программисты.
* Компактность, понятность, код на XAML относительно легко поддерживать.

При компиляции приложения в Visual Studio код в xaml-файлах также компилируется в бинарное представление кода xaml, которое называется BAML (Binary Application Markup Language). И затем код baml встраивается в финальную сборку приложения - exe или dll-файл.

## 1.4 Пространства имен XAML

При создании кода на языке C#, чтобы нам были доступны определенные классы, мы подключаем пространства имен с помощью директивы using, например, using System.Windows;.

Чтобы задействовать элементы в XAML, мы также подключаем пространства имен. Вторая и третья строчки как раз и представляют собой пространства имен, подключаемые в проект по умолчанию. А атрибут **xmlns** представляет специальный атрибут для определения пространства имен в XML.

2. Практическая часть

Приложение EventClick: Квадрат меняет цвет после воспроизведения эвента “нажатия на объект”

2.1 EventClick

Код XAML

<Window x:Class="EventMouse.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:EventMouse"

mc:Ignorable="d"

Title="Установка свойства события мыши" Height="450" Width="800">

<StackPanel>

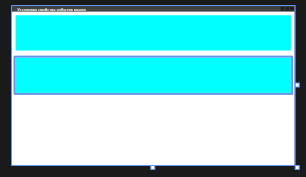
<Rectangle MinHeight="100" MinWidth="100" Fill="Aqua" MouseEnter="Rectange\_MouseEnter" MouseLeave="Rectangle\_MouseLeave" Margin="10"> </Rectangle>

<Rectangle MinHeight="100" MinWidth="100" Fill="Aqua" MouseEnter="Rectange\_MouseEnter" MouseLeave="Rectangle\_MouseLeave" Margin="10"></Rectangle>

</StackPanel>

</Window>

Рисунок 1 - XAML



CS файл

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 EventMouse

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

private void Rectange\_MouseEnter(object sender, MouseEventArgs e)

{

(sender as Rectangle).Fill = Brushes.Red;

}

private void Rectangle\_MouseLeave(object sender, MouseEventArgs e)

{

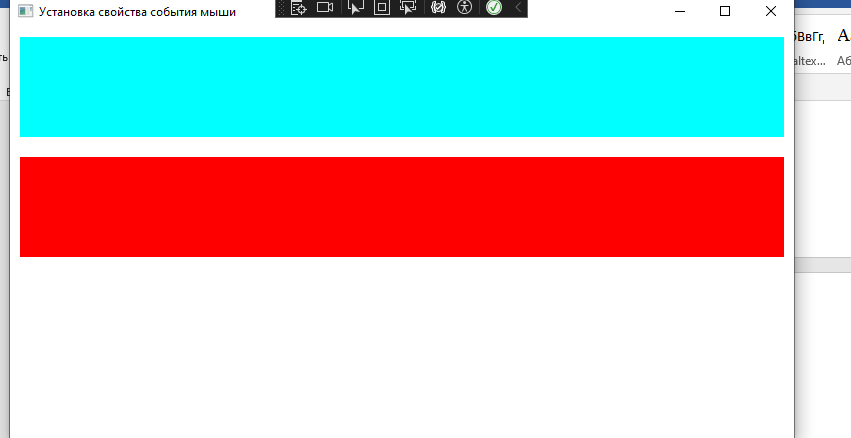
(sender as Rectangle).Fill = Brushes.Aqua;

}

}

}

Рисунок 3 – Работа приложения



2.2 MagicEightBall

MagicEightBall – Магический шар: Работает по принципу вызова заранее написанного ответа, который выбирается случайно кодом. Простейшая функция которая позволяет написать свой текст и получить на него ответ

Код XAML

<Window x:Class="WpfApp3.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:WpfApp3"

mc:Ignorable="d"

Title="Магический шар" Height="350" Width="525">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="324"/>

<ColumnDefinition Width="193"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="140"/>

<RowDefinition Height="auto"/>

<RowDefinition Height="109"/>

</Grid.RowDefinitions>

<Grid.Background>

<LinearGradientBrush EndPoint="0.5, 1" StartPoint="0.5, 0">

<LinearGradientBrush.GradientStops>

<GradientStop Color="Black" Offset="0.299"/>

<GradientStop Color="White" Offset="1"/>

</LinearGradientBrush.GradientStops>

</LinearGradientBrush>

</Grid.Background>

<TextBox Name="txtQuest"

HorizontalAlignment="Stretch"

VerticalAlignment="Stretch"

Margin="10"

TextWrapping="Wrap"

FontFamily="Verdana"

FontSize="20"

Grid.Row="0"

Foreground="#FF646464"

Text="[Введите вопрос.]"

Grid.ColumnSpan="2"/>

<Button HorizontalAlignment="Left"

VerticalAlignment="Center"

Margin="10"

Padding="5"

Grid.Row="1"

Content="Ответить на вопрос"

Click="AnswerClick"

IsDefault="True"/>

<TextBox Name="txtAnswer"

HorizontalAlignment="Stretch"

VerticalAlignment="Stretch"

Margin="10"

TextWrapping="Wrap"

FontFamily="Verdana"

FontSize="20"

IsReadOnly="True"

Grid.Row="2"

Foreground="Blue"

Text="[Ответ тут]"

Grid.ColumnSpan="2"/>

</Grid>

</Window>

Код CS

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

private void AnswerClick(Object sender, RoutedEventArgs e)

{

this.Cursor = Cursors.Wait;

Thread.Sleep(1000);

txtAnswer.Text = answerGenerated.GetRandomAnswer();

this.Cursor = null;

}

}

}

Второй CS файл

namespace WpfApp3

{

public static class answerGenerated

{

public static string[] answer = new string[]

{

"Спросите еще раз",

"еще раз",

"без сомнений",

"нет",

"да",

"ммм, действительно",

"узнайте от себя",

"Нужно больше данных",

"Следущий вопрос",

"Правильно",

"Угадал",

"Неправильно",

"Мой ответ нет"

};

public static string GetRandomAnswer()

{

return answer[new Random().Next(answer.Length)];

}

}

}

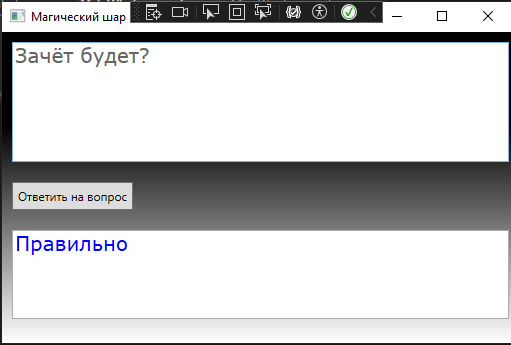


Рисунок – Работа приложения

2.3 Calculator

Простейший калькулятор, работает по принципу обычного декстопного

Калькулятора, работают все функции обычного приложения.

Рисунок 7 - XAML



Рисунок 8 – Код CS

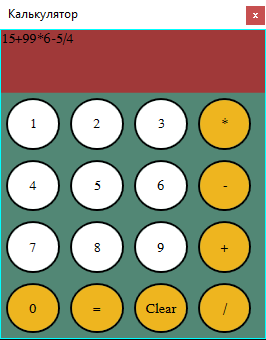
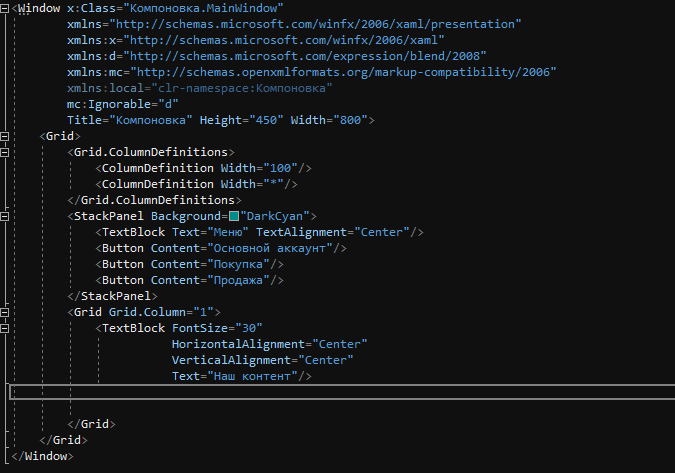


Рисунок 9 – Работа приложения

[C# и WPF | XAML (metanit.com)](https://metanit.com/sharp/wpf/2.php)

2.4 Градиент

2 Квадрата один обычный, другой с градиентом

код - XAML

<Window x:Class="WpfApp1.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:WpfApp1"

mc:Ignorable="d"

Title="Ye" Height="550" Width="800">

<StackPanel>

<Rectangle Fill="Tomato" Width="200" Height="200" Margin="10" />

<Rectangle Width="200" Height="200" Margin="100">

<Rectangle.Fill>

<LinearGradientBrush>

<GradientStop Color="Chocolate" Offset="0.30" />

<GradientStop Color="Aqua" Offset="1" />

</LinearGradientBrush>

</Rectangle.Fill>

</Rectangle>

</StackPanel>

</Window>

Код CS

namespace WpfApp1

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

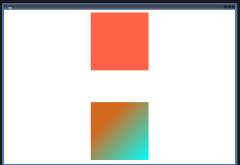
InitializeComponent();

}

}

}

Рисунок 9 – Работа приложения



2.5 Использование Стилей

Код XAMAL

<Window x:Class="WpfApp4.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:WpfApp4"

mc:Ignorable="d"

Title="MainWindow" Height="300" Width="300">

<Window.Resources>

<Style TargetType="Button">

<Setter Property="Background" Value="Red"/>

<Setter Property="Foreground" Value="White"/>

</Style>

<Style TargetType="Button" x:Key="default">

<Setter Property="Background" Value="Silver"/>

<Setter Property="Foreground" Value="Black"/>

</Style>

</Window.Resources>

<UniformGrid VerticalAlignment="Center" Columns="4">

<Button Margin="10"/>

<Button Margin="10" Style="{StaticResource default}"/>

<Button Margin="10" Background="Green" Foreground="White"/>

<Button Margin="10"/>

</UniformGrid>

</Window>

Код CS

namespace WpfApp4

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

}

}

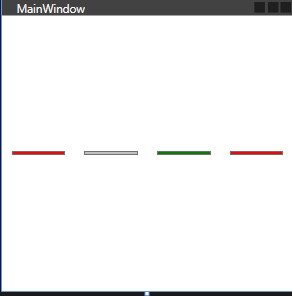


Рисунок приложения

2.6 Компановка

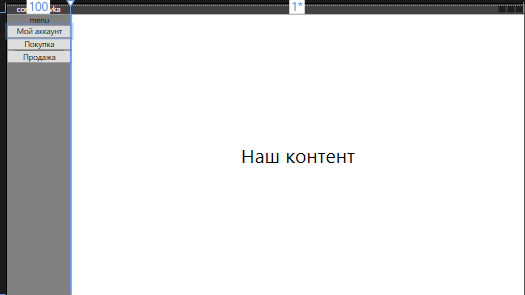


Рисунок приложения

Код XAMAL

<Window x:Class="WpfApp4companovka.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:WpfApp4companovka"

mc:Ignorable="d"

Title="companovka" Height="450" Width="800">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="100"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<StackPanel Background="Gray">

<TextBlock Text="menu" TextAlignment="Center"/>

<Button Content="Мой аккаунт"/>

<Button Content="Покупка"/>

<Button Content="Продажа"/>

</StackPanel>

<Grid Grid.Column="1">

<TextBlock FontSize="30" HorizontalAlignment="Center" VerticalAlignment="Center" Text="Наш контент"/>

</Grid>

</Grid>

</Window>

Код CS

namespace WpfApp4companovka

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

}

}

2.7 Кнопки с разноцветным текстом



Рисунок приложения

Код XAMAL

<Window x:Class="WpfApp5.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:WpfApp5"

mc:Ignorable="d"

Title="MainWindow" Height="450" Width="800">

<Window.Resources>

<Style TargetType="Button" x:Key="Base">

<Setter Property="Foreground" Value="Black"/>

</Style>

<Style TargetType="Button" x:Key="default" BasedOn="{StaticResource Base}">

<Setter Property="Foreground" Value="Blue"/>

</Style>

<Style TargetType="Button" x:Key="default2" BasedOn="{StaticResource Base}">

<Setter Property="Foreground" Value="Red"/>

</Style>

</Window.Resources>

<UniformGrid VerticalAlignment="Center" Columns="4">

<Button Margin="10"/>

<Button Margin="10" Style="{StaticResource default}" Content="Test Text2"/>

<Button Margin="10" Style="{StaticResource default2}" Content="Test Text2"/>

<Button Margin="10"/>

</UniformGrid>

</Window>

Код CS

namespace WpfApp5

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

}

}

2.8 Добавление имён

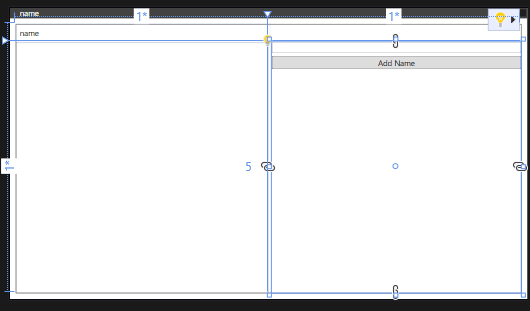


Рисунок приложения

<Window x:Class="WpfApp6.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:WpfApp6"

mc:Ignorable="d"

Title="name" Height="450" Width="800">

<Grid Margin="10">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="auto"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<Label>name</Label>

<ListBox Grid.Row="1" x:Name="LastName" />

<StackPanel Grid.Row="1" Grid.Column="1" Margin="5,0,0,0">

<TextBox x:Name="txtName"/>

<Button x:Name="btnAdd" Margin="0,5,0,0" Click="ButtonAddName">Add Name</Button>

</StackPanel>

</Grid>

</Window>

Код CS

namespace WpfApp6

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

private void ButtonAddName(object sender, RoutedEventArgs e)

{

if(!string.IsNullOrWhiteSpace(txtName.Text) && !LastName.Items.Contains(txtName.Text))

{

LastName.Items.Add(txtName.Text);

txtName.Clear();

}

}

}

}

2.9 Кнопки



Рисунок приложения

<Window x:Class="BTN1.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:BTN1"

mc:Ignorable="d"

Title="MainWindow" Height="300" Width="300" Background="Black">

<!-- Buttons arranged vertically inside a StackPanel. -->

<StackPanel HorizontalAlignment="Left">

<Button>Button 1</Button>

<Button>Button 2</Button>

<Button>Button 3</Button>

</StackPanel>

</Window>

Код CS

namespace BTN1

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

}

} {

LastName.Items.Add(txtName.Text);

txtName.Clear();

}

}

}

}

Код APP.xaml

<Application x:Class="BTN1.App"

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

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

xmlns:local="clr-namespace:BTN1"

StartupUri="MainWindow.xaml">

<Application.Resources>

<LinearGradientBrush x:Key="GrayBlueGradientBrush" StartPoint="0,0" EndPoint="1,1">

<GradientStop Color="DarkGray" Offset="0" />

<GradientStop Color="#CCCCFF" Offset="0.5" />

<GradientStop Color="DarkGray" Offset="1" />

</LinearGradientBrush>

<GradientStopCollection x:Key="MyGlassGradientStopsResource">

<GradientStop Color="WhiteSmoke" Offset="0.2" />

<GradientStop Color="Transparent" Offset="0.4" />

<GradientStop Color="WhiteSmoke" Offset="0.5" />

<GradientStop Color="Transparent" Offset="0.75" />

<GradientStop Color="WhiteSmoke" Offset="0.9" />

<GradientStop Color="Transparent" Offset="1" />

</GradientStopCollection>

<LinearGradientBrush x:Key="MyGlassBrushResource" StartPoint="0,0" EndPoint="1,1" Opacity="0" GradientStops="{StaticResource MyGlassGradientStopsResource}" />

<Style TargetType="{x:Type Button}">

<Setter Property="Background" Value="{StaticResource GrayBlueGradientBrush}"/>

<Setter Property="Width" Value="80" />

<Setter Property="Margin" Value="10" />

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="{x:Type Button}">

<Grid Width="{TemplateBinding Width}" Height="{TemplateBinding Height}" ClipToBounds="True">

<Rectangle x:Name="outerRectangle" HorizontalAlignment="Stretch" VerticalAlignment="Stretch" Stroke="{TemplateBinding Background}" RadiusX="20" RadiusY="20" StrokeThickness="5" Fill="Transparent" />

<Rectangle x:Name="innerRectangle" HorizontalAlignment="Stretch" VerticalAlignment="Stretch" Stroke="Transparent" StrokeThickness="20" Fill="{TemplateBinding Background}" RadiusX="20" RadiusY="20" />

<Rectangle x:Name="glassCube" HorizontalAlignment="Stretch" VerticalAlignment="Stretch" StrokeThickness="2" RadiusX="10" RadiusY="10" Opacity="0" Fill="{StaticResource MyGlassBrushResource}" RenderTransformOrigin="0.5,0.5">

<Rectangle.Stroke>

<LinearGradientBrush StartPoint="0.5,0" EndPoint="0.5,1">

<LinearGradientBrush.GradientStops>

<GradientStop Offset="0.0" Color="LightBlue" />

<GradientStop Offset="1.0" Color="Gray" />

</LinearGradientBrush.GradientStops>

</LinearGradientBrush>

</Rectangle.Stroke>

<Rectangle.RenderTransform>

<TransformGroup>

<ScaleTransform />

<RotateTransform />

</TransformGroup>

</Rectangle.RenderTransform>

<Rectangle.BitmapEffect>

<BevelBitmapEffect />

</Rectangle.BitmapEffect>

</Rectangle>

<DockPanel Name="myContentPresenterDockPanel">

<ContentPresenter x:Name="myContentPresenter" Margin="20" Content="{TemplateBinding Content}" TextBlock.Foreground="Black" />

</DockPanel>

</Grid>

<ControlTemplate.Triggers>

<Trigger Property="IsMouseOver" Value="True">

<Setter Property ="Rectangle.Stroke" TargetName="outerRectangle" Value="{DynamicResource {x:Static SystemColors.HighlightBrushKey}}" />

<Setter Property="Rectangle.Opacity" Value="1" TargetName="glassCube" />

<Setter Property="ContentPresenter.BitmapEffect" TargetName="myContentPresenter">

<Setter.Value>

<BlurBitmapEffect Radius="1"/>

</Setter.Value>

</Setter>

</Trigger>

<Trigger Property="IsFocused" Value="true">

<Setter Property="Rectangle.Opacity" Value="1" TargetName="glassCube" />

<Setter Property="Rectangle.Stroke" TargetName="outerRectangle" Value="{DynamicResource {x:Static SystemColors.HighlightBrushKey}}" />

<Setter Property="Rectangle.Opacity" Value="1" TargetName="glassCube" />

</Trigger>

<EventTrigger RoutedEvent="Mouse.MouseEnter">

<EventTrigger.Actions>

<BeginStoryboard Name="mouseEnterBeginStoryboard">

<Storyboard>

<DoubleAnimation Storyboard.TargetName="glassCube"

Storyboard.TargetProperty="(Rectangle.RenderTransform).(TransformGroup.Children)[0].(ScaleTransform.ScaleX)"

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

<DoubleAnimation Storyboard.TargetName="glassCube"

Storyboard.TargetProperty="(Rectangle.RenderTransform).(TransformGroup.Children)[0].(ScaleTransform.ScaleY)"

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

</Storyboard>

</BeginStoryboard>

</EventTrigger.Actions>

</EventTrigger>

<EventTrigger RoutedEvent="Mouse.MouseLeave">

<EventTrigger.Actions>

<StopStoryboard BeginStoryboardName="mouseEnterBeginStoryboard" />

</EventTrigger.Actions>

</EventTrigger>

<EventTrigger RoutedEvent="Button.Click">

<EventTrigger.Actions>

<BeginStoryboard>

<Storyboard>

<DoubleAnimation Storyboard.TargetName="glassCube"

Storyboard.TargetProperty=

"(Rectangle.RenderTransform).(TransformGroup.Children)[1].(RotateTransform.Angle)"

To="360" Duration="0:0:0.5" />

</Storyboard>

</BeginStoryboard>

</EventTrigger.Actions>

</EventTrigger>

</ControlTemplate.Triggers>

</ControlTemplate>

</Setter.Value>

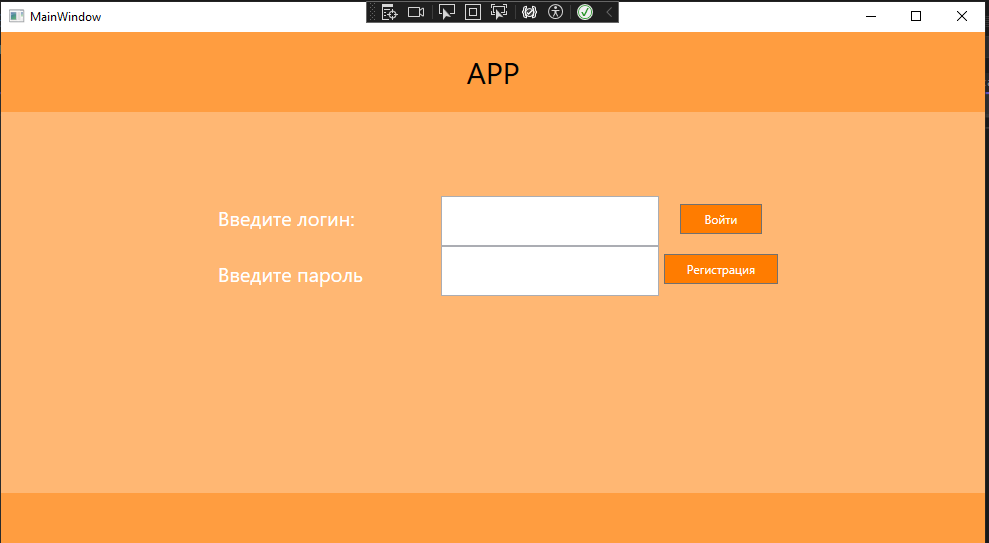
</Setter>

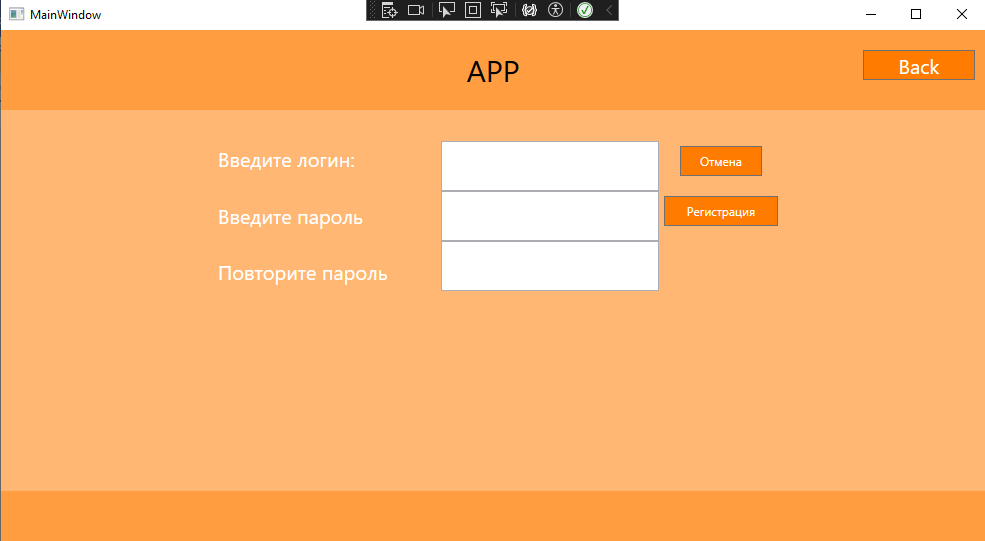
</Style>

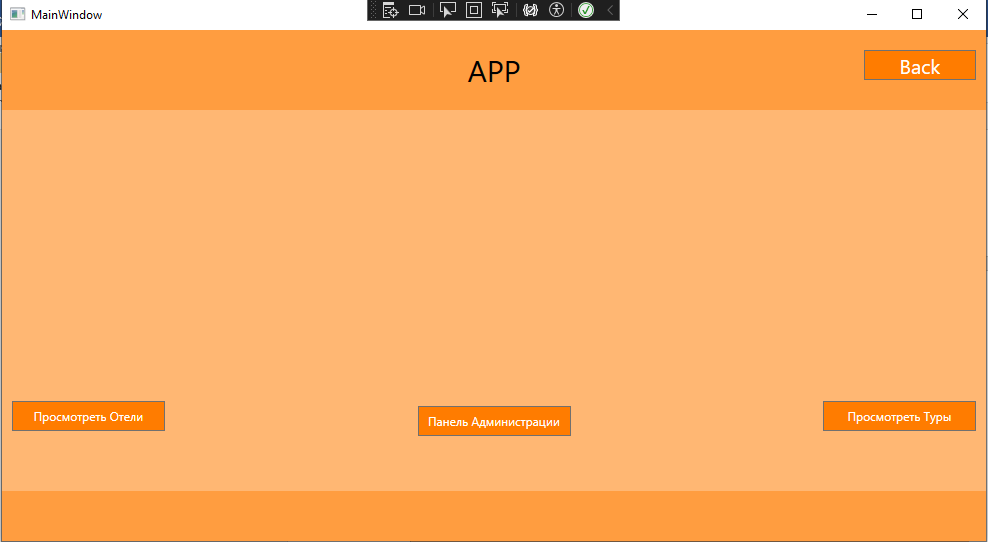
</Application.Resources>

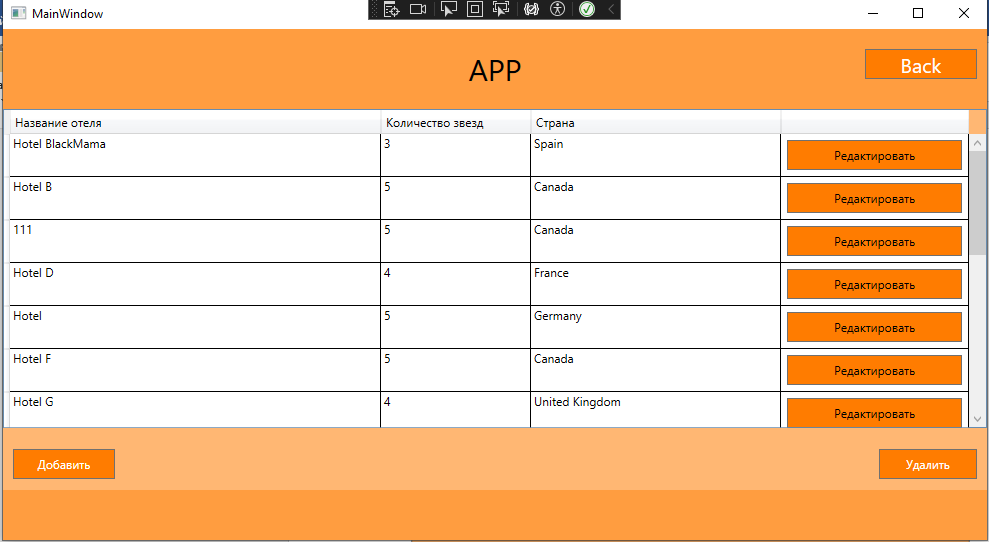
</Application>

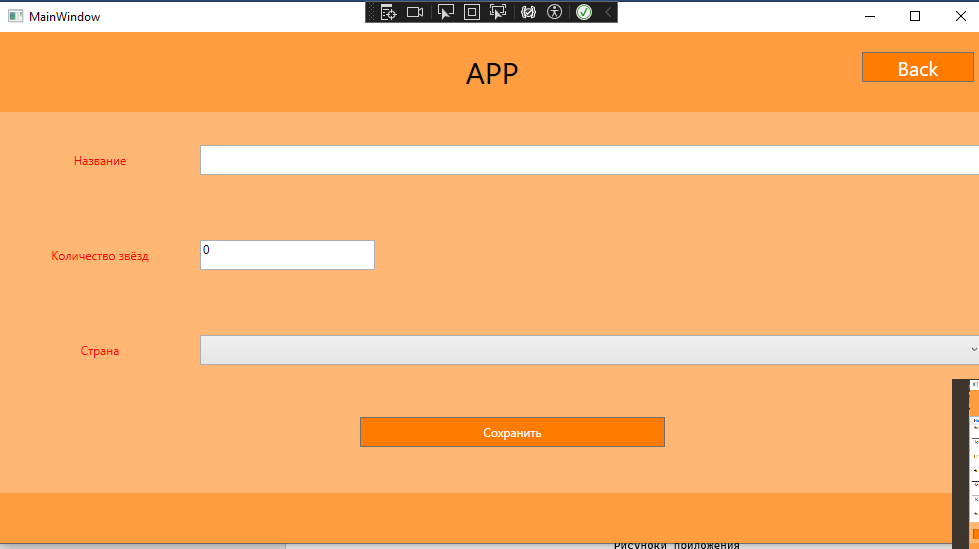
2.9 Финальное приложение с базой данных

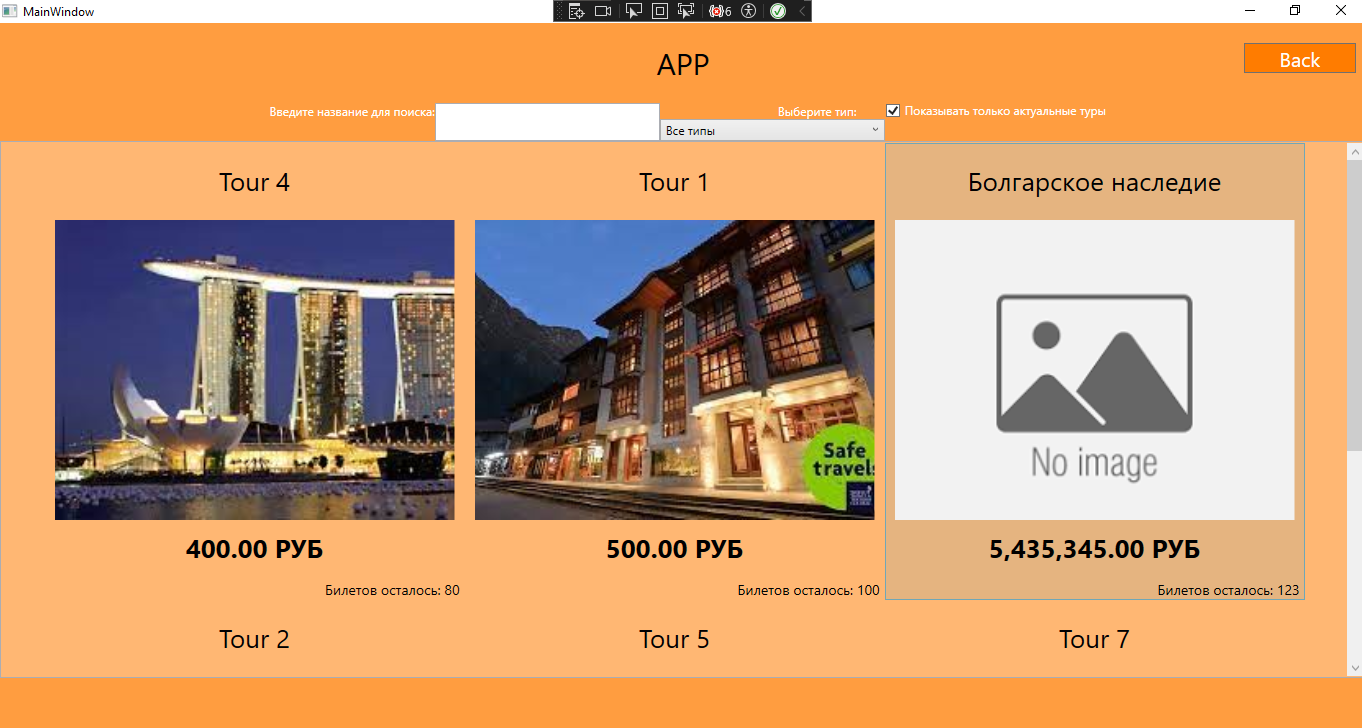




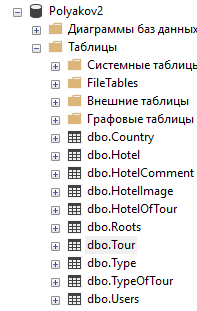


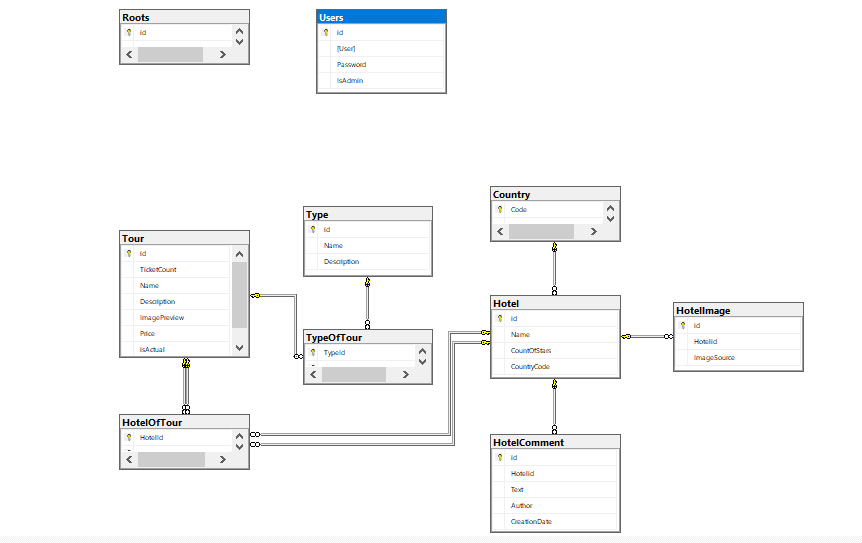






Рисунки приложения





База данных

XAML

<Window x:Class="MakeCarcas.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:MakeCarcas"

mc:Ignorable="d"

Title="MainWindow" Height="550" Width="1000">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="80"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="50"/>

</Grid.RowDefinitions>

<Image Source="Resources/123.png" HorizontalAlignment="Left"></Image>

<TextBlock Text="APP" FontSize="30" HorizontalAlignment="Center" VerticalAlignment="Center"></TextBlock>

<Grid Grid.Row="2" Background="#FF9D40"></Grid>

<Grid Background="#FF9D40" Panel.ZIndex="-2"></Grid>

<Frame NavigationUIVisibility="Hidden" ContentRendered="MainFrame\_ContentRendered" Grid.Row="1" Name="MainFrame" Background="#FFB773"></Frame>

<Button Foreground="White" Content="Back" Name="BtnBack" Width="112" HorizontalAlignment="Right" Click="BtnBack\_click" Margin="0,0,10,10" FontSize="20"/>

</Grid>

</Window>

XAML Логин

<Page x:Class="MakeCarcas.LoginPage"

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

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

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

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

xmlns:local="clr-namespace:MakeCarcas"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="LoginPage">

<Grid >

<Grid.RowDefinitions>

<RowDefinition/>

<RowDefinition/>

</Grid.RowDefinitions>

<StackPanel HorizontalAlignment="Center" Orientation="Horizontal" VerticalAlignment="Bottom" Grid.Row="0">

<StackPanel HorizontalAlignment="Center" Orientation="Vertical" VerticalAlignment="Center">

<Label Foreground="White" Content="Введите логин:" HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="20" Width="218" Margin="10"/>

<Label Foreground="White" Content="Введите пароль" VerticalAlignment="Center" HorizontalAlignment="Center" Width="218" FontSize="20" Margin="10"/>

</StackPanel>

<StackPanel HorizontalAlignment="Center" Orientation="Vertical" VerticalAlignment="Center">

<TextBox x:Name="login" HorizontalAlignment="Center" VerticalAlignment="Center" Width="218" Height="50" />

<PasswordBox x:Name="password" VerticalAlignment="Center" HorizontalAlignment="Center" Width="218" Height="50"/>

</StackPanel>

<StackPanel HorizontalAlignment="Center" Orientation="Vertical" VerticalAlignment="Top" Grid.Row="1">

<Button Foreground="White" x:Name="Loginer" Click="Loginer\_Click" Content="Войти" Width="82" Margin="15"/>

<Button Foreground="White" x:Name="Register" Click="Register\_Click" Content="Регистрация" Width="114"/>

</StackPanel>

</StackPanel>

</Grid>

</Page>

XAML Туры

<Page x:Class="MakeCarcas.TourPage"

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

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

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

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

xmlns:local="clr-namespace:MakeCarcas"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="TourPage">

<Grid Background="#FF9D40">

<Grid.RowDefinitions>

<RowDefinition Height="auto" />

<RowDefinition Height="\*" />

</Grid.RowDefinitions>

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

<StackPanel Orientation="Horizontal">

<TextBlock Foreground="White" Text="Введите название для поиска:" Width="175" TextAlignment="Right" />

<TextBox Width="225" Name="TBoxSearch" TextChanged="TBoxSearch\_TextChanged" />

</StackPanel>

<StackPanel>

<TextBlock Foreground="White" Text="Выберите тип: " Width="175" TextAlignment="Right" />

<ComboBox Width="225" Name="ComboType" SelectionChanged="ComboType\_SelectionChanged" DisplayMemberPath="Name" />

</StackPanel>

<CheckBox Foreground="White" x:Name="CheckActual" Checked="CheckActual\_Checked" Unchecked="CheckActual\_Checked" Content="Показывать только актуальные туры"

HorizontalAlignment="Center" />

</WrapPanel>

<ListView Grid.Row="1" Name="LViewTours" ScrollViewer.HorizontalScrollBarVisibility="Disabled" HorizontalContentAlignment="Center" Background="#FFB773">

<ListView.ItemsPanel>

<ItemsPanelTemplate>

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

</ItemsPanelTemplate>

</ListView.ItemsPanel>

<ListView.ItemTemplate>

<DataTemplate>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="70" />

<RowDefinition Height="310" />

<RowDefinition Height="auto" />

<RowDefinition Height="auto" />

</Grid.RowDefinitions>

<Image Width="400" Grid.Row="1" Stretch="UniformToFill" HorizontalAlignment="Center" Margin="5">

<Image.Source>

<Binding Path="ImagePreview">

<Binding.TargetNullValue>

<ImageSource>Resources/MissingPage.png</ImageSource>

</Binding.TargetNullValue>

</Binding>

</Image.Source>

</Image>

<TextBlock Text="{Binding Name}" VerticalAlignment="Center" TextAlignment="Center" Width="390" TextWrapping="Wrap" HorizontalAlignment="Center"

Margin="5 5" FontSize="26" Grid.Row="0" />

<TextBlock Text="{Binding Price, StringFormat={}{0:N2} РУБ}" Grid.Row="2" Margin="5 5 5 15" HorizontalAlignment="Center" FontSize="26" FontWeight="Bold" />

<TextBlock Text="{Binding TicketCount, StringFormat={}Билетов осталось: {0}}" Grid.Row="3" FontSize="14" HorizontalAlignment="Right" />

<TextBlock Text="{Binding ActualText}" Grid.Row="3" FontSize="14" HorizontalAlignment="Left" />

</Grid>

</DataTemplate>

</ListView.ItemTemplate>

</ListView>

</Grid>

</Page>

XAML Регистрация

<Page x:Class="MakeCarcas.RegisterPage"

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

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

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

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

xmlns:local="clr-namespace:MakeCarcas"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="RegisterPage">

<Grid >

<Grid.RowDefinitions>

<RowDefinition/>

<RowDefinition/>

</Grid.RowDefinitions>

<StackPanel HorizontalAlignment="Center" Orientation="Horizontal" VerticalAlignment="Bottom" Grid.Row="0">

<StackPanel HorizontalAlignment="Center" Orientation="Vertical" VerticalAlignment="Center">

<Label Foreground="White" Content="Введите логин:" HorizontalAlignment="Center" VerticalAlignment="Center" FontSize="20" Width="218" Margin="10"/>

<Label Foreground="White" Content="Введите пароль" VerticalAlignment="Center" HorizontalAlignment="Center" Width="218" FontSize="20" Margin="10"/>

<Label Foreground="White" Content="Повторите пароль" VerticalAlignment="Center" HorizontalAlignment="Center" Width="218" FontSize="20" Margin="10"/>

</StackPanel>

<StackPanel HorizontalAlignment="Center" Orientation="Vertical" VerticalAlignment="Center">

<TextBox x:Name="login" HorizontalAlignment="Center" VerticalAlignment="Center" Width="218" Height="50" />

<PasswordBox x:Name="password" VerticalAlignment="Center" HorizontalAlignment="Center" Width="218" Height="50"/>

<PasswordBox x:Name="password\_copy" VerticalAlignment="Center" HorizontalAlignment="Center" Width="218" Height="50"/>

</StackPanel>

<StackPanel HorizontalAlignment="Center" Orientation="Vertical" VerticalAlignment="Top" Grid.Row="1">

<Button Foreground="White" x:Name="Cancel" Click="Cancel\_Click" Content="Отмена" Width="82" Margin="15"/>

<Button Foreground="White" x:Name="Register" Click="Register\_Click" Content="Регистрация" Width="114"/>

</StackPanel>

</StackPanel>

</Grid>

</Page>

XAML Окно выбора других окон

<Page x:Class="MakeCarcas.IntermediatePage"

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

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

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

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

xmlns:local="clr-namespace:MakeCarcas"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="IntermediatePage">

<Grid Background="#FFB773">

<Grid.RowDefinitions>

<RowDefinition Height="75"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="50"/>

</Grid.RowDefinitions>

<Button Foreground="White" Content="Просмотреть Отели" Grid.Row="1" HorizontalAlignment="Left" VerticalAlignment="Bottom" Name="BtnHotel" Click="BtnHotel\_click" Margin="10,0,0,10" Width="153"></Button>

<Button Foreground="White" Content="Панель Администрации" Grid.Row="1" HorizontalAlignment="Center" VerticalAlignment="Bottom" Name="BtnAdminPanel" Click="BtnAdminPanel\_Click" Visibility="Hidden" Width="153"></Button>

<Button Foreground="White" Content="Просмотреть Туры" Grid.Row="1" HorizontalAlignment="Right" VerticalAlignment="Bottom" Name="BtnTour" Click="BtnTour\_click" Margin="0,0,10,10" Width="153"></Button>

</Grid>

</Page>

XAML Отели

<Page x:Class="MakeCarcas.HotelPage"

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

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

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

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

xmlns:local="clr-namespace:MakeCarcas"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="HotelPage" IsVisibleChanged="Page\_IsVisibleChanged">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="377\*"/>

<RowDefinition Height="73\*"/>

</Grid.RowDefinitions>

<DataGrid x:Name="DGridHotels" AutoGenerateColumns="False" IsReadOnly="True" SelectionChanged="DGridHotels\_SelectionChanged" Background="#FFB773">

<DataGrid.Columns>

<DataGridTextColumn Header="Название отеля" Binding="{Binding Name}" Width="\*" ></DataGridTextColumn>

<DataGridTextColumn Header="Количество звезд" Binding="{Binding CountOfStars}" Width="150"></DataGridTextColumn>

<DataGridTextColumn Header="Страна" Binding="{Binding Country.Name}" Width="250"></DataGridTextColumn>

<DataGridTemplateColumn Width="auto">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button Content="Редактировать" Name="BtnEdit" Click="BtnEdit\_click"></Button>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

</DataGrid.Columns>

</DataGrid>

<Button Foreground="White" Content="Добавить" Grid.Row="1" HorizontalAlignment="Left" Name="BtnAdd" Click="BtnAdd\_click" Margin="10,10,0,0" Width="102"></Button>

<Button Foreground="White" Content="Удалить" Grid.Row="1" HorizontalAlignment="Right" Name="BtnDelete" Click="BtnDelete\_click" Margin="0,10,10,0" Width="98"></Button>

</Grid>

</Page>

Код XAML Изменения данных об отелях

<Page x:Class="MakeCarcas.AddEditPage"

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

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

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

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

xmlns:local="clr-namespace:MakeCarcas"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="AddEditPage">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="200"/>

<ColumnDefinition Width="1\*"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<TextBlock Text="Название" VerticalAlignment="Center" HorizontalAlignment="Center" Foreground="Red"></TextBlock>

<TextBlock Text="Количество звёзд" Grid.Row="1" VerticalAlignment="Center" HorizontalAlignment="Center" Foreground="Red"></TextBlock>

<TextBlock Text="Страна" Grid.Row="2" VerticalAlignment="Center" HorizontalAlignment="Center" Foreground="#FFFF0202"></TextBlock>

<TextBox Text="{Binding Name}" MaxLength="100" Grid.Column="1" Height="30"></TextBox>

<TextBox Text="{Binding CountOfStars}" Grid.Column="1" Grid.Row="1" Width="175" HorizontalAlignment="Left" Height="30"></TextBox>

<ComboBox SelectedItem="{Binding Country}" Grid.Row="2" Grid.Column="1" x:Name="ComboCountries" DisplayMemberPath="Name" Height="30"></ComboBox>

<Button Foreground="White" Content="Сохранить" Grid.Row="3" Name="BtnSave" Click="BtnSave\_click" Grid.Column="1" Margin="160,10,319,37" Width="Auto"/>

</Grid>

</Page>

Код XAML какиеПользователя есть

<Page x:Class="MakeCarcas.AdminPage"

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

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

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

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

xmlns:local="clr-namespace:MakeCarcas"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="AdminPage" IsVisibleChanged="Page\_IsVisibleChanged">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="377\*"/>

<RowDefinition Height="73\*"/>

</Grid.RowDefinitions>

<DataGrid x:Name="DGridUser" AutoGenerateColumns="False" IsReadOnly="True" SelectionChanged="DGridUsers\_SelectionChanged" Background="#FFB773">

<DataGrid.Columns>

<DataGridTextColumn Header="Имя" Binding="{Binding User}" Width="\*" ></DataGridTextColumn>

<DataGridTextColumn Header="Пароль" Binding="{Binding Password}" Width="150"></DataGridTextColumn>

<DataGridTextColumn Header="Имеет права администратора" Binding="{Binding IsAdmin}" Width="250"></DataGridTextColumn>

<DataGridTemplateColumn Width="auto">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button Content="Редактировать" Name="BtnEdit" Click="BtnEdit\_click"></Button>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

</DataGrid.Columns>

</DataGrid>

<Button Foreground="White" Content="Добавить" Grid.Row="1" HorizontalAlignment="Left" Name="BtnAdd" Click="BtnAdd\_click" Margin="10,10,0,0" Width="102"></Button>

<Button Foreground="White" Content="Удалить" Grid.Row="1" HorizontalAlignment="Right" Name="BtnDelete" Click="BtnDelete\_click" Margin="0,10,10,0" Width="98"></Button>

</Grid>

</Page>

Код XAML Редактирование данныхз пользователя

<Page x:Class="MakeCarcas.AdminPanelPage"

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

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

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

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

xmlns:local="clr-namespace:MakeCarcas"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="AdminPanelPage">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="200"/>

<ColumnDefinition Width="1\*"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<TextBlock Text="Имя" VerticalAlignment="Center" HorizontalAlignment="Center" Foreground="Red"></TextBlock>

<TextBlock Text="Пароль" Grid.Row="1" VerticalAlignment="Center" HorizontalAlignment="Center" Foreground="Red"></TextBlock>

<TextBlock Text="Админские права" Grid.Row="2" VerticalAlignment="Center" HorizontalAlignment="Center" Foreground="#FFFF0202"></TextBlock>

<TextBox Text="{Binding User}" MaxLength="100" Grid.Column="1" Height="30"></TextBox>

<TextBox Text="{Binding Password}" Grid.Column="1" Grid.Row="1" Width="175" HorizontalAlignment="Left" Height="30"></TextBox>

<TextBox Text="{Binding IsAdmin}" Grid.Row="2" Grid.Column="1" Height="30"></TextBox>

<Button Foreground="White" Content="Сохранить" Grid.Row="3" Name="BtnSave" Click="BtnSave\_click" Grid.Column="1" Margin="160,10,319,37" Width="Auto"/>

</Grid>

</Page>

Код CS

namespace MakeCarcas

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

MainFrame.Navigate(new LoginPage());

Manager.MainFrame = MainFrame;

}

public static void Loginuspeshen(Users user)

{

Manager.MainFrame.Navigate(new IntermediatePage(user));

}

private void BtnBack\_click(object sender, RoutedEventArgs e)

{

Manager.MainFrame.GoBack();

}

private void MainFrame\_ContentRendered(object sender, EventArgs e)

{

if (MainFrame.CanGoBack)

{

BtnBack.Visibility = Visibility.Visible;

}

else

{

BtnBack.Visibility = Visibility.Hidden;

}

}

}

}

Код CS Логин

namespace MakeCarcas

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

MainFrame.Navigate(new LoginPage());

Manager.MainFrame = MainFrame;

}

public static void Loginuspeshen(Users user)

{

Manager.MainFrame.Navigate(new IntermediatePage(user));

}

private void BtnBack\_click(object sender, RoutedEventArgs e)

{

Manager.MainFrame.GoBack();

}

private void MainFrame\_ContentRendered(object sender, EventArgs e)

{

if (MainFrame.CanGoBack)

{

BtnBack.Visibility = Visibility.Visible;

}

else

{

BtnBack.Visibility = Visibility.Hidden;

}

}

}

}

Код CS Окно для выбора окон

namespace MakeCarcas

{

/// <summary>

/// Логика взаимодействия для IntermediatePage.xaml

/// </summary>

public partial class IntermediatePage : Page

{

private Users user;

public IntermediatePage()

{

InitializeComponent();

}

public IntermediatePage(Users user)

{

InitializeComponent();

Console.WriteLine($"{user.IsAdmin}");

if (user.IsAdmin == "Yes")

BtnAdminPanel.Visibility = Visibility.Visible;

else

BtnAdminPanel.Visibility = Visibility.Hidden;

this.user = user;

}

private void BtnHotel\_click(object sender, RoutedEventArgs e)

{

Manager.MainFrame.Navigate(new HotelPage());

}

private void BtnTour\_click(object sender, RoutedEventArgs e)

{

Manager.MainFrame.Navigate(new TourPage());

}

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

{

Manager.MainFrame.Navigate(new AdminPage());

}

}

}Код CS Отели

namespace MakeCarcas

{

/// <summary>

/// Логика взаимодействия для HotelPage.xaml

/// </summary>

public partial class HotelPage : Page

{

public HotelPage()

{

InitializeComponent();

DGridHotels.ItemsSource = Polyakov2Entities.GetContext().Hotel.ToList();

}

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

{

Manager.MainFrame.Navigate(new AddEditPage(null));

}

private void BtnEdit\_click(object sender, RoutedEventArgs e)

{

Manager.MainFrame.Navigate(new AddEditPage((sender as Button).DataContext as Hotel));

}

private void BtnDelete\_click(object sender, RoutedEventArgs e)

{

var hotelForRemoving = DGridHotels.SelectedItems.Cast<Hotel>().ToList();

if (MessageBox.Show($"Вы точно хотите удалить следующие {hotelForRemoving.Count()} элементов ?" , "Внимание", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

{

try

{

Polyakov2Entities.GetContext().Hotel.RemoveRange(hotelForRemoving);

Polyakov2Entities.GetContext().SaveChanges();

MessageBox.Show("Данные удаленны!");

DGridHotels.ItemsSource = Polyakov2Entities.GetContext().Hotel.ToList();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

}

private void BtnAdd\_click(object sender, RoutedEventArgs e)

{

Manager.MainFrame.Navigate(new AddEditPage(null));

}

private void Page\_IsVisibleChanged(object sender, DependencyPropertyChangedEventArgs e)

{

if (Visibility == Visibility.Visible)

{

Polyakov2Entities.GetContext().ChangeTracker.Entries().ToList().ForEach(p => p.Reload());

DGridHotels.ItemsSource = Polyakov2Entities.GetContext().Hotel.ToList();

}

}

private void DGridHotels\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

}

}

}

Код CS Изменения данных об отелях

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 MakeCarcas

{

/// <summary>

/// Логика взаимодействия для AddEditPage.xaml

/// </summary>

public partial class AddEditPage : Page

{

private Hotel \_currentHotel = new Hotel();

public AddEditPage(Hotel selectedHotel)

{

InitializeComponent();

if (selectedHotel != null)

\_currentHotel = selectedHotel;

DataContext = \_currentHotel;

ComboCountries.ItemsSource = Polyakov2Entities.GetContext().Country.ToList();

}

private void BtnSave\_click(object sender, RoutedEventArgs e)

{

StringBuilder errors = new StringBuilder();

if (string.IsNullOrWhiteSpace(\_currentHotel.Name))

errors.AppendLine("Укажите название отеля");

if (\_currentHotel.CountOfStars < 1 || \_currentHotel.CountOfStars > 5)

errors.AppendLine("Количество звёзд - число от 1 до 5");

if (\_currentHotel.Country == null)

errors.AppendLine("Выберите страну");

if (errors.Length > 0)

{

MessageBox.Show(errors.ToString());

return;

}

if (\_currentHotel.id == 0)

Polyakov2Entities.GetContext().Hotel.Add(\_currentHotel);

try

{

Polyakov2Entities.GetContext().SaveChanges();

MessageBox.Show("Информация сохранена!");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

}

}

Код СS какиеПользователя есть

namespace MakeCarcas

{

/// <summary>

/// Логика взаимодействия для AdminPage.xaml

/// </summary>

public partial class AdminPage : Page

{

public AdminPage()

{

InitializeComponent();

}

private void BtnEdit\_click(object sender, RoutedEventArgs e)

{

Manager.MainFrame.Navigate(new AdminPanelPage((sender as Button).DataContext as Users));

}

private void BtnAdd\_click(object sender, RoutedEventArgs e)

{

Manager.MainFrame.Navigate(new AdminPanelPage(null));

}

private void BtnDelete\_click(object sender, RoutedEventArgs e)

{

var UserForRemoving = DGridUser.SelectedItems.Cast<Users>().ToList();

if (MessageBox.Show($"Вы точно хотите удалить следующие {UserForRemoving.Count()} элементов ?", "Внимание", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

{

try

{

Polyakov2Entities.GetContext().Users.RemoveRange(UserForRemoving);

Polyakov2Entities.GetContext().SaveChanges();

MessageBox.Show("Данные удаленны!");

DGridUser.ItemsSource = Polyakov2Entities.GetContext().Users.ToList();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

}

private void DGridUsers\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

}

private void Page\_IsVisibleChanged(object sender, DependencyPropertyChangedEventArgs e)

{

if (Visibility == Visibility.Visible)

{

Polyakov2Entities.GetContext().ChangeTracker.Entries().ToList().ForEach(p => p.Reload());

DGridUser.ItemsSource = Polyakov2Entities.GetContext().Users.ToList();

}

}

}

}

Код CS Редактирование данныхз пользователя

namespace MakeCarcas

{

/// <summary>

/// Логика взаимодействия для AdminPanelPage.xaml

/// </summary>

public partial class AdminPanelPage : Page

{

private Users \_currentUser = new Users();

public AdminPanelPage(Users selectedUser)

{

InitializeComponent();

if (\_currentUser != null)

\_currentUser = selectedUser;

DataContext = \_currentUser;

}

private void BtnSave\_click(object sender, RoutedEventArgs e)

{

StringBuilder errors = new StringBuilder();

if (string.IsNullOrWhiteSpace(\_currentUser.User))

errors.AppendLine("Укажите Имя пользователя");

if (string.IsNullOrWhiteSpace(\_currentUser.Password))

errors.AppendLine("Укажите Пароль пользователя");

if (\_currentUser.IsAdmin == null)

errors.AppendLine("Выберите Права пользователя");

if (errors.Length > 0)

{

MessageBox.Show(errors.ToString());

return;

}

if (\_currentUser.id == 0)

Polyakov2Entities.GetContext().Users.Add(\_currentUser);

try

{

Polyakov2Entities.GetContext().SaveChanges();

MessageBox.Show("Информация сохранена!");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

}

}

# **ЗАКЛЮЧЕНИЕ**

Поставленные во время учебной практики задачи выполнены:

* Закреплены и выполнены практические задания с «WPF
* Разработан проект сподключенной базой данных
* Сделан дизайн приложения.
* Закреплены знания и выполнены практические задания с «WPF»
* Разработана база данных для своего приложения.

При прохождении учебной практики особых трудностей не возникало. Интерфейс в приложении довольно понятный.

Данная учебная практика является хорошим практическим опытом для дальнейшей самостоятельной деятельности. За время пройденной практики я ознакомился с тонкостями использовния «WPF». Закрепил свои практические умения на практике.

Опыт, полученный мной на учебной практике, в дальнейшем пригодится в профессионально-карьерной деятельности и просто в повседневной жизни.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Официальная документация WPF: Официальная документация Microsoft по WPF содержит подробную информацию о различных аспектах разработки приложений на WPF. Вы можете найти здесь примеры кода, руководства по использованию элементов управления и другую полезную информацию. Ссылка: <https://docs.microsoft.com/en-us/dotnet/desktop/wpf/?view=net-desktop-5.0>
2. WPF Tutorial: WPF Tutorial - это онлайн-ресурс, который предоставляет обширное руководство по WPF с примерами кода и объяснениями. Здесь вы найдете информацию о различных аспектах WPF, включая разметку XAML, стили, шаблоны элементов управления и многое другое. Ссылка: <https://www.wpf-tutorial.com/>
3. Stack Overflow: Stack Overflow - это популярный вопросно-ответный сайт для программистов. Здесь вы можете найти множество вопросов и ответов, связанных с WPF. Вы можете задавать свои вопросы или искать ответы на уже существующие вопросы. Ссылка: <https://stackoverflow.com/questions/tagged/wpf>
4. CodeProject: CodeProject - это онлайн-сообщество разработчиков, где вы можете найти множество статей, примеров кода и проектов, связанных с WPF. Здесь вы можете найти полезные ресурсы, которые помогут вам в разработке приложений на WPF. Ссылка: <https://www.codeproject.com/>
5. WPF Disciples: WPF Disciples - это сообщество разработчиков, которые увлечены WPF и связанными с ним технологиями. Они поддерживают блог, в котором публикуют статьи, руководства и примеры кода, связанные с WPF. Вы можете найти здесь полезные материалы и получить поддержку от сообщества. Ссылка: <http://www.wpfdisciples.com/>
6. WPFpedia: WPFpedia - это онлайн-энциклопедия по WPF, где вы найдете множество статей, примеров кода и руководств. Здесь вы можете найти информацию о различных аспектах WPF, включая элементы управления, стили, анимацию и многое другое. Ссылка: <https://wpfpedia.com/>
7. Channel 9: Channel 9 - это платформа Microsoft, где вы можете найти видеоуроки, доклады и презентации о различных технологиях, включая WPF. Здесь вы можете найти видеоуроки, которые помогут вам лучше понять и использовать WPF. Ссылка: <https://channel9.msdn.com/>
8. WPF-Samples GitHub репозиторий: WPF-Samples - это репозиторий на GitHub, предоставляющий множество примеров кода и проектов, связанных с WPF. Здесь вы можете найти примеры кода, которые помогут вам изучить различные аспекты WPF. Ссылка: <https://github.com/Microsoft/WPF-Samples>
9. WPF Blog: WPF Blog - это блог разработчика, который публикует статьи, руководства и примеры кода, связанные с WPF. Здесь вы можете найти полезные материалы, которые помогут вам в разработке приложений на WPF. Ссылка: <https://www.wpfsharp.com/>
10. MSDN Magazine: MSDN Magazine - это онлайн-журнал, который публикует статьи и руководства о различных технологиях Microsoft, включая WPF. Здесь вы можете найти статьи, которые помогут вам лучше понять и использовать WPF. Ссылка: <https://msdn.microsoft.com/magazine/default.aspx>