

В.В. Подбельский

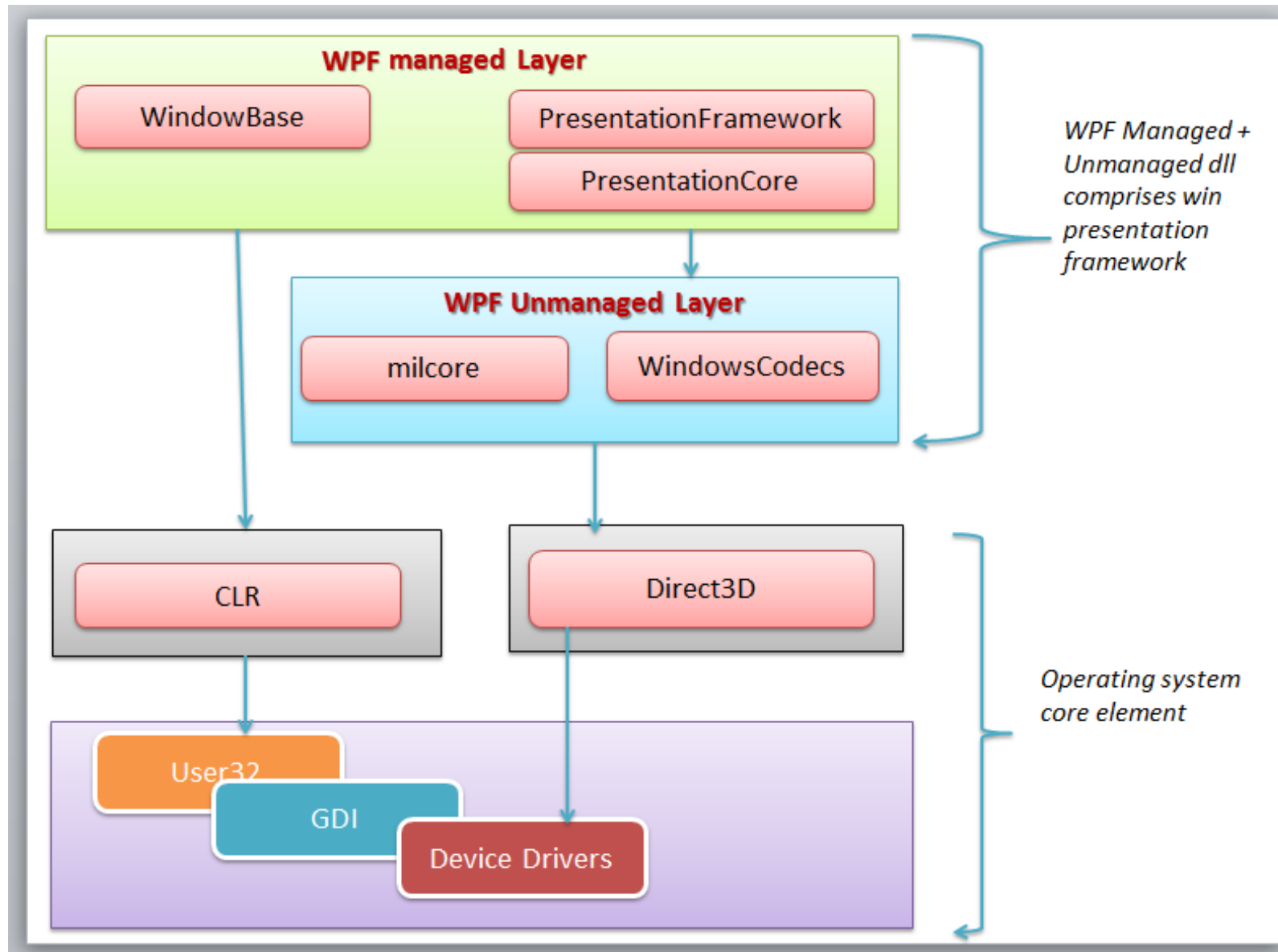
Использованы иллюстрации пособия Daniel Solis, Illustrated C#

Иллюстрации к курсу лекций по дисциплине «Программирование на C#»

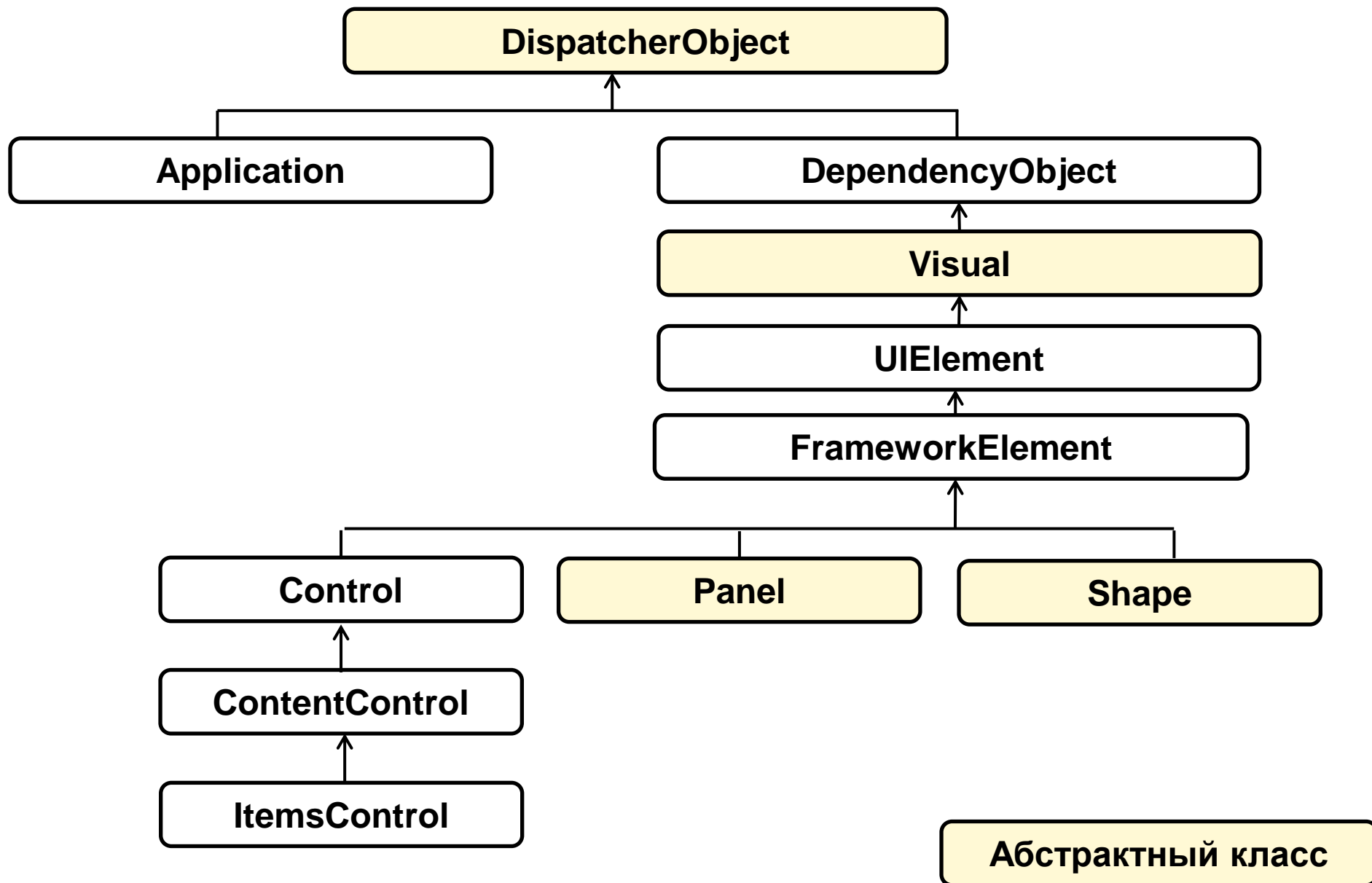
12. Часть 2

Архитектура WPF

Архитектура WPF



Фундаментальные Классы WPF



Важные Пространства Имян WPF

`System.Windows`
`System.Windows.Controls`
`System.Windows.Converters`
`System.Windows.Data`
`System.Windows.Documents`
`System.Windows.Ink`
`System.Windows.Input`
`System.Windows.Markup`
`System.Windows.Media`
`System.Windows.Resources`
`System.Windows.Shapes`
`System.Windows.Threading`

Важно:

`System.Windows.Forms.Button` – класс Windows Forms;

`System.Windows.Controls.Button` – класс WPF.

XAML и XML

XML - eXtensible Markup Language – «расширяемый язык разметки».

XML-дескрипторы или **теги**:

<имя_тега атрибуты...> – открывающий тег;

</имя_тега> – закрывающий тег;

<имя_тега атрибуты... /> – самозакрывающийся.

XML-элемент:

<имя_тега атрибуты...> [содержимое] </имя_тега>

XAML - eXtensible Application Markup Language – «расширяемый язык разметки приложений».

Назначение Тегов в XML

- Указывают границы участка (фрагмента) текста;
- Определяют роль (назначение) выделяемого фрагмента в тексте документа;
- Определяют расположение (позицию) фрагмента относительно других участков текста;
- Определяют вложенность фрагментов текста;
- Определяют связи фрагмента текста с ресурсами, размещёнными за пределами документа.

Синтаксически Верный XML Документ

```
<!-- xml-Example.xml -->
```

```
<Label FontSize = "30" Background="LightGray">
```

```
    Label - это "ярлык"!
```

```
</Label>
```

Escape-Последовательности в XML

Общий синтаксис XML escape-последовательностей:

`&<код>;`

- код в десятичной системе

Символьный Код	Числовой Код	Символ
<code>&amp;</code>	<code>&#038;</code>	Амперсанд (&)
<code>&lt;</code>	<code>&#060;</code>	Открывающая угловая скобка (<)
<code>&gt;</code>	<code>&#062;</code>	Закрывающая угловая скобка (>)
<code>&apos;</code>	<code>&#39;</code>	Одинарная кавычка (')
<code>&quot;</code>	<code>&#34;</code>	Двойная кавычка (")

XAML

Атрибут определения пространства имен в среде выполнения Windows:

<http://schemas.microsoft.com/winfx/2006/xaml/presentation>

Атрибут определения пространства имен XAML:

<http://schemas.microsoft.com/winfx/2006/xaml>

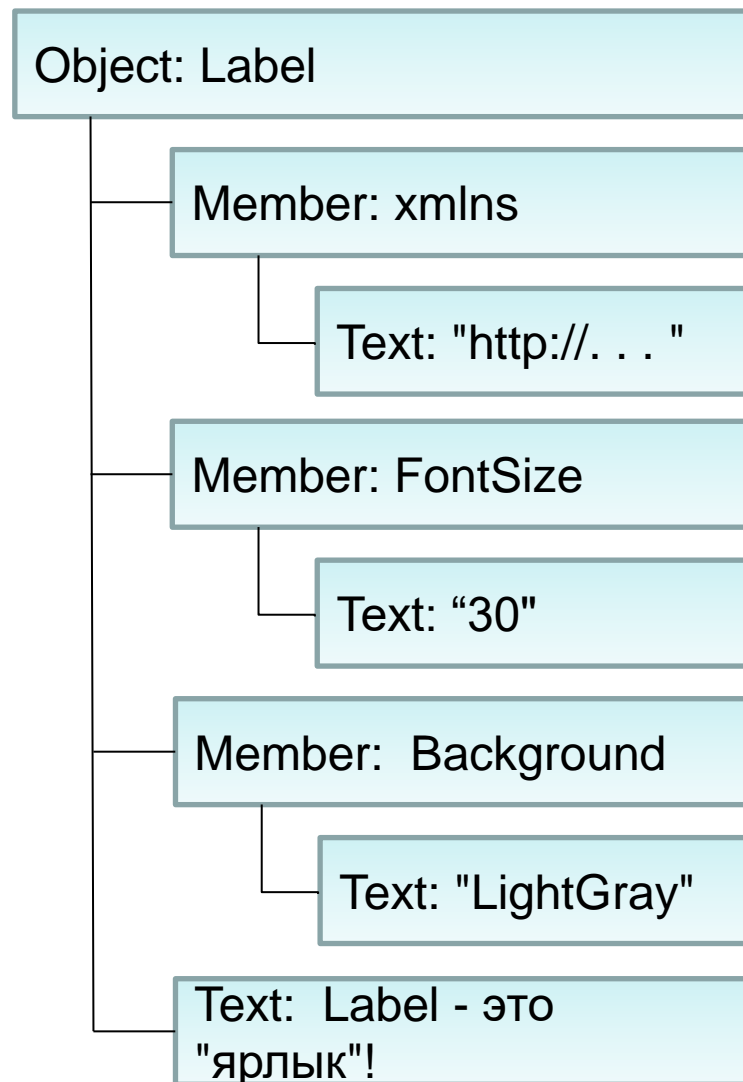
«Общие сведения о языке XAML (WPF)»:

<https://docs.microsoft.com/ru-ru/dotnet/desktop/wpf/xaml/?view=netdesktop-5.0>

Иерархия Объектов в XAML

```
<Label  
  xmlns="http://. . . "  
  FontSize = "30"  
  Background="LightGray">  
  Label - это "ярлык!"  
</Label>
```

Объекты
Члены (объектов)
Текст



Объявление Свойств Объектных Элементов

1) Синтаксис атрибутов:

имя_атрибута="значение_свойства"

2) Синтаксис элементов свойств:

<имя_типа.имя_свойства атрибуты>

</имя_типа.имя_свойства>

3) Самозакрывающаяся форма:

<имя_типа.имя_свойства атрибуты.../>

4) Синтаксис элементов свойств:

<имя_типа.имя_свойства>

Значение_свойства

</имя_типа.имя_свойства>

Элемент Label в Иерархии Классов

System.Object

System.Windows.Threading.DispatcherObject

System.Windows.DependencyObject

System.Windows.Media.Visual

System.Windows.UIElement

System.Windows.FrameworkElement

System.Windows.Controls.Control

System.Windows.Controls.ContentControl

System.Windows.Controls.Label

Основные Свойства Label

Name – имя элемента;

Type – тип элемента;

Background – оформление (цвет) фона;

Foreground – оформление (цвет) содержимого;

Content – содержимое;

Width – ширина;

Height – высота;

HorizontalAlignment – вертикальное выравнивание;

VerticalAlignment – горизонтальное выравнивание;

FontFamily - шрифт (семейство шрифта);

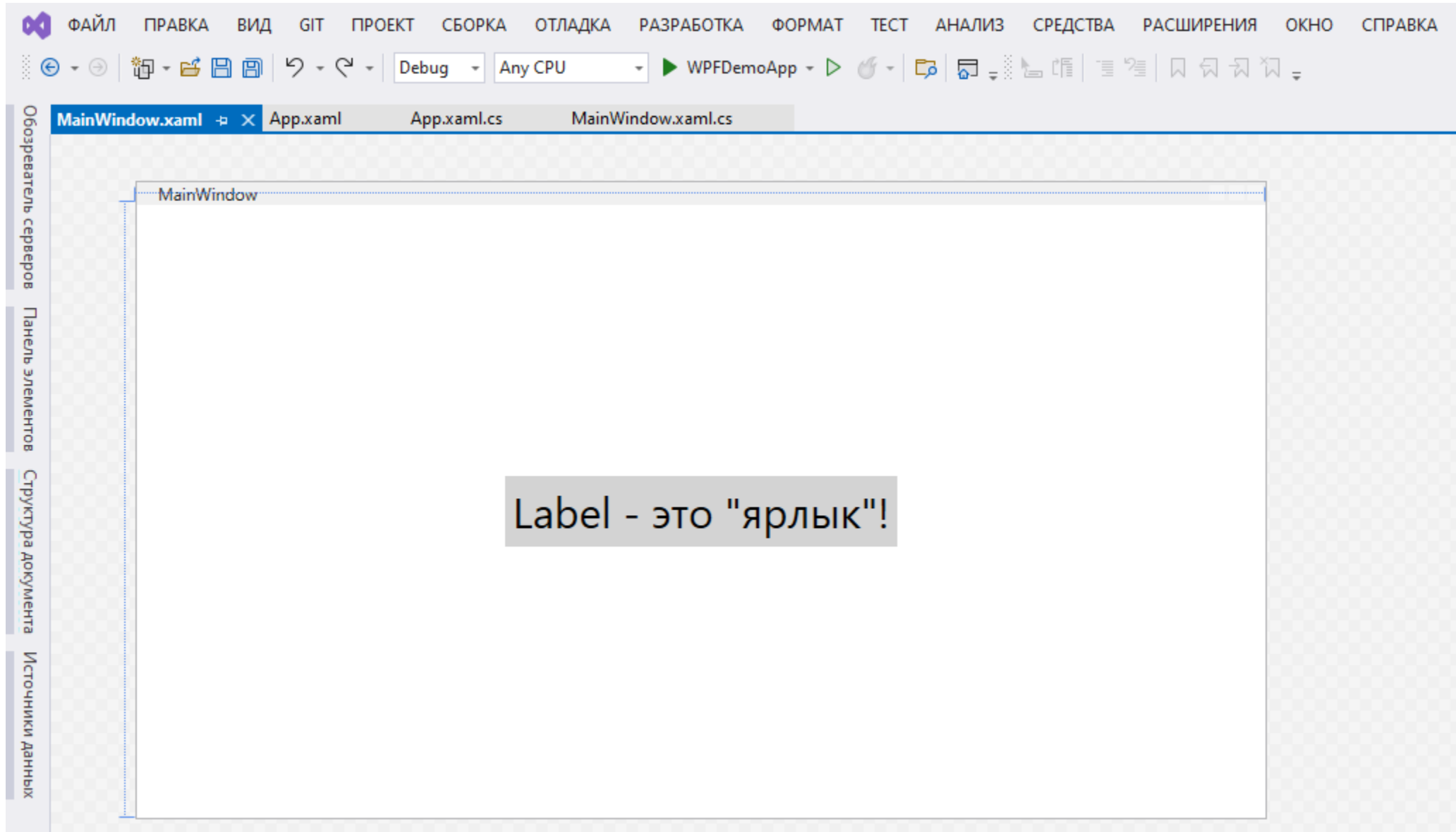
FontSize - размер (кегель - размер по вертикали);

FontStyle - декорирование (подчеркивание, курсив...).

Добавление Элементов в XAML: Label

```
<Window x:Class="WPFDemoApp.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:WPFDemoApp"
    mc:Ignorable="d"
    Title="MainWindow" Height="450" Width="800">
    <Grid>
        <Label xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation">
            <Label.FontSize>30</Label.FontSize>
            <Label.Background>LightGray</Label.Background>
            <Label.VerticalAlignment>Center</Label.VerticalAlignment>
            <Label.HorizontalAlignment>Center</Label.HorizontalAlignment>
            Label - это "ярлык"!
        </Label>
    </Grid>
</Window>
```

Отображение Документа с Label в Visual Studio



Основы технологии WPF

1. Классы поддержки (резервные типы)
2. Имена XAML-элементов
3. Атрибуты XAML-элементов и свойства объектов
4. Имя атрибута совпадает с именем члена объекта
5. XAML-документ – дерево XAML-элементов
6. Однократность вложения XAML-элементов
7. Корень дерева: Window, Page, Application, NavigationWindow. . .

Определения свойства-коллекции объектного элемента

Свойство-коллекция:

<тип...>

<тип.коллекция>

<тип_элемента_коллекции атрибуты />

...

<тип_элемента_коллекции атрибуты />

</тип.коллекция>

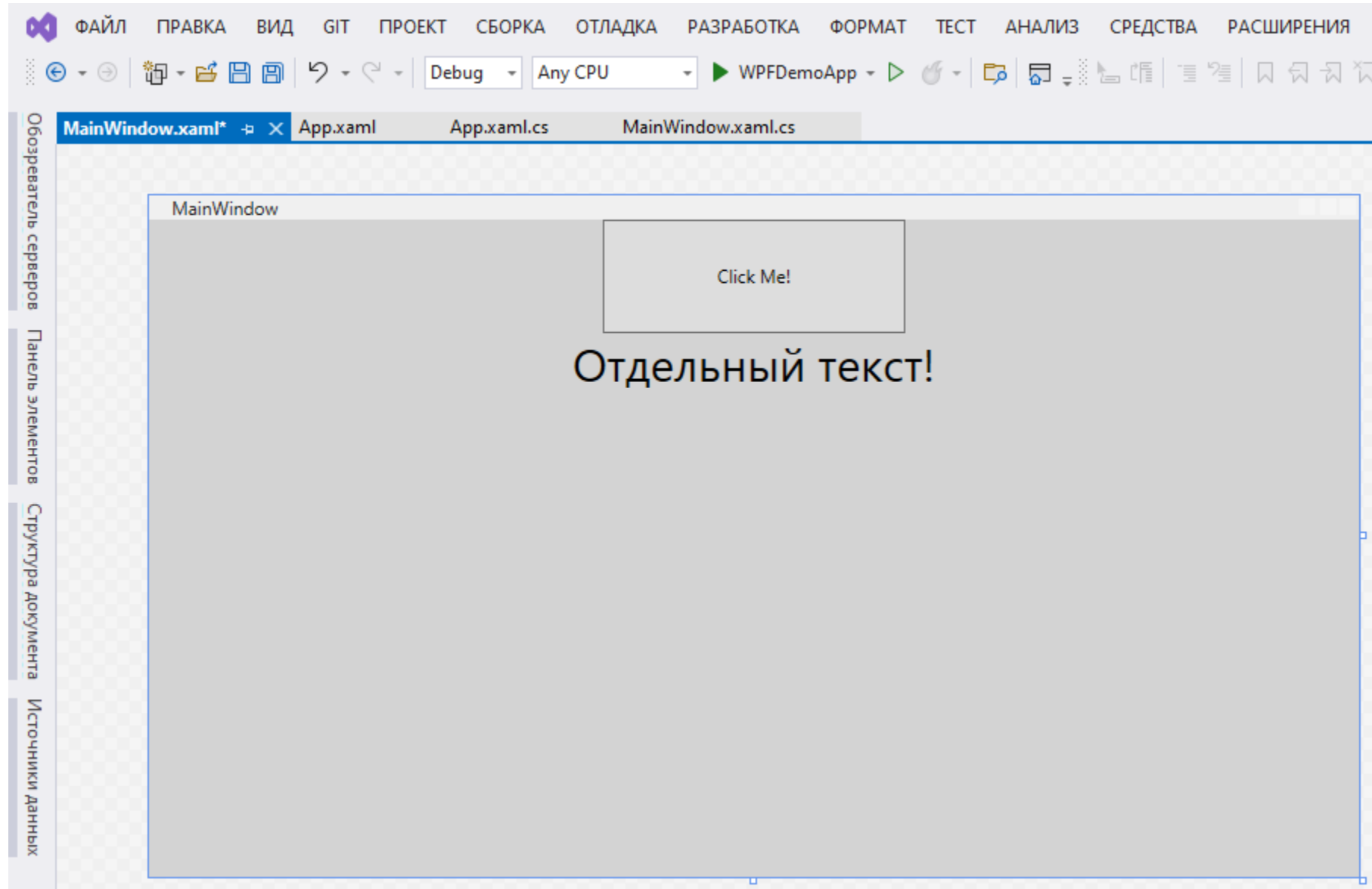
</тип...>

Добавление Элементов в XAML: StackPanel

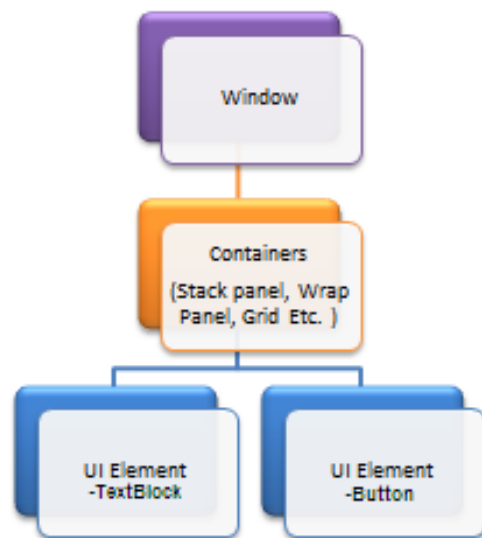
```
<Window x:Class="WPFDemoApp.MainWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:local="clr-namespace:WPFDemoApp"
    mc:Ignorable="d"
    Title="MainWindow" Height="450" Width="800">

    <StackPanel xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation" Background="LightGray">
        <Button Canvas.Top="30" Canvas.Left="40"
            Width="200" Height="75">Click Me!
        </Button>
        <TextBlock Text="Отдельный текст!" FontSize="30" HorizontalAlignment="Center"/>
    </StackPanel>
</Window>
```

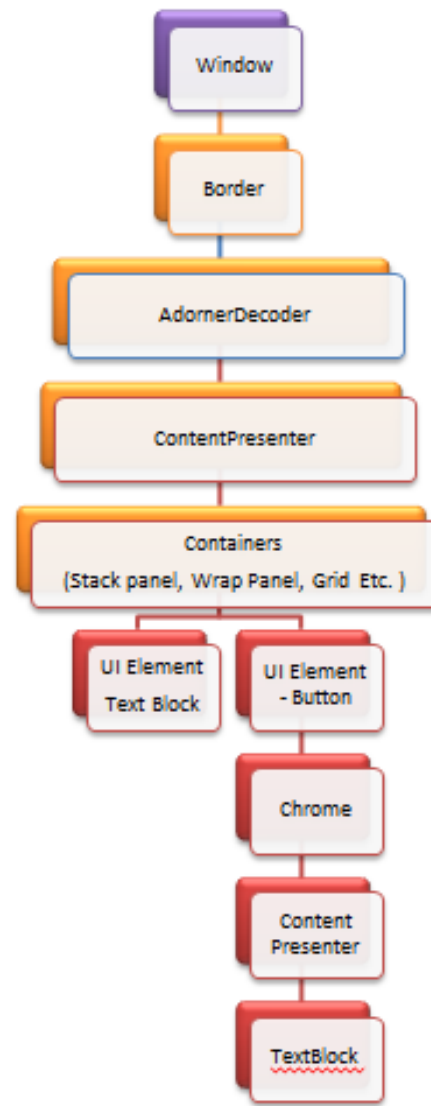
Отображение Документа с StackPanel в Visual Studio



Логическое и Визуальное Дерево Документа



Logical Tree



Visual Tree

Содержимое XAML-Элемента

Content (базовый класс ContentControl, производные классы: Window, Button, Label, RadioButton, ToolTip, ...);

Items (базовый класс ItemsControl, производные классы: Menu, ListBox, ComboBox, TabControl, ...);

Children (базовый класс Panel, производные классы: Grid, StackPanel, DockPanel, Canvas, UniformGrid, WrapPanel);

Header и **Items** (базовый класс HeaderedItemsControl, производные классы: MenuItem, ToolBar, TreeViewItem);

Child (базовый класс Decorator, производные классы: Border, InkPresenter, ViewBox).

Текстовое Содержимое XAML-Элемента

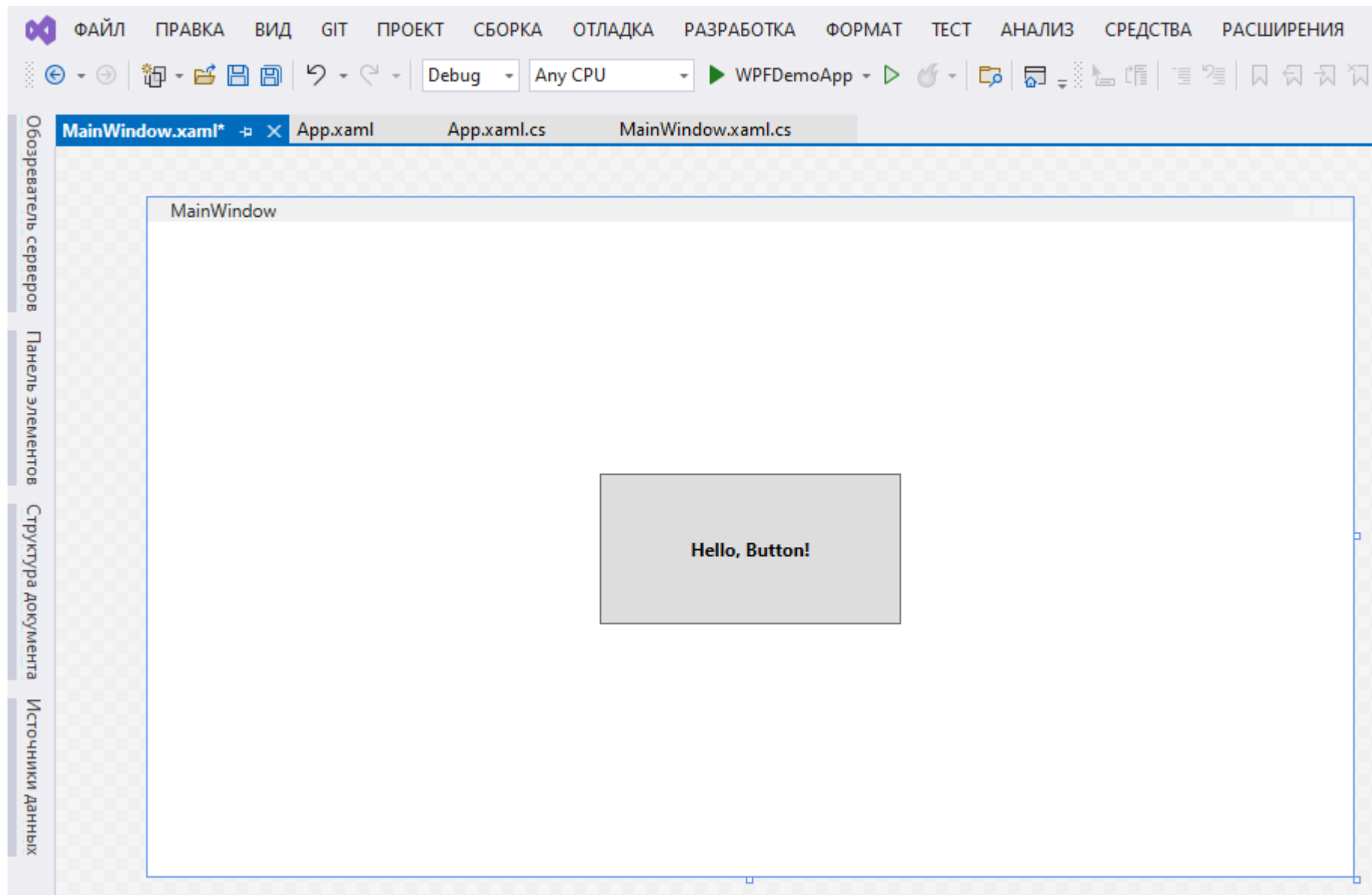
1. `<Button Width="200" Height="100" FontWeight="Bold">
 Hello, Button!
</Button>`

2. `<Button Width="200" Height="100" FontWeight="Bold"
 Content="Hello, Button!"/>`

3. `<Button Width="200" Height="100" FontWeight="Bold">
 <Button.Content>Hello, Button!</Button.Content>
</Button>`

4. `<Button Width="200" Height="100" FontWeight="Bold"
 Button.Content="Hello, Button!">
</Button>`

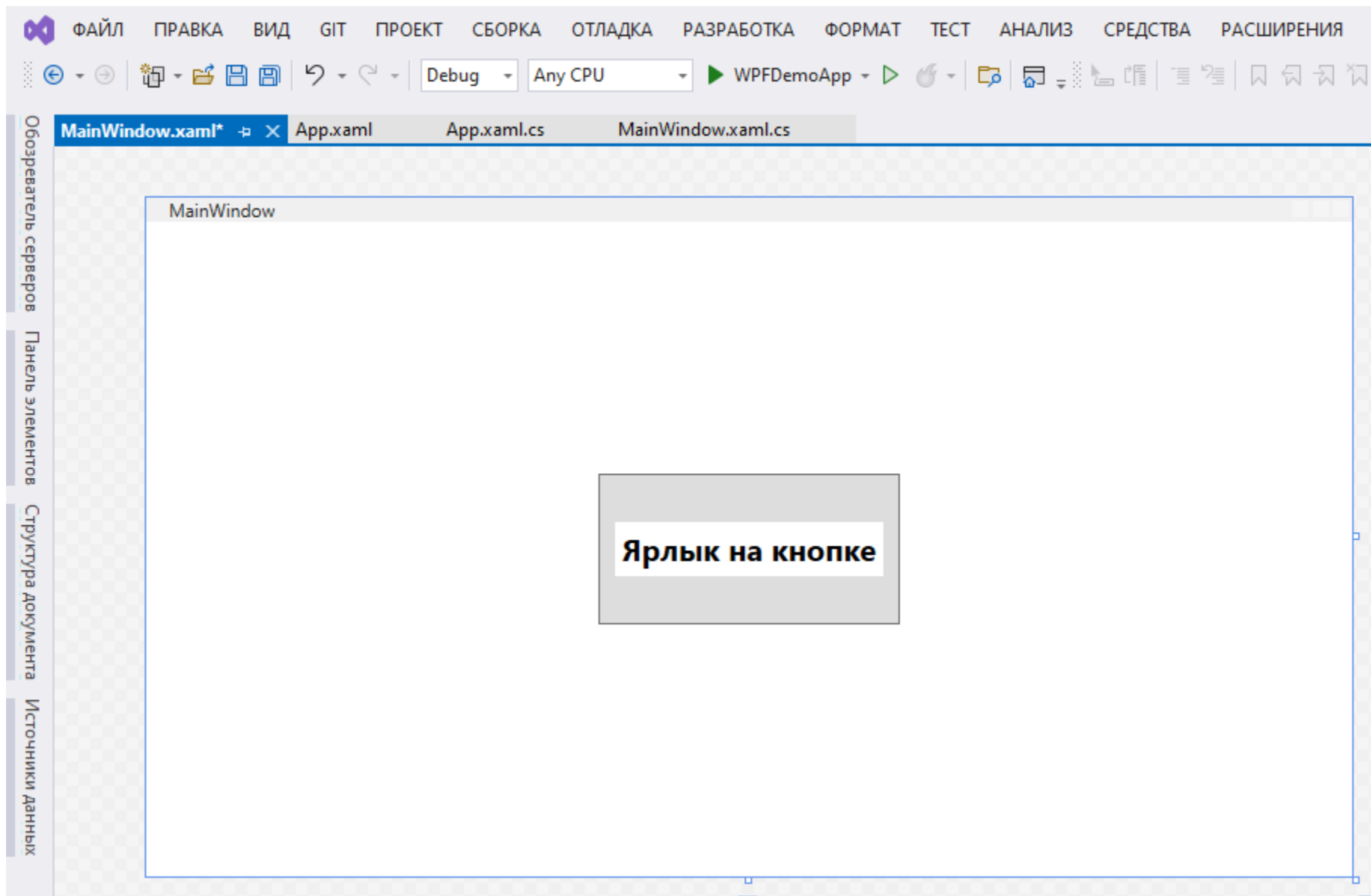
Визуализация Текстового Содержимого



Содержимое – Вложенные XAML-Элементы

```
<Button Width="200" Height="100" FontWeight="Bold">  
    <Button.Content>  
        <Label FontSize = "20" Background="White">  
            Ярлык на кнопке  
        </Label>  
    </Button.Content>  
</Button>
```


Если содержимое – это XAML элемент



Содержимое – Контейнер (Коллекция)

<тип...>

<тип.коллекция>

<тип_элемента_коллекции атрибуты>

Содержимое элемента коллекции

</тип_элемента_коллекции>

...

<тип_элемента_коллекции атрибуты>

Содержимое элемента коллекции

</тип_элемента_коллекции>

</тип.коллекция>

</тип...>

Контейнеры для компоновки

- Grid;
- StackPanel;
- DockPanel;
- Canvas;
- WrapPanel.

Базовый Класс Panel в Иерархии

System.Object

System.Windows.Threading.DispatcherObject

System.Windows.DependencyObject

System.Windows.Media.Visual

System.Windows.UIElement

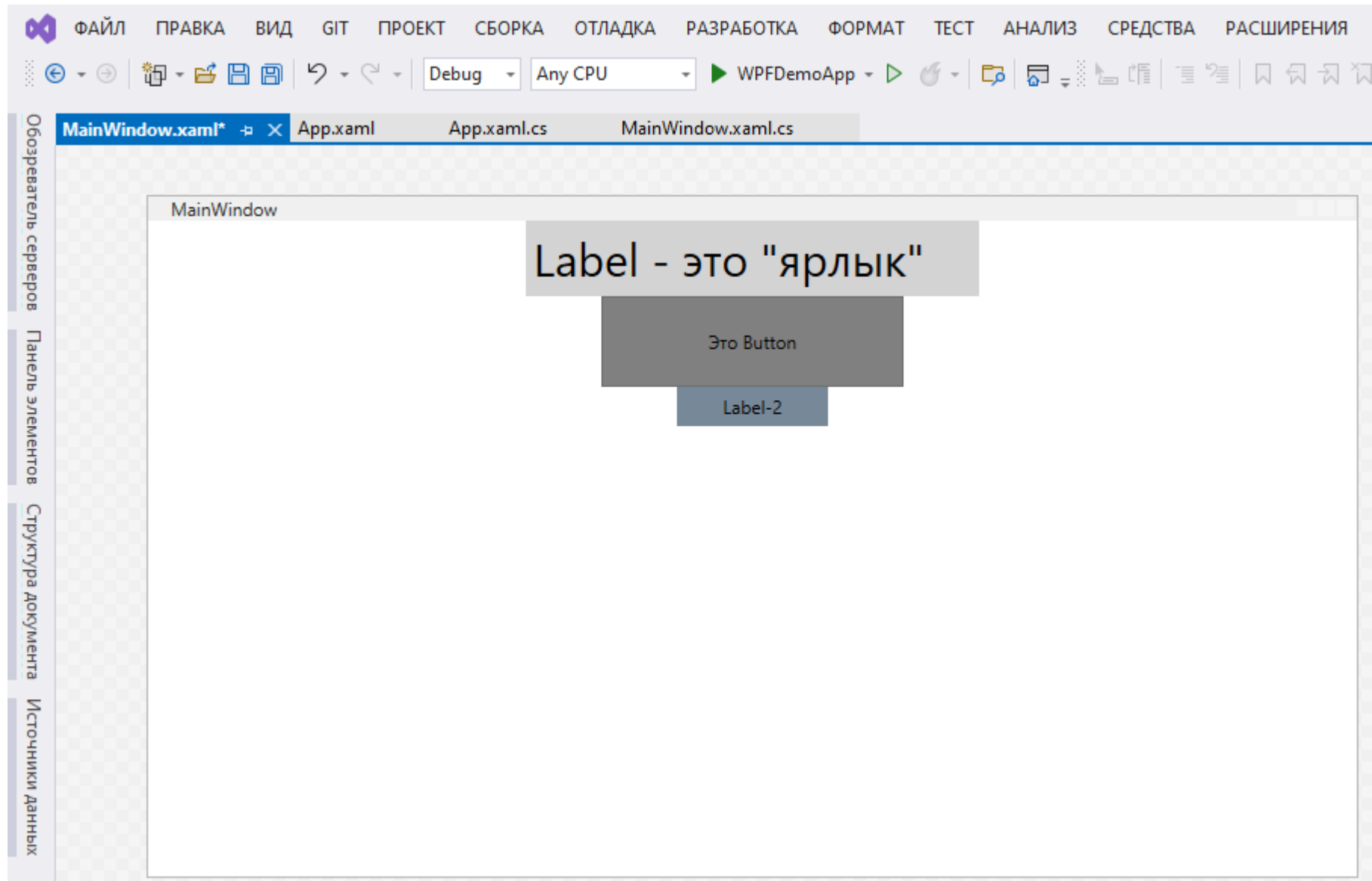
System.Windows.FrameworkElement

System.Windows.Controls.Panel

Добавление Элементов в XAML: Коллекция

```
<Window x:Class="WPFDemoApp.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:WPFDemoApp"
    mc:Ignorable="d"
    Title="MainWindow" Height="450" Width="800">
    <StackPanel xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation" >
        <StackPanel.Children>
            <Label Width="300" FontSize="30" Background="LightGray">Label - это "ярлык"
        </Label>
            <Button Width="200" Height="60" Background="Gray">
                Это Button
            </Button>
            <Label Width="100" HorizontalContentAlignment="Center"
                Background="LightSlateGray">Label-2</Label>
        </StackPanel.Children>
    </StackPanel>
</Window>
```

Отображение Документа с Коллекцией в Visual Studio



Размер и Позиционирование XAML-Документов

Width – явно заданное значение ширины изображения;

Height – явно заданное значение высоты изображения;

MinWidth – минимальное значение ширины изображения;

MaxWidth - максимальное значение ширины изображения;

MinHeight - минимальное значение высоты изображения;

MaxHeight - максимальное значение высоты изображения;

HorizontalAlignment – определяет горизонтальное позиционирование;

VerticalAlignment – определяет вертикальное позиционирование;

Margin – определяет внешние отступы для изображения;

Padding – определяет внутренние отступы для содержимого.

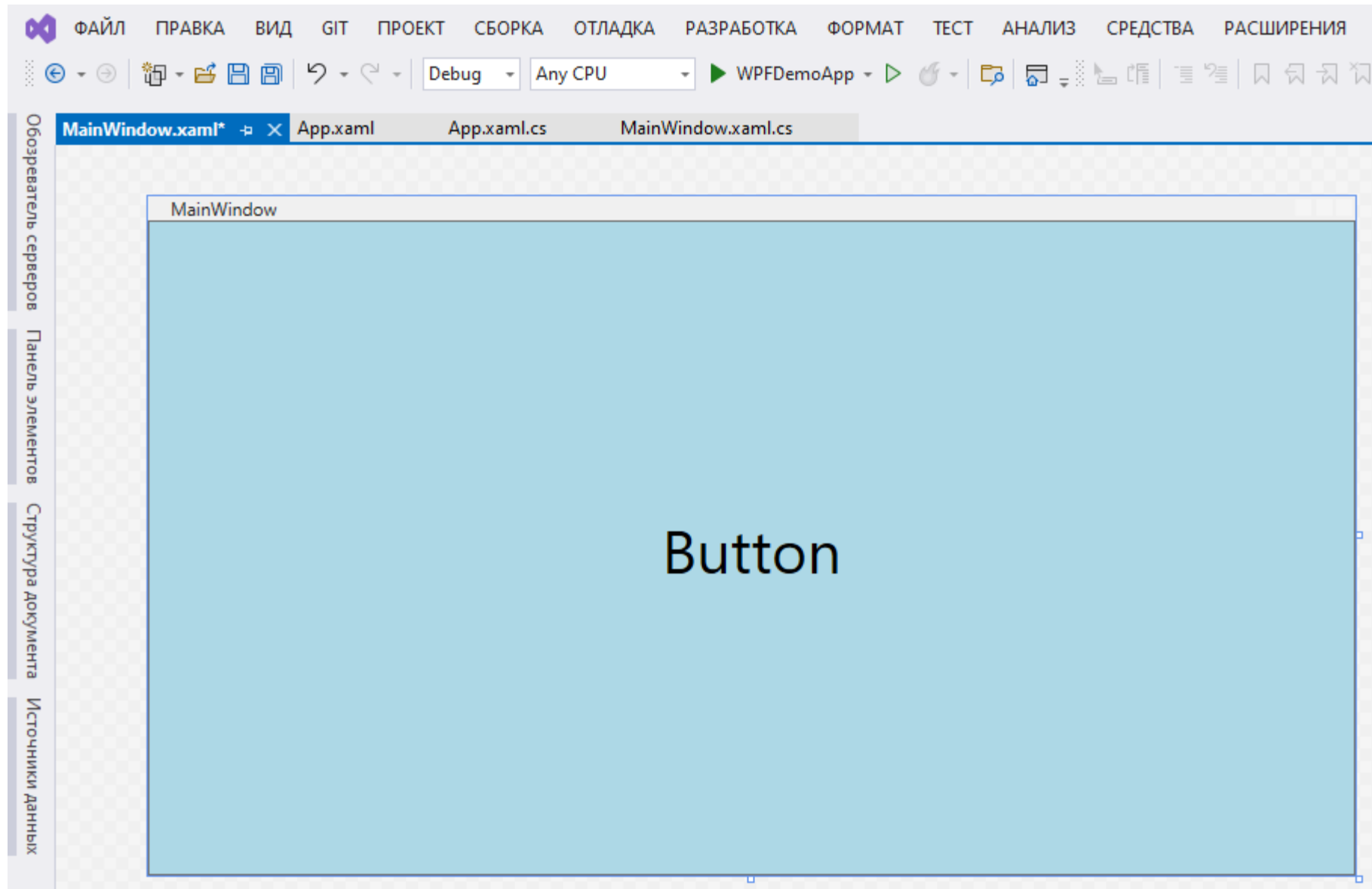
Растяжение Элементов по Ширине и Высоте

Обратите внимание, что по умолчанию:

HorizontalAlignment == VerticalAlignment == Stretch

```
<Window x:Class="WPFDemoApp.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:WPFDemoApp"
  mc:Ignorable="d"
  Title="MainWindow" Height="450" Width="800">
  <Button Background="LightBlue"
    FontSize = "40" Content="Button"/>
</Window>
```

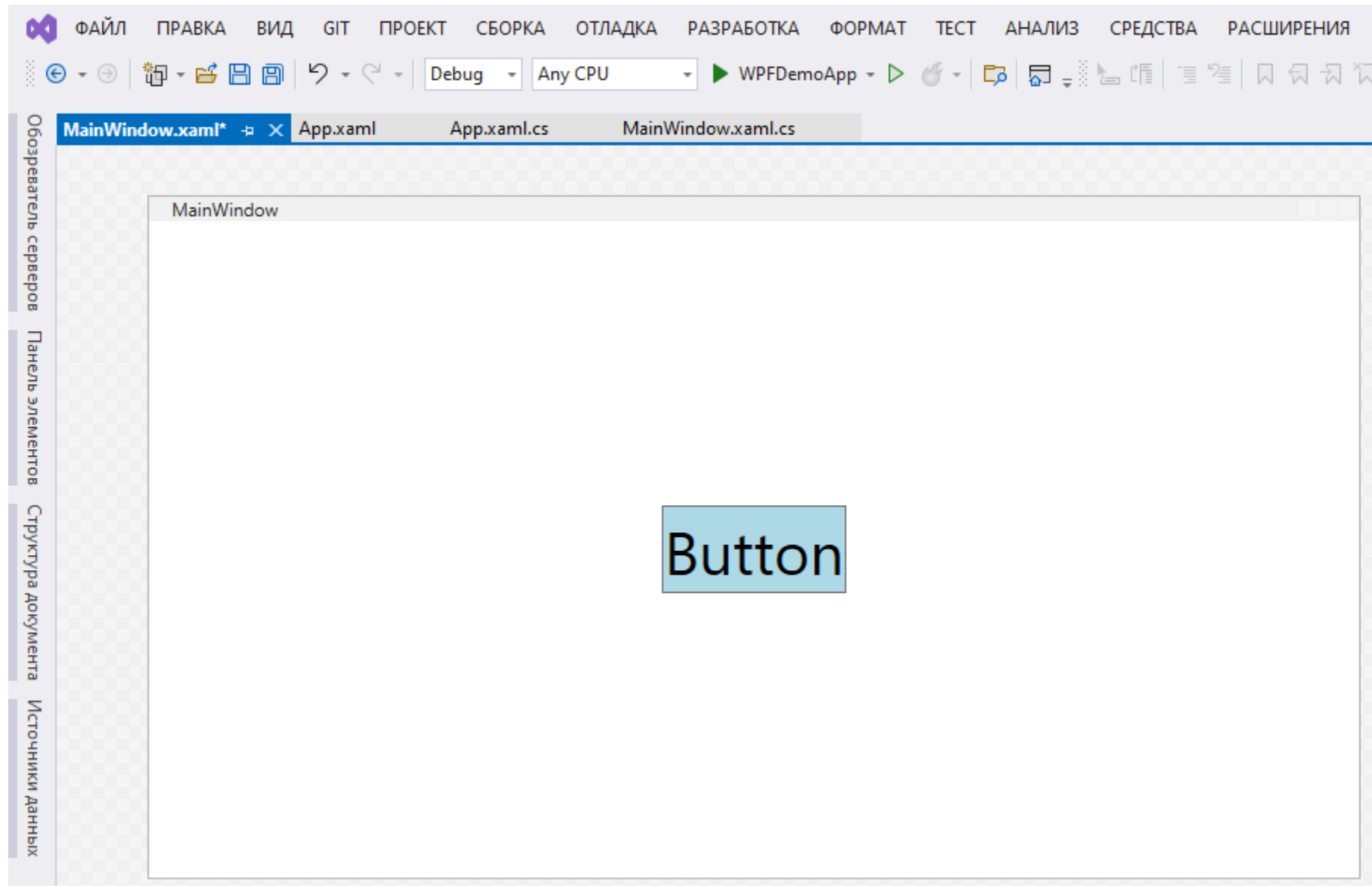

Кнопка Растянута по Ширине и по Высоте



Явно Заданное Положение Кнопки

```
<Window x:Class="WPFDemoApp.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:WPFDemoApp"
  mc:Ignorable="d"
  Title="MainWindow" Height="450" Width="800">
  <Button Background="LightBlue"
    VerticalAlignment="Center" HorizontalAlignment="Center"
    FontSize = "40" Content="Button"/>
</Window>
```

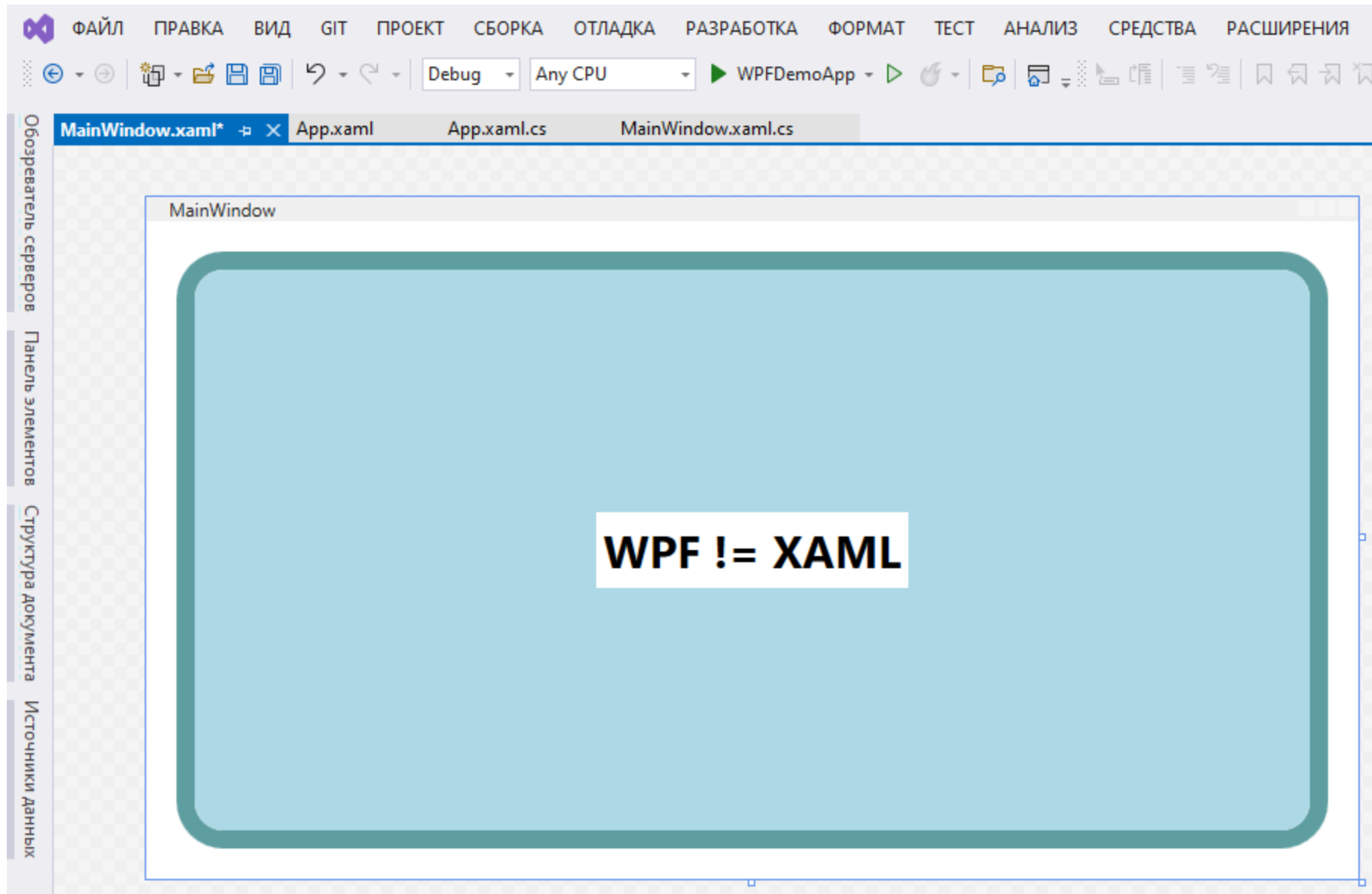
Положение Кнопки Задано Явно по Центру



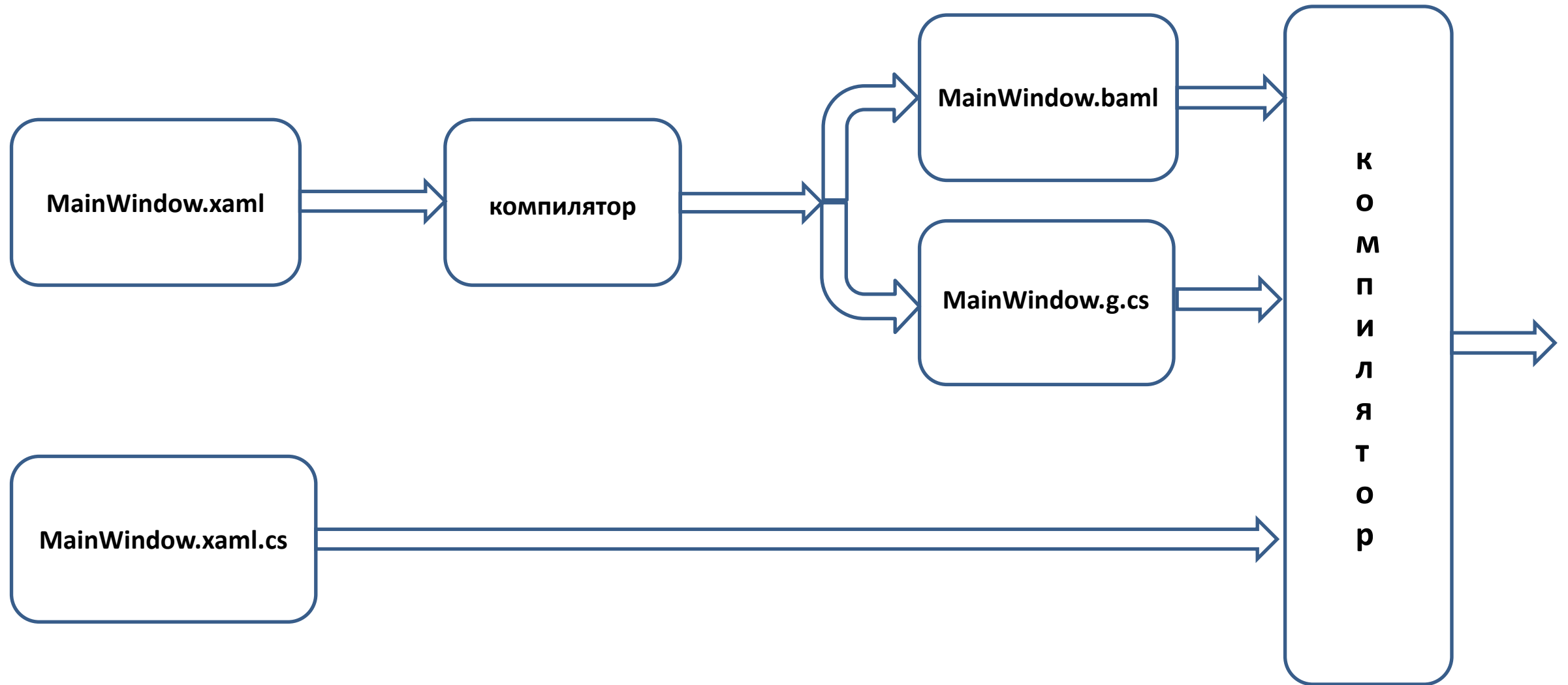
Добавление Элементов в XAML: Border

```
<Window x:Class="WPFDemoApp.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:WPFDemoApp"
    mc:Ignorable="d"
    Title="MainWindow" Height="450" Width="800">
    <Border Background="LightBlue"
        BorderBrush="CadetBlue" BorderThickness="12"
        CornerRadius="25"
        Margin="20" Padding="20">
        <Label Content = "WPF != XAML"
            Background = "White" FontSize = "30"
            FontWeight = "Bold"
            HorizontalAlignment = "Center"
            VerticalAlignment = "Center"/>
    </Border>
</Window>
```

Отображение Документа с Border в Visual Studio



Компиляция WPF-Приложения: Этап 1



Компиляция WPF-Приложения: Этап 2

