

Министерство образования и молодежной политики Свердловской области
ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Специальность: 09.02.07 Информационные системы и программирование

Практическая работа №6
по дисциплине «Технология разработки программного обеспечения»
на тему «Windows Presentation Foundation»

Выполнил:

студент гр. ПР-31

_____/Пинигина Д.А.

проверил:

_____/Гариев Д.О.

дата:

оценка:

СОДЕРЖАНИЕ

ЦЕЛЬ	3
НАЧАЛО РАБОТЫ С WINDOWS PRESENTATION FOUNDATION	4
СОЗДАНИЕ ЭЛЕМЕНТОВ В WPF	5
Изменение размеров окна	6
Создание кнопки	8
Линейная градиентная заливка	9
ФОН ОКНА	12
ВЫВОД	16

ЦЕЛЬ

Цель: научиться работать с WPF, узнать, как и где писать код для создания элементов, научиться создавать градиенты и элементы в WPF.

НАЧАЛО РАБОТЫ С WINDOWS PRESENTATION FOUNDATION

Перед тем как начать делать задания, нам нужно создать проект. Открываем Visual Studio, нажимаем *Создание проекта*, перед нами появится выбор предложенных шаблонов от консольного приложения до WPF.

Именно WPF нужен для работы. Выбираем *Приложение WPF (.NET Framework)* и нажимаем Далее (Рисунок 1).

Далее указываем название проекта, путь расположения проекта, платформу. Таким образом мы создали проект (Рисунок 2).

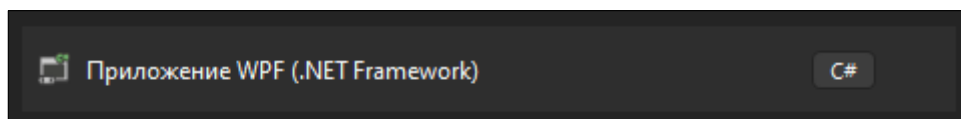


Рисунок 1 — Приложение WPF

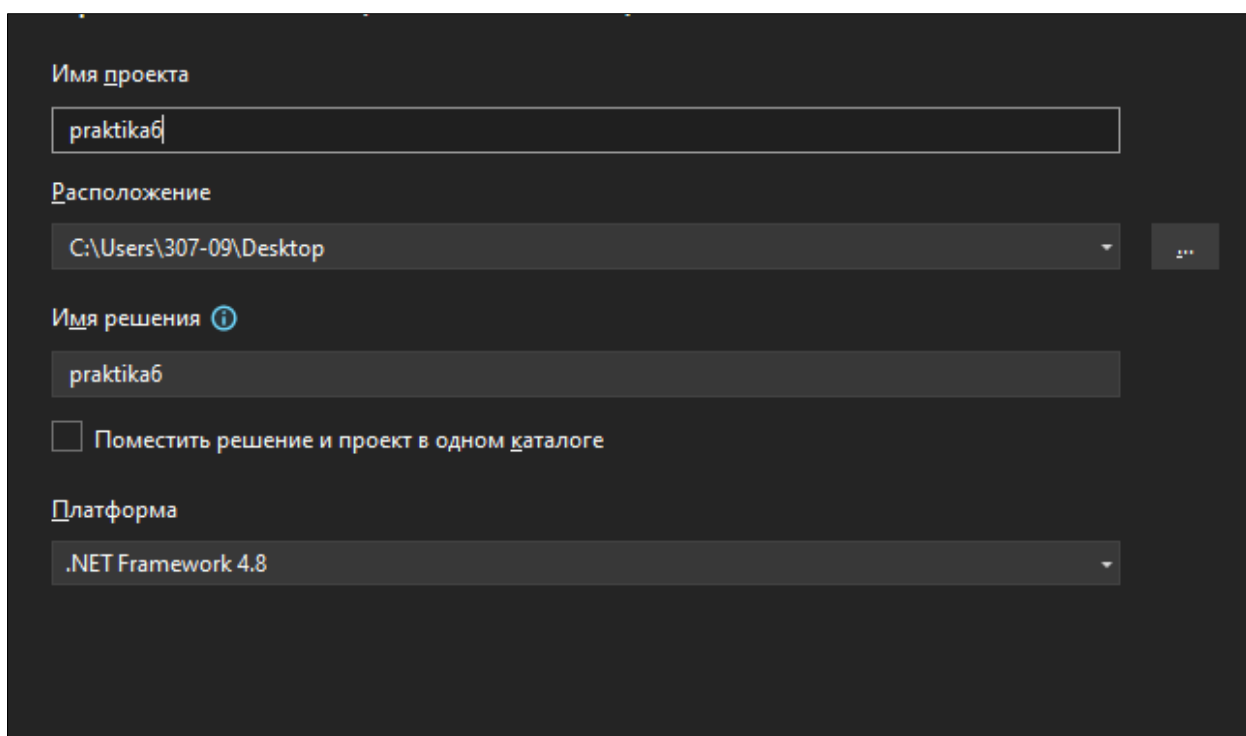


Рисунок 2 — Создание проекта

СОЗДАНИЕ ЭЛЕМЕНТОВ В WPF

После того, как мы выполнили предыдущие шаги, перед нами откроется пустое окно в *Конструкторе* (Рисунок 3). Скоро оно перестанет быть пустым.



Рисунок 3 — Вид окна по умолчанию

Во вкладке XAML код имеет следующий вид:

```
<Window x:Class="praktika6.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:praktika6"
        mc:Ignorable="d">
```

```
Title="MainWindow" Height="450" Width="800">  
<Grid>
```

```
</Grid>
```

```
</Window>
```

Или же (Рисунок 4):

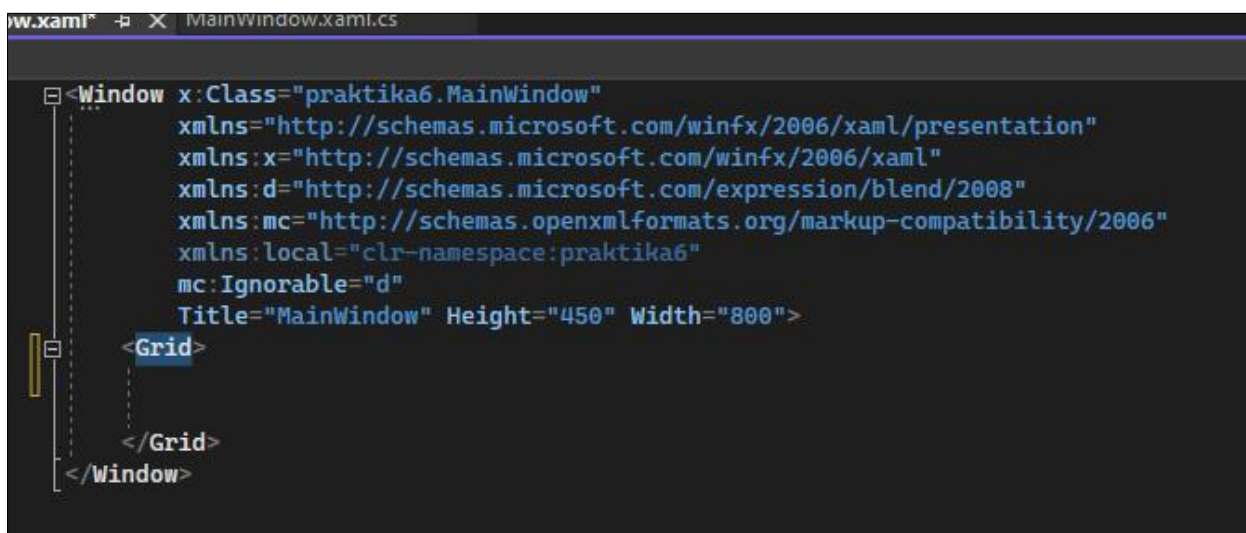


Рисунок 4 — Код в XAML по умолчанию

Тут уже содержатся пространства, позволяющие работать с элементами XAML. Даже указывается высота и ширина, которые надо будет поменять на следующих этапах. Далее код будет изменяться.

Изменение размеров окна

Перед этим стоит поменять высоту и ширину окна.

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

Высота будет равна 350, а ширина составляет 525 (Рисунок 5).

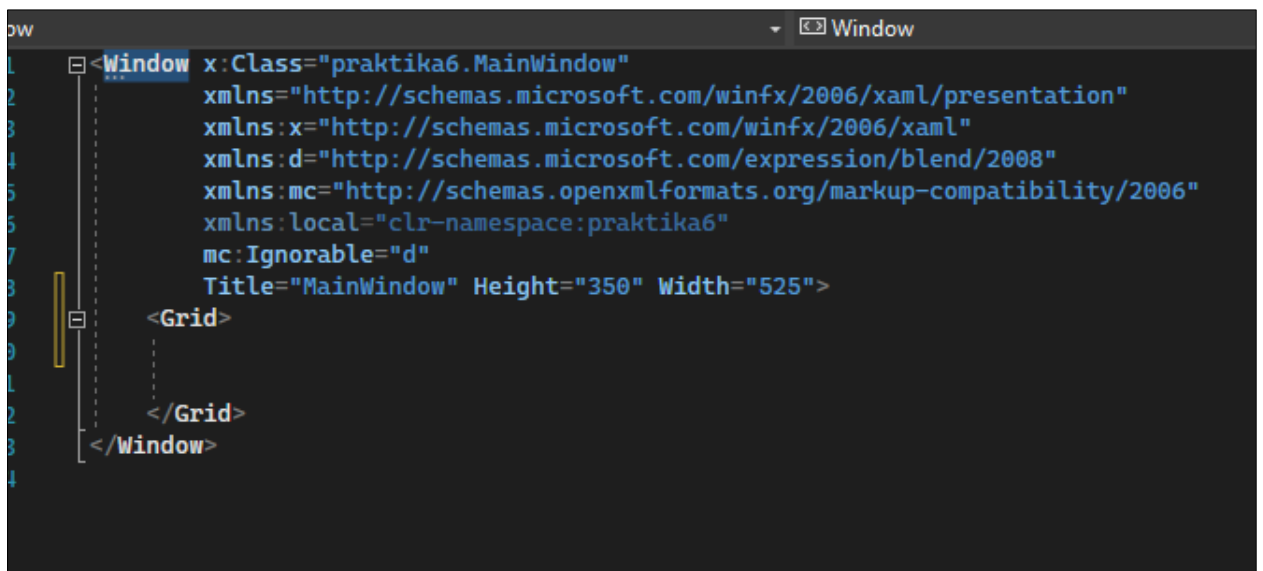


Рисунок 5 — Изменение высоты и ширины окна

Внешний вид окна станет меньше, чем до этого (Рисунок 6).

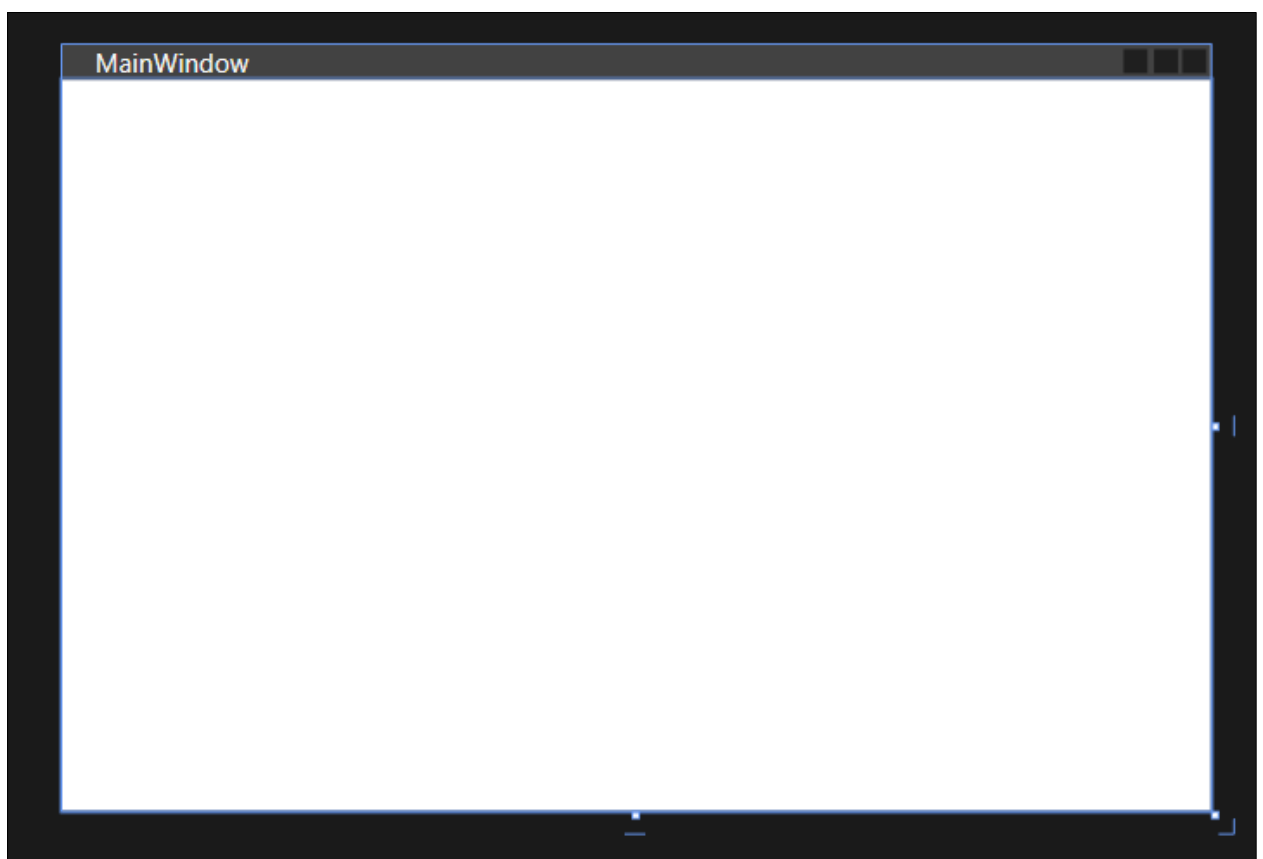


Рисунок 6 — Внешний вид полученного окна

Создание кнопки

Теперь создадим кнопку. Для этого мы должны разместить следующий код:

```
<Grid>

    <Button x:Name="Bt1"

        HorizontalAlignment="Center"
        VerticalAlignment="Center"
        Width="150"
        Height="30"
        FontSize="17"
        Content="Обычная кнопка"
        Foreground="#006699"
        Background="#f0f0f0"
        BorderBrush="#303030"></Button>

</Grid>
```

Или же (Рисунок 7):

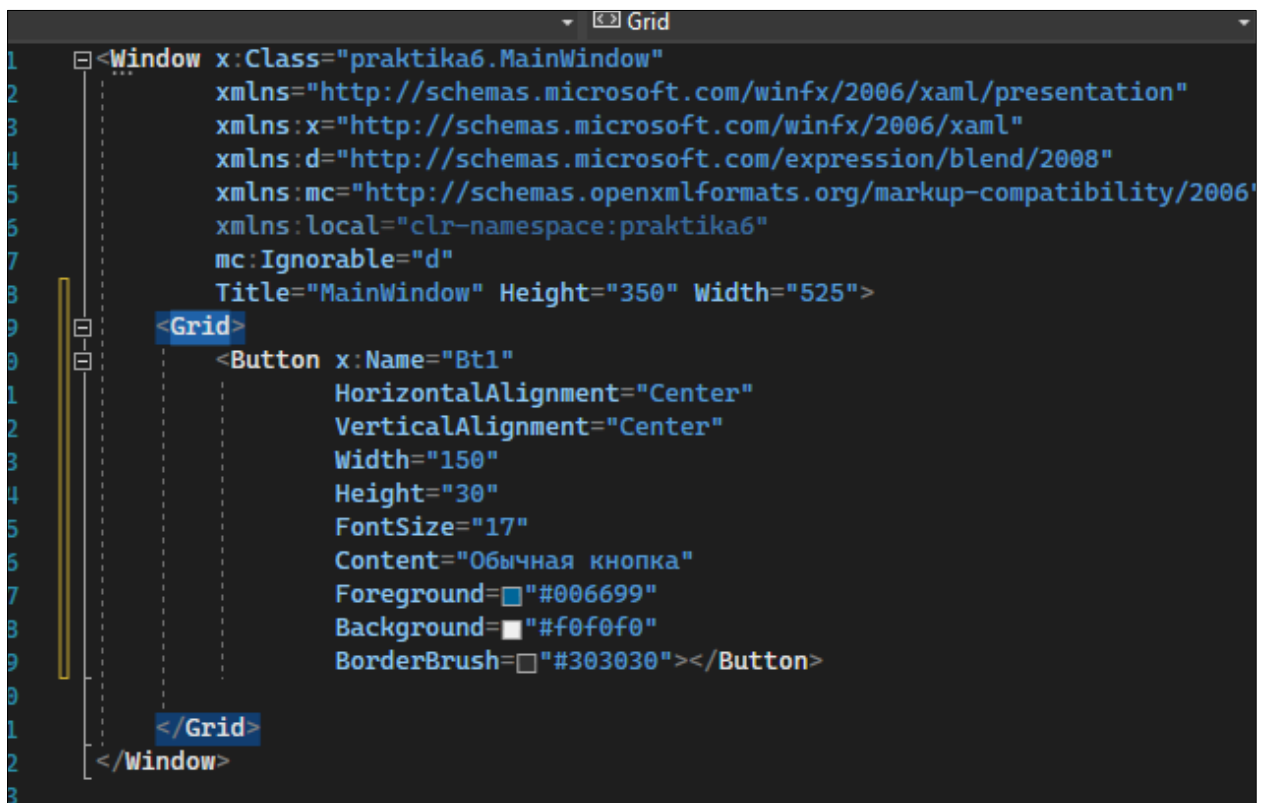


Рисунок 7 — Код кнопки

У элементов есть атрибуты, которые могут принимать определённые значения.

Например: *HorizontalAlignment* может иметь значение *Center*. Это значит, элемент будет выровнен по горизонтали относительно центра. С *VerticalAlignment* идентичная ситуация, только элемент будет выровнен по вертикали относительно центра.

В *Конструкторе* появится кнопка с белым фоном, черными краями, голубым текстом (Рисунок 8). На следующих этапах фон кнопки изменится.

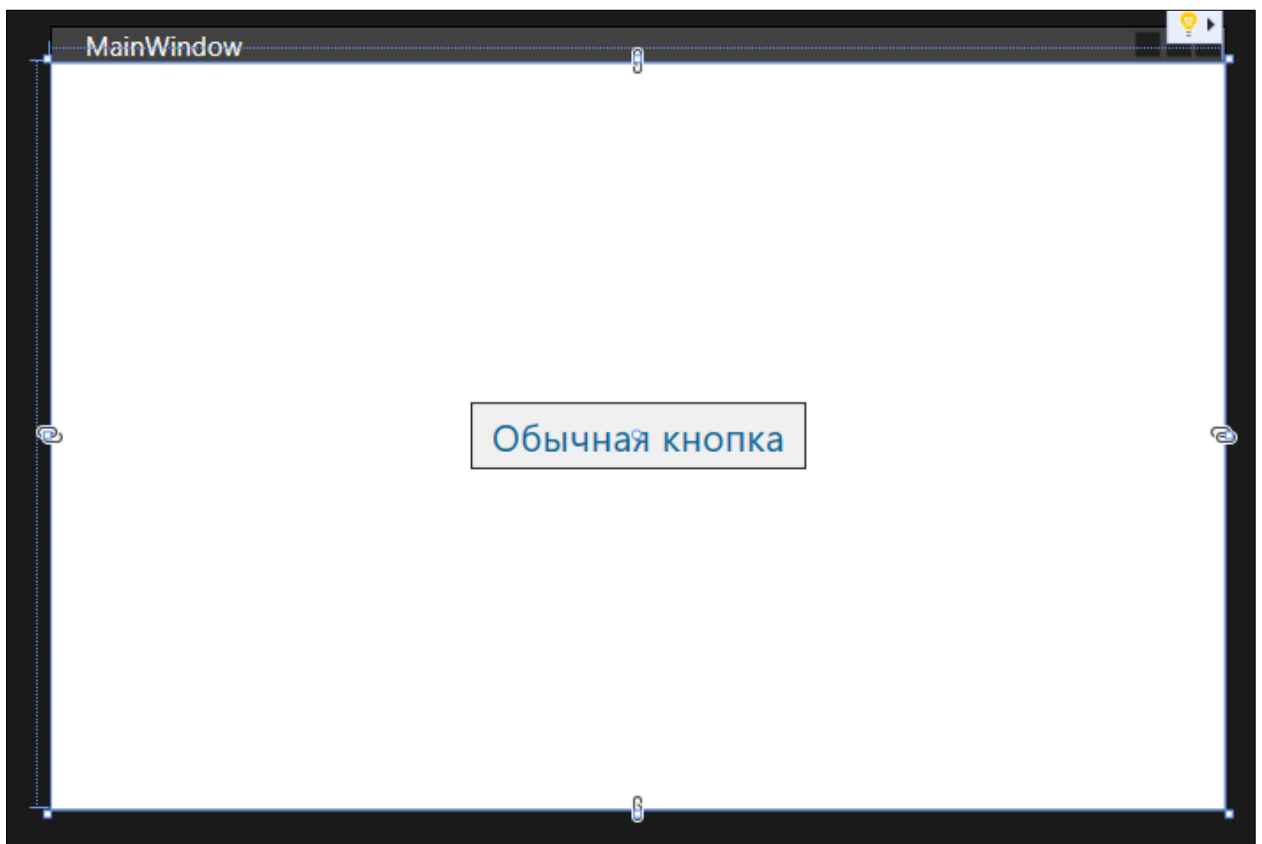


Рисунок 8 — Внешний вид кнопки

Линейная градиентная заливка

В элементе *Button* мы запишем следующую конструкцию:

```
<Button.Background>
```

```
<LinearGradientBrush>
```

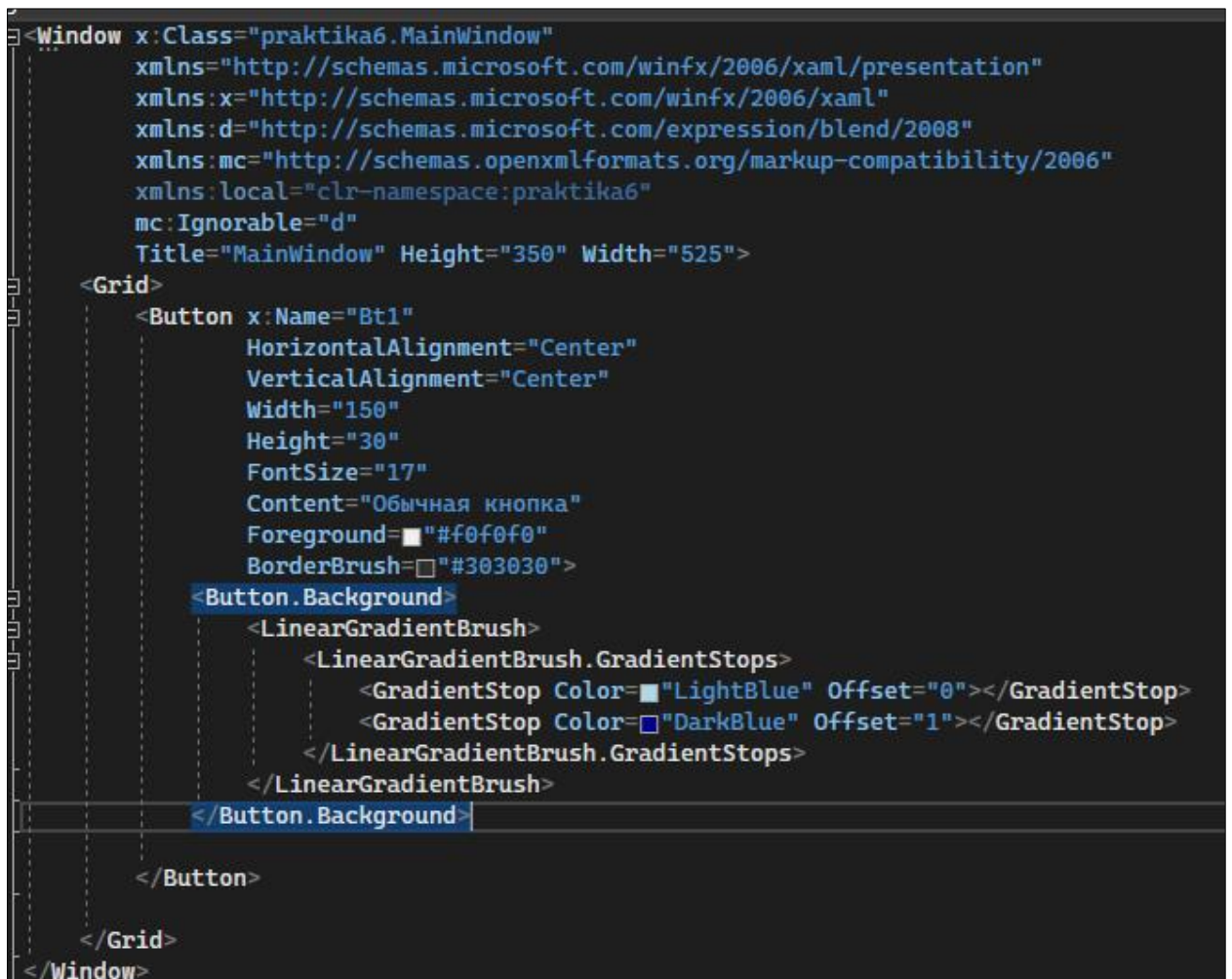
```

        <LinearGradientBrush.GradientStops>
            <GradientStop Color="LightBlue"
Offset="0"></GradientStop>
            <GradientStop Color="DarkBlue"
Offset="1"></GradientStop>
        </LinearGradientBrush.GradientStops>
    </LinearGradientBrush>
</Button.Background>

```

Именно она открывает нам доступ к созданию линейной градиентной заливки. *Offset* помогает указать отрезки от начальной и конечной точки.

Также для этого есть рисунок (Рисунок 9).



```

<Window x:Class="praktika6.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:praktika6"
  mc:Ignorable="d"
  Title="MainWindow" Height="350" Width="525">
    <Grid>
        <Button x:Name="Bt1"
            HorizontalAlignment="Center"
            VerticalAlignment="Center"
            Width="150"
            Height="30"
            FontSize="17"
            Content="Обычная кнопка"
            Foreground="■" "#f0f0f0"
            BorderBrush="□" "#303030">
            <Button.Background>
                <LinearGradientBrush>
                    <LinearGradientBrush.GradientStops>
                        <GradientStop Color="■" "LightBlue" Offset="0"></GradientStop>
                        <GradientStop Color="□" "DarkBlue" Offset="1"></GradientStop>
                    </LinearGradientBrush.GradientStops>
                </LinearGradientBrush>
            </Button.Background>
        </Button>
    </Grid>
</Window>

```

Рисунок 9 — Код линейной градиентной заливки кнопки

Внешний вид кнопки изменится из белого фона на линейный градиент с цветами *LightBlue* и *DarkBlue* (Рисунок 10).

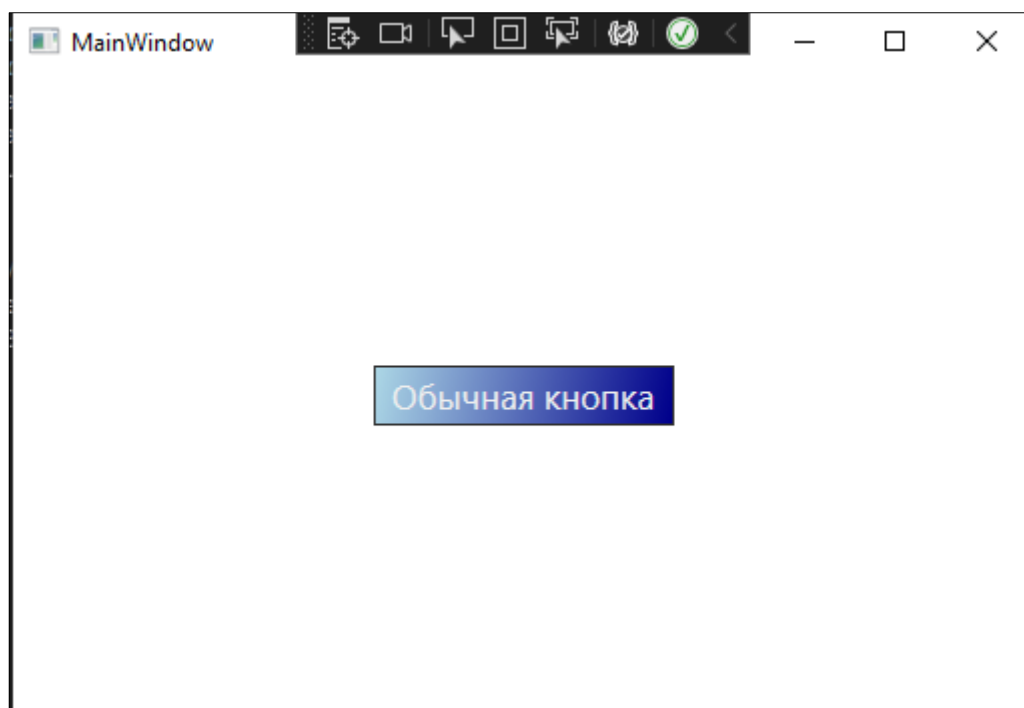


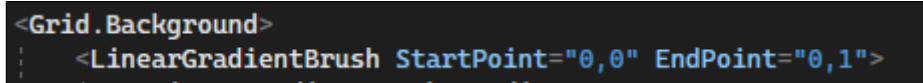
Рисунок 10 — Окончательный внешний вид кнопки

ФОН ОКНА

Для создания вертикального градиента фона в элементе *Grid.Background* нужно указать начальную точку и конечную точку. Код примерно такой:

```
<LinearGradientBrush StartPoint="0,0" EndPoint="0,1"/>
```

Или же как показано на скриншоте (Рисунок 11).



```
<Grid.Background>  
  <LinearGradientBrush StartPoint="0,0" EndPoint="0,1">
```

Рисунок 11 — Указание начальной и конечной точек

Далее повторяем нечто подобное, что вышло с кнопкой, только в результате должен получиться более сложный градиент из четырёх точек, но всё ещё состоящий из двух цветов. Сверху и снизу будет синий, а по середине будет голубой.

Примерно такой код должен получиться для градиента фона:

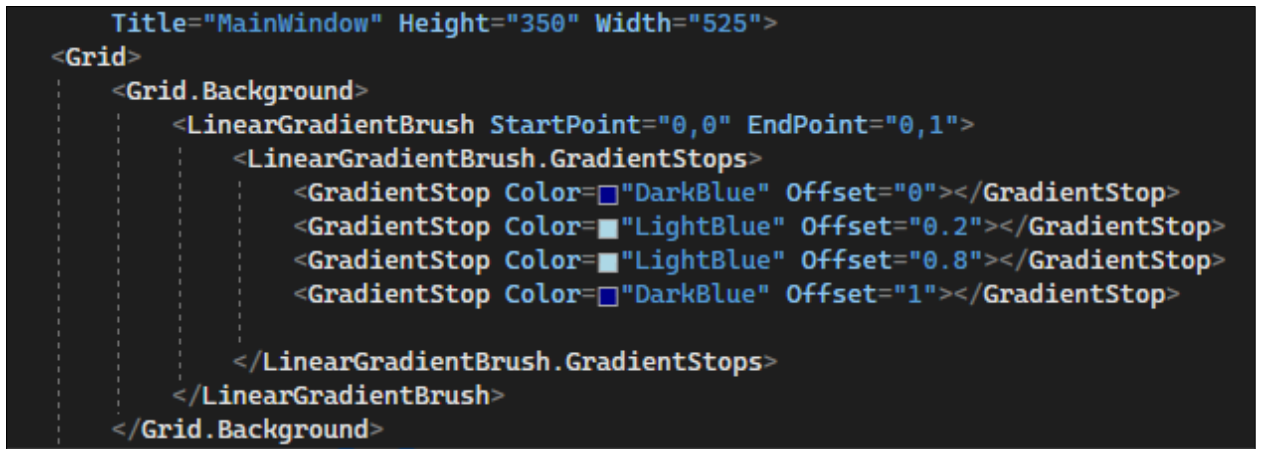
```
<Grid.Background>  
  <LinearGradientBrush StartPoint="0,0" End-  
Point="0,1">  
    <LinearGradientBrush.GradientStops>  
      <GradientStop Color="DarkBlue" Off-  
set="0"></GradientStop>  
      <GradientStop Color="LightBlue" Off-  
set="0.2"></GradientStop>  
      <GradientStop Color="LightBlue" Off-  
set="0.8"></GradientStop>  
      <GradientStop Color="DarkBlue" Off-  
set="1"></GradientStop>  
    </LinearGradientBrush.GradientStops>
```

```

        </LinearGradientBrush>
    </Grid.Background>

```

Или же как на скриншоте (Рисунок 12).



```

Title="MainWindow" Height="350" Width="525">
<Grid>
    <Grid.Background>
        <LinearGradientBrush StartPoint="0,0" EndPoint="0,1">
            <LinearGradientBrush.GradientStops>
                <GradientStop Color="DarkBlue" Offset="0"></GradientStop>
                <GradientStop Color="LightBlue" Offset="0.2"></GradientStop>
                <GradientStop Color="LightBlue" Offset="0.8"></GradientStop>
                <GradientStop Color="DarkBlue" Offset="1"></GradientStop>
            </LinearGradientBrush.GradientStops>
        </LinearGradientBrush>
    </Grid.Background>

```

Рисунок 12 — Код для вертикального градиента фона

В итоге код выглядит следующим образом:

```

<Window x:Class="praktika6.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:praktika6"
        mc:Ignorable="d"
        Title="MainWindow" Height="350" Width="525">
    <Grid>
        <Grid.Background>
            <LinearGradientBrush StartPoint="0,0" End-
Point="0,1">
                <LinearGradientBrush.GradientStops>

```

```

        <GradientStop Color="DarkBlue" Offset="0"></GradientStop>
        <GradientStop Color="LightBlue" Offset="0.2"></GradientStop>
        <GradientStop Color="LightBlue" Offset="0.8"></GradientStop>
        <GradientStop Color="DarkBlue" Offset="1"></GradientStop>

    </LinearGradientBrush.GradientStops>
</LinearGradientBrush>
</Grid.Background>
<Button x:Name="Bt1"
    HorizontalAlignment="Center"
    VerticalAlignment="Center"
    Width="150"
    Height="30"
    FontSize="17"
    Content="Обычная кнопка"
    Foreground="#f0f0f0"
    BorderBrush="#303030">
    <Button.Background>
        <LinearGradientBrush>
            <LinearGradientBrush.GradientStops>
                <GradientStop Color="LightBlue"
Offset="0"></GradientStop>
                <GradientStop Color="DarkBlue"
Offset="1"></GradientStop>
            </LinearGradientBrush.GradientStops>

```

```
</LinearGradientBrush>  
</Button.Background>  
  
</Button>  
  
</Grid>  
</Window>
```

Конечный результат в *Конструкторе* состоит из фона и кнопки. Оба эти элемента с градиентами (Рисунок 13).

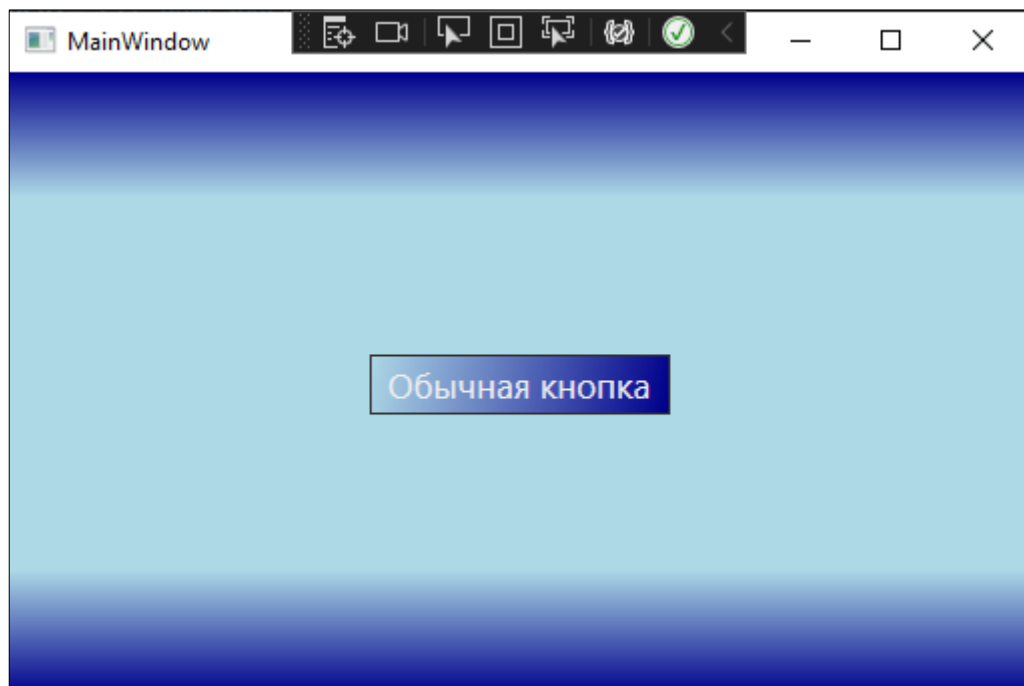


Рисунок 13 — Конечный результат

ВЫВОД

Вывод: мы научились работать с WPF, узнали, как и где писать код для создания элементов, научились создавать градиенты и элементы в WPF.