

## Практическое занятие №4

### Тема: Изменение внешнего вида окна

Окна необычной формы часто являются товарным знаком современных прикладных приложений вроде редакторов фотографий, программ для создания кинофильмов и MP3-проигрывателей; скорее всего, они будут встречаться в WPF-приложениях даже более часто.

В создании базового приложения нестандартной формы в WPF нет ничего сложного. Однако создание привлекательного профессионально выглядящего окна необычной формы требует немалых усилий – и, нередко, привлечения талантливого дизайнера графики для создания эскизов и фоновой графики.

Базовая процедура для создания окна нестандартной формы подразумевает выполнение следующих шагов:

- 1) Установите для свойства `Window.AllowsTransparency` значение `true`.
- 2) Установите для свойства `Window.WindowStyle` значение `None`, чтобы скрыть не клиентскую область окна (рамку голубого цвета). Если этого не сделать, при попытке показать окно появится ошибка `InvalidOperationException`.
- 3) Установите для фона (свойства `Background`) прозрачный цвет (цвет `Transparent`, значение альфа-канала которого равно нулю). Или же сделайте так, чтобы для фона использовалось изображение, имеющее прозрачные области (с нулевым значением альфа-канала).

Эти три шага эффективно удаляют стандартный внешний вид окна. Для обеспечения эффекта окна необычной формы далее необходимо предоставить какое-то непрозрачное содержимое, имеющее нужную форму. Здесь возможны перечисленные ниже варианты:

- 1) Предоставить фоновую графику, используя файл такого формата, который поддерживает прозрачность. Например, для фона можно

использовать файл PNG. Это простой прямолинейный подход, и он очень удобен, если приходится работать с дизайнерами, которые не разбираются в XAML. Однако из-за того, что окно будет визуализироваться с большим количеством пикселей и более высокими системными параметрами DPI фоновая графика может приобрести искаженный вид. Это также может представлять проблему и в случае разрешения пользователю изменять размеры окна.

2) Использовать доступные в WPF функции для рисования формы, чтобы создать фон с векторным содержимым. Такой подход исключает потерю качества, какими бы ни были размеры окна и настройка DPI системы. Однако в этом случае наверняка потребуется использовать средство проектирования, поддерживающее XAML, такое как Expression Blend.

3) Использовать более простой WPF-элемент, имеющий необходимую форму. Например, окно с замечательными скругленными углами можно создать с помощью элемента Border. Такой подход позволяет создавать окна с современным внешним видом в стиле Office без применения каких-либо дизайнерских навыков.

Покажем на примере, как переопределить шаблон окна, чтобы оно имело нестандартную прямоугольную форму.

Воспользуемся третьим вариантом изменения формы, т.е. используем элемент Border со скругленными углами для придания окну отчетливого внешнего вида.

Для этого необходимо указать следующие параметры:

**CornerRadius="0,150,0,150"** – устанавливает радиус скругления по четырем углам (если задать только один параметр, то получим окружность);

**Opacity="0.7"** – устанавливает прозрачность окна;

**Background="LightBlue"** – устанавливает цвет окна.

В результате имеем следующий 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="Круглое окно" Height="250" Width="300"
    AllowsTransparency="True" WindowStyle="None"
Background="Transparent">
    <Window.Template>
        <ControlTemplate TargetType="Window">
            <Border Name="newBorder" CornerRadius="0,150,0,150"
Opacity="0.7" Background="LightBlue">
                <Grid>
                    <Grid.RowDefinitions>
                        <RowDefinition Height="Auto" />
                        <RowDefinition />
                        <RowDefinition Height="Auto" />
                    </Grid.RowDefinitions>
                    <!--Заголовок-->
                    <TextBlock Text="{TemplateBinding Title}"
FontWeight="Bold" HorizontalAlignment="Center"

MouseLeftButtonDown="TextBlock_MouseLeftButtonDown"/>
                    <!--Основное содержание-->
                    <Border Grid.Row="1">
                        <AdornerDecorator>
                            <ContentPresenter />
                        </AdornerDecorator>
                    </Border>
                    <!--Элемент захвата и изменения размера - работает
только для прямоугольных окон-->
                    <ResizeGrip Grid.Row="2"
HorizontalAlignment="Right" VerticalAlignment="Bottom"
Visibility="Collapsed" IsTabStop="False"

/>
                </Grid>
            </Border>
        </ControlTemplate>
    </Window.Template>
    <Grid>
        <Button x:Name="closeButton" Content="Закреть"
Click="closeButton_Click"
Width="80" Height="30" Background="LightPink" />
    </Grid>
</Window>

```

Важно здесь отметить установку трех свойств элемента Window:

- AllowsTransparency: позволяет сделать форму прозрачной, невидимой
- WindowStyle: позволяет убрать у окна стиль при установке значения «None»
- Background: устанавливает прозрачный фон с помощью значения Transparent

И также в файле кода C# изменим код окна, определив необходимые обработчики событий:

```
private void closeButton_Click(object sender, RoutedEventArgs e)
{
    this.Close();
}

//Служит для перемещения окна за заголовок формы
private void TextBlock_MouseLeftButtonDown(object sender,
MouseButtonEventArgs e)
{
    this.DragMove();
}
```

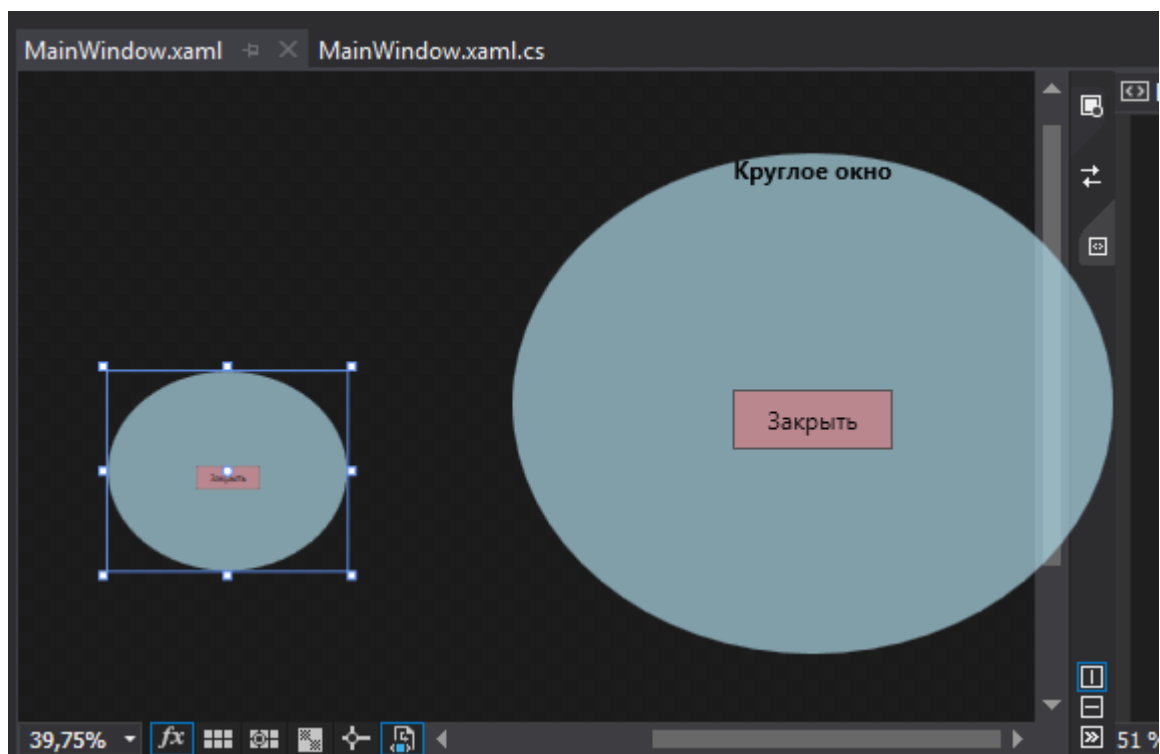


Рисунок – Результат работы программы

## Индивидуальные задания

Согласно своему варианту (таблица 4.1) создать приложение с нестандартной формой главного окна.

Таблица 4.1 – Индивидуальное задание

Вариант	Вид формы главного окна
1	Астраханская область
2	Республика Башкортостан
3	Владимирская область
4	Волгоградская область
5	Воронежская область
6	Ивановская область
7	Республика Калмыкия
8	Кировская область
9	Курганская область
10	Липецкая область
11	Республика Марий Эл
12	Республика Мордовия
13	Нижегородская область
14	Омская область
15	Оренбургская область
16	Пензенская область
17	Пермский край
18	Ростовская область
19	Рязанская область
20	Самарская область
21	Саратовская область
22	Свердловская область
23	Тамбовская область
24	Республика Татарстан
25	Тюменская область
26	Республика Удмуртия
27	Ульяновская область
28	Ханты-Мансийский автономный округ - Югра
29	Челябинская область
30	Республика Чувашия