# Intro

## Intro

<!--x:Class="WPFApplication.MainWindow" - определение класса с кодом окна

Подключения пространств имен для создания элементов управления в разметке.

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

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

<Window x:Class="WPFApplication.MainWindow"

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

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

Title="Главное окно"

Height="350"

Width="525">

<Window.Background>

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

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

<GradientStop Color="#FFA3BB0C" Offset="1"/>

</LinearGradientBrush>

</Window.Background>

<Grid>

<!--Блок текста-->

<TextBlock Width="100"

Height="24"

Margin="210,93,207,202"

Foreground="Red"

Text="Введите свое имя">

</TextBlock>

<!--Поле ввода-->

<TextBox Name="textBoxName"

Width="100"

Height="24"

Margin="202,139,201,148" Text="" AcceptsTab="False">

</TextBox>

<!--Кнопка-->

<Button Width="100"

Height="24"

Content="Ok"

Margin="202,182,200,105"

Click="Button\_Click">

<Button.Background>

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

<GradientStop Color="#FFF3F3F3" Offset="0"/>

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

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

<GradientStop Color="#FF80D639" Offset="1"/>

</LinearGradientBrush>

</Button.Background>

</Button>

</Grid>

</Window>

## Gradient Rectangle

<Window x:Class="PropertiesInMarkUp.MainWindow"

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

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

Title="Установка свойств в разметке" Height="280" Width="525">

<StackPanel>

<!--Установка значения свойства Fill через атрибут-->

<Rectangle Fill="Green" Width="100" Height="100">

</Rectangle>

<!--Установка значения свойства Fill через вложенный элемент-->

<Rectangle Width="100" Height="100" Margin="10">

<Rectangle.Fill>

<LinearGradientBrush>

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

<GradientStop Color="#FFE2B82B" Offset="1"></GradientStop>

</LinearGradientBrush>

</Rectangle.Fill>

</Rectangle>

</StackPanel>

</Window>

## Events

<Window x:Class="EventsInMarkup.MainWindow"

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

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

Title="Установка события в разметке. События мыши." Height="350" Width="525">

<StackPanel>

<!--Прямоугольникам установлены обработчики на события мыши:

MouseEnter - курсор попал в область прямоугольника.

MouseLeave - курсор вышел за область прямоугольника. -->

<Rectangle MinHeight="100" MinWidth="100" Fill="Green" Margin="10"

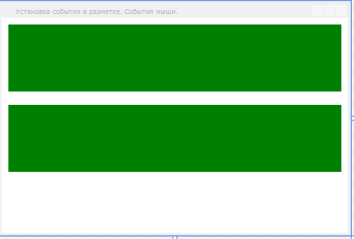
MouseEnter="Rectangle\_MouseEnter" MouseLeave="Rectangle\_MouseLeave"></Rectangle>

<Rectangle MinHeight="100" MinWidth="100" Fill="Green" Margin="10"

MouseEnter="Rectangle\_MouseEnter" MouseLeave="Rectangle\_MouseLeave"></Rectangle>

</StackPanel>

</Window>



## Grid

<Window x:Class="\_8Ball.MainWindow"

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

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

Title="Магический 8 Ball" Height="350" Width="525">

<!--

Объект Grid Определяет гибкую область сетки, состоящую из столбцов и строк.

-->

<Grid>

<!--

RowDefinitions - Коллекция которая определяет

строки для отображения - RowDefinition.

-->

<Grid.RowDefinitions>

<!--

Создаем три строки(RowDefinition) в коллекции - RowDefinitions.

(Height="100\*"):

100 - Фиксированные размер.

100\* - Указывает, что начальный размер ряда значение до звздочки, но размер может пропорционально менятся при изменении размера окна.

Auto - Указывает что размер данного элемента будет подобран под размер

самого большого объекта в контейнере.

-->

<RowDefinition Height="100\*" />

<RowDefinition Height="Auto" />

<RowDefinition Height="100\*" />

</Grid.RowDefinitions>

<!--

Используем сложную кисть для заливки фона градиентом.

Добавляем для этого дескриптор - Background.

-->

<Grid.Background>

<!--

LinearGradientBrush - Заполняет область линейным градиентом.

Линейный градиент определяет градиент вдоль прямой линии.

Конечные точки линии определяются свойствами StartPoint и EndPoint

линейного градиента.

Кисть LinearGradientBrush рисует свои GradientStops вдоль этой линии.

-->

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

<!-- Коллекция свойств GradientStop -->

<LinearGradientBrush.GradientStops>

<!--

GradientStop имеет свойства:

Offset - Задает позици цвета.

Color - Задает цвет.

-->

<GradientStop Offset="1" Color="#FFEEEEEE" />

<GradientStop Offset="0.881" Color="#FF302F2F" />

</LinearGradientBrush.GradientStops>

</LinearGradientBrush>

</Grid.Background>

<!--

<TextBox Имя объкта.

Name="txtQuestion"

Выравнивание элемента по горизонтали относительно контейнера.

HorizontalAlignment="Stretch"

Выравнивание элемента по вертикали относительно контейнера.

VerticalAlignment="Stretch"

Выравнивание элемента относительно сторон контейнера.

Margin="10"

Пренос на новую строку в случае если текст не помещается в одной строке.

TextWrapping="Wrap"

Шрифт который будем использовать.

FontFamily="Verdana"

Размер используемого шрифта.

FontSize="20"

Строка в контейнере в которой будет находится данный элемент.

Grid.Row="0"

Цвет используемого шрифта.

Foreground="#FF646464"

Свойство текст.

Text="[Введите свой вопрос.]" />

-->

<TextBox Name="txtQuestion"

HorizontalAlignment="Stretch"

VerticalAlignment="Stretch"

Margin="10"

TextWrapping="Wrap"

FontFamily="Verdana"

FontSize="20"

Grid.Row="0"

Foreground="#FF646464"

Text="[Введите свой вопрос.]" />

<!--

IsDefault - Получает или задает значение, указывающее, является ли Button

кнопкой по умолчанию.

-->

<Button HorizontalAlignment="Left"

VerticalAlignment="Center"

Margin="10,0,10,0"

Padding="5"

Grid.Row="1"

Content="Ответ на вопрос"

Click="Answer\_Click"

IsDefault="True" />

<TextBox Name="txtAnswer"

HorizontalAlignment="Stretch"

VerticalAlignment="Stretch"

Margin="10"

TextWrapping="Wrap"

FontFamily="Verdana"

FontSize="20"

IsReadOnly="True"

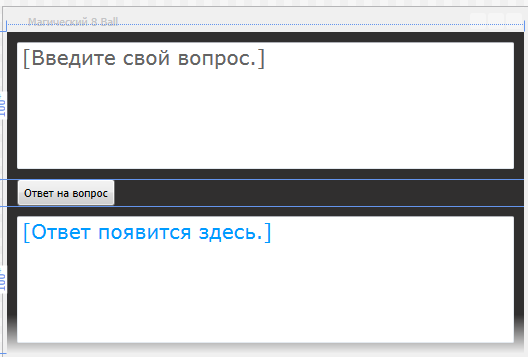
Grid.Row="2"

Foreground="#FF009BFF"

Text="[Ответ появится здесь.]" />

</Grid>

</Window>



# Layout

## Grid

<Grid ShowGridLines="True">

<!--Определение колонок таблицы-->

<Grid.ColumnDefinitions>

<ColumnDefinition></ColumnDefinition>

<ColumnDefinition></ColumnDefinition>

</Grid.ColumnDefinitions>

<!--Определение строк таблицы-->

<Grid.RowDefinitions>

<RowDefinition></RowDefinition>

<RowDefinition></RowDefinition>

</Grid.RowDefinitions>

<TextBlock Grid.Row="0" Grid.Column="0"

Padding="10" Margin="20" Background="LightCoral">

Grid.Row="0" Grid.Column="0"</TextBlock>

<TextBlock Grid.Row="0" Grid.Column="1"

Padding="10" Margin="20" Background="LightCyan">

Grid.Row="0" Grid.Column="1"</TextBlock>

<TextBlock Grid.Row="1" Grid.Column="0" Grid.ColumnSpan="2"

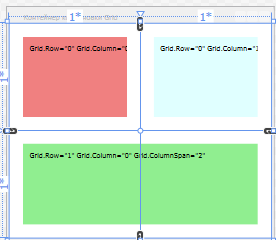
Padding="10" Margin="20" Background="LightGreen">

Grid.Row="1" Grid.Column="0" Grid.ColumnSpan="2"

</TextBlock>

</Grid>

</Window>



## Grid size

<Grid>

<Grid.RowDefinitions>

<!--Высота строки 100-->

<RowDefinition Height="100"></RowDefinition>

<!--Высота строки изменяется пропорционально с изменением окна-->

<RowDefinition Height="\*"></RowDefinition>

<!--Высота равна высоте самого большого элемента определенного в строке-->

<RowDefinition Height="Auto"></RowDefinition>

</Grid.RowDefinitions>

<TextBlock Grid.Row="0"

Background="LightBlue" Width="1000">TextBlock 1 (Height="100")</TextBlock>

<TextBlock Grid.Row="1"

Background="LightGreen">TextBlock 2 (Height="\*")</TextBlock>

<TextBlock Grid.Row="2"

Background="LightCyan" Height="50">TextBlock 3 (Height="Auto")</TextBlock>

</Grid>

</Window>

## StackPanel

<!--StackPanel - Располагает дочерние элементы в одну строку, которую можно ориентировать по горизонтали или по вертикали -->

<StackPanel Orientation="Horizontal">

<TextBlock Background="LightGoldenrodYellow" Padding="10" Margin="10">TextBlock 1</TextBlock>

<TextBlock Background="LightBlue" Padding="10" Margin="10">TextBlock 2</TextBlock>

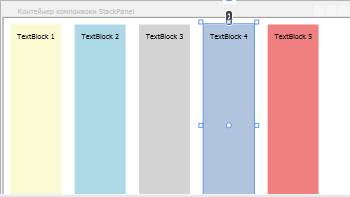
<TextBlock Background="LightGray" Padding="10" Margin="10">TextBlock 3</TextBlock>

<TextBlock Background="LightSteelBlue" Padding="10" Margin="10">TextBlock 4</TextBlock>

<TextBlock Background="LightCoral" Padding="10" Margin="10">TextBlock 5</TextBlock>

</StackPanel>

</Window>



## DockPanel

<!--

DockPanel - определяет область, в которой можно упорядочить дочерние элементы

горизонтально или вертикально относительно друг друга.

DockPanel.Dock - указывает, к какой стороне контейнера следует привязать контрол.

-->

<DockPanel LastChildFill="False">

<TextBlock DockPanel.Dock="Top"

Background="LightBlue"

Padding="10">

DockPanel.Dock="Top"</TextBlock>

<TextBlock DockPanel.Dock="Top"

Background="Aqua"

Padding="10">

Toolbar</TextBlock>

<TextBlock DockPanel.Dock="Bottom"

Background="LightGreen"

Padding="10">

DockPanel.Dock="Bottom"</TextBlock>

<TextBlock DockPanel.Dock="Right"

Background="LightCyan"

Padding="10">

DockPanel.Dock="Right"</TextBlock>

<TextBlock DockPanel.Dock="Left"

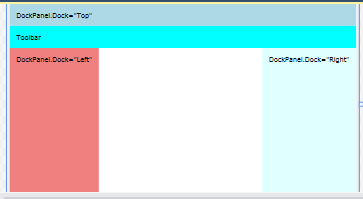
Background="LightCoral"

Padding="10">

DockPanel.Dock="Left"</TextBlock>

</DockPanel>

</Window>



## WrapPanel

<!--WrapPanel - Размещает дочерние элементы последовательно слева направо,

разбивая содержимое до следующей строки на краю содержащего окна.

Последующее размещение происходит последовательно сверху вниз или справа налево,

в зависимости от значения свойства Orientation. -->

<WrapPanel Orientation="Horizontal">

<TextBlock Background="LightGoldenrodYellow" Padding="10" Margin="10">TextBlock 1</TextBlock>

<TextBlock Background="LightBlue" Padding="10" Margin="10">TextBlock 2</TextBlock>

<TextBlock Background="LightGray" Padding="10" Margin="10">TextBlock 3</TextBlock>

<TextBlock Background="LightSteelBlue" Padding="10" Margin="10">TextBlock 4</TextBlock>

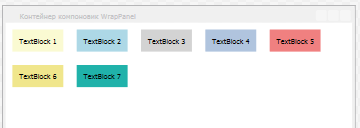
<TextBlock Background="LightCoral" Padding="10" Margin="10">TextBlock 5</TextBlock>

<TextBlock Background="Khaki" Padding="10" Margin="10">TextBlock 6</TextBlock>

<TextBlock Background="LightSeaGreen" Padding="10" Margin="10">TextBlock 7</TextBlock>

</WrapPanel>

</Window>



## UniformGrid

<!--UniformGrid - Предоставляет способ размещения элементов в сетки,

при котором все ячейки имеют одинаковый размер.-->

<UniformGrid Rows="2" Columns="2">

<TextBlock Background="LightGoldenrodYellow" Padding="10" Margin="10">TextBlock 1</TextBlock>

<TextBlock Background="LightBlue" Padding="10" Margin="10">TextBlock 2</TextBlock>

<TextBlock Background="LightGray" Padding="10" Margin="10">TextBlock 3</TextBlock>

<TextBlock Background="LightSteelBlue" Padding="10" Margin="10">TextBlock 4</TextBlock>

</UniformGrid>

</Window>



## Canvas

<!--Canvas - Определяет область, в рамках которой можно явно расположить дочерние элементы

путем использования координат, являющихся относительными к области Canvas.-->

<Canvas>

<TextBlock Canvas.Left="30" Canvas.Top="60"

Height="100" Background="LightCoral"

TextWrapping="WrapWithOverflow" Padding="10">

Canvas.Left="30" Canvas.Top="60"

</TextBlock>

<TextBlock Canvas.Left="160" Canvas.Top="120"

Height="100" Width="200" Background="LightGreen"

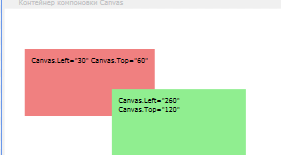
TextWrapping="WrapWithOverflow" Padding="10">

Canvas.Left="260" Canvas.Top="120"

</TextBlock>

</Canvas>

</Window>



## IncCanvas

<Window x:Class="\_07\_IncCanvas.MainWindow"

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

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

Title="Контейнер компоновик InkCanvas" Height="350" Width="525" Loaded="Window\_Loaded">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="Auto"></RowDefinition>

<RowDefinition></RowDefinition>

<RowDefinition Height="22"></RowDefinition>

</Grid.RowDefinitions>

<StackPanel Margin="5" Orientation="Horizontal">

<TextBlock Margin="5" Text="Режим редактирования: " />

<ComboBox Name="lstEditingMode"

VerticalAlignment="Center"

HorizontalAlignment="Stretch"

Margin="0,0,10,0" />

</StackPanel>

<!--

InkCanvas - Определяет область, которая рисует и отображает рукописные штрихи.

-->

<InkCanvas Name="inkCanvas"

Grid.Row="1"

Background="LightBlue"

EditingMode="{Binding ElementName=lstEditingMode, Path=SelectedItem}"

Gesture="inkCanvas\_Gesture">

<Button InkCanvas.Left="10" InkCanvas.Top="10">Button</Button>

<TextBox InkCanvas.Left="100" InkCanvas.Top="32">TextBox</TextBox>

</InkCanvas>

<TextBox Name="gestureName"

HorizontalAlignment="Stretch"

VerticalAlignment="Bottom"

Grid.Row="2" />

</Grid>

</Window>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

private void inkCanvas\_Gesture(object sender, InkCanvasGestureEventArgs e)

{

String gestures = "";

// Выборка "предпологаемых" гестур.

foreach (GestureRecognitionResult res in e.GetGestureRecognitionResults())

gestures += res.ApplicationGesture.ToString() + "; ";

// Отображаем название гестуры.

gestureName.Text = gestures;

}

private void Window\_Loaded(object sender, RoutedEventArgs e)

{

// Выборка всех режимов редактирования InkCanvas.

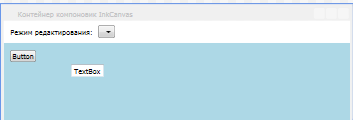
foreach (InkCanvasEditingMode mode in Enum.GetValues(typeof(InkCanvasEditingMode)))

lstEditingMode.Items.Add(mode);

lstEditingMode.SelectedItem = inkCanvas.EditingMode;

}

}



## GridSplitter

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="\*" />

<RowDefinition Height="\*" />

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*" />

<ColumnDefinition Width="Auto" />

<ColumnDefinition Width="\*" />

</Grid.ColumnDefinitions>

<TextBlock Grid.Row="0" Grid.Column="0" Background="LightGoldenrodYellow" Padding="10">

TextBlock 1</TextBlock>

<TextBlock Grid.Row="1" Grid.Column="0" Background="LightGray" Padding="10">

TextBlock 2</TextBlock>

<GridSplitter Grid.Row="0" Grid.Column="1" Grid.RowSpan="2" Width="3"

HorizontalAlignment="Stretch" VerticalAlignment="Stretch" Background="Black"></GridSplitter>

<!--Вложенный Grid-->

<Grid Grid.Row="0" Grid.Column="2" Grid.RowSpan="2">

<Grid.RowDefinitions>

<RowDefinition Height="\*"></RowDefinition>

<RowDefinition Height="Auto"></RowDefinition>

<RowDefinition Height="\*"></RowDefinition>

</Grid.RowDefinitions>

<TextBlock Grid.Row="0" Grid.Column="0" Background="LightGreen" Padding="10">

TextBlock 1</TextBlock>

<TextBlock Grid.Row="2" Grid.Column="0" Background="LightBlue" Padding="10">

TextBlock 2</TextBlock>

<GridSplitter Grid.Row="1" Grid.Column="0" Height="3"

HorizontalAlignment="Stretch" VerticalAlignment="Stretch" Background="Black"></GridSplitter>

</Grid>

</Grid>

</Window>

