

WPF

(Windows Presentation Foundation)

Особенности WPF

XAML (eXtensible Application Markup Language) - язык разметки, используемый для инициализации объектов в технологиях на платформе .NET

Использование векторной графики

Базируется на DirectX

Декларативная реализация с помощью XAML

Трехмерные объекты

Привязка данных

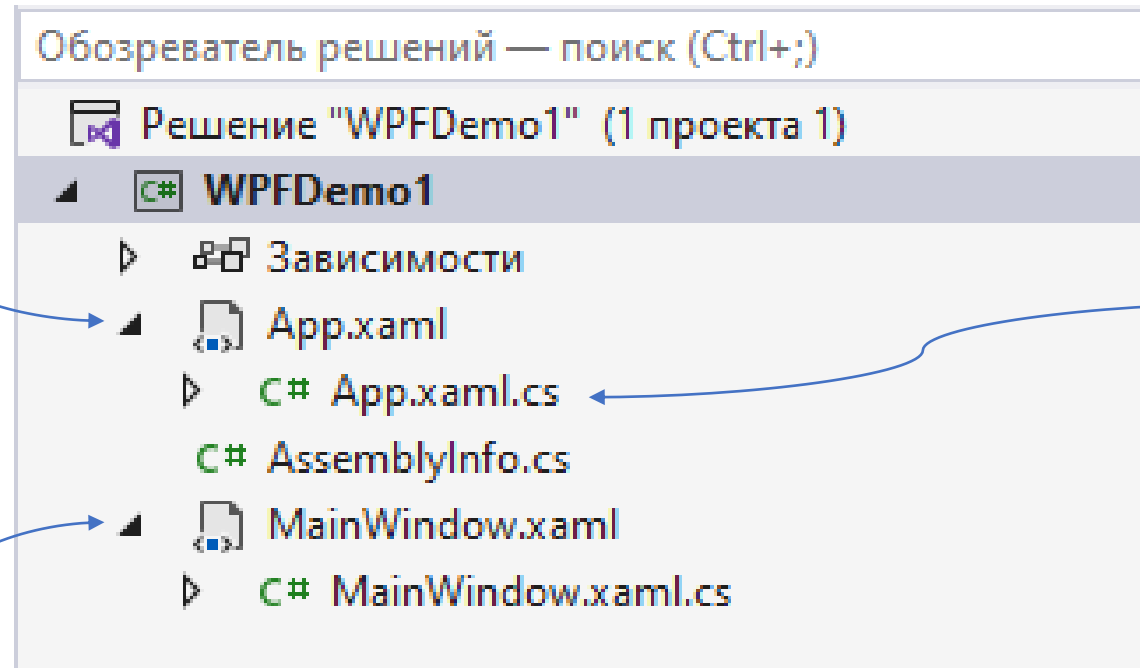
Использование стилей, тем

Обратная связь с WinForms

Структура проекта

Задаёт файл окна,
открывающееся
при запуске

Одно из
множества
будущих окон
приложения



глобальный файл для
всего приложения

Пространства имен XAML

Содержит компоненты
WPF (пространство имен
по умолчанию)

Содержит параметры
компонентов характерные
для XAML

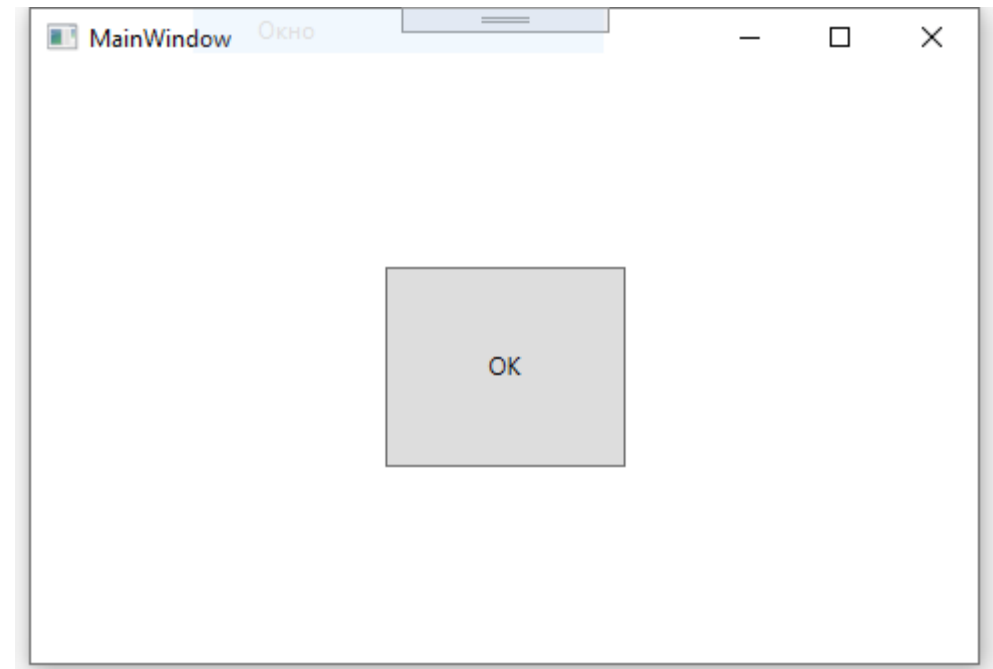
```
<Window x:Class="WPFDemo1.MainWindow"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
xmlns:local="clr-namespace:WPFDemo1"
mc:Ignorable="d"
Title="MainWindow" Height="450" Width="800">
    <Grid>
    </Grid>
</Window>
```

обеспечивает
режим
совместимости
разметок XAML

Локальное
пространство имен
(текущий проект)

- Компоновка реализуется с помощью контейнеров
- Каждый контейнер обладает собственной логикой компоновки
- При объявлении элемента внутри окна (без панелей) он размещается в центре окна
- Окно может содержать только один элемент компоновки

```
<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">
  <Button Height="100" Width="120">OK</Button>
</Window>
```



Использование спецсимволов в XAML

СИМВОЛ	Код
<	<
>	>
&	&
"	"

```
<Button Content="<"Hello">" />
```



```
<Button Content="&lt;&quot;Hello&quot;&gt;" />
```

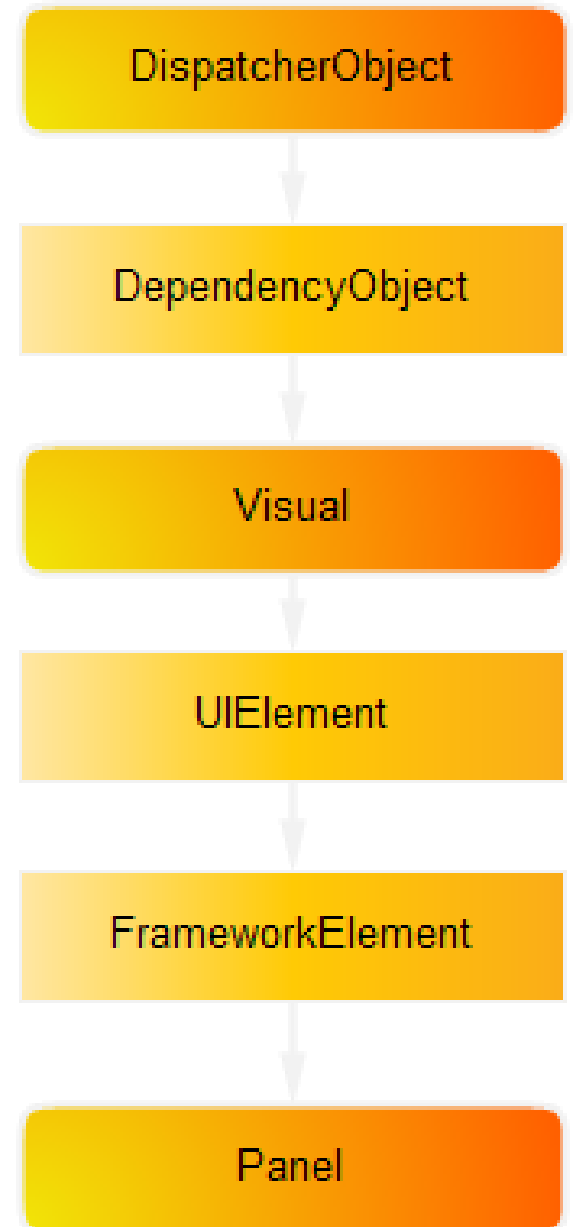


```
<"Hello">
```

Контейнеры компоновки

- предназначены для построения гибкого адаптивного визуального представления

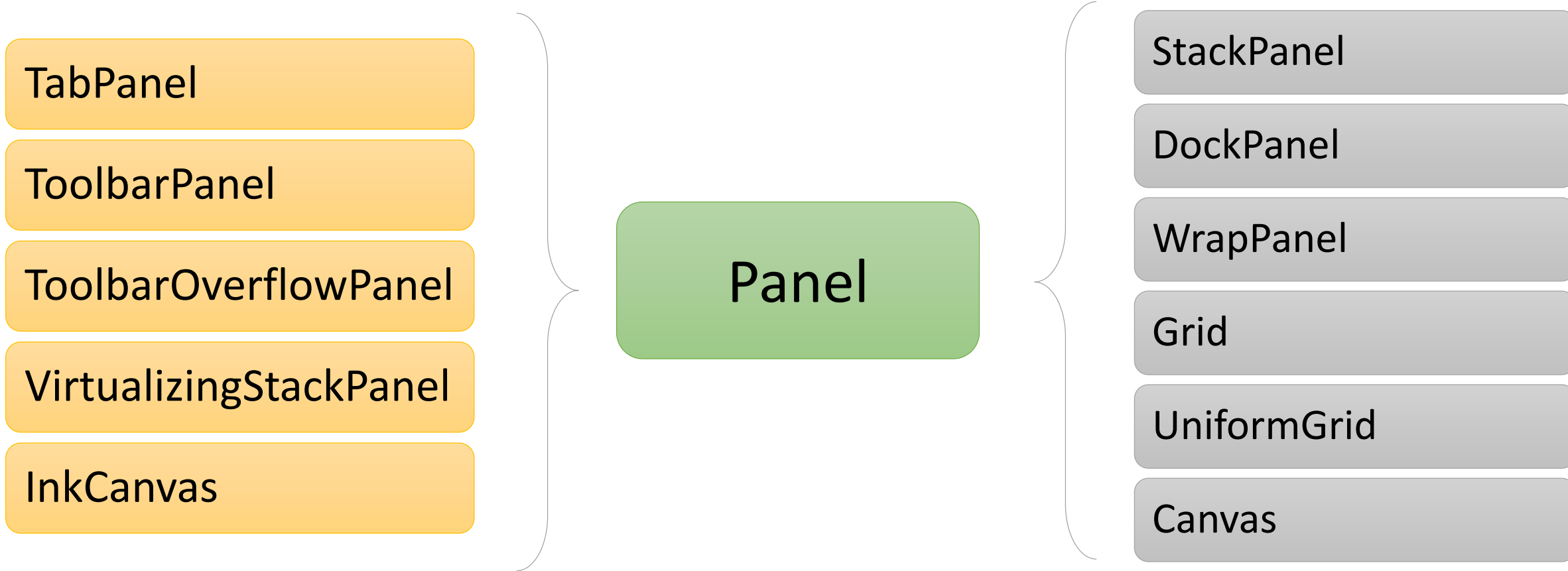
Все контейнеры компоновки WPF являются панелями, которые унаследованы от абстрактного класса **System.Windows.Controls.Panel**.



Свойства класса Panel

- Background - Кисть, используемая для рисования фона панели.
- Children - Коллекция элементов, находящихся в панели.
- IsItemsHost - Булевское значение, которое равно true, если панель используется для отображения элементов, ассоциированных с ItemsControl

Производные класса Panel



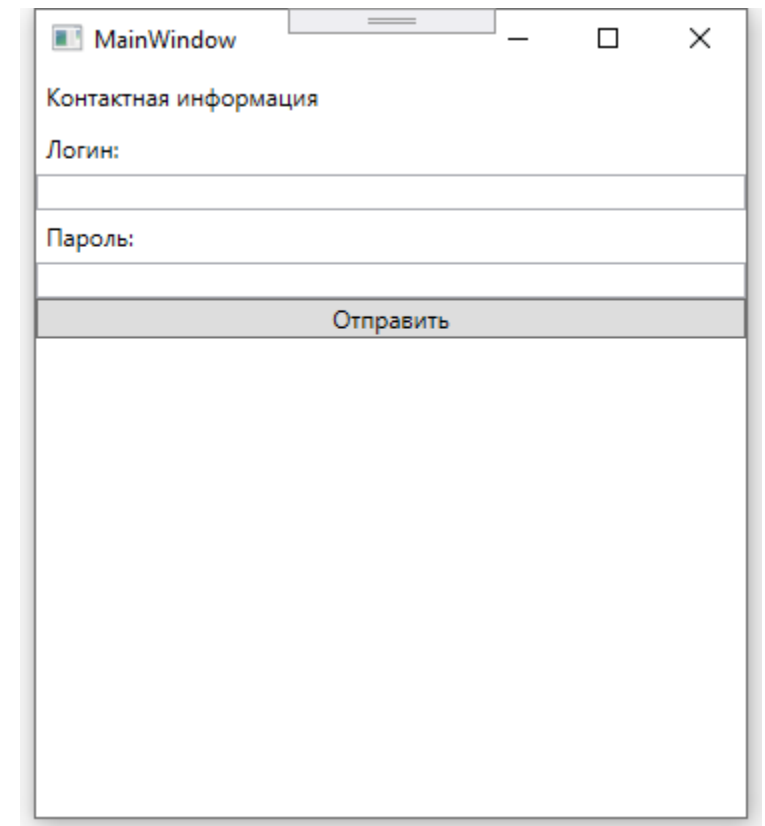
StackPanel

- Элемент компоновки, который отображает содержимое в виде стопки (горизонтально или вертикально – по умолчанию)

```
<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">
  <StackPanel>
    <Label>Контактная информация</Label>
    <Label>Логин:</Label>
    <TextBox></TextBox>
    <Label>Пароль:</Label>
    <TextBox></TextBox>
    <Button>Отправить</Button>
  </StackPanel>

</Window>

<StackPanel Orientation="Horizontal">
```

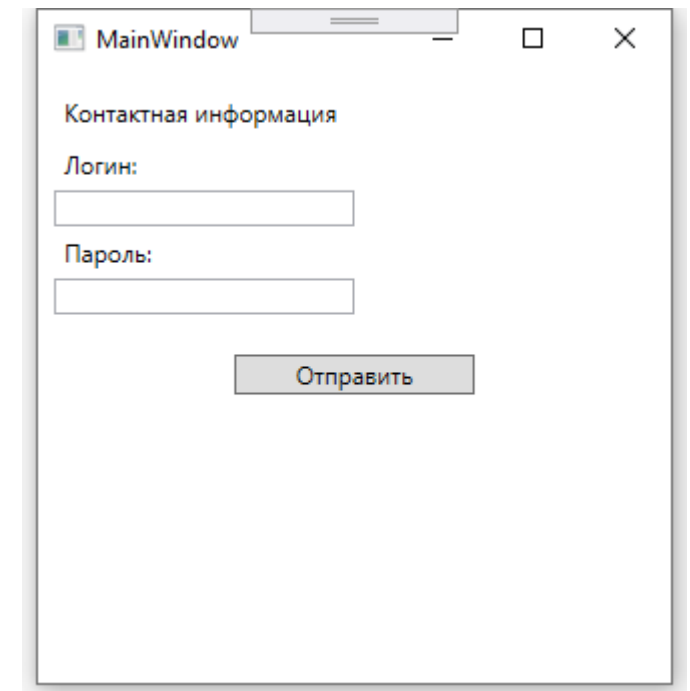


Свойства компоновки

- HorizontalAlignment, VerticalAlignment - Определяет позиционирование дочернего элемента внутри контейнера компоновки,
- Margin - Добавляет некоторое пространство вокруг элемента.
- MinWidth, MinHeight - Устанавливает минимальные размеры элемента.
- MaxWidth, MaxHeight - Устанавливает максимальные размеры элемента.
- Width, Height - Явно устанавливают размеры элемента

```
<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">
  <StackPanel Margin="8">
    <Label>Контактная информация</Label>
    <Label>Логин:</Label>
    <TextBox MaxWidth="250" MinWidth="150" HorizontalAlignment="Left"></TextBox>
    <Label>Пароль:</Label>
    <TextBox MaxWidth="250" MinWidth="150" HorizontalAlignment="Left"></TextBox>
    <Button Margin="20" MaxWidth="120" HorizontalAlignment="Stretch" MinWidth="100"
Content="Отправить"></Button>
  </StackPanel>

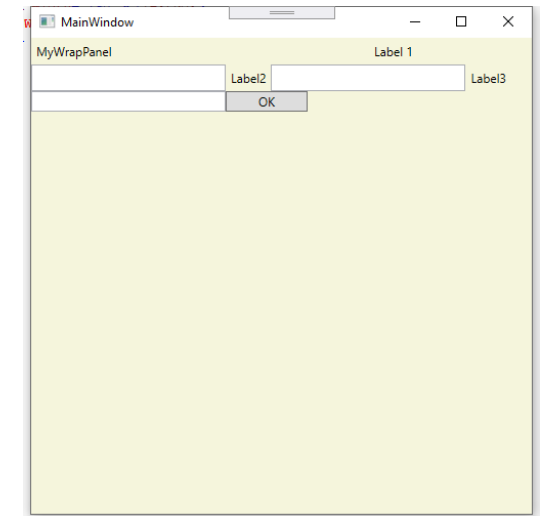
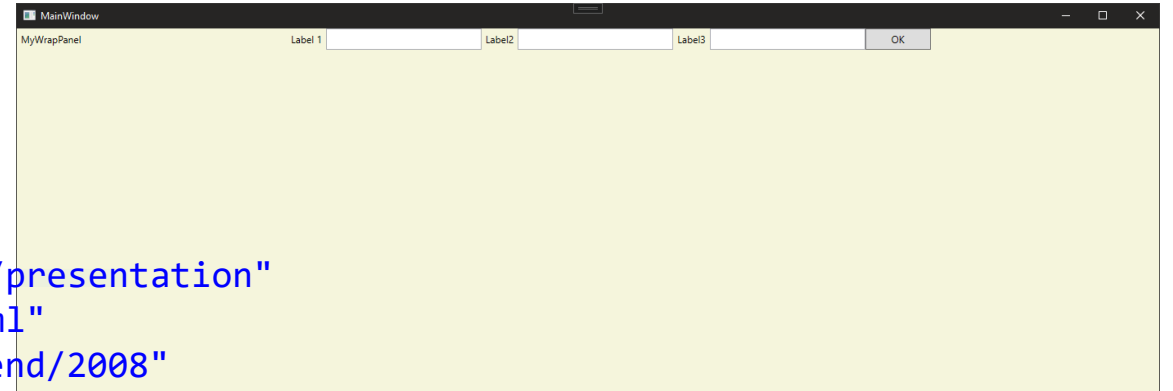
</Window>
```



WrapPanel

позволяет определить содержимое, которое обтекает панель при изменении размеров окна

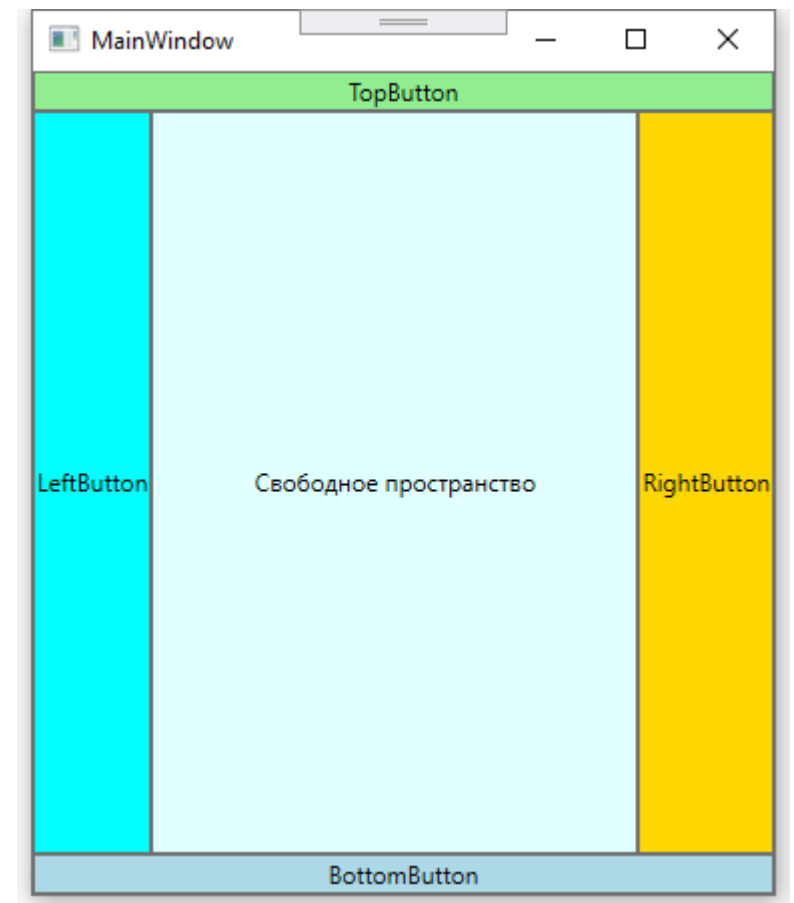
```
<Window x:Class="WpfApp5.MainWindow"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
  xmlns:local="clr-namespace:WpfApp5"
  mc:Ignorable="d"
  Title="MainWindow" Height="450" Width="800">
  <WrapPanel Background="Beige">
    <Label Content="MyWrapPanel" Width="330" />
    <Label Content="Label 1" />
    <TextBox Width="190"></TextBox>
    <Label Content="Label2"></Label>
    <TextBox Width="190"></TextBox>
    <Label Content="Label3"></Label>
    <TextBox Width="190"></TextBox>
    <Button Width="80">OK</Button>
  </WrapPanel>
</Window>
```



DockPanel

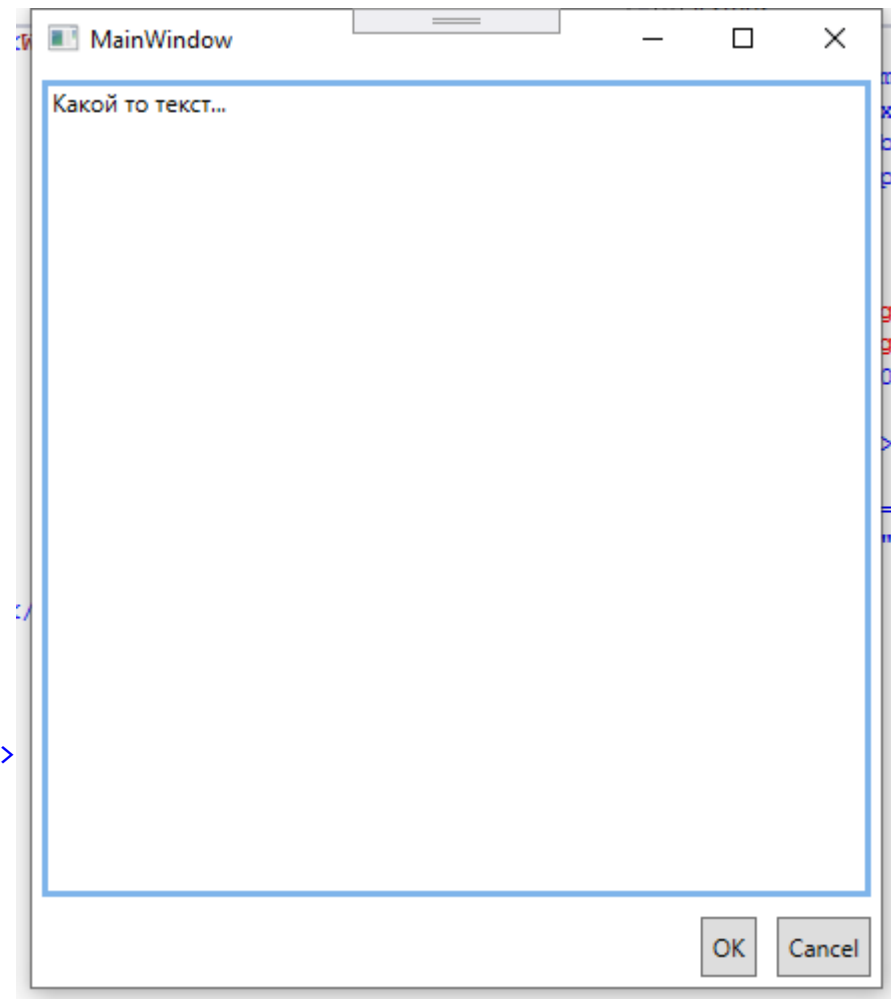
панель растягивает элементы управления вдоль одной из внешних границ.
Дочерние элементы определяют сторону через свойство Dock

```
<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">
  <DockPanel>
    <Button DockPanel.Dock="Top" Background="LightGreen">TopButton</Button>
    <Button DockPanel.Dock="Bottom" Background="LightBlue">BottomButton</Button>
    <Button DockPanel.Dock="Left" Background="Aqua">LeftButton</Button>
    <Button DockPanel.Dock="Right" Background="Gold">RightButton</Button>
    <Button Background="LightCyan">Свободное пространство</Button>
  </DockPanel>
</Window>
```



Вложение контейнеров

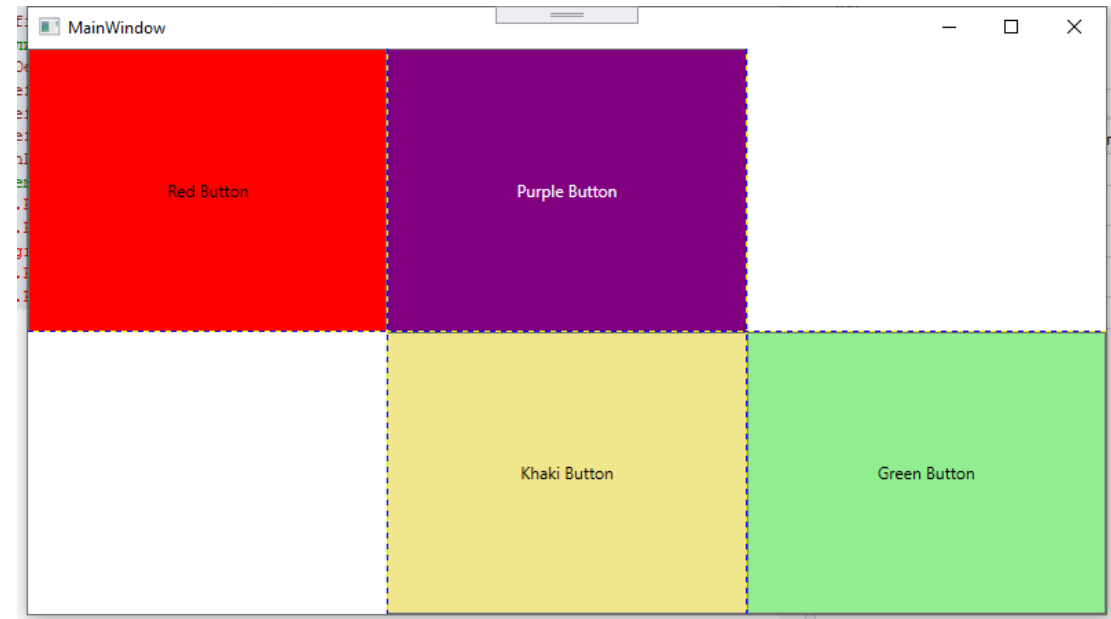
```
<Window x:Class="WpfApp5.MainWindow"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
  xmlns:local="clr-namespace:WpfApp5"
  mc:Ignorable="d"
  Title="MainWindow" Height="450" Width="800">
  <DockPanel LastChildFill="True" MinWidth="400" MinHeight="300">
    <StackPanel DockPanel.Dock="Bottom" HorizontalAlignment="Right"
      Orientation="Horizontal" MinHeight="40">
      <Button Margin="5" Padding="5">OK</Button>
      <Button Margin="5" Padding="5">Cancel</Button>
    </StackPanel>
    <TextBox DockPanel.Dock="Top" Margin="5" FontSize="12"
      BorderBrush="LightBlue" BorderThickness="3">Какой то текст...</TextBox>
  </DockPanel>
</Window>
```



Grid

- располагает элементы в невидимой сетке строк и колонок

```
<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">
  <Grid ShowGridLines="True">
    <!-- Устанавливаем две строки -->
    <Grid.RowDefinitions>
      <RowDefinition/>
      <RowDefinition/>
    </Grid.RowDefinitions>
    <!-- Устанавливаем три столбца -->
    <Grid.ColumnDefinitions>
      <ColumnDefinition/>
      <ColumnDefinition/>
      <ColumnDefinition/>
    </Grid.ColumnDefinitions>
    <!-- Размещаем элементы в сетке -->
    <Button Grid.Row="0" Grid.Column="0" Background="Red">Red Button</Button>
    <Button Grid.Row="0" Grid.Column="1" Background="Purple"
      Foreground="White">Purple Button</Button>
    <Button Grid.Row="1" Grid.Column="1" Background="Khaki">Khaki Button</Button>
    <Button Grid.Row="1" Grid.Column="2" Background="LightGreen">Green Button</Button>
  </Grid>
</Window>
```



Требования к компоновке:

- Элементы не должны иметь явно установленных размеров.
- Элементы не указывают свою позицию в экранных координатах.
- Контейнеры компоновки "разделяют" доступное пространство между своими дочерними элементами.
- Контейнеры компоновки могут быть вложенными.

Самостоятельное задание

- Произвести анализ принципов работы всех контейнеров, их назначения. Результат оформить в виде сводной таблицы
- Создать проект с реализацией адаптивного интерфейса со всеми перечисленными контейнерами (по окну на каждый контейнер). Заполнение контейнера элементами управления на ваш выбор.
- Адаптивность интерфейса подразумевает при изменении размеров окна пропорциональное изменение вложенных элементов