

КОМПОНОВКА ЭЛЕМЕНТОВ В WPF

- **Компоновка** - важнейший фактор, обеспечивающий удобство работы с приложением на различных устройствах
- WPF содержит встроенные панели, которые помогают избежать неприятностей с компоновкой.

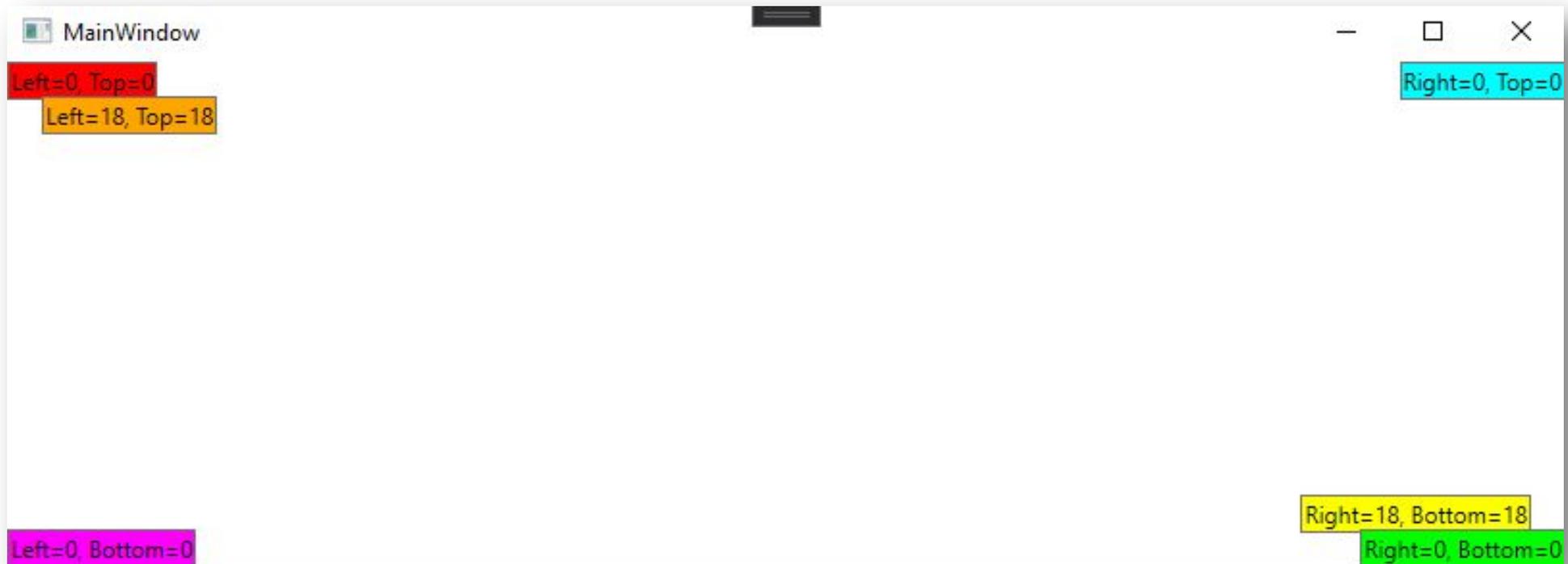
Основные встроенные панели:

- 1) Canvas
- 2) StackPanel
- 3) WrapPanel
- 4) DockPanel
- 5) Grid

Панель Canvas

- **Canvas(холст)** - самая простая панель. Canvas поддерживает только «классическое» позиционирование элементов путем явного задания координат. Панель Canvas позволяет задавать координаты относительно любого угла.
- Позиционирование элемента на холсте осуществляется с помощью присоединенных свойств: Left, Top, Right и Bottom.

```
<Canvas>
  <Button Background=■"Red">Left=0, Top=0</Button>
  <Button Canvas.Left="18" Canvas.Top="18" Background=■"Orange">Left=18, Top=18</Button>
  <Button Canvas.Right="18" Canvas.Bottom="18" Background=■"Yellow">Right=18, Bottom=18</Button>
  <Button Canvas.Right="0" Canvas.Bottom="0" Background=■"Lime">Right=0, Bottom=0</Button>
  <Button Canvas.Right="0" Canvas.Top="0" Background=■"Aqua">Right=0, Top=0</Button>
  <Button Canvas.Left="0" Canvas.Bottom="0" Background=■"Magenta">Left=0, Bottom=0</Button>
</Canvas>
```



ПРЕДУПРЕЖДЕНИЕ

- Для элемента нельзя задавать более двух присоединенных свойств Canvas!
- При попытке одновременно установить свойства Canvas.Left и Canvas.Right последнее будет проигнорировано. А при попытке одновременно установить свойства Canvas.Top и Canvas.Bottom будет проигнорировано Canvas.Bottom. Таким образом, невозможно пристыковать элемент более чем к одному углу холста.

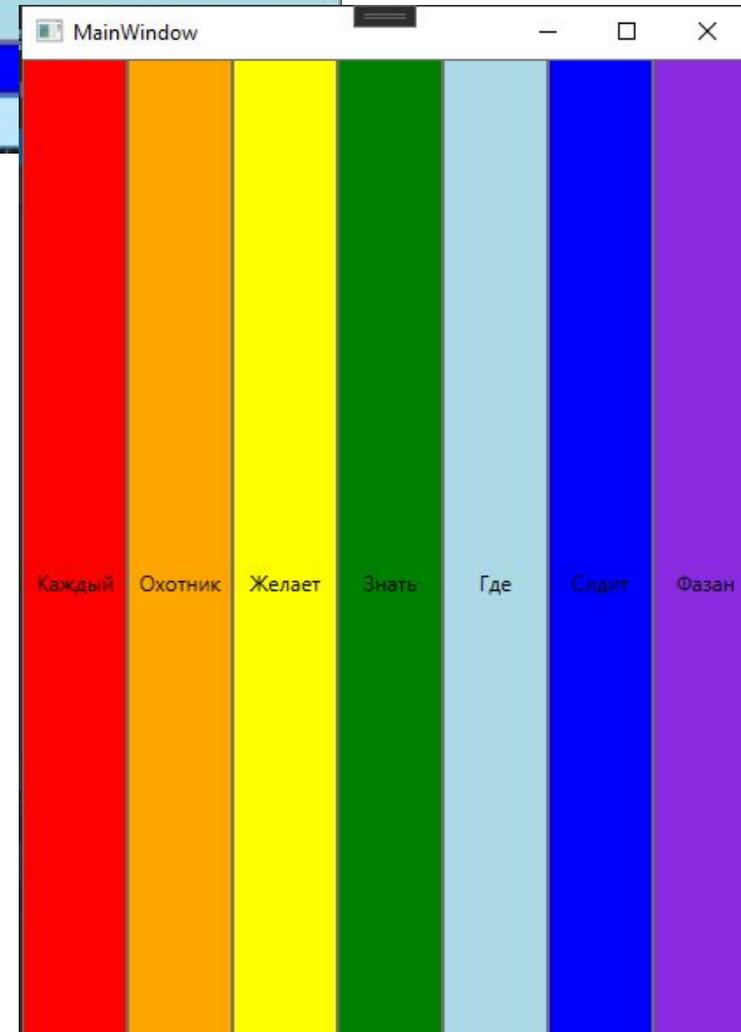
Свойство	Допустимо ли внутри Canvas
Margin	Частично. Для двух сторон, использованных для позиционирования элемента (по умолчанию Top и Left), к значениям присоединенных свойств прибавляются соответствующие значения двух из четырех полей
HorizontalAlignment и VerticalAlignment	Нет. Элементам назначается в точности та величина вертикального выравнивания, которая им необходима
LayoutTransform	Да. Отличается от RenderTransform тем, что при использовании LayoutTransform элементы всегда отстоят на заданное расстояние от выбранного угла Canvas

Панель StackPanel

- Панель StackPanel популярна из-за своей простоты и удобства. Как следует из названия, она последовательно размещает своих потомков в виде стопки.
- **StackPanel и размещение справа налево**
- Если свойство FlowDirection равно RightToLeft, то для панели StackPanel с горизонтальной ориентацией сборка стопки производится справа налево, а не слева направо, как в случае по умолчанию.



- В отсутствие присоединенных свойств, единственный способ организовать дочерние элементы - воспользоваться свойством панели `Orientation`, которое может принимать значение `Horizontal` или `Vertical`. По умолчанию подразумевается ориентация `Vertical`.



Свойство	Допустимо ли внутри Canvas
Margin	Частично. Для двух сторон, использованных для позиционирования элемента (по умолчанию Top и Left), к значениям присоединенных свойств прибавляются соответствующие значения двух из четырех полей
HorizontalAlignment и VerticalAlignment	Частично, поскольку выравнивание игнорируется в направлении сборки стопки (так как дочерним элементам отводится ровно столько места, сколько им необходимо). Если Orientation="Vertical", то игнорируется значение VerticalAlignment. Если Orientation="Horizontal", то игнорируется значение HorizontalAlignment
LayoutTransform	Да. Отличается от RenderTransform тем, что при использовании LayoutTransform оставшиеся в стопке элементы сдвигаются вниз, чтобы освободить место. При комбинировании компоновки Stretch преобразованием RotateTransform или SkewTransform, применяемым в режиме LayoutTransform, растяжение происходит, только если угол кратен

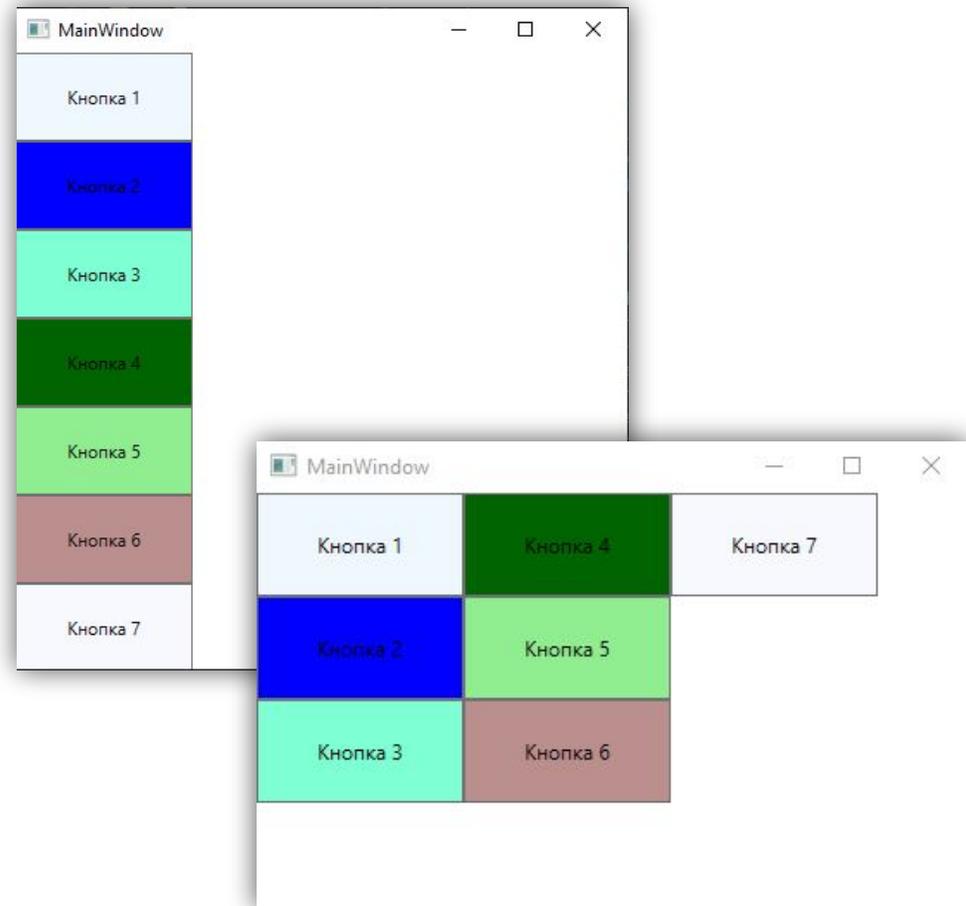
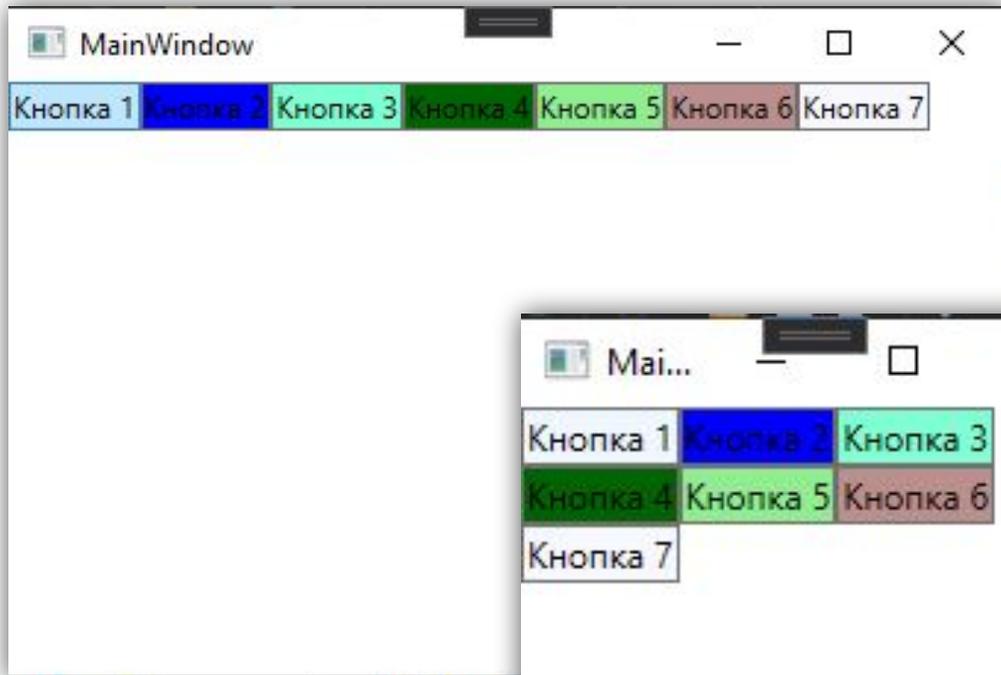
Панель WrapPanel

- Панель WrapPanel похожа на StackPanel. Но помимо организации дочерних элементов в стопку она создает новые строки или столбцы, когда для одной стопки не хватает места. Это полезно для отображения заранее неизвестного числа элементов, когда компоновка должна отличаться от простого списка.
- Как и StackPanel, панель WrapPanel не обладает присоединенными свойствами для управления положением элементов. В классе WrapPanel определены три свойства, контролирующие его поведение.

Свойства:

- **Orientation**- аналогично одноименному свойству `StackPanel` с тем отличием, что по умолчанию подразумевается значение `Horizontal`. Панель с горизонтальной ориентацией выглядит, как вид Эскизы страниц в Проводнике Windows: элементы располагаются один за другим слева направо, а когда место кончается, переходят на следующую строку. Панель с вертикальной ориентацией выглядит, как вид Список в Проводнике Windows: элементы располагаются один под другим, а когда место кончается, начинается новый столбец.
- **ItemHeight**- единая высота для всех дочерних элементов. Каким образом каждый потомок распоряжается этой высотой, зависит от значений его свойств `VerticalAlignment`, `Height` и пр. Элементы, ширина которых превышает `ItemHeight`, отсекаются.
- **ItemWidth**- единая ширина для всех дочерних элементов. Каким образом каждый потомок распоряжается этой шириной, зависит от значений его свойств `HorizontalAlignment`, `Width` и пр. Элементы, высота которых превышает `ItemWidth`, отсекаются.

Orientation = "Horizontal"



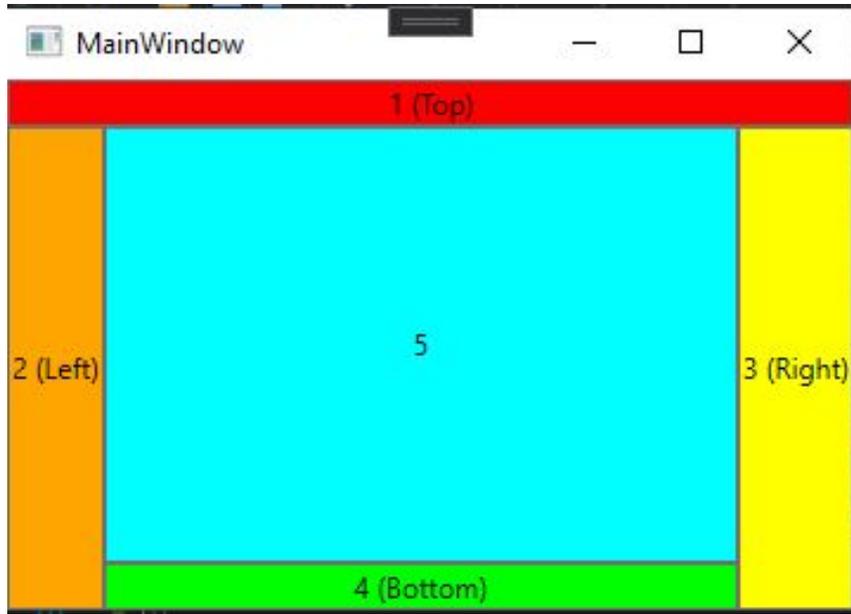
Orientation = "Vertical"

Свойство	Допустимо ли внутри Canvas
Margin	Да. Поля учитываются, когда WrapPanel вычисляет размеры элементов, чтобы определить подразумеваемую по умолчанию ширину или высоту стопки
HorizontalAlignment и VerticalAlignment	Частично. Выравнивание можно задавать в направлении, противоположном направлению роста стопки, как и в случае StackPanel. Но выравнивание может быть полезно и в направлении роста стопки, если значение ItemHeight или ItemWidth таково, что в элементе имеется дополнительное пространство для выравнивания
LayoutTransform	Да. Отличается от RenderTransform тем, что при использовании LayoutTransform оставшиеся элементы сдвигаются, чтобы освободить место, но только если не установлено свойство ItemHeight или ItemWidth (в зависимости от ориентации). При комбинировании компоновки Stretch с преобразованием RotateTransform или SkewTransform, применяемым в режиме LayoutTransform, растяжение происходит, только если угол кратен 90°, как и в случае StackPanel

Панель DockPanel

- Панель DockPanel дает простой способ пристыковки элемента к одной из сторон, растягивая его на всю имеющуюся ширину или высоту. (Отличие от Canvas заключается в том, что элементы пристыковываются не к одному углу, а ко всей стороне.) Кроме того, DockPanel позволяет расположить один элемент так, чтобы он занял все место, свободное от пристыкованных элементов. В классе DockPanel определено присоединенное свойство Dock, с помощью которого дочерние элементы могут управлять своим положением. Оно может принимать четыре значения: Left (подразумевается по умолчанию, если свойство Dock не задано явно), Top, Right и Bottom.
- Отметим, что у свойства Dock нет значения Fill, означающего, что нужно заполнить оставшееся место. Вместо этого действует соглашение о том, что все оставшееся место отдается последнему дочернему элементу, добавленному в DockPanel, если только свойство LastChildFill не равно false. Если LastChildFill равно true (по умолчанию), то значение свойства Dock, заданное для последнего добавленного элемента, игнорируется. Если же оно равно false, то последний элемент можно пристыковать к любой стороне (по умолчанию к левой, Left).

Порядок добавления дочерних элементов на панель имеет значение, потому что каждому потомку выделяется все оставшееся место на стороне, к которой он пристыковывается. (Можно провести аналогию с эгоистом, который, первым сев в кресло в самолете или в аудитории, занимает оба подлокотника.)

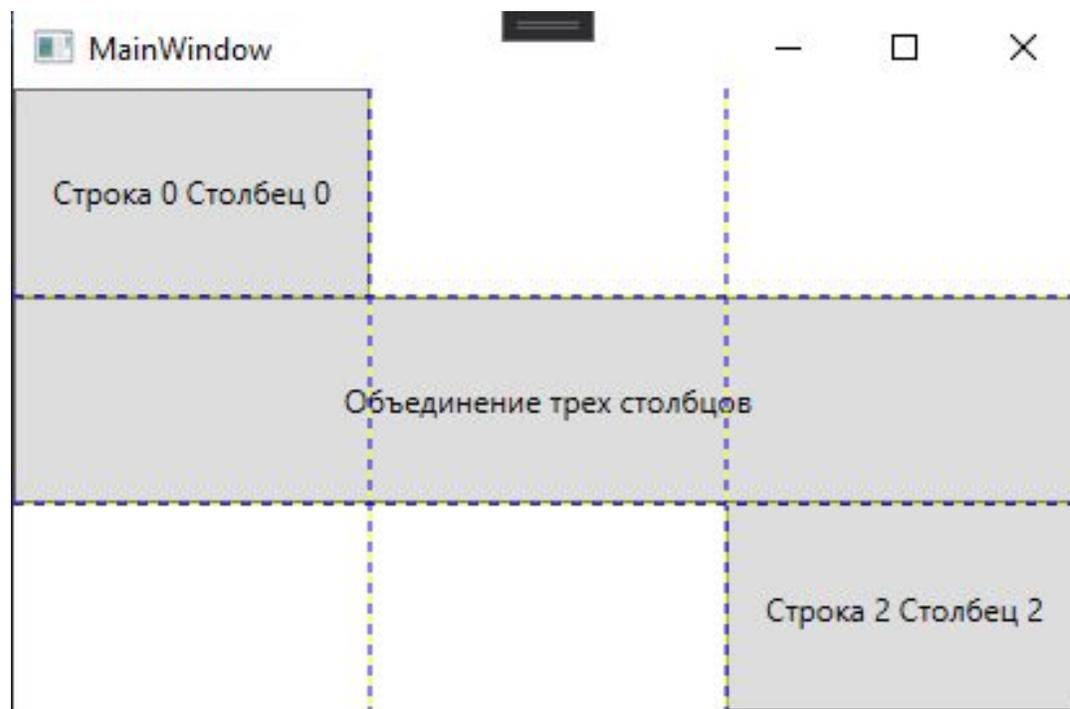


Свойство	Допустимо ли внутри Canvas
Margin	Да. Свойство Margin определяет, сколько места оставлять между элементом и стороной панели, а также промежуток между самими элементами
HorizontalAlignment и VerticalAlignment	Частично. Как и в случае StackPanel, выравнивание в направлении пристыковки игнорируется. Иначе говоря, если Dock равно Left или Right, то не имеет смысла задавать свойство HorizontalAlignment, а если Top или Bottom то свойство VerticalAlignment. Однако для элемента, заполняющего оставшееся пространство, применимы оба свойства, HorizontalAlignment и VerticalAlignment
LayoutTransform	Да. Отличается от RenderTransform тем, что при использовании LayoutTransform оставшиеся элементы сдвигаются, чтобы освободить место. При комбинировании компоновки Stretch с преобразованием RotateTransform или SkewTransform, применяемым в режиме LayoutTransform, растяжение происходит, только если угол кратен 90°, за исключением элемента, заполняющего оставшееся пространство (потому что он может растягиваться в обоих направлениях)

Панель Grid

- Grid(сетка) - самая гибкая из всех панелей и, пожалуй, наиболее употребительная.
- Она позволяет расположить дочерние элементы в несколько строк и несколько столбцов, не полагаясь на режим автоматического переноса (как в WrapPanel). Кроме того, предоставляется ряд интересных способов управления строками и столбцами. Работа с панелью Grid очень напоминает использование элемента TABLE в HTML.
- Он содержит столбцы и строки, количество которых задает разработчик. Для определения строк используется свойство **RowDefinitions**, а для определения столбцов - свойство **ColumnDefinitions**

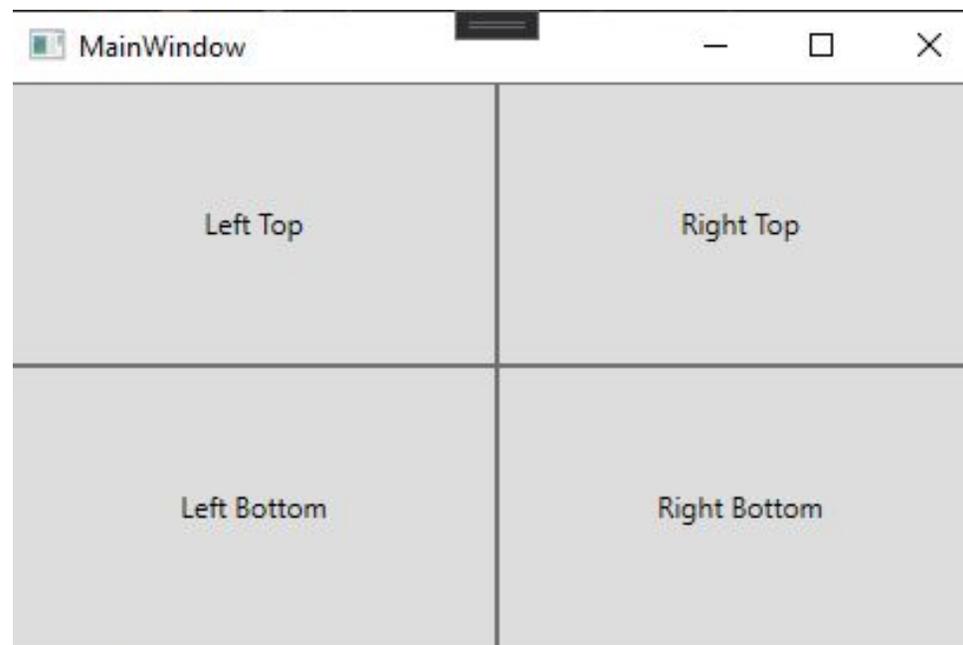
- Чтобы задать позицию элемента управления с привязкой к определенной ячейке Grid, в разметке элемента нужно прописать значения свойств **Grid.Column** и **Grid.Row**, тем самым указывая, в каком столбце и строке будет находиться элемент. Кроме того, если мы хотим растянуть элемент управления на несколько строк или столбцов, то можно указать свойства **Grid.ColumnSpan** и **Grid.RowSpan**, как в следующем примере:



UniformGrid

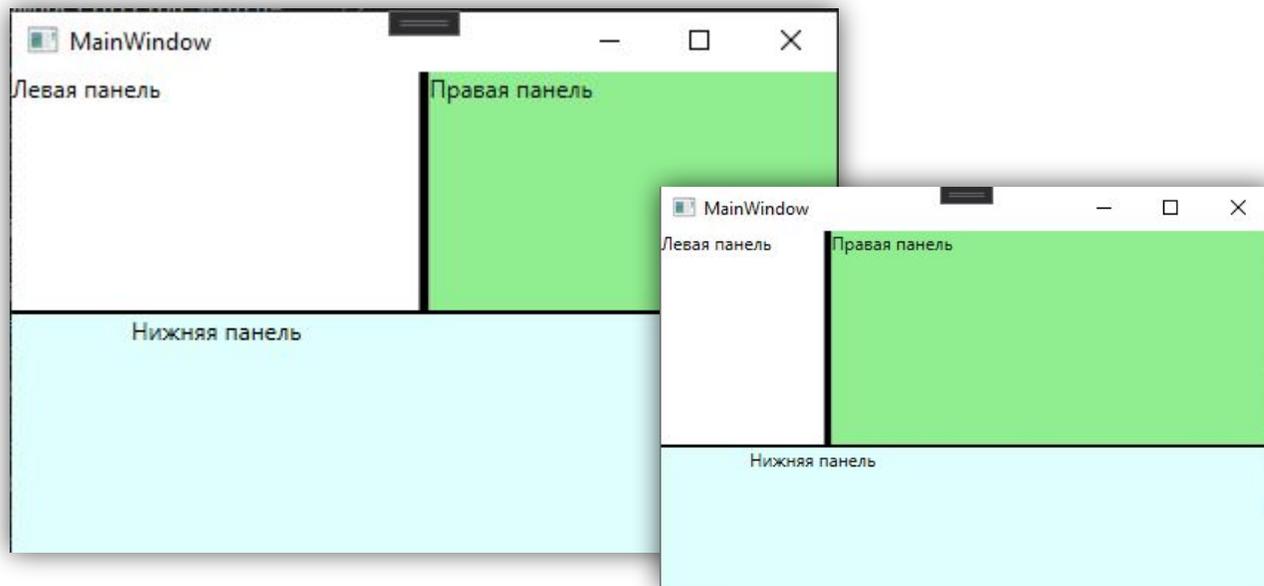
- Аналогичен контейнеру Grid контейнер **UniformGrid**, только в этом случае все столбцы и строки одинакового размера и используется упрощенный синтаксис для их определения:

- ```
<UniformGrid Rows="2" Columns="2">
```
- ```
  <Button Content="Left Top" />
```
- ```
 <Button Content="Right Top" />
```
- ```
  <Button Content="Left Bottom" />
```
- ```
 <Button Content="Right Bottom" />
```
- ```
</UniformGrid>
```



Элемент GridSplitter

- Элемент **GridSplitter** помогает создавать интерфейсы наподобие элемента **SplitContainer** в WinForms, только более функциональные. Он представляет собой некоторый разделитель между столбцами или строками, путем сдвига которого можно регулировать ширину столбцов и высоту строк. В качестве примера можно привести стандартный интерфейс проводника в Windows, где разделительная полоса отделяет древовидный список папок от панели со списком файлов.



Если свойство **ShowsPreview** равно **False**, то изменение границ кнопок будет происходить сразу же при перемещении сплитера. Если же оно равно **True**, тогда изменение границ будет происходить только после того, как перемещение сплитера завершится, и при перемещении сплиттера мы увидим его проекцию.

ВОПРОСЫ:

- 1) Какие виды панелей Вам известны?
- 2) Опишите особенности каждой панели (и её свойства)?
- 3) Придумайте ситуации в которых можно применить ту или иную панель.
- 4) Что такое GridSplitter?

