# Компоновка

Чтобы перейти уже непосредственно к созданию красивых интерфейсов и их компонентов, сначала необходимо познакомиться с компоновкой. Компоновка (layout) представляет собой процесс размещения элементов внутри контейнера. Возможно, вы обращали внимание, что одни программы и веб-сайты на разных экранах с разным разрешением выглядят по-разному: где-то лучше, где-то хуже. В большинстве своем такие программы используют жестко закодированные в коде размеры элементов управления. WPF уходит от такого подхода в пользу так называемого "резинового дизайна", где весь процесс позиционирования элементов осуществляется с помощью компоновки.

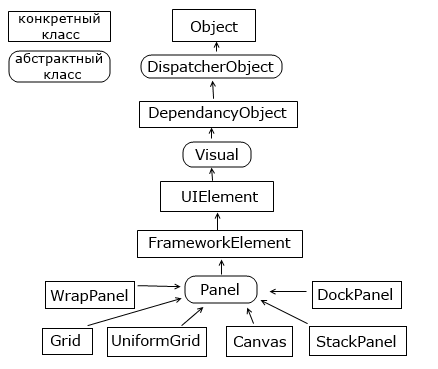
Благодаря компоновке мы можем удобным нам образом настроить элементы интерфейса, позиционировать их определенным образом. Например, элементы компоновки в WPF позволяют при ресайзе - сжатии или растяжении масштабировать элементы, что очень удобно, а визуально не создает всяких шероховатостей типа незаполненных пустот на форме.

В WPF компоновка осуществляется при помощи специальных контейнеров. Фреймворк предоставляет нам следующие контейнеры: **Grid**, **UniformGrid**, **StackPanel**, **WrapPanel**, **DockPanel** и **Canvas**.

Различные контейнеры могут содержать внутри себя другие контейнеры. Кроме данных контейнеров существует еще ряд элементов, такие как TabPanel, которые могут включать другие элементы и даже контейнеры компоновки, однако на саму компоновку не столь влияют в отличие от выше перечисленных. Кроме того, если нам не хватает стандартных контейнеров, мы можем определить свои с нужной нам функциональностью.

Контейнеры компоновки позволяют эффективно распределить доступное пространство между элементами, найти для него наиболее предпочтительные размеры.

Все выше перечисленные контейнеры компоновки наследуются от абстрактного класса Panel, а само дерево наследования можно представить следующим образом:



Наглядным примером применения компоновки может служить сделанное в первой главе приложение калькулятора, где мы не указываем явным образом размеры кнопок, а полагаемся на контейнер Grid, который и осуществляет всю расстановку и масштабирование дочерних элементов.

### Процесс компоновки

Процесс компоновки проходит два этапа: измерение (measure) и расстановка (arrange). На этапе измерения контейнер производит измерение предпочтительного для дочерних элементов места. Однако не всегда контейнер имеет достаточно места, чтобы расставить все элементы по их предпочтительным размером, поэтому их размеры приходится усекать. Затем происходит этап непосредственной расстановки дочерних элементов внутри контейнера.

**Canvas**

Элемент управления Canvas поддерживает абсолютное позиционирование и предоставляет минимум встроенных возможностей макета для содержащихся в нем элементов управления. Элемент управления Canvas позволяет размещать содержащиеся в нем элементы управления со сдвигом относительно любого угла панели. Canvas предоставляет четыре свойства для содержащихся в нем элементов управления: Top, Bottom, Right и Left. Элементы управления, содержащиеся в элементе управления Canvas, должны определять одно свойство по горизонтали и одно свойство по вертикали, тем самым назначая угол, от которого будет задаваться смещение. Например, если элемент управления определяет значения для свойств Top и Right, он будет поддерживать постоянное расстояние от правого верхнего угла. Если указаны значения для более чем одного свойства по горизонтали или по вертикали, то одно из значений не будет обрабатываться.

**DockPanel**

Элемент управления DockPanel обеспечивает поддержку закрепления для упрощения расположения панелей инструментов и других элементов управления, которые требуется закрепить вдоль одной из границ панели. Элемент управления DockPanel предоставляет свойство с именем DockStyle для содержащихся в нем элементов управления, которое определяет, как они будут расположены.

**Grid**

Наиболее знакомым контейнерным элементом управления является элемент управления Grid. По умолчанию каждое новое окно Window, открытое в Windows Presentation Foundation (WPF) для Visual Studio (конструктор), включает элемент управления Grid. Элемент управления Grid позволяет позиционировать элементы управления внутри ячеек, задаваемых пользователем.

**StackPanel**

Элемент управления StackPanel располагает содержащиеся в нем элементы управления либо в вертикальном столбце, либо в горизонтальной строке, в зависимости от значения свойства Orientation. Если в элемент управления StackPanel добавлено больше элементов управления, чем может быть отображено по ширине StackPanel, лишние элементы обрезаются и не отображаются.

**WrapPanel**

Элемент управления WrapPanel аналогичен элементу управления StackPanel в том, что в нем элементы управления располагаются в столбце или в строке в зависимости от свойства Orientation. В дополнение к подобному расположению, элемент управления WrapPanel поддерживает возможность переноса в следующий столбец или строку для содержащихся в нем элементов управления. Таким образом, если в элемент управления WrapPanel добавлено больше элементов, чем может быть отображено по ширине WrapPanel, они переносятся в дополнительный столбец или строку.