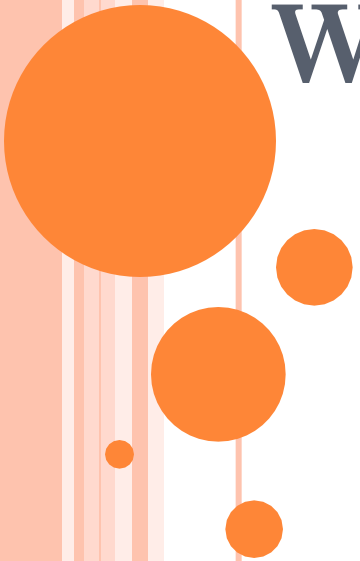


Windows Presentation Foundation 4

Базовые сведения

Windows Presentation Foundation (**WPF**) — система для построения клиентских приложений Windows с визуально привлекательными возможностями взаимодействия с пользователем, графическая (презентационная) подсистема в составе .NET Framework (начиная с версии 3.0), использующая язык XAML.

Преимущества WPF

Технология WPF существенно упрощает создание любых пользовательских интерфейсов. При этом интерфейс может создаваться относительно независимо от остального приложения, что позволяет дизайнерам гораздо эффективнее участвовать в процессе разработки приложения.

Особенности WPF

- WPF предустановлена в Windows Vista, Windows 7, Windows 8.
- WPF предоставляет средства для создания визуального интерфейса, включая язык XAML (Extensible Application Markup Language), элементы управления, привязку данных, макеты, двухмерную и трёхмерную графику, анимацию, стили, шаблоны, документы, текст, мультимедиа и оформление.

Особенности WPF

- В основе WPF лежит векторная система визуализации, не зависящая от разрешения устройства вывода и созданная с учётом возможностей современного графического оборудования.
- Графической технологией, лежащей в основе WPF, является DirectX, в отличие от Windows Forms, где используется GDI/GDI+.
  - производительность WPF выше, чем у GDI+ за счёт использования аппаратного ускорения графики через DirectX

Технология Silverlight и WPF

Технологию Silverlight является подмножеством WPF и дополнительно включает несколько фундаментальных классов из каркаса .NET Framework (встроенные типы данных, классы коллекций и т. д.).

Если требуется функциональность, которая существует только в полной версии .NET, то рекомендуется использовать WPF. Если необходима возможность выполнять приложение на компьютерах Mac или от устройствах, отличных стандартного ПК, используется Silverlight. то

На любой платформе Silverlight обеспечивает единую технологию построения интерфейса.

Создание WPF-приложения

Создание WPF приложения начинается с процедуры задания размеров и положений элементов управления (и других элементов) – *компоновки или вёрстки макета*.

В WPF имеется богатая инфраструктура компоновки. В ее основе лежит механизм взаимодействия между элементами-родителями и их потомками.

Панели WPF

Родительские элементы, поддерживающие компоновку нескольких детей, называются *панелями*.

WPF пять основных встроенных панелей (все в пространстве имен System.Windows.Controls) в порядке возрастания сложности (и полезности):

- Canvas;
- StackPanel;
- WrapPanel;
- DockPanel;
- Grid.

Виды панелей

1)Canvas (холст) - самая простая панель. Использовать для организации пользовательского интерфейса не рекомендуется.

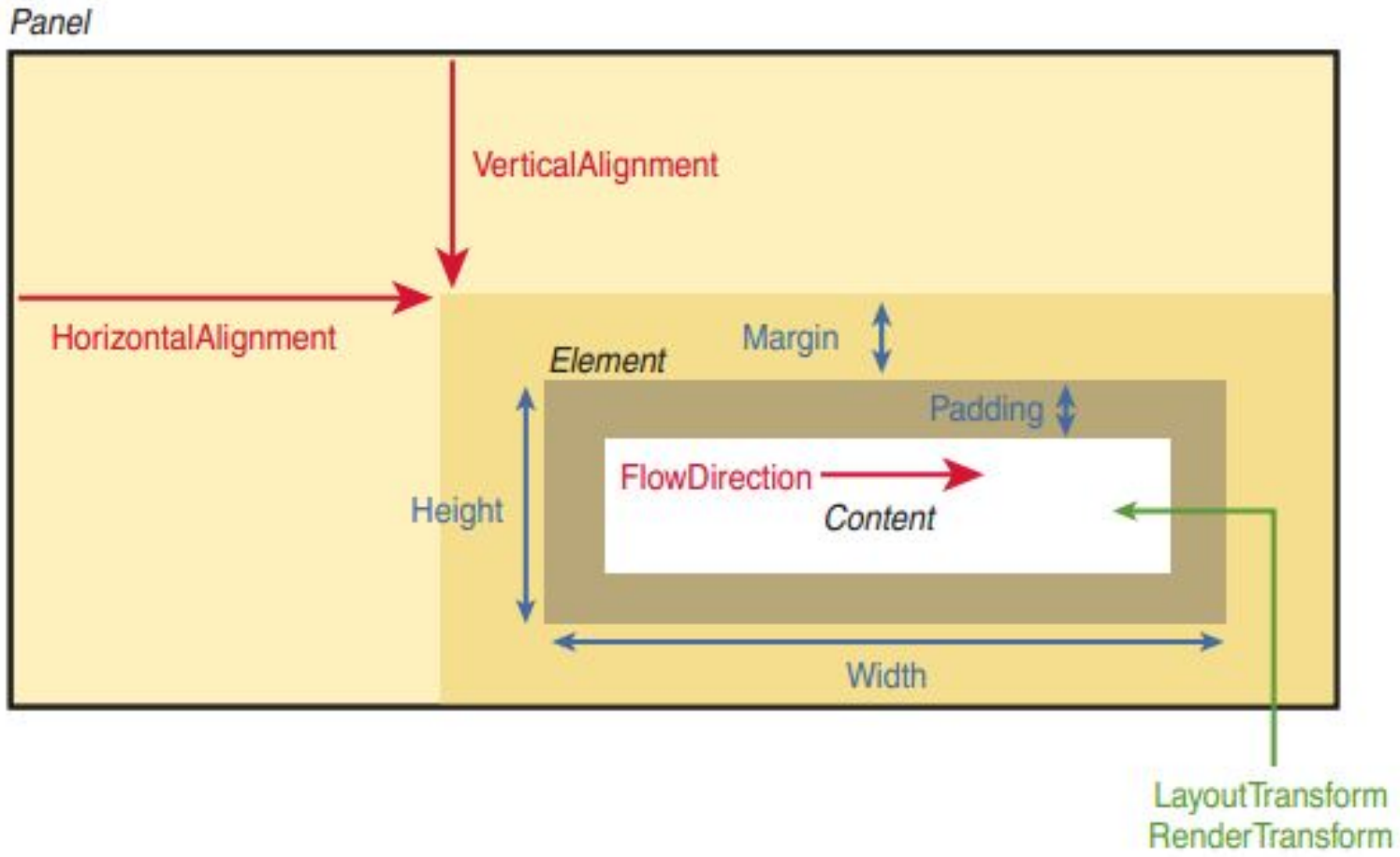
2)StackPanel последовательно размещает своих потомков в виде стопки.

3)Панель WrapPanel похожа на StackPanel, но при нехватке места для одной стопки создает новые строки или столбцы.

4)Панель DockPanel дает простой способ пристыковки элемента к одной из сторон, растягивая его на всю имеющуюся ширину или высоту.

5)Grid(сетка) - самая гибкая из всех панелей и, пожалуй, наиболее употребительная. В проектах VisualStudio и ExpressionBlend панель Grid используется по умолчанию. Она позволяет расположить дочерние элементы в несколько строк и несколько столбцов, не полагаясь на режим автоматического переноса.

Свойства панелей



Свойства Height и Width

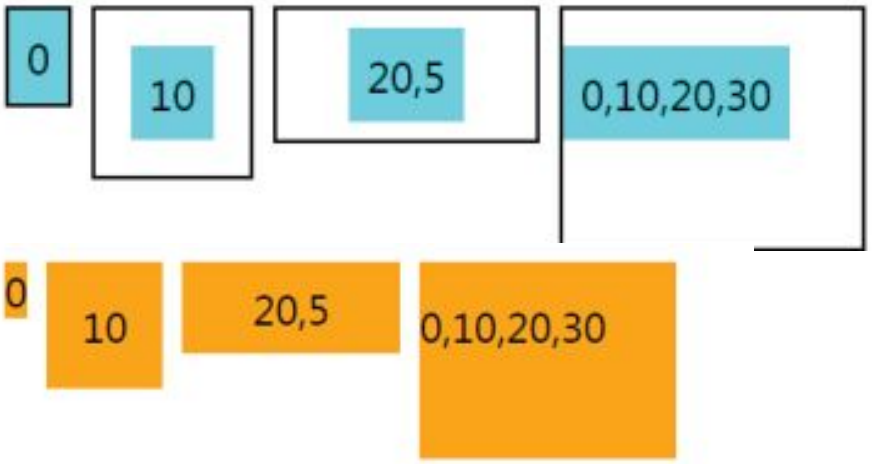
Во всех классах, производных от FrameworkElement, есть свойства Height (высота) и Width (ширина) (типа double), а также MinHeight, MaxHeight, MinWidth и MaxWidth, которыми можно пользоваться для задания допустимых диапазонов значений. Все эти свойства можно задавать в любой комбинации как в процедурном коде, так и в XAML.

Свойства Margin и Padding

Свойства Margin и Padding тоже связаны с размером элемента.

Margin задает *внешнее* поле вокруг элемента, а Padding - *внутренний* отступ между содержимым элемента и его границами.

Результат установки разных свойств Margin и Padding



Выравнивание

С помощью свойств HorizontalAlignment и VerticalAlignment элементможет управлять распределением избыточного пространства, выделенного ему родителем. Значениями свойств являются одноименные перечисления, которые определены в пространстве имен System.Windows:

- HorizontalAlignment - Left, Center, Right, Stretch
- VerticalAlignment - Top, Center, Bottom, Stretch

Пример:

Применение преобразований

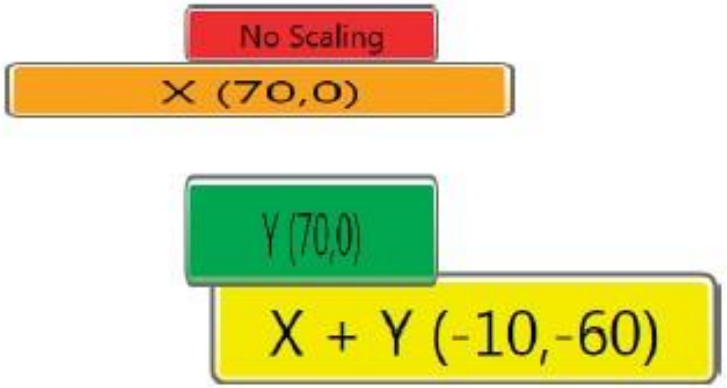
- В WPF имеется ряд встроенных классов двумерных геометрических преобразований (производных от System.Windows.Media.Transform), которые позволяют изменять размер и положение элементов независимо от ранее рассмотренных свойств.

- Встроенные двумерные преобразования, определенные в пространстве имен System.Windows.Media:
  - RotateTransform;
  - ScaleTransform;
  - SkewTransform;

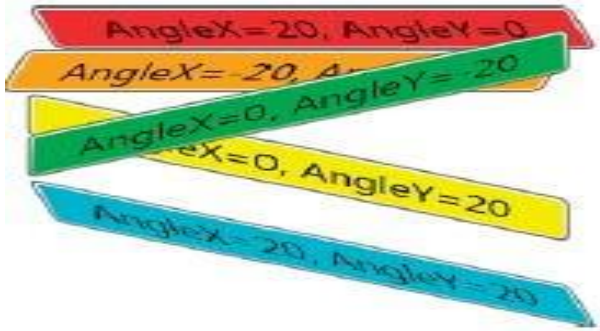
Примеры преобразований

RotateTransform

ScaleTransform



SkewTransform



Маршрутизируемые события и команды

Рассмотрим две важных составных части инфраструктуры WPF - маршрутизируемые события и команды.

Маршрутизируемые события

Сгенерированное маршрутизируемое событие может распространяться вверх или вниз по визуальному и логическому дереву, достигая каждого элемента простым и естественным образом без написания дополнительного кода.

Маршрутизация событий позволяет большинству приложений вообще не задумываться о наличии визуального дерева и является основой механизма композиции элементов в WPF.

События жестко связаны с деталями конкретных действий пользователя (например, нажатие кнопки или выбор элемента ListBoxItem из списка). Для различных устройств ввода определены свои события. Существуют события:

- клавиатуры;
- мыши;
- стилуса.

Команды

В WPF существуют команды – это более абстрактная и слабо связанная версия событий. Команды представляют действия независимо от того, как они выглядят в пользовательском интерфейсе. Каноническими примерами служат команды Cut (Вырезать), Copy (Копировать) и Paste (Вставить). Мощь механизма команд основывается на трех основных особенностях:

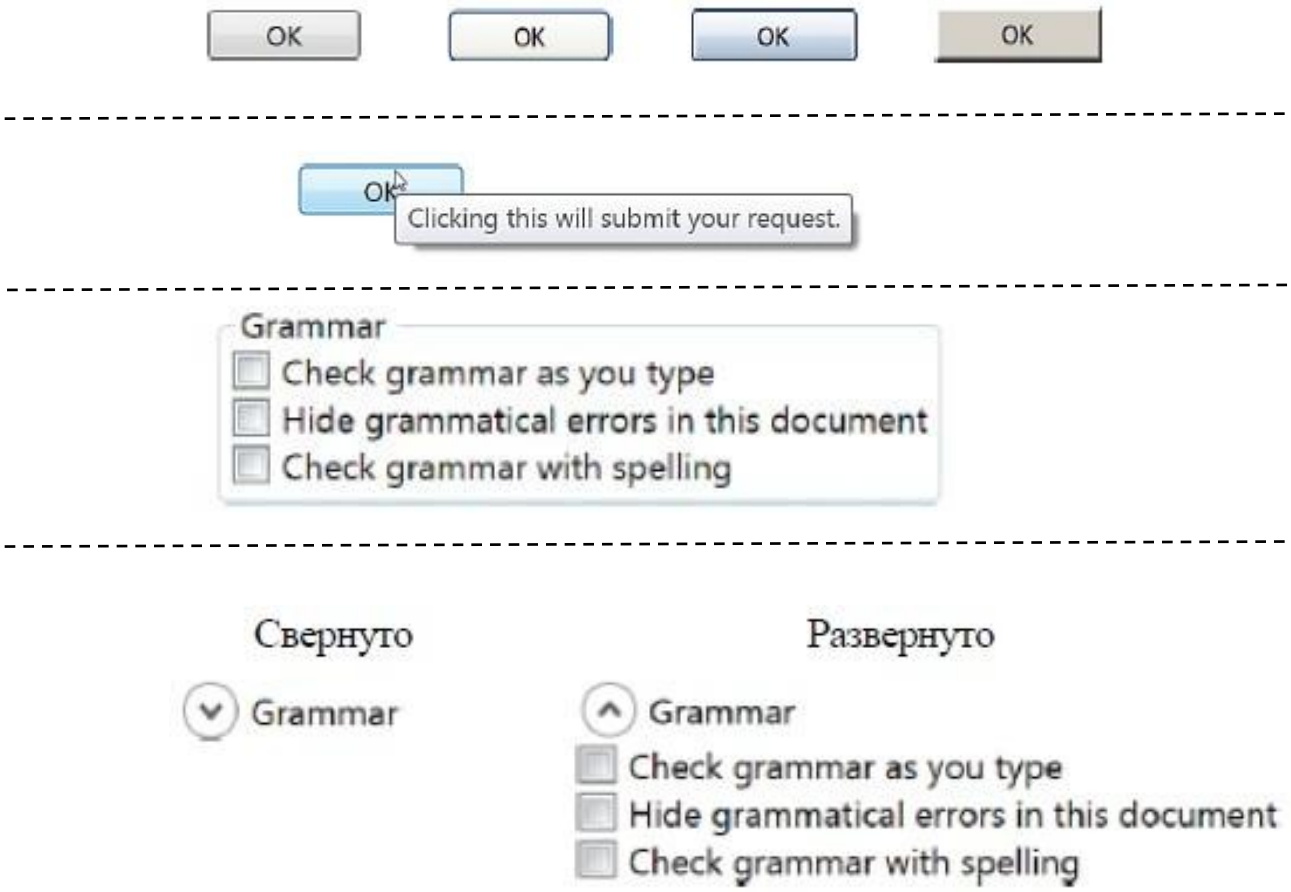
- в WPF определено много встроенных команд;
- в команды встроена автоматическая поддержка жестов ввода (например, сочетаний клавиш);
- встроенное поведение некоторых элементов управления WPF уже ориентировано на те или иные команды.

Элементы управления

Есть триосновных разновидности однодетных элементов управления:

- Кнопки
- Простые контейнеры
- Контейнеры с заголовками

Примеры элементов управления



Двумерная графика

Основное отличие WPF от GDI - то, что в WPF применяется графическая система, работающая полностью в режиме *запоминания,* а не *непосредственной визуализации.*

В системе, работающей в режиме запоминания, можно формулировать высокоуровневые указания, например: «Помести синий квадрат размером 10x10 точку (0,0), - и система сама запомнит и будет поддерживать это состояние. На самом деле это означает: «Помести синий квадрат размером 10x10 в точку (0,0) *и следи за тем, чтобы он там оставался*». В дальнейшем нет необходимости возиться с недействительными областями и перерисовкой, а это экономит немало времени.

Двумерная графика

Существует тритипа данных: Drawing, Visualи Shape.

Drawing - это просто описания путей и фигур с ассоциированными кистями Brush для заливки и контура.

Визуальное представление Visual — это один из способов нарисовать объект Drawing на экране, но класс Visual открывает также возможность низкоуровневого и менее ресурсоемкого подхода к рисованию, позволяющего обходиться вообще без объектов Drawing.

Фигуры Shape - это готовые объекты Visual, предлагающие самый простой (но и самый ресурсоемкий) подход к рисованию на экране.

Трёхмерная графика

Трехмерная графика (или 3D-графика) полностью интегрирована в платформу WPF, многие встречающиеся в ней концепции пересекаются с двумерной графикой и другими компонентами. Как и в случае двумерной графики, возможности 3D доступны как из процедурного кода, так и из XAML- разметки. Чтобы отобразить в WPF трехмерную сцену, необходимо построить граф объектов. После того как сцена описана, система берет на себя ответственность за ее визуализацию и перерисовку в нужные моменты времени.

Трёхмерная графика

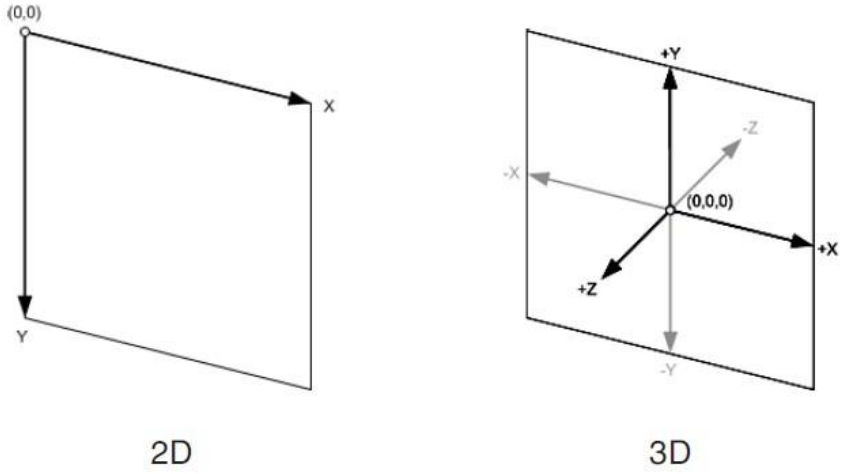
Задача трехмерной графики - создание по трехмерным моделям двумерных изображений, которые можно было бы отобразить на некоем устройстве вывода, например на экране монитора.

Хотя большинство 3D-классов - прямые обобщения двумерного API, есть два понятия, не имеющих аналогов в двумерном мире:

- Материалы Material и источники света Light
- Камеры Camera

Камера

В WPF для управления тем, что появится в объекте Viewport3D, необходимо поместить на 3D-сцену виртуальную камеру (объект Camera). Для этого следует позиционировать и ориентировать камеру в мировой системе координат (иногда для краткости ее называют *мировым пространством).* На рисунке показаны двумернаяи трехмерная системы координат, применяемые в WPF.



2D 3D

Освещение

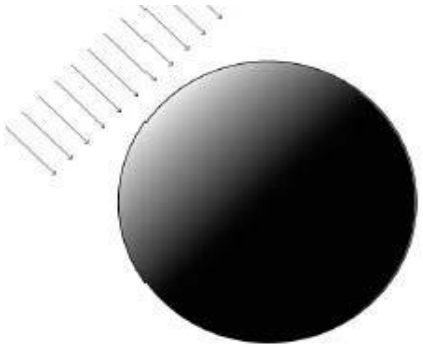
Освещение складывается из трех основных компонентов:

- объекты Light, являющиеся источниками света;
- объекты Material, то есть материалы, по-разному отражающие свет в камеру;
- геометрия (Geometry) модели, определяющая углы падения и отражения света.

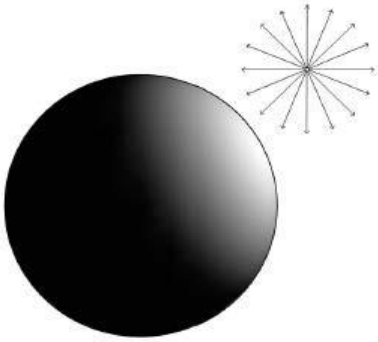
Рассмотрим различные источникиосвещения, поддерживаемые WPF.

Виды источников света

DirectionalLight (направленный источник света) — расположен в бесконечности, освещает сцену параллельными лучами света. Этот класс аппроксимирует удаленный источник света, например Солнце.

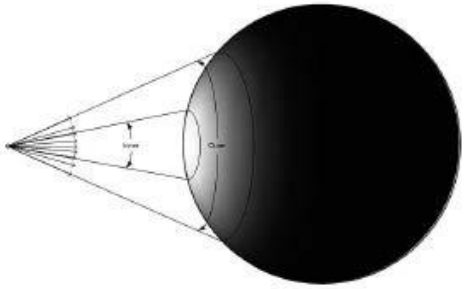


PointLight (точечный источник) — излучает свет равномерно во всех направлениях. Яркость света убывает с увеличением расстояния от источника. Класс PointLight аппроксимирует нефокусированные источники света, например электрические лампочки.

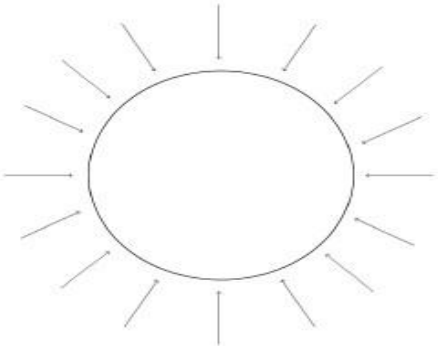


Виды источников света

SpotLight (прожектор) - испускает конус света. Яркость убывает с увеличением расстояния от источника. Класс SpotLight аппроксимирует фокусированные источники света, например луч фонаря.



AmbientLight (рассеянный свет) — освещает все поверхности равномерно. Яркий источник рассеянного света создает плоские изображения из-за отсутствия теней. Однако не слишком яркий источник аппроксимирует эффект рассеянного освещения, созданного диффузно отражающими поверхностями на сцене.



Анимация, аудио и видео

Анимация

В WPF у анимации есть следующее изменение значения свойства во времени (цвета, размера, положения и т.п.).

Аудио

Средства работы с аудио представлены следующими классами:

- SoundPlayer
- SoundPlayerAction
- MediaPlayer
- MediaElement и MediaTimeline

Видео

Поддержка видео в WPF основана на классе MediaPlayer, а также на аудио, видео и других сопутствующих ему классах MediaElement и MediaTimeline. Таким образом, все форматы файлов, поддерживаемые Windows Media Player (WMV, AVI, MPG и прочие), поддерживаются и WPF-приложениями.