1)Технология WPF является часть экосистемы платформы .NET и представляет собой подсистему для построения графических интерфейсов.

Преимущества WPF:

Использование традиционных языков .NET-платформы - C#, F# и VB.NET для создания логики приложения.

Независимость от разрешения экрана: поскольку в WPF все элементы измеряются в независимых от устройства единицах, приложения на WPF легко масштабируются под разные экраны с разным разрешением.

Аппаратное ускорение графики.

Богатые возможности по созданию различных приложений.

Также стоит учитывать, что по сравнению с приложениями на Windows Forms объем программ на WPF и потребление ими памяти в процессе работы в среднем несколько выше.

2) XAML (eXtensible Application Markup Language) - язык разметки, используемый для инициализации объектов в технологиях на платформе .NET. Данный язык используется прежде всего для создания пользовательского интерфейса.

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

4) Все элементы управления могут быть условно разделены на несколько подгрупп:

Элементы управления содержимым, например кнопки (Button), метки (Label)

Специальные контейнеры, которые содержат другие элементы, но в отличие от элементов Grid или Canvas не являются контейнерами компоновки - ScrollViewer,GroupBox

Декораторы, чье предназначение создание определенного фона вокруг вложенных элементов, например, Border или Viewbox.

Элементы управления списками, например, ListBox, ComboBox.

Текстовые элементы управления, например, TextBox, RichTextBox.

Элементы, основанные на диапазонах значений, например, ProgressBar, Slider.

Элементы для работ с датами, например, DatePicker и Calendar.

Остальные элементы управления, которые не вошли в предыдущие подгруппы, например, Image.

5)Свойство зависимости нужны для привязки данных, стили, анимации.

если мы хотим применять свойства зависимостей, то нам надо унаследовать свой класс от абстрактного класса DependencyObject.

Свойства зависимости **позволяют WPF вычислять значение на основе нескольких внешних значений**, что может быть очень важно для анимации и служб привязки данных.

6) Маршрутизируемые события — это события с большими транспортными возможностями.

Маршрутизируемые события позволяют обработать событие в одном элементе (например, в метке), хотя оно возникло в другом (например, в изображении внутри этой метки). Как и в случае свойств зависимости, маршрутизируемые события можно употреблять и традиционным способом — подключив обработчик событий с нужной сигнатурой

Основные группы событий:

События клавиатуры

События мыши

События стилуса

События сенсорного экрана

События жизненного цикла

7)Resource Dictionary – словарь ресурсов, которые добавляются к ресурсам приложения.

8)+

9) Паттерн "Команда" (Command) позволяет инкапсулировать запрос на выполнение определенного действия в виде отдельного объекта.

Когда надо передавать в качестве параметров определенные действия, вызываемые в ответ на другие действия.

Когда необходимо обеспечить выполнение очереди запросов, а также их возможную отмену.