1.Свойства зависимости - свойства, которые обеспечивают взаимодействия с ЭУ. Они нужны для привязки данных, стилей, анимаций.  
2.статический метод Register класса DependencyProperty (имя свойства, тип свойства, тип владельца свойства, а также функции обратного вызова для валидации и преобразования значений.  
3.Делегат ValidateValueCallback используется для проверки допустимости значения.   
4.Делегат CoerceValueCallback используется для принудительного приведения значения к определенному диапазону. Он позволяет корректировать значение.  
5.Direct: События, которые напрямую передаются от источника к цели визуального дерева.  
Bubbling: События, которые всплывают от цели к корню визуального дерева.  
Tunneling: События, которые опускаются от корня визуального дерева к цели.  
6.статический метод Register класса RoutedEvent () (имя события, стратегия маршрутизации, тип делегата для обработки событий, тип владельца события). дальше в создаем событие организуя механизм подписки и отписки  
7.Концепция Command - абстракция, которая позволяет разделять логику действия от его представления на ЭУ. Преимущества команд включают отделение логики от интерфейса и повторное использование команд.  
8.Команды используются путем привязки их к элементам управления, таким как кнопки, меню и др. Затем команды могут быть выполнены из интерфейса пользователя, без явного вызова методов из кода обработчика событий.  
9.определяем статическое поле только для чтения и инициализируем это свойство с помощью создания объекта конструктора класса RoutedUICommand, который принимает имя команды, отображаемое имя и владельца команды. дальшее создаем привязку с помощью метода Add в котором создаем экземпляр с помощью конструктора CommandBinding который принимает 1 параметр - сама команда, 2 - обработчик, который будет вызываться при выполнении этой команды