**08. 단일 자식 엘리먼트**

- FrameworkElement나 Control을 상속받는 많은 클래스들은 자식을 갖는다. 그리고 이런 자식들을 수용하기 위해서 통상 1개의 프로퍼티와 4개의 메소드를 오버라이딩한다.

1. **VisualChildrenCount**   
   엘리먼트가 관리하는 자식의 수를 알 수 있게 한다.
2. **GetVisualChild**엘리먼트가 인덱스에 대응하는 자식을 반환할 수 있다.  
   \*null을 반환해서는 안된다. ( 인덱스가 부정확할 경우에는 예외 상황을 던져야 함. )
3. **MeasureOverride**엘리먼트의 요구 크기를 계산하고 그 크기를 반환한다.  
   자식을 가진 엘리먼트는 그자식에 대한 필요한 크기도 고려해야 한다.
4. **ArrangeOverride**엘리먼트에 대한 최종 레이아웃 크기를 가리키는 Size객체를 받는다.
5. **OnRender**엘리먼트를 그릴 수 있다.(OnPaint와 유사… 다시 그려야 할 경우 호출됨)

# - Decorator

# 테코레이터 패턴에서는 객체에 추가적인 요건을 동적으로 첨가한다. 데코레이터는 서브클래스를 만드는 것을 통해서 기능을 유연하게 확장할 수 있는 방법을 제공한다.

# - ControlTemplate

템플릿은 엘리먼트들을 연결시킬뿐만 아니라, 엘리먼트의 프로퍼티에 어떤 변화가 생길 때 그에 대해 반응하는 트리거(triggers)를 정의한다.

# 실제 엘리먼트를 생성하지 않는다. ( FrameworkElementFactory 클래스를 통해 수행) \* 프로퍼티 지정(의존 프로퍼티), 이벤트 핸들러 지정(라우티드 이벤트)

# 여러 개의 FactoryElementFactory를 사용해 부모 자식 관계 설정가능.

# Tirgger 는 조건부 작업을 수행 Setter와 연관시켜 수행.

# (예제3 참조)