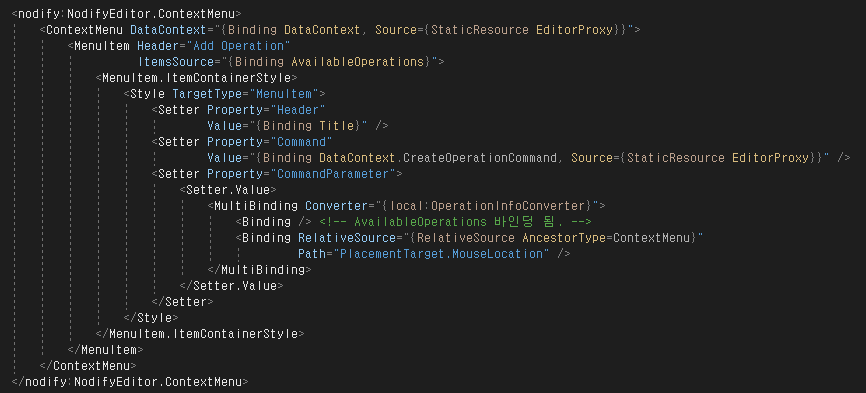
1. 바인딩.

* 기본적으로 MVVM 디자인 패턴을 따라서, 프로젝트가 만들어졌다. 따라서 여기서는 바인딩하는 개념을 잘 익혀야 한다.

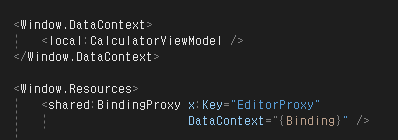
1. 작동 방식에 대한 내용

* NodifyEditor.ContextMenu 를 통해서 노드가 생성된다.



* 바인딩을 하기 위해서 EditorProxy 를 씀. <ContextMenu DataContext = “{Binding DataContext, Source={StaticResource EditorProxy}}”>

구문에서 DataContext 는 EditorProxy 클래스의 property 임.



1. ConnectionCompletedCommand (NodifyEditor 와 연관 되는 부분임)

-

CreateConnection

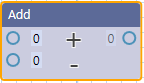
VM

NodifyEditor

1. 바인딩 되는 객체와 변환 방식

* itemsSource 에 매핑되는 방식
* ContextMenu 에서 Command 프로퍼티에 매핑은 private void CreateOperation(CreateOperationInfoViewModel arg) 로 매핑된다. 여기서 입력 파라미터의 경우 CreateOperationInfoViewModel 이다.
* 여기서, MenuItem 의 ItemsSource 는 NodifyObservableCollection<OperationInfoViewModel> AvailableOperations 로 바인딩 되어 있다.
* AvailableOperations 의 경우는 Constructor 에서 이미 OperationInfoViewModel 의 요소들을 리스트에 다 넣은 상태이다. (매핑시 Constructor가 구동됨으로)
* 이제 converter 를 통해서 OperationInfoConverter 를 마우스 위치정보를 추가하여, CreateOperationInfoViewModel 로 만들어 준다.
* 그래서 최종적으로 매핑된 private void CreateOperation(CreateOperationInfoViewModel arg) 에 입력파라미터로 들어가게 되고 Execute(object parameter) 가 수행된다.
* CreateOperation 를 수행하면 public NodifyObservableCollection<OperationViewModel> Operations 에 추가하는 형태가 되고 이것은 NodifyEditor.ItemsSource 에 매핑된다.
* ~~ConnectionCompletedCommand 매핑되는 방식~~
* ~~ICommand 인터페이스를 상속받은 개체를 통해서 delegate 함수를 제작하고 이를 통해서 NodifyEditor 의 Command Dependency Properties 의 ConnectionCompletedCommandProperty에 매핑되는 구조이다.~~
* ~~NodifyEditor 내부에서 해당 프로퍼티는 NodifyEditor 에서 마우스 왼쪽 버튼을 놓았을 때 발생하는 이벤트에서 맨처음 발동되기 시작한다. (OnMouseLeftButtonUp) 해당 이벤트 핸들러는 Connector 에 있다.~~

Connector



* 여기서 파란 동그라미가 커넥터(코드의 객체는 Connector 라고 하고 구현된 객체는 한글로 커넥터로 나타낸다) 이다.
* 커넥터를 마우스 왼쪽 버튼을 누르고 때면(Connector.OnMouseLeftButtonUp) -> Connector.OnConnectorDragCompleted 함수를 호출하고 해당함수에서 PendingConnectionCompletedEvent 의 이벤트를 발생시킨다.이때 해당 이벤트는 NodifyEditor의 컨스트럭터에서 AddHandler 로 이벤트 핸들러를 등록 시켜주었다.

AddHandler(Connector.PendingConnectionCompletedEvent, new PendingConnectionEventHandler(OnConnectionCompleted));

그래서, NodifyEditor의 OnConnectionCompleted 함수에서 처리가 되고 여기서 ConnectionCompletedCommand 프로퍼티에 매핑된 함수를 실행하게 된다.

동일하게 마우스 버튼을 누루고 때면 동일한 이벤트가 발생하는데, 이때, PendingConnection 에서 OnPendingConnectionCompleted 함수가 실행된다.

바로 아래 구문 때문이다.

Editor.AddHandler(Connector.PendingConnectionCompletedEvent, new PendingConnectionEventHandler(OnPendingConnectionCompleted), true);