**Событийно-ориентированноепрограммирование(event-drivenprogramming)**— парадигма программирования, в которой выполнение программы определяется событиями—действиямипользователя(клавиатура,мышь),сообщениямидругих программипотоков,событиямиоперационнойсистемы(например,поступлением сетевого пакета).

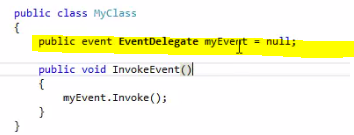
**События**-это особый тип многоадресных делегатов, которые можно вызвать только из класса или структуры, в которой они объявлены (класс издателя)

**«Чем отличается event от delegate?»** - о том, что «delegate – это класс, а event – это член класса с типом delegate» не казал вообще никто.



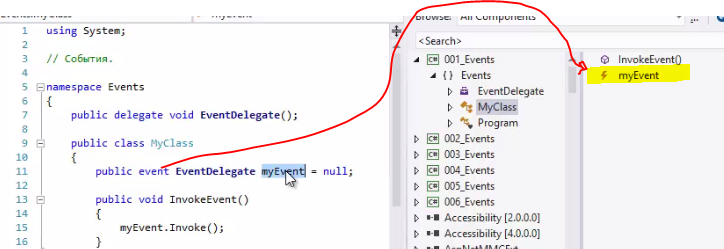
Делегат может использоваться самостоятельно либо в контексте событийной модели

В теле класса создаем открытое событие с именем myEvent типа EventDelegate



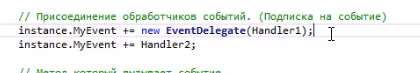
Ключевое слово Evet в ObjectBrowser отображает молнию, первая особенность

События пишутся с большой буквы



Подписка на событие

Во втором примере используем технику предположения делегата без EventDelegate для событийной модели



Запрещено из вне(вне теле класса) вызывать событие

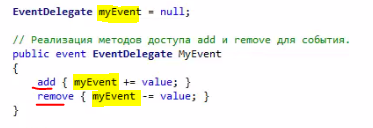
Вторая особенность



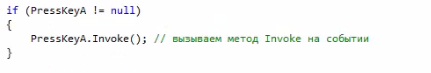
Открепляем метод обработки события



Реализация метод add и remove для события



Проверка обработчика события на NULL

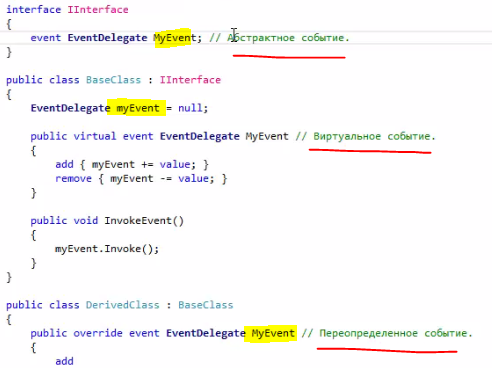


События могут быть

Абстратными

Виртуальными

Можно переопределять

,

На событие можно подписать анонимный метод, но просто отписаться от него не получиться



События вызванное при нажатии кнопки на форме

Первым параметром приходит объект на который нажали,

Вторым чем нажали (если нажать ENTER то событие вызвано клавиатурой)

