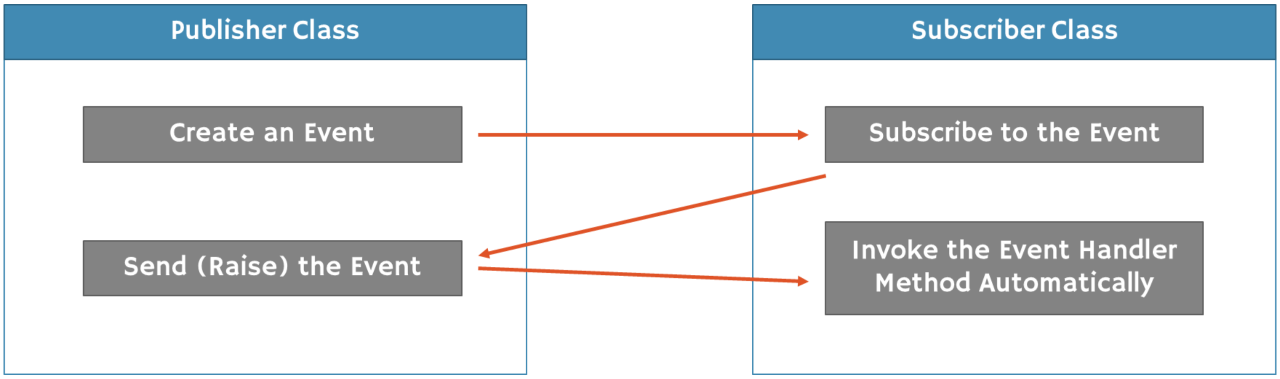
.NET > C#.NET > OOP

Events

- by Harsha Vardhan

Events

- Events enable a class to send notifications to other classes, when something occurs.
- The class that sends (or raises) events (notifications), is called as "publisher class".
- The class that receives (or subscribes or handles) events (notifications), is called as "subscriber class".
- Publisher class sends events; Subscriber class receives events.



Overview of Events

- The Publisher class creates an event.
- The Subscriber class subscribes to the event; that means an "event handler" method is created in the subscriber class. The "event handler" method is nothing but, the method which is dedicated to be executed when the event is raised.
- The publisher class can send (raise) events.
- Every time, when the event is raised by the publisher, the corresponding "event handler" method executes automatically.

Syntax of Events

Create a Delegate

```
public delegate ReturnType DelegateTypeName(paraml, param2, ...);
```

Create an Event in Publisher Class

```
class  Publisher
{
     private  MyDelegateType  myEvent;

     public  event  MyDelegateType  MyEvent
     {
         add
         {
             myEvent += value;
         }
         remove
         {
             myEvent -= value;
         }
     }
}
```

Raise the event in Publisher Class

```
if (myEvent != null)  myEvent(arg1, arg2, ...);
```

Create Event Handler Method in Subscriber Class

```
class  Subscriber
{

     public  ReturnType  EventHandlerMethodName(param1, param2, ...)
     {
          Method body here
     }

}
```

Subscribe to the Event [Inside or Outside the Subscriber class]

```
MyEvent += EventHandlerMethodName;
```

Rules for Events

- The event should be created based on the delegate. That means, the event accepts the methods that are having specific parameters and return type, defined in the delegate.
- An event can have multiple subscribers.
- A subscriber can subscribe multiple events from multiple publishers.
- Events are basically signals to inform to other classes, that some important thing happened in the publisher class.
- Events are special kind of "multi-cast delegates", which can be raised only within the same class, in which they are created.
- Events can be static, virtual, sealed and abstract.
- Events will not be raised (throws exception), if there is no at least one subscriber.
- Events can be defined in interfaces.