

Event چیست ؟

یک Event در اصل لیستی از Delegate ها است. از نظر تکنیکی این یک Multicast Delegate است. بهر حال ، کلمه رزرو شده Event به کامپایلر اجازه میدهد تا برای کنترل نحوه دسترسی به Delegate آن را کپسوله کند.

شیئی که Event را درون خود دارد ناشر (Publisher) نامیده میشود. شیئی که Event را کنترل (Handle) میکند مشترک (Subscriber) نامیده میشود. یک Event به ناشر اجازه میدهد تا یک کنترل کننده Event مشترک را بدون اینکه چیزی در مورد مشترک بداند فراخوانی کند.

برای مثال ، یک شی دکمه با یک Event کلیک میتواند تمام ClickEventHandlers ها را بدون دانستن نام کلاس یا متدی که Event را کنترل میکند، فراخوانی کند.

این مسئولیت مشترک است که مشترک Event میشود. ناشر به سادگی Event را در زمان لازم فعال میکند. هر زمان که Event فعال شد ، هر Delegate که در لیست آن Event باشد فراخوانی خواهد شد.

تذکر :

کلمه رزرو شده Event ، رخ دادن Delegate را کپسوله میکند و کامپایلر در خارج از کلاس فقط اجازه عملیات += و -= را میدهد. در آخر فقط ناشر میتواند مستقیماً به شی Delegate دسترسی داشته باشد و آن را صدا بزند یا لیست Delegate ها را ویرایش کند.

نمونه های کد :

قراردادهای Event مخصوص .Net :

- Delegate هایی که برای یک نوع Event بکار میروند دارای پسوند " EventHandler " هستند.

- EventArgs های مخصوص، دارای پسوند EventArgs هستند و از EventArgs به ارث برده می شوند.

- Delegate ها همیشه باید دارای دو پارامتر باشند : یکی Object Sender و دیگری EventArgs

نامهای مثال :

```
// Event Handler Delegate Declaration
public delegate void MouseClickedEventHandler ( object sender, MouseClickedEventArgs e ) ;
// Event Args Declaration
public class MouseClickedEventArgs : EventArgs
{
    // ...
}
// Event Declaration
public event MouseClickedEventHandler MouseClicked;
```

تعریف یک Event ساده

```
public event EventHandler Changed;
```

تعریف یک Event به نام Mouse Clicked

```
public event MouseClickEventHandler MouseClicked;
```

مشترک یک Event شدن

```
public void Subscribe()
{
    //Subscribe to the event
    btnCancel.Click += new EventHandler( CancelClick );
}
// The event handler
void CancelClick( object sender, EventArgs e)
{
    // ...
}
```

تعريف یک Delegate و یک Event با EventArgs مخصوص

// Declare a delegate inside the namespace

```
public delegate void StatusChangedEventHandler( object sender, StatusChangedEventArgs e );
```

// StatusChangedEventArgs is at the bottom of this example

// The event publisher

```
public class ClassWithStatus
```

```
{
```

```
    // Declare the event
```

```
    public event StatusChangedEventHandler StatusChanged;
```

```
}
```

// Declare the custom event args class

// It should inherit from EventArgs

```
public class StatusChangedEventArgs : EventArgs
```

```
{
```

```
    private readonly bool _isDone;
```

```
    public StatusChangedEventArgs( bool isDone )
```

```
    {
```

```
        _isDone = isDone;
```

```
    }
```

```
    public bool IsDone
```

```
    {
```

```
        get
```

```
        {
```

```
            return _isDone;
```

```
        }
```

```
    } }
```

تعریف یک Event که از EventHandler از نوع Generic و از EventArgs مخصوص استفاده میکند :

```
// The event publisher
public class TreeNode
{
    // Declare the event
    public event EventHandler<ExpandedEventArgs> Expanded;
}

// The custom event args class
public class ExpandedEventArgs : EventArgs
{
    private readonly TreeNode _expandingNode;
    public StatusChangedEventArgs( TreeNode expandingNode )
    {
        _expandingNode = expandingNode;
    }
    public TreeNode ExpandingNode
    {
        get
        {
            return _expandingNode;
        }
    }
}
}
```

ترجمه شده از سایت CSharp-Online.net