



Empowering Digital Skills For The Jobs Of The Future



by



# Academy .NET

# Eventi

# Sommario

- Gli eventi
- EventHandler
- Definizione di un evento
- Generazione di un evento
- Creazione classe EventArgs personalizzata

# Eventi

Button

Click Event

- Un evento è un'operazione in cui un oggetto notifica un altro oggetto al verificarsi di una determinata attività.
- Svolgono un ruolo centrale nel framework .NET

# Eventi

- Gli **eventi** segnalano il verificarsi di un'azione/notifica.
- Gli oggetti che generano eventi non hanno bisogno di conoscere esplicitamente l'oggetto che gestirà l'evento. (si possono avere più **handlers** o gestori)

# EventHandler

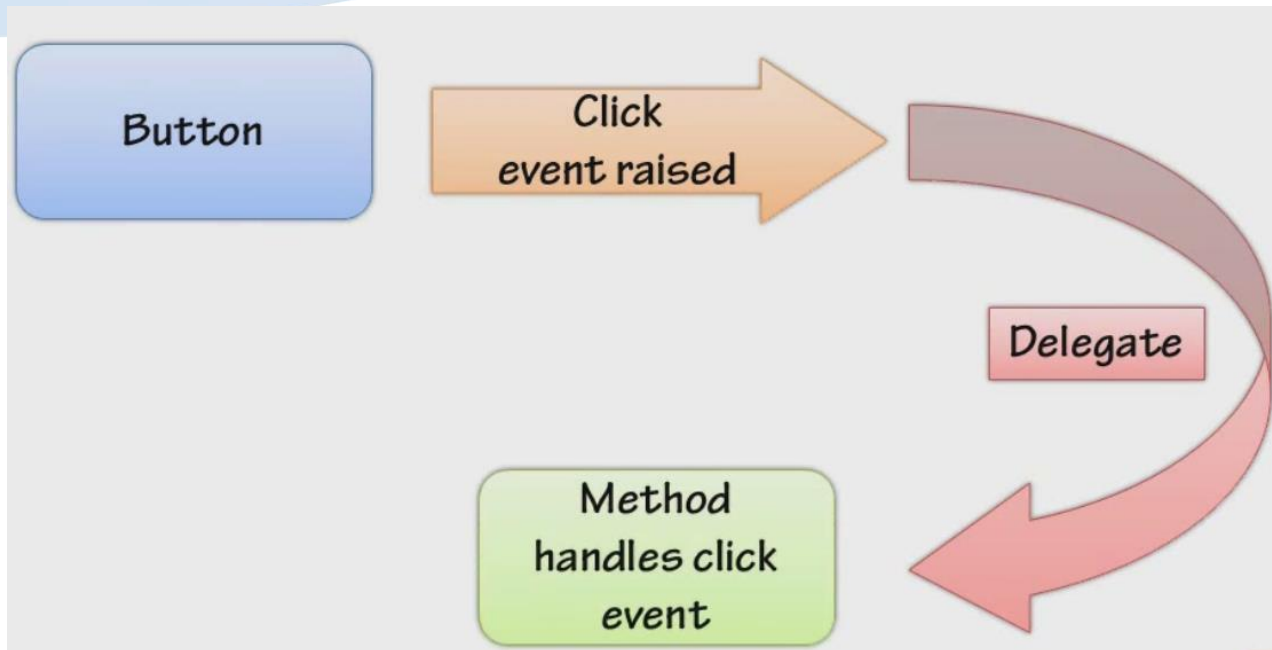
Il delegate EventHandler è responsabile della ricezione e del trattamento dei dati.

Normalmente riceve due parametri:

- **Sender**
- **EventArgs**

**EventArgs** è responsabile dell'incapsulamento dei dati degli eventi.

# EventHandler



```
public void ButtonClick(object sender, EventArgs e)
{
    // Handling of button click occurs here
}
```



# Eventi

## Delegate:

```
public delegate WorkPerformerHandler(int hours, WorkType workType);
```

## Istanza delegate:

```
WorkPerformerHandler delegate1 = new WorkPerformerHandler(WorkPerformed1);
```

## Handler:

```
static void WorkPerformed1(int hours, WorkType workType)
{
    Console.WriteLine("WorkPerformed called" + hours.ToString());
}
```

Classe con proprietà:  
Code  
GoToMeeting  
Scrum

# Invocazione delegate

```
delegate1(5, WorkType.Code);
```

```
WorkPerformedHandler delegate1 = new WorkPerformedHandler(WorkPerformed1)  
WorkPerformedHandler delegate2 = new WorkPerformedHandler(WorkPerformed2)  
  
delegate1 += delegate2;  
delegate1(5, WorkType.GoToMeeting);
```

Con una sola chiamata si posso notificare istantaneamente più handlers.

# Definizione di un evento

- Per definire un evento si utilizza la parola chiave **event**

```
public event WorkPerformerHandler WorkPerformer;
```

Delegate



A blue arrow points from the word 'Delegate' in the box below to the 'WorkPerformerHandler' type in the code snippet above, indicating that 'WorkPerformerHandler' is a delegate type.

# Generazione di un evento

Gli eventi vengono generati nel seguente modo:

```
if (WorkPerformer != null)
{
    WorkPerformer(8, WorkType.Scrum);
}
```

Se il delegate è **null**, si otterrà un'eccezione.

# Creazione classe EventArgs personalizzata

- Quando è necessario passare dati personalizzati, la classe **EventArgs** può essere estesa.

```
private void SubmitButton_Click(object sender, EventArgs e)
{
    MessageBox.Show("Clicked me");
}
```

# Creazione classe EventArgs personalizzata

- Se sono necessari molti dati per essere generati in caso, possono essere incapsulati in un oggetto **EventArgs** ereditato.
- Utilizzo di **EventHandler<T>**

```
public class WorkPerformedEventArgs : EventArgs
{
    public int Hours {get;set;}

    public WorkType workType {get;set;}
}
```

# Creazione classe EventArgs personalizzata

```
public delegate void WorkPerformedHandler(int Hours, WorkType workType)

public class Worker
{
    public event WorkPerformedHandler workPerformed;
}
```

.NET utilizza una classe **EventHandler<T>** generica che può essere utilizzata al posto di un delegate personalizzato:

```
public event EventHandler<WorkPerformedEventArgs> WorkPerformed;
```

# Domande & approfondimenti



# Academy .NET