

# Basi Programmazione - Gestione Eventi

Un **evento**

è un **messaggio inviato** da un oggetto per segnalare l'**occorrenza di un'azione**.

**L'azione** può essere causata dall'interazione dell'utente, ad esempio la pressione di un pulsante, oppure generata da altre logiche di programma, ad esempio la modifica di un valore della proprietà.

# Basi Programmazione - Gestione Eventi

```
class Counter
{
    public event EventHandler ThresholdReached;

    protected virtual void OnThresholdReached(EventArgs e)
    {
        EventHandler handler = ThresholdReached;
        if (handler != null)
        {
            handler(this, e);
        }
    }

    // provide remaining implementation for the class
}
```

# Basi Programmazione - Gestione Eventi

```
using System;

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            Counter c = new Counter(new Random().Next(10));
            c.ThresholdReached += c_ThresholdReached;

            Console.WriteLine("press 'a' key to increase total");
            while (Console.ReadKey(true).KeyChar == 'a')
            {
                Console.WriteLine("adding one");
                c.Add(1);
            }

            static void c_ThresholdReached(object sender, EventArgs e)
            {
                Console.WriteLine("The threshold was reached.");
                Environment.Exit(0);
            }
        }
    }
}
```

```
class Counter
{
    private int threshold;
    private int total;

    public Counter(int passedThreshold)
    {
        threshold = passedThreshold;
    }

    public void Add(int x)
    {
        total += x;
        if (total >= threshold)
        {
            OnThresholdReached(EventArgs.Empty);
        }
    }

    protected virtual void OnThresholdReached(EventArgs e)
    {
        EventHandler handler = ThresholdReached;
        if (handler != null)
        {
            handler(this, e);
        }
    }

    public event EventHandler ThresholdReached;
}
}
```

# Basi Programmazione - Gestione Delegate

Un **delegato** è un tipo che contiene un **riferimento a un metodo**.

Un delegato è dichiarato con una **firma che mostra il tipo restituito e i parametri** per i metodi a cui fa riferimento e può contenere riferimenti solo per i metodi che corrispondono alla firma.

Un delegato è pertanto equivalente a un **puntatore a funzione** indipendente dai tipi o un callback.

Una dichiarazione di delegato è sufficiente per definire una classe delegata.

# Basi Programmazione - Gestione Delegate

```
// Create a method for a delegate.
```

```
public static void DelegateMethod(string message)
{
    System.Console.WriteLine(message);
}
```

```
// Instantiate the delegate.
```

```
Del handler = DelegateMethod;
```

```
// Call the delegate.
```

```
handler("Hello World");
```

# Basi Programmazione - Gestione Delegate

C#

```
public void MethodWithCallback(int param1, int param2, Del callback)
{
    callback("The number is: " + (param1 + param2).ToString());
}
```

È quindi possibile passare il delegato creato in precedenza a tale metodo:

C#

```
MethodWithCallback(1, 2, handler);
```