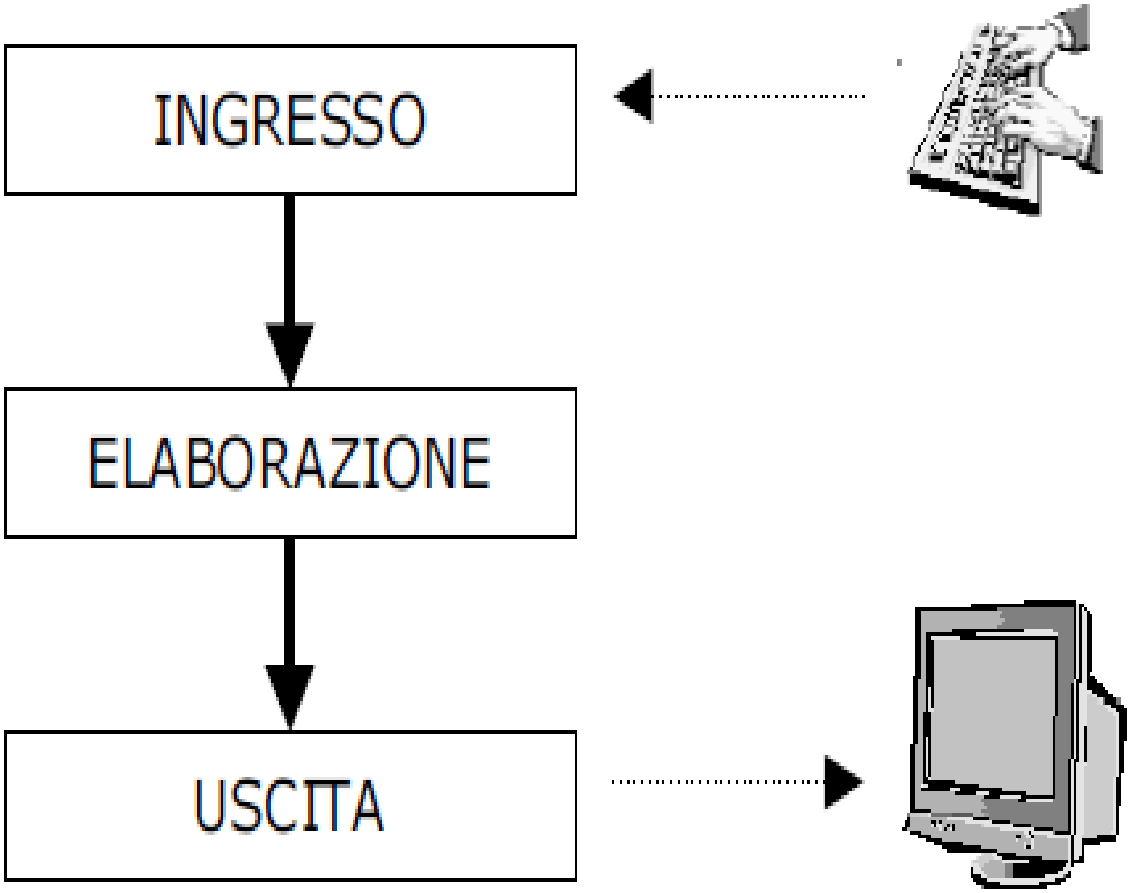
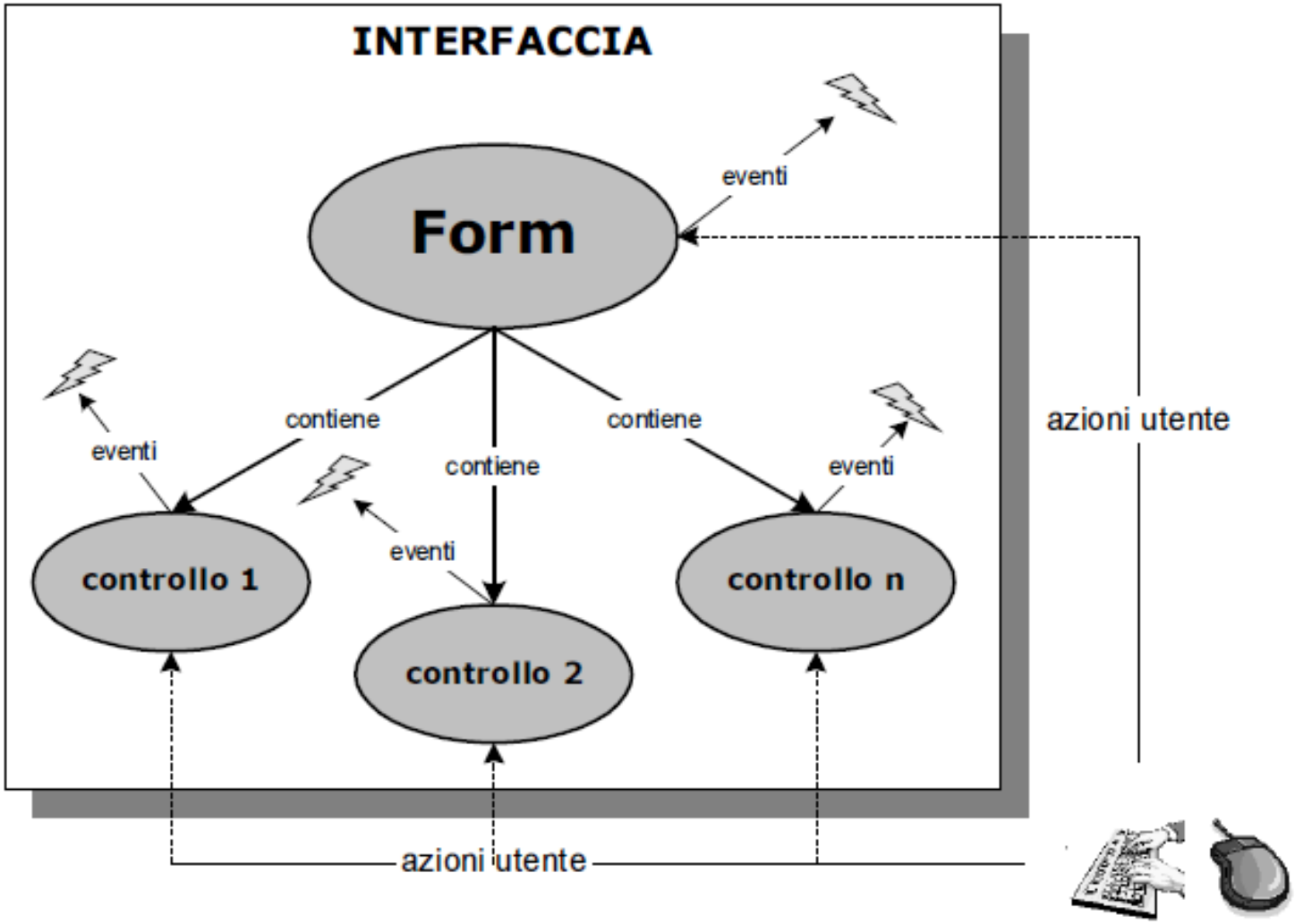
# interfacce grafiche con C#

Prof. Francesco Accarino
IIS Altiero Spinelli Sesto San Giovanni

# Schema di funzionamento di una tipica «Applicazione Console».



# Schema funzionale di una tipica applicazione grafica (winform)



## Controlli (Bottoni, Caselle di testo RadioButton ecc.)

I controlli sono oggetti visuali, oggetti, cioè, in grado di "visualizzare se stessi" sullo schermo. In effetti, non esistono metodi specifici per la visualizzazione dei controlli, essi lo fanno da sé; tutto ciò che deve fare il programmatore è:

a) creare il controllo;

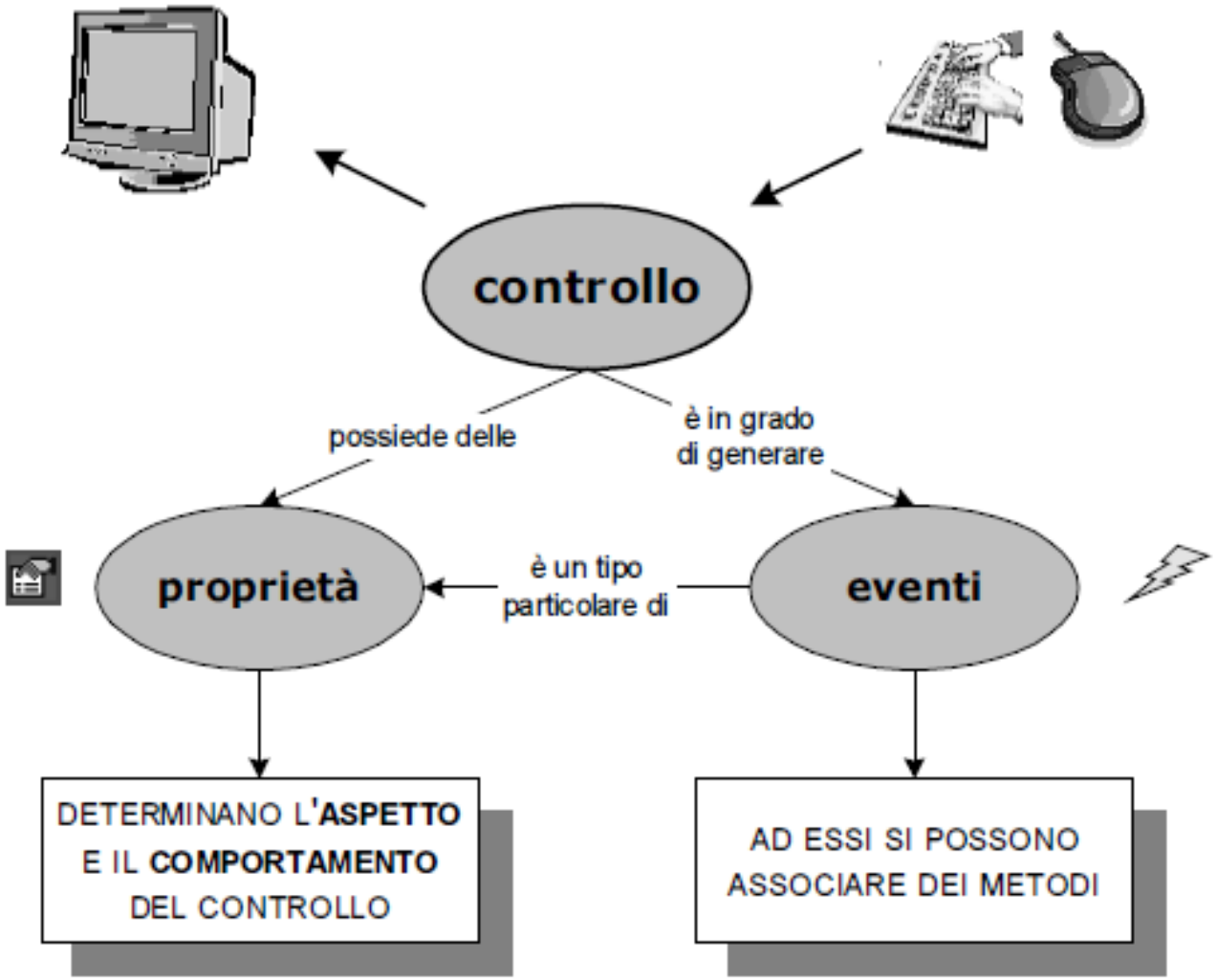
b) definirne l'aspetto, impostando opportunamente le sue «proprietà»

c) definire e associare dei metodi agli eventi del controllo che si desidera gestire.

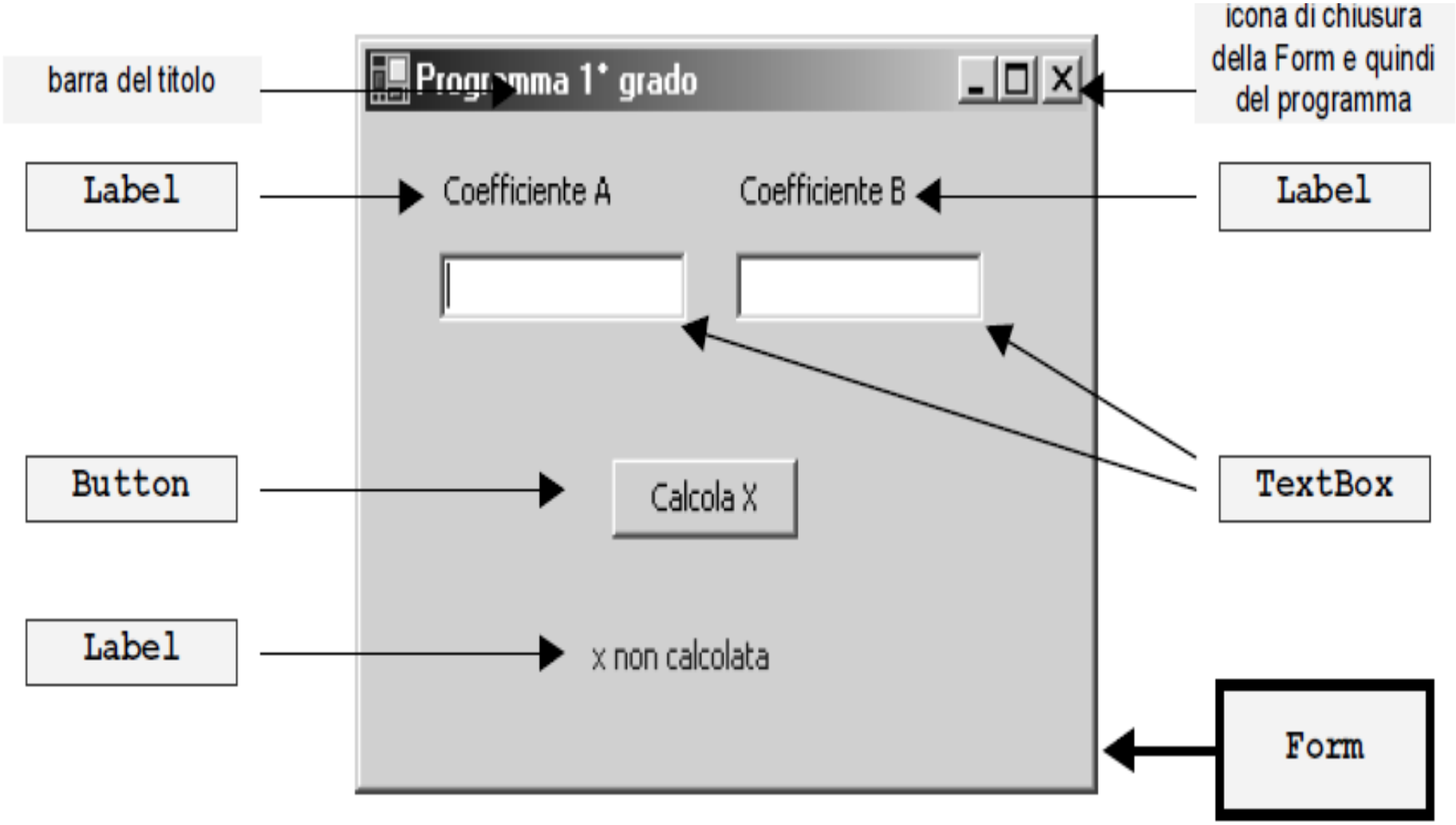
d) aggiungere il controllo all'interfaccia.

Dopodiché esso sarà sempre visibile e in grado di ricevere le azioni dell'utente, almeno fino a quando non sarà eliminato dall'interfaccia oppure temporaneamente reso invisibile o disabilitato.

# Rappresentazione schematica del rapporto tra controllo, proprietà ed eventi.

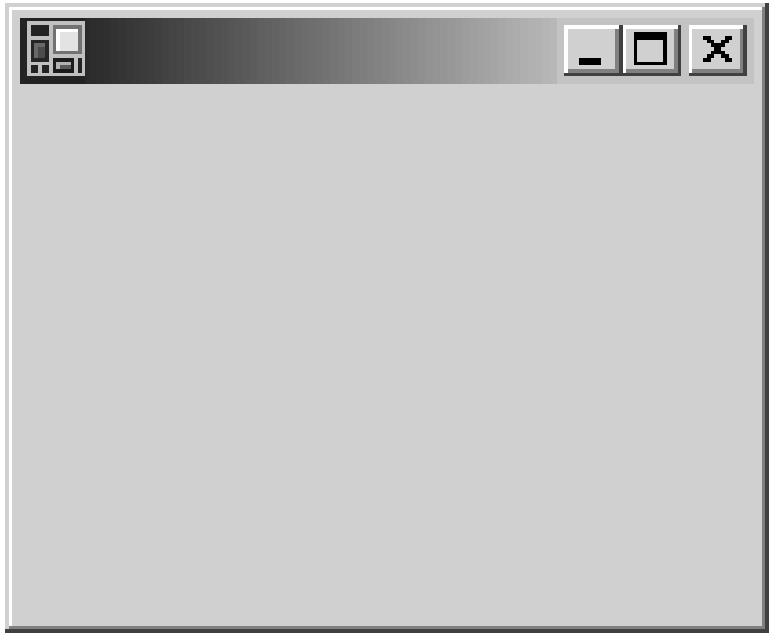


# Form contenitore di Controls

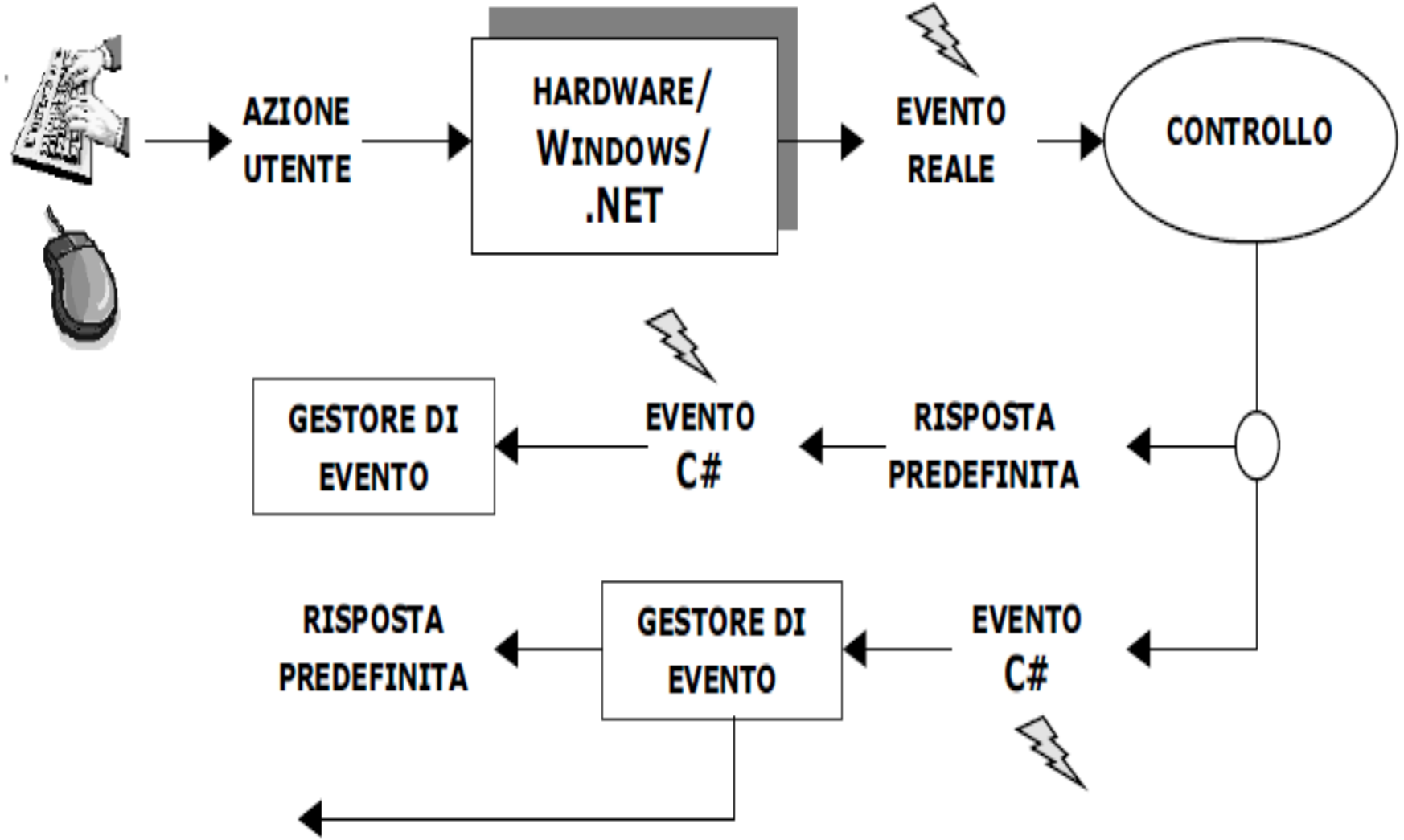


# Scheletro di base di un programma

```
using System;
using System.Windows.Forms;
class MainForm: Form
{
        public MainForm()
        {
        }
        public static void Main()
        {
            Application.Run( new MainForm() );
        }
}
```



# Gestione degli eventi

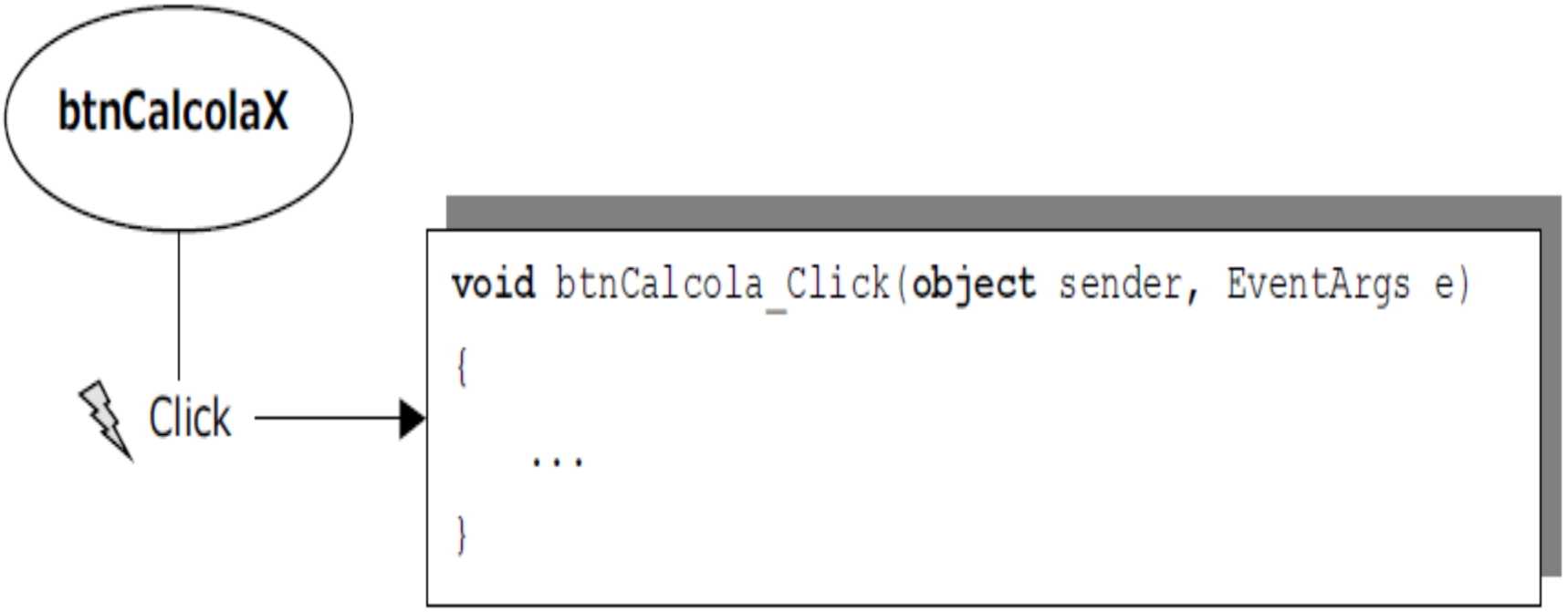


# Designare un gestore di evento

```
btnCalcolaX.Click += new EventHandler(btnCalcolaX_Click);
```

produce un risultato che può essere così schematizzato:

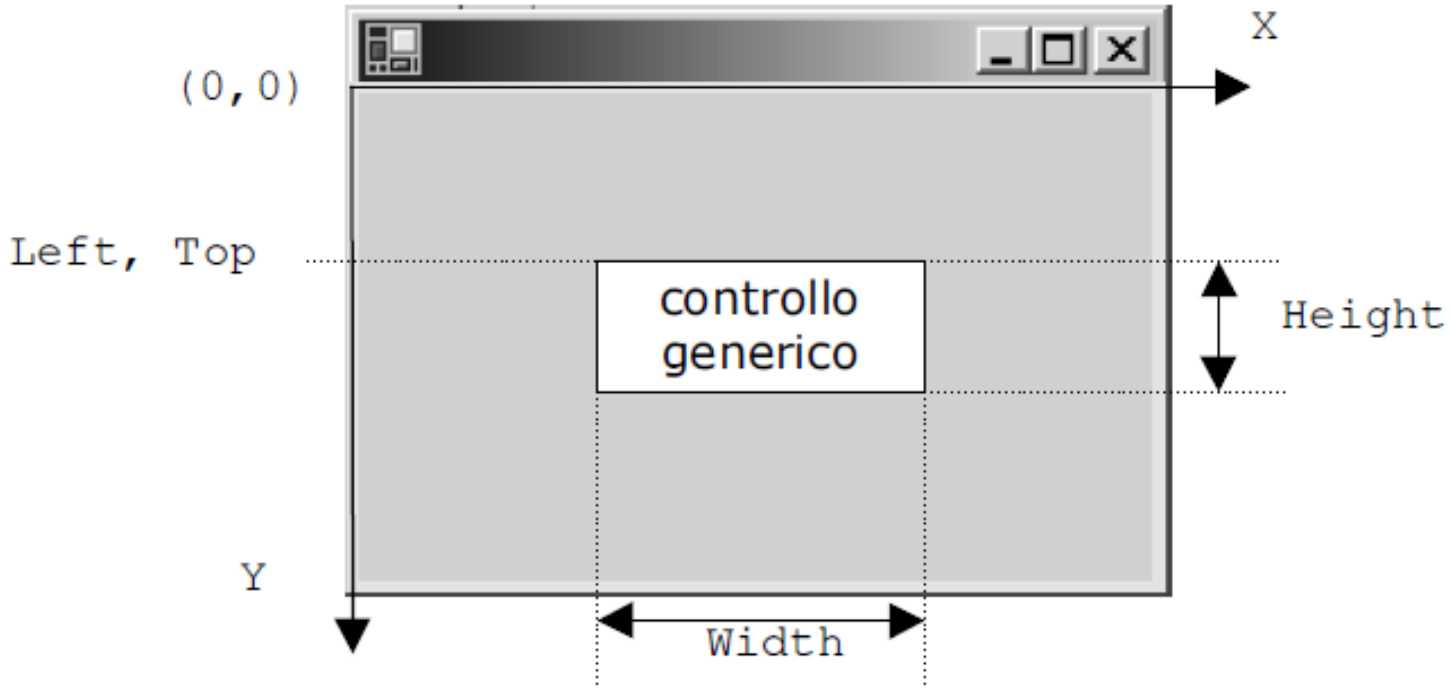
Schema associazione tra metodo `btnCalcolaX_Click` e l'evento `Click`.



# Posizione e dimensioni dei controlli

La posizione di un controllo nella client area può essere espressa in due modi:

a) attraverso la coppia di proprietà Left (ascissa) e Top (ordinata);
b) attraverso la proprietà Location, che appartiene a un particolare tipo di dato: il Point.



# Posizione e dimensioni dei controlli

Ad esempio, il seguente codice (il bottone b si suppone già creato):

```
b.Left = 100;
b.Top = 150;
```

equivale all'istruzione:

```
b.Location = new Point(100, 150);
```

che può essere tradotta in: «crea un oggetto di tipo Point e assegnalo alla proprietà Location».
E' possibile accedere, ma non modificare, alle singole coordinate memorizzate nella proprietà Location:

```
lblEsempio.Left = b.Location.X + 50; // ok: accesso all'ascissa di b
lbl.Location.Y = b.Location.Y; // errore!
```

Infine, è possibile utilizzare variabili di tipo Point.

```
Point coordinate = new Point(100, 100);
lblEsempio.Location = coordinate;
```

# Posizione e dimensioni dei controlli

la proprietà Size memorizza le dimensioni del controllo sotto forma di una coppia di valori – Width e Height

Ad esempio, il seguente codice (il bottone b si suppone già creato):

```
b.Width = 50;
b.Height = 25;
```

equivale all'istruzione:

```
b.Size = new Size(50, 25);
```

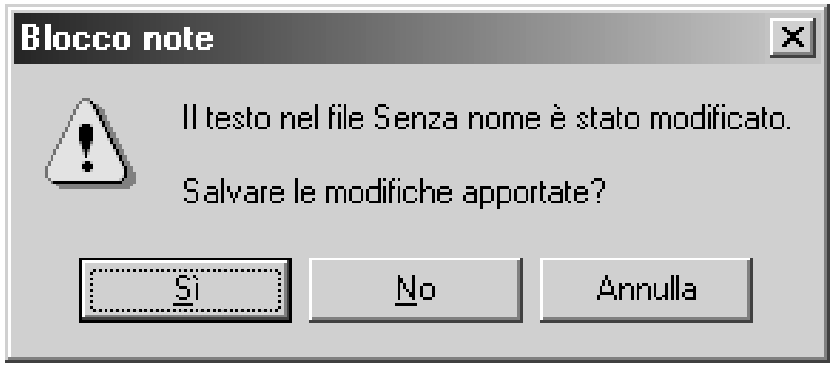
che può essere tradotta in: «crea un oggetto di tipo Size e assegnalo alla proprietà Size».

è possibile utilizzare variabili di tipo Size.

```
Size dimensioni = new Size(50, 40);
lblEsempio.Size = dimensioni;
```

# VISUALIZZARE MESSAGGI: CLASSE «MessageBox»

In molte situazioni si presenta la necessità di comunicare all'utente lo stato di una certa elaborazione, oppure di chiedere una conferma prima di procedere o meno all'esecuzione di una determinata operazione.




L'esempio mostra un particolare tipo di form, chiamato «message dialog», che mediante tre Button chiede all'utente di selezionare una fra tre alternative possibili. La principale caratteristica di una message dialog è che assume il pieno controllo sulle azioni dell'utente; in altre, parole, finché la dialog non viene chiusa non è possibile accedere al resto dell'interfaccia.

Le message dialog sono fondamentalmente impiegate per servire i seguenti scopi:

1) visualizzare un messaggio informativo;
2) visualizzare un messaggio informativo di errore;
3) visualizzare un messaggio di avvertimento o di richiesta di conferma

Una message dialog viene visualizzata mediante l'esecuzione del metodo Show() della classe MessageBox.

L'invocazione di tale metodo può assumere varie forme, in base al numero di elementi con i quali si intende caratterizzare la message dialog.

Il suo prototipo è:

```
DialogResult Show(string testo,
                  string titoloopz,
                  MessageBoxButtons buttonsopz,
                  MessageBoxIcon iconopz);
```

dove:

testo ➡ rappresenta il messaggio da visualizzare;

titolo ➡ rappresenta il testo da visualizzare sulla barra del titolo della message dialog;

buttons ➡ indica quali bottoni visualizzare;

icon ➡ indica con quale icona caratterizzare la message dialog.

## Esempio di message dialog informativa.


Tale risultato si ottiene mediante l'invocazione:

```
MessageBox.Show("Questo è un messaggio informativo");
```

## Esempio di message dialog informativa con testo sulla barra del titolo.


Ciò si ottiene specificando come secondo argomento il testo da visualizzare sulla barra del titolo:

```
MessageBox.Show("Questo è un messaggio informativo", "Questo è il titolo");
```

Esempio di message dialog informativa con icona «Information».


Ciò richiede di specificare altri due parametri, poiché è impossibile visualizzare una message dialog con icona senza specificare anche quali bottoni visualizzare:

```
MessageBox.Show("Questo è un messaggio informativo",
                "Questo è il titolo", MessageBoxButtons.OK, MessageBoxIcon.Information);
```

| ICONA - TIPO MessageBoxIcon | DESCRIZIONE |
|---|---|
| Asterisk | Viene visualizzata l'icona «informazione». |
| Error | Viene visualizzata l'icona «errore» |
| Esclamation | Viene visualizzata l'icona «punto esclamativo». |
| Hand | Viene visualizzata l'icona «mano». |
| Information | Viene visualizzata l'icona «informazione». |
| Question | Viene visualizzata l'icona «punto di domanda». |
| Stop | Viene visualizzata l'icona «cartello di stop». |
| Warning | Viene visualizzata l'icona «punto esclamativo». |

# Elaborare la risposta dell'utente

Mediante le message dialog puramente informative la comunicazione avviene sempre e solo dal programma verso l'utente. Quest'ultimo può solo confermare di aver "ricevuto" il messaggio.
Diverso è il caso delle message dialog di avvertimento o conferma. Con esse si chiede all'utente di selezionare una tra più scelte possibili, rappresentate dai bottoni visualizzati. In questo caso, ovviamente, il programma dovrà elaborare in qualche modo la risposta dell'utente.

Segue un frammento di codice che visualizza una message dialog di conferma ed elabora successivamente il valore ritornato dal metodo Show(), che rappresenta appunto la risposta dell'utente:

```
DialogResult scelta;
scelta = MessageBox.Show("Confermi l'operazione?","Questo è il titolo",
                                    MessageBoxButtons.YesNo,
                                    MessageBoxIcon.Warning);

if (scelta == DialogResult.Yes)
{
...// effettua l'operazione di cui è stata richiesta la conferma
}
```

# Elenco dei possibili valori prodotti del metodo Show().

| RIPOSTA UTENTE - TIPO DialogResult | DESCRIZIONE |
|---|---|
| Abort | E' stato cliccato il bottone «Termina». |
| Cancel | E' stato cliccato il bottone «Annulla». |
| Ignore | E' stato cliccato il bottone «Ignora». |
| No | E' stato cliccato il bottone «No». |
| OK | E' stato cliccato il bottone «Ok». |
| Retry | E' stato cliccato il bottone «Riprova». |
| Yes | E' stato cliccato il bottone «Si». |

# Elenco dei possibili valori del parametro buttons.

| BOTTONI<br>-<br>TIPO MessageBoxButtons | DESCRIZIONE |
|---|---|
| AbortRetryIgnore | Sono visualizzati i bottoni «Termina», «Riprova», «Ignora». |
| OK | E' visualizzato il bottone «OK». |
| OKCancel | Sono visualizzati i bottoni «OK», «Annulla». |
| RetryCancel | Sono visualizzati i bottoni «Riprova», «Annulla». |
| YesNo | Sono visualizzati i bottoni «Si», «No». |
| YesNoCancel | Sono visualizzati i bottoni «Si», «No», «Annulla». |

# Esercitazione 2: Realizzare un'applicazione come quella mostrata in figura: