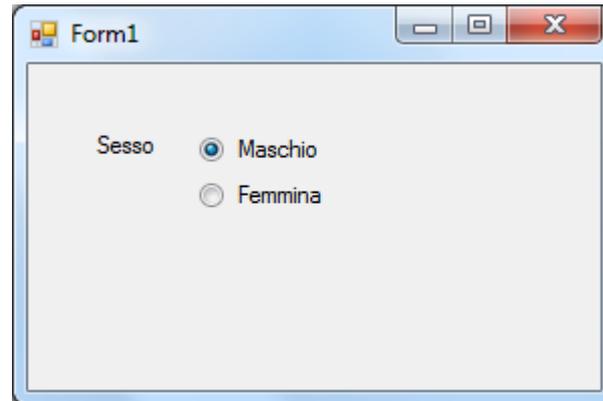

RadioButton E CheckBox

Prof. Francesco Accarino
IIS Altiero Spinelli Sesto San Giovanni

Classe «RadioButton»

Un RadioButton svolge in pratica la funzione di un interruttore, il quale può trovarsi soltanto nello stato «accesso» oppure nello stato «spento»; analogamente, un RadioButton può assumere due soli stati possibili: checked o unchecked («marcato» o «non marcato»).

Un RadioButton funziona sempre in collaborazione con altri RadioButton, insieme ai quali definisce la lista dei possibili stati che può assumere un determinato valore; in ogni momento uno solo tra i RadioButton può essere checked, determinando dunque lo stato del valore in questione.



Per questo motivo, benché non rappresenti in sé un errore formale, inserire nel form un solo RadioButton è semplicemente privo di senso.

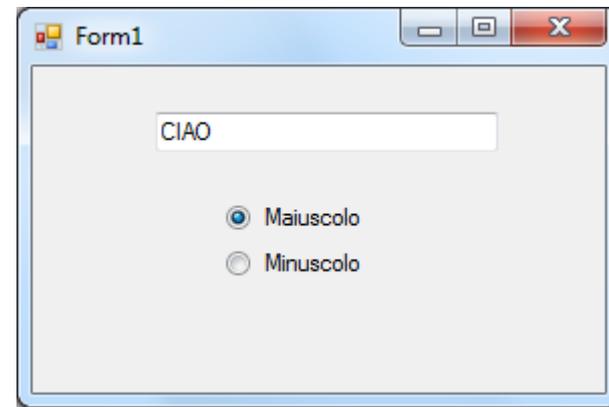
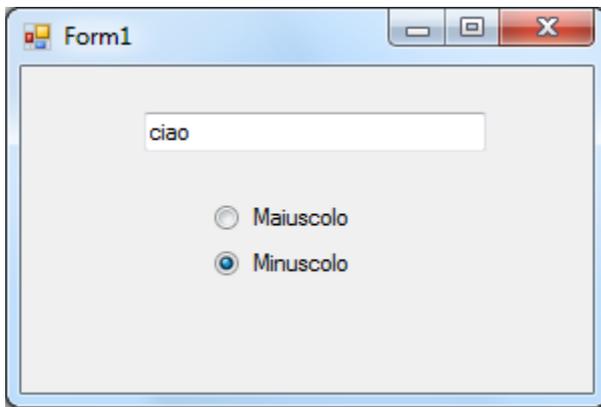
Proprietà

La proprietà più importante di un RadioButton è la proprietà **Checked**, la quale ne memorizza lo stato: «marcato» (**true**), «non marcato» (**false**).

Eventi

Di norma non è necessario gestire gli eventi sollevati dai RadioButton, poiché essi elaborano autonomamente le azioni dell'utente, impostando in modo appropriato il proprio stato, «marcato» o «non marcato», in relazione allo stato degli altri RadioButton.

E' invece necessario gestire gli eventi di sola notifica **Click** o **CheckedChanged** se si desidera che lo stato di uno o più controlli venga aggiornato immediatamente in relazione al cambiamento dello stato del RadioButton in questione.

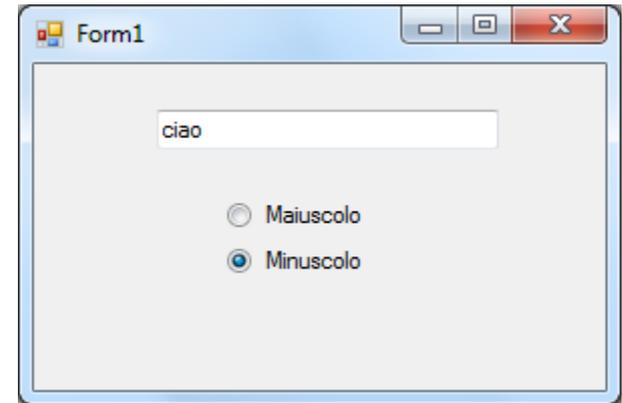


Eventi Gestori di eventi loro associazione

```
this.rbuMaiuscolo.CheckedChanged += new  
System.EventHandler(Testo_CheckedChanged);
```

```
this.rbuMinuscolo.CheckedChanged += new  
System.EventHandler(Testo_CheckedChanged);
```

```
void Testo_CheckedChanged(object sender, EventArgs e)  
{  
    if (rbuMaiuscolo.Checked == true)  
        txtEsempio.CharacterCasing = CharacterCasing.Upper;  
    else  
        txtEsempio.CharacterCasing = CharacterCasing.Lower;  
}
```



RAGGRUPPARE I CONTROLLI

Esistono due controlli, **Panel** e **GroupBox**, il cui ruolo non è quello di “dialogare” con l’utente ma di fungere da «contenitori» (container) per gli altri controlli, **TextBox**, **Label**, **RadioButton**, eccetera. Come «contenitori» entrambi svolgono una funzione analoga a quella del form, cioè “ospitano” altri controlli e possono influenzare il loro aspetto e il loro comportamento.

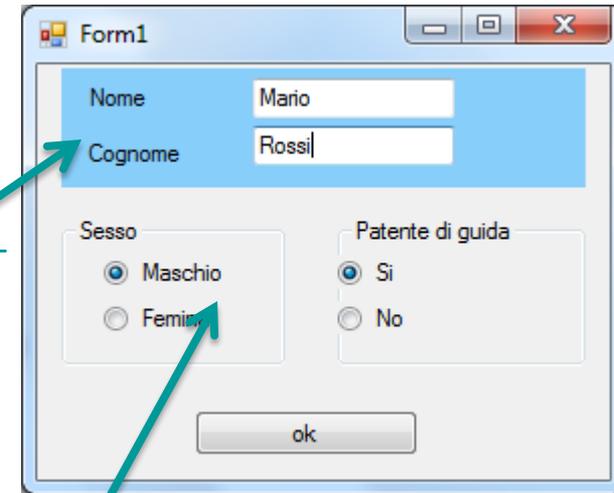
Esistono svariate ragioni per raggruppare dei controlli inserendoli in un Panel o un GroupBox, tra le quali:

- ❑ creare, ad esempio mediante un riquadro o un particolare colore di sfondo, delle aree visivamente separate dal resto dell’interfaccia, dedicate ad ospitare dei controlli logicamente correlati tra loro;
- ❑ facilitare l’aggiornamento dello stato di alcune proprietà appartenenti a controlli logicamente correlati tra loro. Ad esempio, impostando a false la proprietà Enabled di un Panel vengono automaticamente disabilitati tutti i controlli in esso contenuti;
- ❑ creare insiemi di RadioButton funzionalmente indipendenti tra loro.

Esempio

```
private System.Windows.Forms.Panel pnlNominativo;  
this.pnlNominativo = new System.Windows.Forms.Panel();  
this.pnlNominativo.BackColor = System.Drawing.Color.LightSkyBlue;  
this.pnlNominativo.Controls.Add(this.txtCognome);  
this.pnlNominativo.Controls.Add(this.label2);  
this.pnlNominativo.Controls.Add(this.txtNome);  
this.pnlNominativo.Controls.Add(this.label1);  
this.pnlNominativo.Location = new System.Drawing.Point(12, 2);  
this.pnlNominativo.Name = "panel1";  
this.pnlNominativo.Size = new System.Drawing.Size(260, 60);  
this.pnlNominativo.TabIndex = 5;
```

```
private System.Windows.Forms.GroupBox gpbSceltaSesso;  
this.gpbSceltaSesso = new System.Windows.Forms.GroupBox();  
this.gpbSceltaSesso.Controls.Add(this.rbuMaschio);  
this.gpbSceltaSesso.Controls.Add(this.rbuFemmina);  
this.gpbSceltaSesso.Location = new System.Drawing.Point(12, 77);  
this.gpbSceltaSesso.Name = "groupBox1";  
this.gpbSceltaSesso.Size = new System.Drawing.Size(111, 75);  
this.gpbSceltaSesso.TabIndex = 6;  
this.gpbSceltaSesso.TabStop = false;  
this.gpbSceltaSesso.Text = "Sesso";
```



Se non avessimo creato i due Group Box non sarebbe stato possibile fare la doppia scelta perché i quattro bottoni si sarebbero comportati come un unico gruppo e quindi uno solo dei quattro sarebbe stato attivo

Esercizio 4 sviluppare un'applicazione come quella mostrata in figura

The image displays two windows from a Windows application. The first window, titled 'Form1', contains a form with the following fields and options:

- Nome: Mario
- Cognome: Rossi
- Sesso: Maschio, Femina
- Patente di guida: Si, No
- ok button

The second window is a summary dialog box with the following text:

Ti chiami: Mario Rossi
sei: Maschio
Hai la Patente

An arrow points from the 'ok' button in the first window to the 'OK' button in the second window.

Cliccando su ok viene visualizzato il riassunto delle scelte effettuate

Anche un **CheckBox**, in modo simile a un **RadioButton**, svolge la funzione di un interruttore, il quale può trovarsi soltanto nello stato «**accesso**» oppure nello stato «**spento**» e cioè «**marcato**» o «**non marcato**».

Diversamente dai **RadioButton**, un **CheckBox** non agisce in collaborazione con altri **CheckBox**, esso, infatti, definisce lo stato, **checked** o **unchecked**, di una opzione che è indipendente da eventuali altre opzioni rappresentate nell'interfaccia.

Proprietà

Analogamente al controllo **RadioButton**, la classe **CheckBox** definisce la proprietà **Checked**, attraverso la quale è possibile conoscere e impostare il suo stato:

«**marcato**» (**true**), «**non marcato**» (**false**).

Eventi

Analogamente a quanto accade per i **RadioButton**, di norma non è necessario gestire gli eventi sollevati da un **CheckBox**. E' comunque appropriato gestire gli eventi **Click** o **CheckedChanged** se si desidera che lo stato di uno o più controlli venga aggiornato immediatamente in relazione al cambiamento dello stato del **CheckBox** in questione.

Esercizio 5: Sviluppare un'applicazione come quella mostrata in figura:

The figure shows three sequential screenshots of a Windows application window titled "Form1".

- Left screenshot:** The form is empty. The "Nome" and "Cognome" fields are blank. The "Laurea" and "Diploma Secondaria" checkboxes are unchecked. The "Laurea" section has empty "Facoltà" and "Anno" fields. The "Diploma" section has empty "Scuola" and "Anno" fields.
- Middle screenshot:** The "Laurea" checkbox is checked. The "Facoltà" field contains the text "Ingegneria".
- Right screenshot:** Both "Laurea" and "Diploma Secondaria" checkboxes are checked. The "Facoltà" field contains "Ingegneria", the "Anno" field contains "2012", and the "Scuola" field contains "IIS Altiero Spinelli".

Si sfrutta ancora il **GroupBox** non solo per migliorare l'aspetto dell'applicazione ma anche per abilitare velocemente gruppi di controlli sulla notifica dell'evento **CheckedChanged** dei **CheckBox**.

Esercizio 5 Continua

Form1

Nome Franco

Cognome Rossi

Laurea Diploma Secondaria

Laurea

Facoltà Ingegneria Anno 2012

Diploma

Scuola IIS Altiero Spinelli Anno 2005

ok

```
private void cbuLaurea_CheckedChanged(object sender, EventArgs e)
{
    if (cbuLaurea.Checked == true)
        gpbLaurea.Enabled = true;
    else
        gpbLaurea.Enabled = false;
}
```

```
private void cbuSecondaria_CheckedChanged(object sender, EventArgs e)
{
    if (cbuSecondaria.Checked == true)
        gpbDiploma.Enabled = true;
    else
        gpbDiploma.Enabled = false;
}
```

Esercizio 5 Continua

Form1

Nome Franco
Cognome Rossi

Laurea Diploma Secondaria

Laurea
Facoltà Anno
Ingegneria 2012

Diploma
Scuola Anno
IIS Altiero Spinelli 2005

ok

Riepilogo

Nome Franco
Cognome Rossi

Diploma
Scuola Anno
IIS Altiero Spinelli 2005

Laurea
Facoltà Anno
Ingegneria 2012

Confirma Annulla

conferma dei dati inseriti

 I tuoi dati sono stati acquisiti

OK

Facendo Click sul bottone ok viene visualizzato il form di riepilogo dei dati sul quale se si clicca su Conferma abbiamo il messaggio di conferma e quindi la chiusura del form; se invece si clicca su annulla si chiude semplicemente il form

Esercizio 5 Continua

```
private void btnOk_Click(object sender, EventArgs e)
{
    Riepilogo riepilogo = new Riepilogo();
    riepilogo.txtNome.Text = this.txtNome.Text;
    riepilogo.txtCognome.Text = this.txtCognome.Text;

    if (cbuSecondaria.Checked == true)
    {
        riepilogo.gpbDiploma.Enabled = true;
        riepilogo.txtScuola.Text = this.txtScuola.Text;
        riepilogo.txtAnnoDiploma.Text = this.txtAnnoDiploma.Text;
    }

    if (cbuLaurea.Checked == true)
    {
        riepilogo.gpbLaurea.Enabled = true;
        riepilogo.txtAnnoLaurea.Text = this.txtAnnoLaurea.Text;
        riepilogo.txtfacolta.Text = this.txtFacolta.Text;
    }

    riepilogo.Show();
}

private void btnConferma_Click(object sender, EventArgs e)
{
    MessageBox.Show("I tuoi dati sono stati acquisiti", "conferma dei dati inseriti", MessageBoxButtons.OK, MessageBoxIcon.Information);
    this.Close();
}

private void btnAnnulla_Click(object sender, EventArgs e)
{
    this.Close();
}
```

Form1

Nome Franco
Cognome Rossi

Laurea Diploma Secondaria

Laurea
Facoltà Anno
Ingegneria 2012

Diploma
Scuola Anno
IIS Altiero Spinelli 2005

ok

Riepilogo

Nome Franco
Cognome Rossi

Diploma
Scuola Anno
IIS Altiero Spinelli 2005

Laurea
Facoltà Anno
Ingegneria 2012

Conferma Annulla