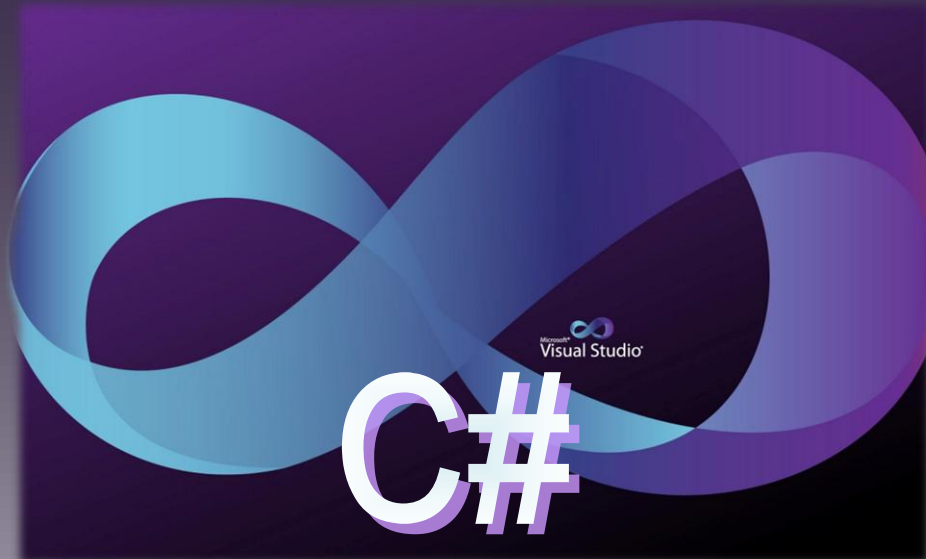


**ISTITUTO TECNICO INDUSTRIALE
G. M. ANGIOY
SASSARI**



CORSO DI PROGRAMMAZIONE

UNO SGUARDO ALL'INTERFACCIA VISUALE

DISPENSA 04.01

04-01_Interfaccia_Visuale_[ver_15]



Questa dispensa è rilasciata sotto la licenza Creative Common CC BY-NC-SA. Chiunque può copiare, distribuire, modificare, creare opere derivate dall'originale, ma non a scopi commerciali, a condizione che venga riconosciuta la paternità dell'opera all'autore e che alla nuova opera vengano attribuite le stesse licenze dell'originale.

Versione del: **07/11/2015**

Revisione numero: **15**

Prof. Andrea Zoccheddu
Dipartimento di Informatica

DIPARTIMENTO INFORMATICA E TELECOMUNICAZIONI





INTERFACCIA UTENTE

INTERFACCIA UTENTE

CONCETTI DI BASE

Quando si costruisce un progetto ci si riferisce spesso all'utente. Per utente si intende la persona fisica ideale che potenzialmente andrà a utilizzare il programma e che dovrà interagire con esso. Per esempio un utente potrebbe essere la commessa del negozio, il contabile dell'azienda, il giocatore al computer, il navigatore su internet.

Il programmatore, il progettista dell'applicazione, deve sempre pensare al suo utente ideale quando costruisce il progetto; oltre al nucleo del progetto deve anche costruire l'interfaccia utente, cioè i modi con cui l'utente potrà usare il programma.

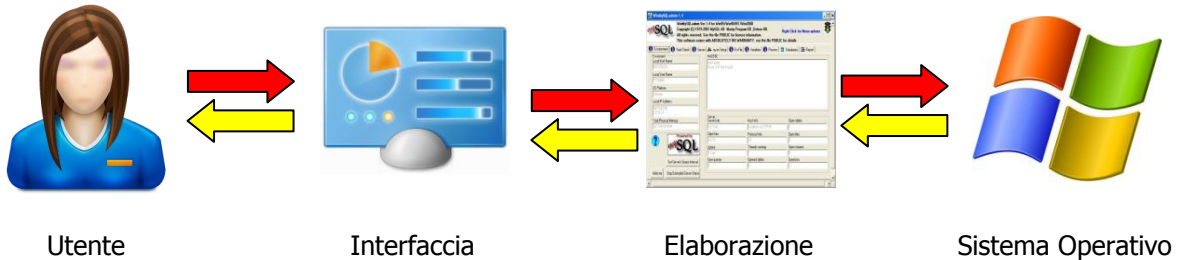
Quando si parla di **interfaccia utente** ci si riferisce a quella parte di un'applicazione che è preposta alla gestione della comunicazione con l'utente, consentendo a questo di immettere le informazioni da elaborare e di ricevere il risultato di tale elaborazione.

Il termine interfaccia denota **due significati** distinti, anche se correlati, a seconda della prospettiva secondo la quale lo si considera:

- dal punto di vista dell'utente, l'interfaccia è costituita dalla rappresentazione a video delle informazioni e dalle specifiche modalità di scambio;
- dal punto di vista del progettista, l'interfaccia è rappresentata dalla parte di programma che assolve alle funzioni di ricezione delle azioni dell'utente e di risposta del programma, visualizzando su video.

Ogni programma può pertanto essere costituito da due parti logiche:

- la parte interfaccia, che assolve alla funzione di comunicare con l'utente e dunque di scambiare i dati oggetto dell'elaborazione;
- la parte elaborazione, costituita dal codice che presiede alla vera e propria gestione interna dei dati;



COMUNICAZIONE CON L'UTENTE

Sebbene esistano diversi modi di comunicazione con l'utente, in questa dispensa prediligeremo una comunicazione visuale (GUI), guidata dagli eventi (event-driven).

Una comunicazione diversa da quella esaminata è quella di interfaccia console, che sfrutta funzioni specifici di acquisizione (ingresso) e di visualizzazione (uscita) dei dati. Questo tipo di interfaccia è ancora molto diffuso negli ambienti di sviluppo e di amministrazione di sistema.

Il modello di comunicazione visuale ha come punto di forza una rappresentazione visiva potente e versatile, che sfrutta colori e forme come interazione con l'utente. Il sistema visuale divenuto standard, è incentrato su finestre che possono ospitare altri controlli; e ciascun controllo è preposto a un tipo di comunicazione con



l'utente. Oltre alla tastiera, l'utente può usare il mouse come strumento preminente di gestione dell'interfaccia.

Il modello di gestione di interfaccia guidato dagli eventi si basa sull'assunto che gli elementi visuali interagiscano con l'utente reagendo a situazioni. Per evento si intende una situazione che può verificarsi sia a causa del comportamento dell'utente (sposta il mouse, preme tasti, digita sulla tastiera, ecc) sia per altre evenienze esterne (clock del timer, segnali dalla scheda di rete, interruzioni del sistema operativo, progresso di stampanti o risposte del file system).

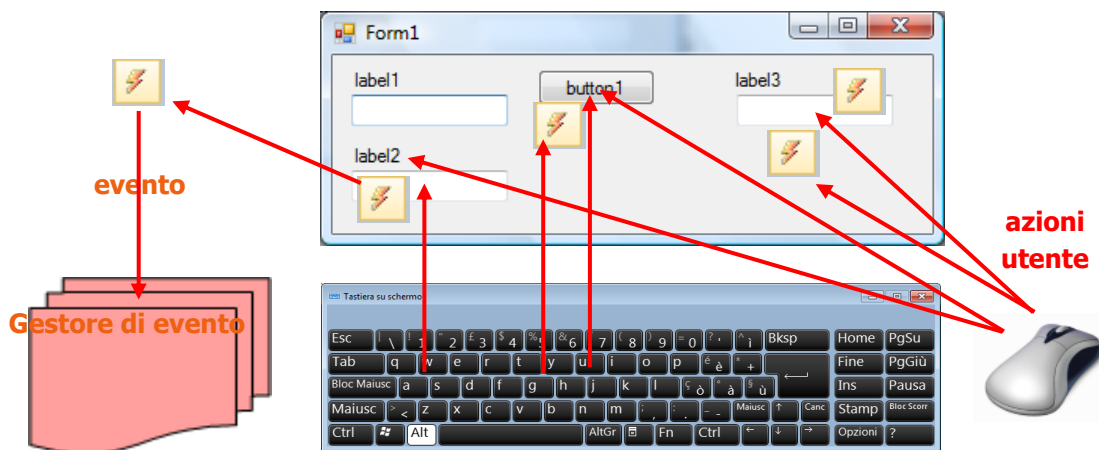
Gli elementi visuali (come finestre, pulsanti, menu o caselle di testo) hanno la capacità di rilevare (accorgersi di) gli eventi pertinenti e di poter reagire con una gestione opportuna. Per esempio, quando un utente fa clic sulla intestazione di un comando menu, normalmente questo apre la varietà delle scelte possibili e infine può scatenare una reazione controllata (es. File | Chiudi che normalmente consente di chiudere il file aperto).

Per progettare interfacce di questo tipo occorre tenere presente che gli eventi non si verificano in sequenza ordinata, ma possono accavallarsi in modo talvolta imprevedibile; per esempio in un videogioco l'utente (il giocatore) può trovarsi a competere in concorrenza con altri eventi guidati dal computer (timer o casualità) o persino da altri utenti. Questo tipo di programmazione può essere pensato come una programmazione concorrente, sebbene il parallelismo di azioni multiple richieda una programmazione più specifica.

STRUTTURA DI UNA INTERFACCIA GRAFICA

Il modello di interfaccia grafica | guidata dagli eventi (che da qui in poi chiameremo GUI, Graphic User Interface) è articolata con un'architettura piuttosto diversa da quello tipico delle Applicazioni Console. Per le esigenze di questo corso, esamineremo quattro aspetti fondamentali di una GUI:

- la finestra visuale (il form) principale che è normalmente associata all'applicazione;
- gli oggetti visuali (i controlli), contenuti nella finestra grafica;
- le situazioni generate (gli eventi) dalla finestra e dai controlli visuali;
- i metodi (i gestori di eventi) che reagiscono e gestiscono gli eventi.



Architettura di una GUI



I CONTROLLI

INTRODUZIONE AI CONTROLLI BASE

I CONTROLLI ANALIZZATI

Da questa dispensa si inizierà una rapida e sintetica panoramica dei controlli che è possibile usare in semplici applicazioni visuali. Ovviamente l'esame sarà superficiale e parziale e non intende esaurire l'argomento, ma solo aiutare il programmatore acerbo a intuire le potenzialità dell'ambiente e a scegliere quali intende approfondire per proprio conto.

LA LIBRERIA DEI CONTROLLI

L'ambiente di sviluppo basato sul .NET Framework offre un cospicuo numero di elementi visuali e non visuali per creare interfacce utente. I componenti sono elementi di Windows che riescono a mediare la comunicazione tra un utente e il sistema operativo dove l'applicazione è attiva. I controlli sono costruiti in un pacchetto che è noto come libreria di controlli di Windows Presentation Foundation (WPF).

La libreria WPF contiene informazioni sui controlli disponibili. In particolare i controlli che puoi usare sono i seguenti:

- | | | |
|----------------------------|----------------|-----------------------|
| ■ Border | ■ Frame | ■ Barra multifunzione |
| ■ BulletDecorator | ■ Grid | ■ RichTextBox |
| ■ Button | ■ GridSplitter | ■ ScrollBar |
| ■ Calendar | ■ GroupBox | ■ ScrollView |
| ■ Canvas | ■ Image | ■ Separator |
| ■ CheckBox | ■ Label | ■ Slider |
| ■ ComboBox | ■ ListBox | ■ StackPanel |
| ■ ContextMenu | ■ ListView | ■ StatusBar |
| ■ DataGrid | ■ Menu | ■ TabControl |
| ■ DatePicker | ■ Panel | ■ TextBlock |
| ■ DockPanel | ■ PasswordBox | ■ TextBox |
| ■ DocumentViewer | ■ Popup | ■ ToolBar |
| ■ Expander | ■ ProgressBar | ■ ToolTip |
| ■ FlowDocumentPageViewer | ■ PrintDialog | ■ TreeView |
| ■ FlowDocumentReader | ■ RadioButton | ■ WrapPanel |
| ■ FlowDocumentScrollViewer | ■ RepeatButton | ■ Viewbox |

In questo percorso di studi comunque non saranno analizzati tutti i controlli, ma solo alcuni. Ho evidenziato in giallo quelli che saranno descritti nel corso anche se non approfonditamente.

COS'È UN CONTROLLO

Un controllo è un elemento utile per la gestione dell'interfaccia utente. Ogni controllo è un oggetto. Poiché gli elementi di Windows sono costruiti a partire da elementi base, essi costituiscono una gerarchia. L'insieme dei controlli assomiglia cioè a un albero zoologico, dove un discendente è costruito a partire da un antenato.

Sebbene esista una classe base chiamata Control che serve come modello per costruire la maggior parte dei controlli, è utile sapere che esistono controlli che non derivano da Control.

Molti controlli però hanno Proprietà, Metodi ed Eventi simili, poiché spesso discendono dalla stessa classe base. Questo meccanismo potrà essere chiarito solo dopo aver studiato la programmazione ad oggetti (OOP).



La libreria WPF fornisce molti dei più comuni componenti dell'interfaccia utente che sono utilizzati in quasi tutte le applicazioni Windows; tra i controlli comuni figurano ad esempio Button, Label, TextBox, Menu e ListBox.

Questi oggetti sono denominati controlli. Sebbene WPF SDK usi per tradizione il termine "controllo" per indicare in modo generale qualsiasi classe di oggetto che rappresenta un oggetto visibile in un'applicazione, è importante notare che un oggetto non deve derivare dalla classe Control per essere un controllo visibile.

Le classi che derivano dalla classe Control dispongono di un oggetto ControlTemplate che consente al programmatore che usa un controllo di modificarne radicalmente l'aspetto senza dover creare necessariamente una nuova classe derivata.

BUTTON (PULSANTE)

Un controllo Button è costruito per reagire all'input dell'utente tramite mouse, tastiera o altro dispositivo di input e genera un evento Click. Un oggetto Button è un componente di base della GUI e può essere ospitato in un contenitore come per esempio un controllo Panel o un GroupBox.

Un'istanza di Button può trovarsi in stato normale, attivo o premuto.

CHECKBOX (CASELLA DI SELEZIONE)

Un oggetto CheckBox è costruito per rappresentare le opzioni che un utente può selezionare o deselezionare tramite la GUI. È possibile usare più controlli CheckBox per offrire all'utente più scelte. Un oggetto CheckBox è un componente di base della GUI e può essere ospitato in un contenitore come per esempio un controllo Panel o un GroupBox.

Un'istanza di CheckBox può trovarsi in stato deselezionato, selezionato o indeterminato.

COMBOBOX (CASELLA A DISCESA)

Un oggetto ComboBox è costruito per fornire all'utente un elenco di opzioni. L'elenco è mostrato e nascosto a seconda che il controllo sia esteso o compresso. Nello stato predefinito l'elenco è compresso, per cui è visualizzata un'unica opzione. L'utente fa clic sulla freccia di discesa per visualizzare l'elenco completo delle opzioni. Il ComboBox può essere possibile per l'utente digitare una opzione personale che non rientra nell'elenco.

Un'istanza di ComboBox può trovarsi in stato predefinito (compresso) o premuto (esteso).

CONTEXTMENU (MENU A COMPARSA)

Un oggetto ContextMenu è costruito per agganciarsi a un altro controllo visuale; questo consente al controllo di visualizzare uno specifico Menu detto contestuale che si riferisce al secondo controllo. In genere, un ContextMenu viene visualizzato dalla GUI al clic destro del mouse o tramite il tasto apposito della tastiera.

Un'istanza di ContextMenu può trovarsi in stato predefinito (nascosto) o attivo (aperto).

Nello stato predefinito, il controllo è nascosto. Quando si preme il pulsante destro del mouse sull'elemento agganciato al menu, il menu compare e si espande mostrando le sue voci.

DATAGRIDVIEW (GRIGLIA)

Il DataGrid è un controllo visuale complesso e articolato che serve per visualizzare dati in forma tabellare. Sarà discusso quando si studieranno gli array bidimensionali (matrici) e i database.

GROUPBOX (CONTENITORE DI RAGGRUPPAMENTO)

Un controllo GroupBox è costruito per ospitare altri controlli creando un gruppo. È un contenitore con un titolo utile per descrivere lo scopo dei controlli raggruppati.



IMAGE (IMMAGINE)

Un controllo Image è costruito per visualizzare immagini bitmap nelle GUI.

LABEL (ETICHETTA DI TESTO)

Un controllo Label è costruito per mostrare frasi e informazioni scritte nella GUI.

I controlli Label hanno sempre un contenuto di solo testo, ma poiché la Label è un oggetto discendente da ContentControl, può contenere anche un oggetto UIElement.

Un controllo Label fornisce un aiuto visivo per le altre parti della GUI. La Label può anche accettare input dell'utente come tasti di scelta rapida (combinazioni di tasti della tastiera).

LISTBOX (ELENCO DI SCELTE)

Un oggetto ListBox è costruito per fornire all'utente un elenco di opzioni. Lo scopo è simile al ComboBox ma è più semplice. L'elenco è sempre visibile (a differenza del ComboBox) e se le opzioni sono molte usa delle barre di scorrimento laterali. Solitamente il controllo risponde al Clic dell'utente sull'opzione scelta. A differenza del ComboBox non è mai possibile per l'utente digitare una opzione personale che non rientra nell'elenco.

MENU (MENU PRINCIPALE)

Il controllo Menu consente l'organizzazione gerarchica di elementi associati a comandi e gestori eventi. Ciascun oggetto Menu può contenere più controlli MenuItem. Ciascun oggetto MenuItem può essere associato ad un comando o un gestore eventi Click. MenuItem può inoltre disporre di più elementi MenuItem come elementi figlio, creando un sottomenu.

Il controllo è in stato predefinito quando nessun dispositivo, ad esempio un puntatore del mouse, si trova su Menu. Il controllo si sposta in stato attivo se il puntatore del mouse è posizionato su Menu. Il controllo si trova in stato premuto quando si preme su un pulsante del mouse sopra il Menu.

PANEL (PANNELLO CONTENITORE)

Il controllo Panel è la classe base per tutti i controlli contenitori di altri elementi. Quando si realizza una GUI, il Panel consente di creare barre di pulsanti o riquadri di controlli.

PASSWORDBOX (CASELLA PER PAROLE SEGRETE)

Il controllo PasswordBox è simile a una TextBox ma è utilizzato per immettere informazioni riservate o private.

POPUP (SUGGERIMENTO A COMPARSA)

Nel controllo Popup viene visualizzato il contenuto in una finestra separata mobile rispetto alla finestra dell'applicazione corrente. A differenza del ToolTip il controllo PopUp è capace di ospitare altri controlli, come immagini. Il controllo si aggancia ad un controllo di riferimento (come il ToolTip) detto controllo padre.

PROGRESSBAR (BARRA DI PROGRESSIONE)

Il controllo ProgressBar serve per indicare lo stato di avanzamento di un'operazione. Il controllo ProgressBar è costituito da una barra colorata che viene riempita gradualmente con il colore di evidenziazione del sistema mentre è in corso l'avanzamento dell'operazione. In effetti la barra indica un visivamente un valore numerico rispetto a valori minimo e massimo possibili (per esempio una percentuale compresa tra zero e cento).

PRINTDIALOG (DIALOGO DI STAMPA)

Il controllo PrintDialog consente di visualizzare una finestra di dialogo di stampa standard prevista per la configurazione automatica di oggetti di stampa.



RADIOBUTTON (PULSANTE ESCLUSIVO)

Un oggetto RadioButton è costruito per consentire l'opzione di scelta mutuamente esclusiva che un utente può selezionare o deselegionare tramite la GUI. I controlli RadioButton in genere vengono raggruppati in modo da offrire agli utenti un'unica scelta tra più opzioni; è possibile selezionare solo un pulsante per volta.

Un oggetto RadioButton è un componente di base della GUI e può essere ospitato in un contenitore come per esempio un controllo Panel o un GroupBox.

Un'istanza di RadioButton può trovarsi in stato deselegionato, selezionato.

RICHTEXTBOX (EDITORE DI TESTO)

L'elemento RichTextBox definisce un controllo di modifica con supporto incorporato per funzionalità come taglia e incolla, presentazione di documenti RTF e selezione di contenuto.

TEXTBOX (CASELLA DI TESTO)

Il controllo TextBox fornisce supporto per l'input di testo di base nelle applicazioni WPF. Questo controllo è stato già discusso in precedenza.

TOOLBAR (BARRA STRUMENTI)

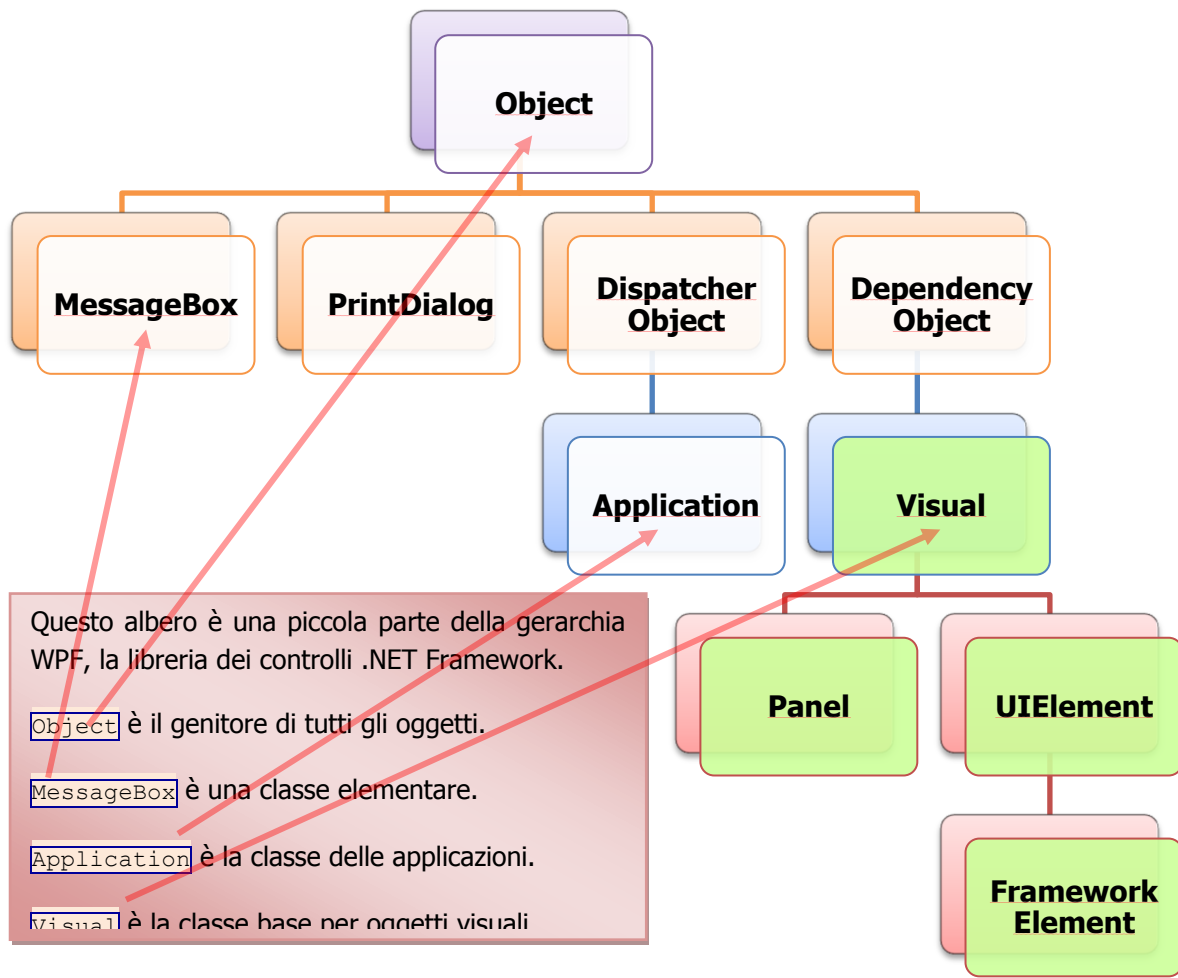
Il controllo ToolBar è un contenitore di un gruppo di comandi o controlli le cui funzioni sono in genere correlate.

CARATTERISTICHE DEI CONTROLLI

UNA GERARCHIA

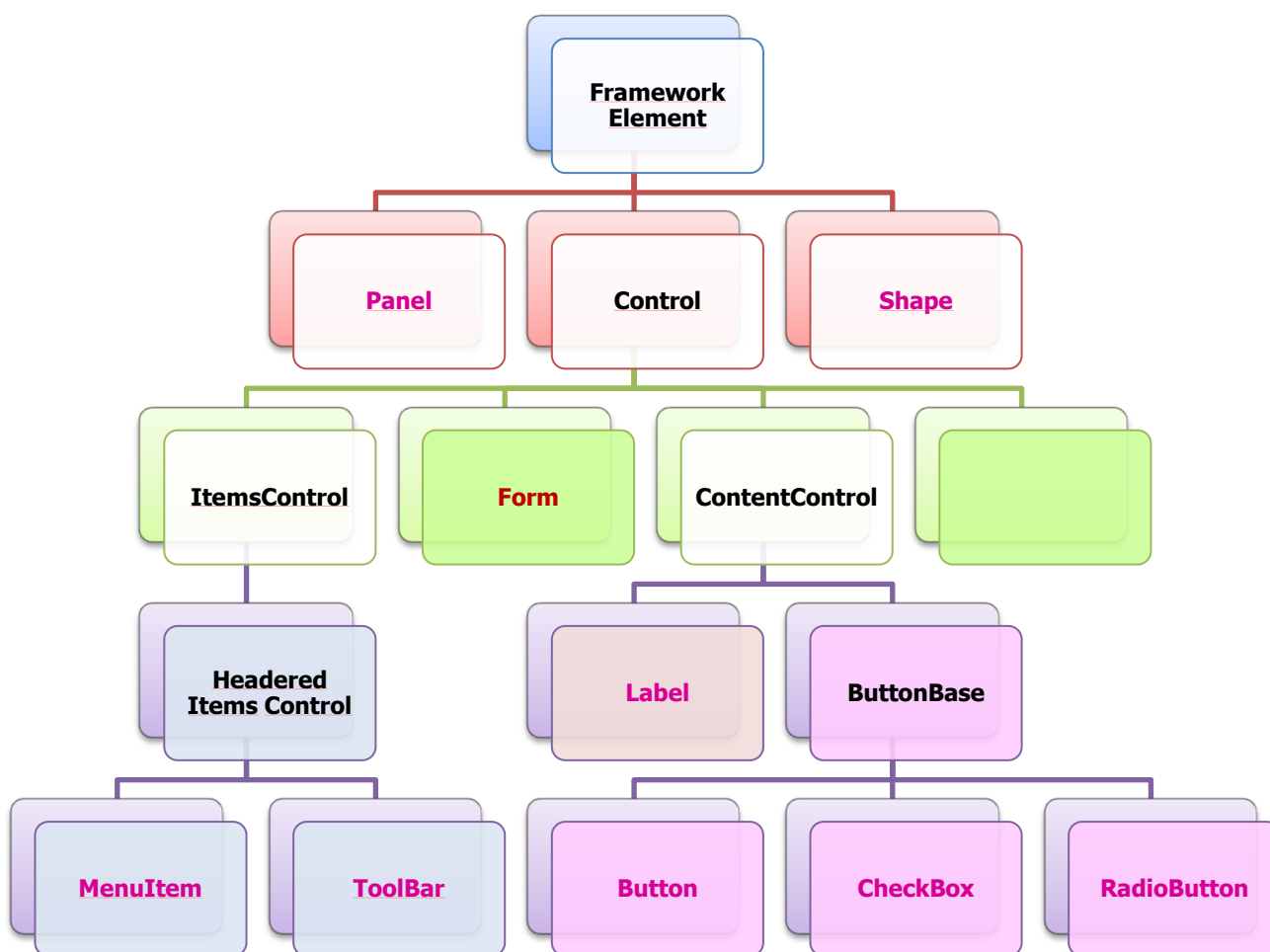
Sebbene la trattazione dei controlli richieda la conoscenza della programmazione ad oggetti, proviamo a introdurre una idea intuitiva della loro costruzione.

In Visual Studio ogni oggetto è costruito a partire da un precedente; per poter immaginare il meccanismo si pensi alla classe Animali, da cui discendono la classe Mammiferi e Rettili; dalla classe Mammiferi discendono i Felini e da questa i Gatti; dalla classe rettili discende la classe delle Lucertole. In modo analogo la classe base di Visual Studio è Object.



A partire dalla classe `Object` si costruiscono le prime classi elementari; si può notare la classe `MessageBox` che è una classe semplice (basata sui messaggi di Windows) e la classe `Visual`.

Dalla classe `FrameworkElement` (si noti che ciascun nome di classe è senza spazi o trattini, ma è una sequenza di lettere Maiuscole e minuscole) si costruiscono alcuni elementi importanti tra cui la classe `Control`. È questa la classe principale che serve per costruire la gran parte dei controlli visuali. Si osservi comunque che altre come `Panel` (base per i contenitori) e `Shape` (base per le figure piane) sono costruite a partire da classi più semplici o mediante classi visive differenti.



Dalla figura si può osservare che il controllo `MenuItem` (la classe che costituisce i singoli elementi di un menu generico, come la voce Salva del Menu File) e il controllo `Button` (il classico pulsante) sono entrambi derivati da `Control` ma attraverso rami differenti della gerarchia. Nella gerarchia sono state omesse alcune classi intermedie (`ToggleButton`).

È interessante sapere che, se una classe dispone di caratteristiche (Proprietà, Metodi e Eventi) allora anche tutte le sue classi discendenti dispongono delle medesime caratteristiche! Questo significa che poiché la classe `Control` possiede la Proprietà `BackColor` (colore di sfondo) allora tutti i controlli che ne derivano (per es. `MenuItem`) possiedono egualmente questa Proprietà.

Tra i controlli possiamo annoverare il `Form`, la finestra di base, che dispone di Proprietà, Metodi ed Eventi (a cui associare dei rispettivi gestori) che sarà discussa più approfonditamente nelle prossime dispense. Possiamo anticipare che il `Form` dispone degli eventi `Mouse000` e `Key000` (con questo pseudonimo si intende l'intero gruppo degli eventi del Mouse e degli eventi Tastiera) e che si apprenderà come gestirli. Poiché però sono eventi comuni ai controlli, anche i `Button`, le `TextBox`, le `Label` e gli altri controlli elementari possono essere gestiti nello stesso modo.

**PROPRIETÀ DEI CONTROLLI**

Come si è già detto, non è intenzione elencare e discutere tutte le caratteristiche dei controlli. Questo elenco è pertanto parziale.

Proprietà	Descrizione
AutoSize	Tutti i controlli hanno AutoSize ma solo le derivate la usano.
BackColor	Indica il colore di sfondo del controllo.
BackgroundImage	Indica l'immagine di sfondo del controllo.
Bottom	È la distanza, in pixel, tra il bordo inferiore del controllo e del contenitore.
CanFocus	Indica se il controllo può ricevere il focus.
CanRaiseEvents	Indica se l'evento può essere sollevato sul controllo.
CanSelect	Indica se il controllo può essere selezionato.
ClientRectangle	È l'area cliente rettangolare del controllo.
ClientSize	È la grandezza (altezza e larghezza) dell'area cliente del controllo.
ContextMenu	Indica il menu contestuale associato al controllo.
ContextMenuStrip	Indica il menu ContextMenuStrip associato al controllo.
Controls	È la collezione di controlli contenuti nel controllo. In effetti un controllo è in grado di ospitare altri controlli; questo è l'elenco dei controlli ospitati.
Cursor	Indica l'aspetto del cursore del mouse quando è sul controllo.
DefaultBackColor	Indica il colore di sfondo di default del controllo.
DefaultCursor	Indica l'aspetto del cursore del mouse del controllo.
DefaultFont	Indica il carattere di default del controllo.
DefaultForeColor	Indica il colore del testo di default mostrato dal controllo.
Enabled	Indica se il controllo può reagire all'interazione utente.
Events	È l'elenco dei gestori di evento agganciati a questo componente. È una proprietà ereditata da Component.
Focused	Indica se il controllo ha l'input focus.
Font	Indica il carattere del testo mostrato dal controllo.
FontHeight	Indica l'altezza del carattere del testo mostrato dal controllo.
ForeColor	Indica il colore del carattere del testo mostrato dal controllo.
Handle	È il puntatore di riferimento del controllo. Per comprendere questa proprietà occorre conoscere il funzionamento delle API di Windows.
HasChildren	Indica se il controllo contiene uno o più figli. Per comprendere questa proprietà occorre capire cosa sono i controlli figli di altri controlli.
Height	Indica l'altezza del controllo (in pixel).
Left	È la distanza, in pixel, tra il bordo sinistro del controllo e del contenitore.
Location	Indica il punto (coordinate) dell'angolo superiore sinistro del controllo rispetto al suo ospite. È possibile anche usare Top e Left, al suo posto.
Margin	Indica la distanza tra i controlli.
ModifierKeys	Indica quali tasti chiave sono attualmente premuti (SHIFT, CTRL e ALT).
MouseButtons	Indica quali pulsanti del mouse sono attualmente premuti (Left, Right).
MousePosition	Indica le coordinate del mouse quando è premuto sul controllo.
Name	Indica il nome del controllo.
Padding	Indica il riempimento del controllo.
Parent	Indica il controllo contenitore del controllo.
Region	Indica la regione associata al controllo. Questa proprietà è utilizzata soprattutto se si vuole ridisegnare l'aspetto del controllo (per es. una finestra rotonda).
Right	È la distanza, in pixel, tra il bordo destro del controllo e del contenitore.



RightToLeft	Indica se gli elementi del controllo sono allineati a destra o sinistra.
ScaleChildren	Indica come sono esposti i controlli figli.
Size	Indica la grandezza del controllo.
TabIndex	Indica l'ordine di tabulazione dei controlli. In effetti i controlli consentono di essere selezionati premendo il tasto TAB. L'ordine è deciso da questi numeri.
TabStop	Indica se il controllo ammette la selezione mediante la tabulazione.
Tag	Indica un dato ininfluenza ma specifico del controllo.
Text	Indica il testo associato al controllo.
Top	È la distanza, in pixel, tra il bordo superiore del controllo e del contenitore.
UseWaitCursor	Indica l'aspetto del cursore di attesa per il controllo e i suoi figli.
Visible	Indica se il controllo è visibile oppure è nascosto.
Width	Indica la larghezza del controllo.

Tutti i controlli possiedono le proprietà sopra elencate; ogni controllo le può usare in maniera specifica, seppure sono intrinsecamente uguali. Per esempio Text è il testo del controllo; il Form però lo visualizza in alto, sulla barra della finestra, mentre il Button lo visualizza al centro.

METODI DEI CONTROLLI

Come si è già detto, non è intenzione elencare e discutere tutte le caratteristiche dei controlli. Questo elenco è pertanto parziale.

Metodo	Descrizione
BringToFront	Porta il controllo in primo piano.
Contains	Rende un valore che indica se è figlio di un altro controllo.
Dispose()	Distrugge il controllo e libera la memoria occupata. È ereditato dal Component.
DoDragDrop	Inizia una operazione DRAGDROP.
DrawToBitmap	Supporta il rendering di una bitmap.
FindForm	Rende il Form che lo ospita.
Focus	Imposta il focus sul controllo.
GetNextControl	Rende il controllo successive nell'ordine di tabulazione.
GetType	Restituisce il Tipo dell'istanza corrente di controllo. Sembra idiota, ma invece è molto utile. Per capire come funziona è indispensabile conoscere la OOP.
Hide	Nasconde l'oggetto (lo rende invisibile) all'utente.
Invalidate()	Invalida la superficie del controllo e obbliga a ridisegnarlo.
IsInputChar	Determina se un carattere è ammissibile come input.
IsInputKey	Determina se un tasto è ammissibile come input.
IsKeyLocked	Determina se i tasti CAPS LOCK, NUM LOCK, o SCROLL LOCK sono attivi.

I successivi metodi riproducono l'effetto di un evento che normalmente crea l'utente.

OnAutoSizeChanged	Solleva l'evento AutoSizeChanged .
OnBackColorChanged	Solleva l'evento BackColorChanged .
OnBackgroundImageChanged	Solleva l'evento BackgroundImageChanged .
OnBackgroundImageLayoutChanged	Solleva l'evento BackgroundImageLayoutChanged .
OnBindingContextChanged	Solleva l'evento BindingContextChanged .
OnCausesValidationChanged	Solleva l'evento CausesValidationChanged .
OnChangeUICues	Solleva l'evento ChangeUICues .
OnClick	Solleva l'evento Click .
OnClientSizeChanged	Solleva l'evento ClientSizeChanged .



OnContextMenuChanged	Solleva l'evento ContextMenuChanged .
OnContextMenuStripChanged	Solleva l'evento ContextMenuStripChanged .
OnControlAdded	Solleva l'evento ControlAdded .
OnControlRemoved	Solleva l'evento ControlRemoved .
OnCreateControl	Solleva l'evento CreateControl method.
OnCursorChanged	Solleva l'evento CursorChanged .
OnDockChanged	Solleva l'evento DockChanged .
OnDoubleClick	Solleva l'evento DoubleClick .
OnDragDrop	Solleva l'evento DragDrop .
OnDragEnter	Solleva l'evento DragEnter .
OnDragLeave	Solleva l'evento DragLeave .
OnDragOver	Solleva l'evento DragOver .
OnEnabledChanged	Solleva l'evento EnabledChanged .
OnEnter	Solleva l'evento Enter .
OnFontChanged	Solleva l'evento FontChanged .
OnForeColorChanged	Solleva l'evento ForeColorChanged .
OnGiveFeedback	Solleva l'evento GiveFeedback .
OnGotFocus	Solleva l'evento GotFocus .
OnHandleCreated	Solleva l'evento HandleCreated .
OnHandleDestroyed	Solleva l'evento HandleDestroyed .
OnHelpRequested	Solleva l'evento HelpRequested .
OnImeModeChanged	Solleva l'evento ImeModeChanged .
OnInvalidated	Solleva l'evento Invalidated .
OnKeyDown	Solleva l'evento KeyDown .
OnKeyPress	Solleva l'evento KeyPress .
OnKeyUp	Solleva l'evento KeyUp .
OnLayout	Solleva l'evento Layout .
OnLeave	Solleva l'evento Leave .
OnLocationChanged	Solleva l'evento LocationChanged .
OnLostFocus	Solleva l'evento LostFocus .
OnMarginChanged	Solleva l'evento MarginChanged .
OnMouseCaptureChanged	Solleva l'evento MouseCaptureChanged .
OnMouseClicked	Solleva l'evento MouseClicked .
OnMouseDoubleClick	Solleva l'evento MouseDoubleClick .
OnMouseDown	Solleva l'evento MouseDown .
OnMouseEnter	Solleva l'evento MouseEnter .
OnMouseHover	Solleva l'evento MouseHover .
OnMouseLeave	Solleva l'evento MouseLeave .
OnMouseMove	Solleva l'evento MouseMove .
OnMouseUp	Solleva l'evento MouseUp .
OnMouseWheel	Solleva l'evento MouseWheel .
OnMove	Solleva l'evento Move .
OnNotifyMessage	Informa il controllo di un messaggio di Windows.
OnPaddingChanged	Solleva l'evento PaddingChanged .
OnPaint	Solleva l'evento Paint .
OnPaintBackground	Ridisegna lo sfondo del controllo.



OnPreviewKeyDown	Solleva l'evento PreviewKeyDown .
OnPrint	Solleva l'evento Paint .
OnQueryContinueDrag	Solleva l'evento QueryContinueDrag .
OnRegionChanged	Solleva l'evento RegionChanged .
OnResize	Solleva l'evento Resize .
OnRightToLeftChanged	Solleva l'evento RightToLeftChanged .
OnSizeChanged	Solleva l'evento SizeChanged .
OnStyleChanged	Solleva l'evento StyleChanged .
OnSystemColorsChanged	Solleva l'evento SystemColorsChanged .
OnTabIndexChanged	Solleva l'evento TabIndexChanged .
OnTabStopChanged	Solleva l'evento TabStopChanged .
OnTextChanged	Solleva l'evento TextChanged .
OnValidated	Solleva l'evento Validated .
OnValidating	Solleva l'evento Validating .
OnVisibleChanged	Solleva l'evento VisibleChanged .
Select()	Attiva il controllo.
SelectNextControl	Attiva il controllo successivo.
SendToBack	Porta il controllo in secondo piano.
Show	Rivela il controllo (al contrario di Hide).
ToString	Rende una stringa che contiene il nome del componente se esiste.

Tutti i controlli possiedono i metodi sopra elencati; ogni controllo li può invocare per causare un effetto specifico. Per esempio Show mostra il controllo, anche se ciascuno appare in modo specifico, per esempio la finestra compare sul video, un Button compare sulla finestra.

EVENTI DEI CONTROLLI

Evento	Descrizione
Click	Si solleva quando si fa clic sul controllo.
ControlAdded	Si solleva se un nuovo controllo è aggiunto a Control.ControlCollection .
ControlRemoved	Si solleva se un nuovo controllo è rimosso da Control.ControlCollection .
Disposed	Si solleva quando il componente (controllo) è distrutto.
DoubleClick	Si solleva quando c'è un doppio clic sul controllo .
DragDrop	Si solleva quando si complete un drag-and-drop.
DragEnter	Si solleva quando un oggetto entra nell'area del controllo.
DragLeave	Si solleva quando un oggetto esce dall'area del controllo.
DragOver	Si solleva quando un oggetto è trascinato sopra il controllo.
EnabledChanged	Si solleva quando cambia la proprietà Enabled .
Enter	Si solleva quando si entra sul controllo.
GiveFeedback	Si solleva durante l'operazione drag-and-drop.
GotFocus	Si solleva quando il controllo ottiene il focus.
HelpRequested	Si solleva quando l'utente chiede aiuto.
KeyDown	Si solleva quando si digita su tastiera mentre il controllo ha il focus.
KeyPress	Si solleva quando si tiene premuto su tastiera mentre il controllo ha il focus.
KeyUp	Si solleva quando si rilascia il tasto da tastiera mentre il controllo ha il focus.
Leave	Si solleva quando il controllo perde il focus di input.
LocationChanged	Si solleva quando il controllo si sposta.
LostFocus	Si solleva quando il controllo perde il focus.
MouseDown	Si solleva quando il mouse fa clic sul controllo.



MouseDown	Si solleva quando il mouse fa doppio clic sul controllo.
MouseDown	Si solleva quando il mouse è sul controllo e si tiene premuto un pulsante.
MouseEnter	Si solleva quando il cursore del mouse va sul controllo.
MouseHover	Si solleva quando il cursore del mouse permane sul controllo.
MouseLeave	Si solleva quando il cursore del mouse esce dal controllo.
MouseMove	Si solleva quando il cursore del mouse si muove sopra il controllo.
MouseUp	Si solleva quando il mouse è sul controllo e rilascia un suo pulsante.
MouseWheel	Si solleva quando il mouse è sul controllo e si muove la rotella.
Move	Si solleva quando il controllo si muove.
TextChanged	Si solleva quando il testo del controllo cambia.
VisibleChanged	Si solleva quando cambia la visibilità del controllo.

NOTE SUI CONTROLLI

Nelle prossime dispense si procederà con lo studio delle caratteristiche dei controlli. È opportuno ribadire ancora una volta che quanto esaminato per un singolo controllo (per. es. il Form) è valido anche per gli altri controlli (es. un Button). Ovviamente se si gestisce un evento del mouse sul Button esso si attiverà solo quando il mouse sarà sul Button stesso.

GERARCHIA DEI CONTROLLI

Di seguito è proposta un albero che illustra la gerarchia dei controlli.

System.Object	---	classe base per tutti gli oggetti
+ System.MarshalByRefObject		
+ System.ComponentModel.Component	---	classe base dei controlli elementari
+ System.Windows.Forms.Control	---	classe base dei controlli visivi
+ System.Windows.Forms.ButtonBase	---	classe base per i pulsanti
+ System.Windows.Forms.Button	---	classe pulsante standard
+ System.Windows.Forms.CheckBox	---	classe pulsante di spunta
+ System.Windows.Forms.RadioButton	---	classe pulsante radio esclusivo
+ System.Windows.Forms.DataGrid	---	classe base per le griglie di dati
+ System.Windows.Forms.DataGridView	---	classe griglia di dati
+ System.Windows.Forms.Form	---	classe base finestra
+ System.Windows.Forms.GroupBox	---	classe contenitore di raggruppamento
+ System.Windows.Forms.ItemsControl	---	classe base per i pulsanti
+ System.Windows.Forms.MenuItem	---	classe pulsante elemento del menu
+ System.Windows.Forms.ToolBar	---	classe barra degli strumenti
+ System.Windows.Forms.Label	---	classe etichetta
+ System.Windows.Forms.ProgressBar	---	classe barra di progressione
+ System.Windows.Forms.ScrollBar	---	classe barre laterali
+ System.Windows.Forms.StatusBar	---	classe barra di stato
+ System.Windows.Forms.TextBoxBase	---	classe base per le caselle di testo
+ System.Windows.Forms.TextBox	---	classe casella di testo
+ System.Windows.Forms.TreeView	---	classe vista ad albero
+ System.Windows.Forms.WebBrowserBase	---	classe base per un web browser



ESERCIZI

ESERCIZI SUI CONTROLLI

CONSTRUISCI QUESTO FORM

Carattere

Tipo Spaziatura e posizione

Tipo di carattere: Calibri
Broadway
Browallia New
BrowalliaUPC
Brush Script MT
Calibri

Stile carattere: Grassetto
Normale
Corsivo
Grassetto
Grassetto Corsivo

Dimensione: 11
8
9
10
11
12

Colore carattere:

Stile sottolineatura:

Colore sottolineatura: Automatico

Effetti

Barrato Ombreggiatura Maiuscoletto
 Barrato doppio Contorno Tutto majuscole
 Apice In rilievo Nascosto
 Pedice Incassato

Anteprima

Calibri

Tipo di carattere TrueType. Lo stesso tipo di carattere verrà utilizzato per lo schermo e per la stampante.

Predefinito... OK Annulla

IN SOSPESO

IN SOSPESO

IN SOSPESO

IN SOSPESO

IN SOSPESO

IN SOSPESO

IN SOSPESO

IN SOSPESO

IN SOSPESO



SOMMARIO

- INTERFACCIA UTENTE.....2**
- CONCETTI DI BASE.....2**
- COMUNICAZIONE CON L'UTENTE.....2**
- Struttura di una interfaccia grafica3
- I CONTROLLI.....4**
- INTRODUZIONE AI CONTROLLI BASE4**
- I Controlli analizzati4
- La Libreria dei controlli4
- Cos'è un controllo4
- BUTTON (Pulsante).....5
- CHECKBOX (Casella di selezione)5
- COMBOBOX (Casella a discesa)5
- CONTEXTMENU (Menu a comparsa)5
- DATAGRIDView (Griglia)5
- GROUPBOX (Contenitore di raggruppamento).....5
- IMAGE (Immagine).....6
- LABEL (Etichetta di testo).....6
- LISTBOX (Elenco di scelte)6
- MENU (Menu principale).....6
- PANEL (Pannello contenitore)6
- PASSWORDBOX (Casella per parole segrete)6
- POPUP (Suggerimento a comparsa)6
- PROGRESSBAR (Barra di progressione)6
- PRINTDIALOG (Dialogo di stampa)6
- RADIOBUTTON (Pulsante esclusivo).....7
- RICHTEXTBOX (Editore di testo).....7
- TEXTBOX (Casella di testo)7
- TOOLBAR (Barra strumenti)7
- CARATTERISTICHE DEI CONTROLLI7**
- Una gerarchia.....7
- Proprietà dei controlli10
- Metodi dei controlli11
- Eventi dei controlli13
- Note sui controlli.....14
- Gerarchia dei controlli14
- ESERCIZI SUI CONTROLLI15**
- Costruisci questo Form15
- In sospeso15
- In sospeso15
- In sospeso15
- In sospeso15
- In sospeso15
- In sospeso15
- In sospeso15
- In sospeso15
- In sospeso15
- In sospeso15

