

Applicazioni grafiche

I grafici all'interno di un form sono realizzati attraverso gli oggetti del *namespace* **System.Drawing.Graphics**.

Il *namespace* è incluso automaticamente nel progetto, quindi non è necessaria la dichiarazione di importazione:

```
Imports System.Drawing.Graphics
```

I due oggetti principali sono:

- **Graphics**, che rappresenta l'area del grafico
- **Pen**, che contiene i parametri di disegno: colore e spessore del tratto.

```
Dim Piano As Graphics  
Dim pen As New Pen(Color.Black, 1)
```

La prima istruzione crea un oggetto grafico, la seconda imposta la penna con il colore nero e lo spessore 1.

All'inizio del programma, associata all'evento *Load* del form, si può inizializzare l'area del grafico con il metodo **CreateGraphics** che il form applica a se stesso (**Me**):

```
Piano = Me.CreateGraphics()
```

Il disegno di una retta è realizzato dal metodo **DrawLine**, che ha come parametri la penna e le coordinate dei punti estremi della retta.

```
Piano.DrawLine(pen, X1, Y1, X2, Y2)
```

Questo metodo non può essere utilizzato nella subroutine di gestione dell'evento *Load* del form. Si osservi che si possono liberare le risorse grafiche utilizzate, quando non servono più, attraverso il metodo **Dispose**:

```
pen.Dispose()
```

dopo aver tracciato la retta

```
Piano.Dispose()
```

quando il programma termina.

Progetto 1

Disegnare una retta nel piano cartesiano.

Il progetto consiste nel tracciare gli assi cartesiani all'interno di un form e successivamente dare la possibilità all'utente di tracciare il grafico di una retta di cui fornisce le coordinate del primo punto A e del secondo punto B.

Dati di input: Coordinate del punto A (XA, YA)
Coordinate del punto B (XB, YB)

Dati di output: retta che unisce i due punti.

Nome del progetto

Retta di tipo Applicazione Windows Form.

Disegno dell'interfaccia grafica

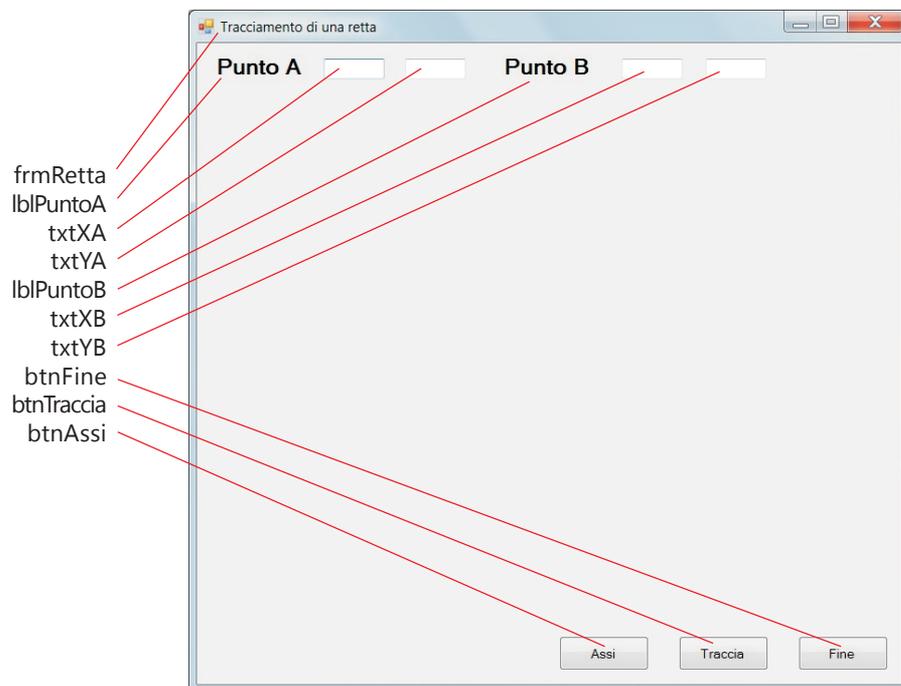
Il programma utilizza tre pulsanti per le operazioni principali:

- tracciamento degli assi (*Assi*)
- tracciamento della retta (*Traccia*)

oltre al pulsante per chiudere il programma (*Fine*).

L'utente inserisce poi le coordinate dei due punti nelle caselle di testo.

Classe	Proprietà dell'oggetto	
Form	Name	frmRetta
	Text	<i>Tracciamento di una retta</i>
Label	Name	lblPuntoA
	Text	<i>Punto A</i>
TextBox	Name	txtXA
TextBox	Name	txtYA
Label	Name	lblPuntoB
	Text	<i>Punto B</i>
TextBox	Name	txtXB
TextBox	Name	txtYB
Button	Name	btnAssi
	Text	<i>Assi</i>
Button	Name	btnTraccia
	Text	<i>Traccia</i>
Button	Name	btnFine
	Text	<i>Fine</i>



Gestione degli eventi

Occorre osservare che un form possiede un riferimento cartesiano intrinseco, avente l'origine nell'angolo in alto a sinistra, e che le sue dimensioni sono determinate dalle proprietà **Size.Width** e **Size.Height**.

Quindi, se si vogliono tracciare gli assi di un sistema di riferimento cartesiano avente l'origine al centro del form, occorre disegnare due linee:

- la prima per l'asse x dal punto (0, Height \ 2) al punto (Width, Height \ 2)
- la seconda per l'asse y dal punto (Width \ 2, Height) al punto (Width \ 2, 0).

L'origine si trova nel punto di coordinate $X0 = \text{Width} \setminus 2$, $Y0 = \text{Height} \setminus 2$.

Per tracciare correttamente la retta passante per i punti dei quali l'utente fornisce le coordinate, occorre poi operare una traslazione.

Se le coordinate dei punti fornite dall'utente sono A (XA, YA) e B(XB, YB), le coordinate trasformate diventano

$(X0 + XA, Y0 - YA)$ e $(X0 + XB, Y0 - YB)$.

Codice Visual Basic

```
Imports System.Drawing.Graphics

Public Class frmRetta
    Dim X0, Y0 As Integer
    Dim X1, Y1, X2, Y2 As Integer
    Dim XA, YA, XB, YB As Integer
    Dim Piano As Graphics

    Private Sub frmRetta_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        Piano = Me.CreateGraphics()
    End Sub

    Private Sub btnAssi_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnAssi.Click
        ' penna nera
        Dim pen As New Pen(Color.Black, 1)
        'asse x
        X1 = 0
        Y1 = Me.Size.Height \ 2
        X2 = Me.Size.Width
        Y2 = Me.Size.Height \ 2
        Piano.DrawLine(pen, X1, Y1, X2, Y2)
        'asse y
        X1 = Me.Size.Width \ 2
        Y1 = Me.Size.Height
        X2 = Me.Size.Width \ 2
        Y2 = 0
        ' disegna la retta
        Piano.DrawLine(pen, X1, Y1, X2, Y2)
        pen.Dispose()
        ' origine
        X0 = Me.Size.Width \ 2
        Y0 = Me.Size.Height \ 2
    End Sub
```

```

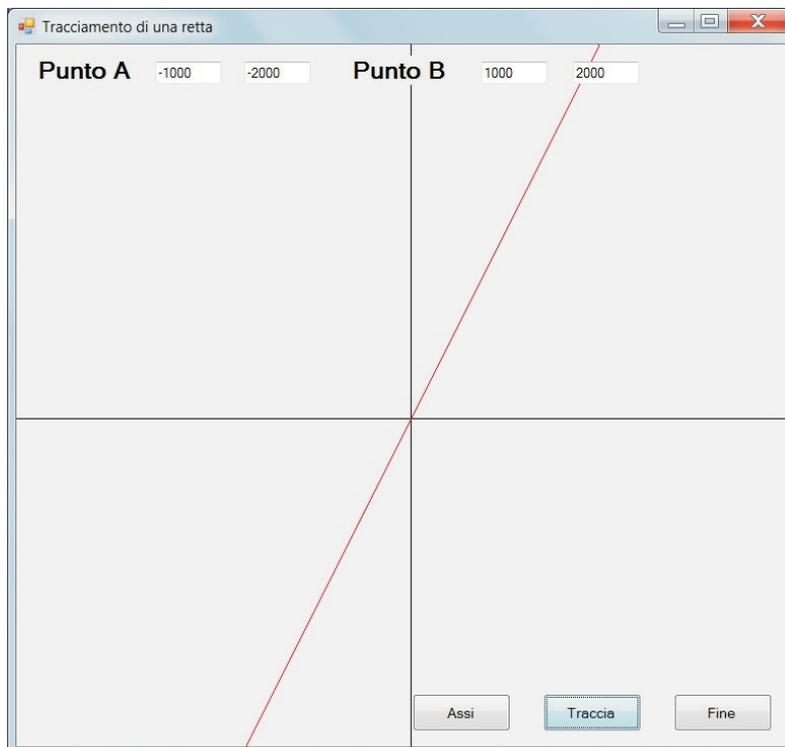
Private Sub btnTraccia_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnTraccia.Click
    ' penna rossa
    Dim pen As New Pen(Color.Red, 1)
    ' coordinate dei punti della retta
    XA = Val(txtXA.Text)
    YA = Val(txtYA.Text)
    XB = Val(txtXB.Text)
    YB = Val(txtYB.Text)
    ' traslazione dei punti
    X1 = X0 + XA
    Y1 = Y0 - YA
    X2 = X0 + XB
    Y2 = Y0 - YB
    ' disegna la retta
    Piano.DrawLine(pen, X1, Y1, X2, Y2)
    pen.Dispose()
End Sub

Private Sub btnFine_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnFine.Click
    Piano.Dispose()
End
End Sub

End Class

```

Prova di esecuzione



Progetto 2

Disegnare il grafico delle funzioni: $y=x$, $y=x^2$, $y=x^4$, $y=e^x$

Le funzioni sono rappresentate sull'intervallo delle ascisse (-1,4) in modo da poter osservare l'andamento delle funzioni nei loro aspetti grafici più significativi.

L'utente, attraverso quattro caselle di controllo, ha la possibilità di selezionare quali grafici rappresentare.

Nome del progetto

Grafici di tipo Applicazione Windows Form.

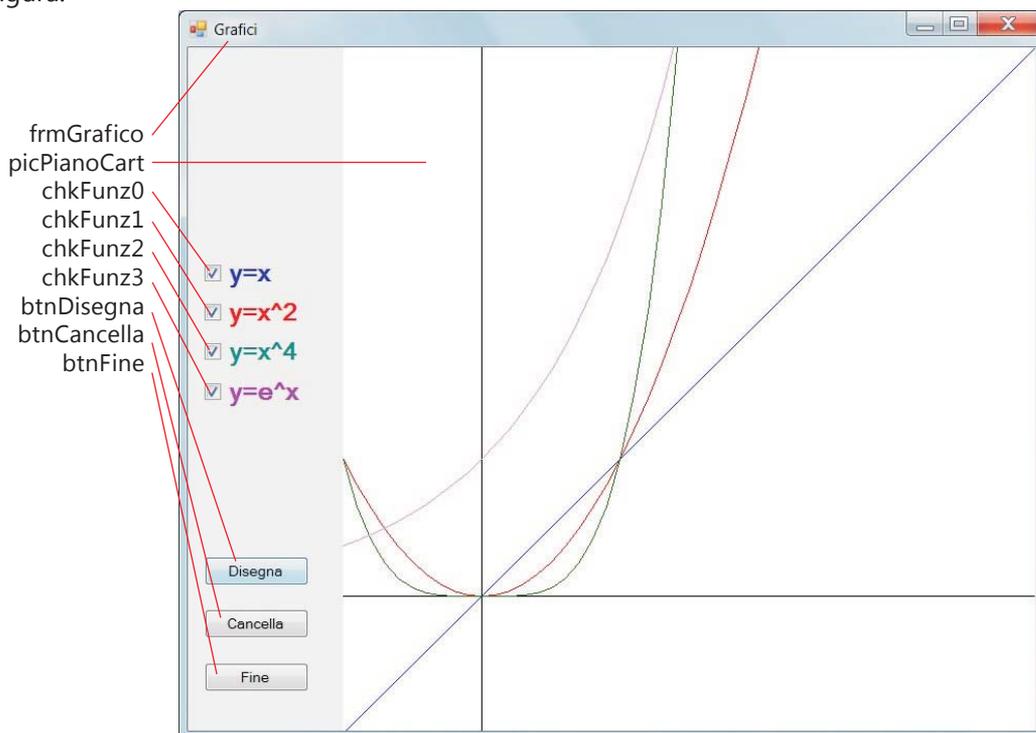
Disegno dell'interfaccia grafica

Il controllo fondamentale di questa applicazione è un *PictureBox* che occupa la maggior parte del form e nel quale viene tracciato il grafico delle funzioni. Il grafico delle funzioni può essere disegnato o tralasciato a seconda delle impostazioni delle quattro *CheckBox*. Ogni voce del *CheckBox* corrisponde ad una funzione: in presenza del segno di spunta viene disegnata la funzione corrispondente.

Sono, inoltre, definiti i tre bottoni di comando *Disegna*, *Cancella* e *Fine*: il primo serve per tracciare il grafico, il secondo per pulire il *PictureBox* e il terzo per chiudere l'applicazione.

Classe	Proprietà dell'oggetto	
Form	Name	frmGrafico
	Text	<i>Grafici</i>
PictureBox	Name	picPianoCart
CheckBox	Name	chkFunz0
	Text	$y=x$
	ForeColor	<i>Blue</i>
CheckBox	Name	chkFunz1
	Text	$y=x^2$
	ForeColor	<i>Rosso</i>
CheckBox	Name	chkFunz2
	Text	$y=x^4$
	ForeColor	<i>Verde</i>
CheckBox	Name	chkFunz3
	Text	$y=e^x$
	ForeColor	<i>Viola</i>
Button	Name	btnDisegna
	Text	<i>Disegna</i>
Button	Name	btnCancella
	Text	<i>Cancella</i>
Button	Name	btnFine
	Text	<i>Fine</i>

Se si sceglie di disegnare tutte le funzioni, l'aspetto finale del *PictureBox* è come quello della figura.



Gestione degli eventi

All'inizio del programma, nella subroutine *frmGrafico_Load*, si calcola l'unità di misura del sistema cartesiano sul video, dividendo la larghezza del controllo *PictureBox* (che è uguale all'altezza) per il numero di ascisse fissato dai valori assegnati alle costanti *minX*, *maxX*.

Gli assi e l'origine del sistema cartesiano del grafico sono determinati come visto prima nel progetto del tracciamento della retta, ricordando che il sistema cartesiano associato al controllo *PictureBox*, attraverso l'oggetto di tipo *Graphics*, ha l'origine nell'angolo in alto a sinistra.

Per determinare quindi le coordinate nel piano reale del video, occorre operare le opportune trasformazioni per l'ascissa e l'ordinata dei punti.

La funzione *F* calcola il valore dell'ordinata e utilizza due parametri: l'ascissa del punto e il numero della funzione.

Per calcolare la funzione $y = e^x$ si usa la funzione **Exp** della classe **Math**.

La procedura *Segmento* disegna il tratto che unisce un punto al successivo secondo l'incremento di 0.1 per l'ascissa.

Per rimuovere il grafico dal video si usa il metodo **Clear** applicato all'oggetto di tipo *Graphics*: il metodo usa un parametro per il colore che viene impostato al colore di sfondo del controllo *PictureBox*.

L'array *GrafSel* di tipo booleano ricorda, per ciascuna funzione, se l'utente ha fatto la selezione con il segno di spunta nella casella di controllo.

Le istruzioni *Piano.DrawLine* contengono le funzioni **CInt**, per la conversione in numeri interi, perché le coordinate dei punti sono di tipo *Double* e i parametri del metodo **DrawLine** devono essere di tipo intero.

Le seguenti costanti, utilizzate nel programma, definiscono la porzione di piano cartesiano su cui tracciare i grafici:

```
minX = -1
maxX = 4
minY = -1
maxY = 4.
```

Il programma utilizza gli oggetti grafici già visti nel progetto precedente per il tracciamento della retta.

Per la rappresentazione di una funzione, la struttura più conveniente è il ciclo *For* con la variabile *x* che va dall'estremo sinistro all'estremo destro dell'intervallo su cui si vuole rappresentare il grafico, con un incremento piccolo della *x*, per esempio 0.1.

All'interno del ciclo *For* in corrispondenza di ciascun valore della *x* si calcola $y=f(x)$ e si disegna il segmento che unisce il punto al successivo, corrispondente all'ascissa incrementata.

Codice Visual Basic

```
Public Class frmGrafico
    ' Estremi del piano cartesiano
    Const minX As Single = -1
    Const maxX As Single = 4
    Const minY As Single = -1
    Const maxY As Single = 4
    ' Larghezza (= altezza) dell'area del grafico
    Dim l As Short
    ' Unità di misura in pixel
    Dim u As Single
    ' Coordinate di un punto
    Dim X0, Y0 As Integer
    Dim X1, Y1, X2, Y2 As Double
    ' Oggetto grafico
    Dim Piano As Graphics
    ' Grafici selezionati
    Dim GrafSel(3) As Boolean

    Private Sub frmGrafico_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        Piano = picPianoCart.CreateGraphics()
        ' Dimensioni del PictureBox
        l = picPianoCart.Size.Width
        ' Unità di misura
        u = l / (maxX - minX)
        ' Origine
        X0 = u
        Y0 = maxY * u
    End Sub

    Public Sub DisegnaAssi()
        Dim pen As New Pen(Color.Black, 1)
        'Asse x
```

```

X1 = X0 + minX * u
Y1 = Y0
X2 = X0 + maxX * u
Y2 = Y0
Piano.DrawLine(pen, CInt(X1), CInt(Y1), CInt(X2), CInt(Y2))
'Asse y
X1 = X0
Y1 = Y0 - minY * u
X2 = X0
Y2 = Y0 - maxY * u
Piano.DrawLine(pen, CInt(X1), CInt(Y1), CInt(X2), CInt(Y2))
pen.Dispose()
End Sub

Public Sub ControllaSelezione()
    If chkFunz0.Checked Then
        GrafSel(0) = True
    Else
        GrafSel(0) = False
    End If
    If chkFunz1.Checked Then
        GrafSel(1) = True
    Else
        GrafSel(1) = False
    End If
    If chkFunz2.Checked Then
        GrafSel(2) = True
    Else
        GrafSel(2) = False
    End If
    If chkFunz3.Checked Then
        GrafSel(3) = True
    Else
        GrafSel(3) = False
    End If
End Sub

Public Sub Segmento(ByVal nf As Integer, ByVal a As Double, ByVal b
As Double, ByVal c As Double, ByVal d As Double)
    Dim col As Color
    Select Case nf
        Case 0
            col = Color.Blue
        Case 1
            col = Color.Red
        Case 2
            col = Color.Green
        Case 3
            col = Color.Violet
    End Select
    Dim pen As New Pen(col, 1)
    Piano.DrawLine(pen, CInt(a), CInt(b), CInt(c), CInt(d))
    pen.Dispose()
End Sub

```

```

Public Function F(ByVal i As Single, ByVal nf As Integer) As Double
    Select Case nf
        Case 0
            F = i
        Case 1
            F = i ^ 2
        Case 2
            F = i ^ 4
        Case 3
            F = Math.Exp(i)
    End Select
End Function

Private Sub btnDisegna_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnDisegna.Click
    Piano.Clear(picPianoCart.BackColor)
    ControllaSelezione()
    DisegnaAssi()
    Dim x As Single
    Dim j As Integer
    For x = minX To maxX Step 0.1
        For j = 0 To 3
            If GrafSel(j) Then
                X1 = X0 + x * u
                Y1 = Y0 - F(x, j) * u
                X2 = X0 + (x + 0.1) * u
                Y2 = Y0 - F(x + 0.1, j) * u
                Segmento(j, X1, Y1, X2, Y2)
            End If
        Next j
    Next x
End Sub

Private Sub btnCancella_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnCancella.Click
    Piano.Clear(picPianoCart.BackColor)
End Sub

Private Sub btnFine_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnFine.Click
    Piano.Dispose()
End Sub

End Class

```