

Laboratorul 2

Elemente de grafică în Visual Basic

Ce ne propunem astăzi?



În acest laborator ne propunem să realizăm o aplicație simplă de grafică similară Microsoft Paint, urmărind familiarizarea cu noțiunea de grafică persistentă precum și cu metodele grafice principale.

Mai pe larg, vom proceda astfel:

Pentru realizarea ferestrei principale a aplicației se utilizează un meniu (MenuStrip), o bară de unelte (ToolStrip) și suprafața pentru desenare (PictureBox) în care utilizatorul va putea desena sau încărca o imagine de tip bitmap (Figura 1).

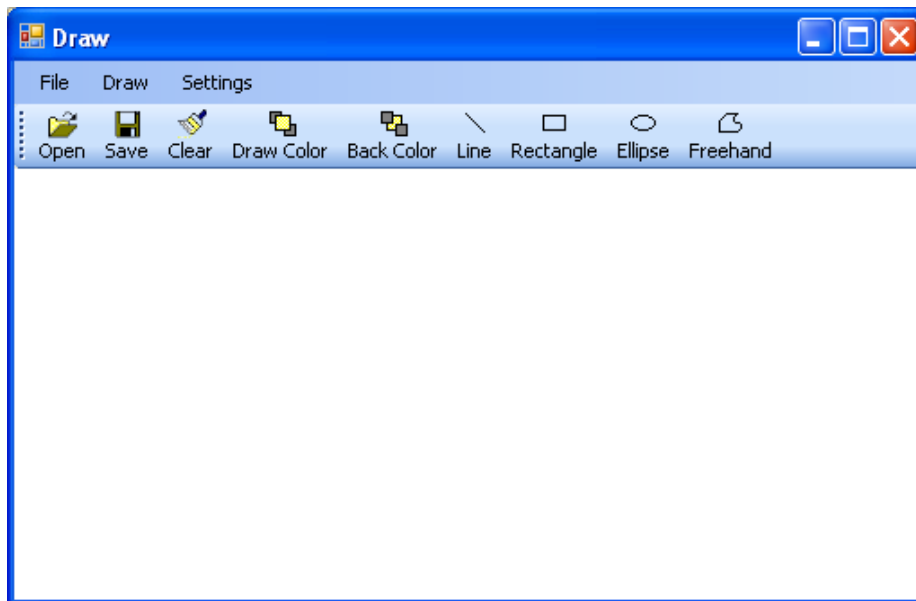


Figura 1. Fereastra principală a aplicației

Pentru ca controlul PictureBox să ocupe întreaga suprafață client a ferestrei aplicației, selectați controlul, dați click pe caseta din colțul dreapta-sus a chenarului de selecție și selectați comanda „Dock in parent container” (vezi Figura 2).

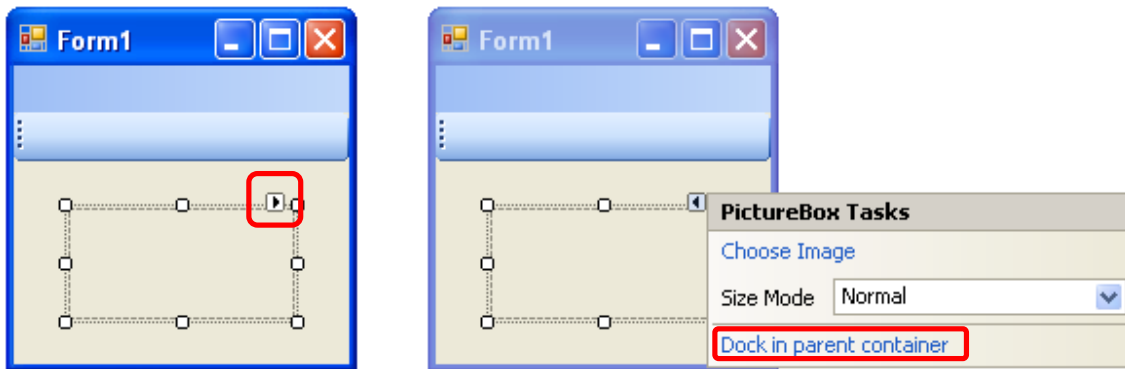


Figura 2. Setarea proprietății „Dock in parent container” pentru controlul PictureBox

Pentru a adăuga o comandă de meniu validă este necesară introducerea textului care va fi afișat în interfața grafică a ferestrei (proprietatea *Text*). Vor fi adăugate comenzile de meniu vizibile în Figura 3.

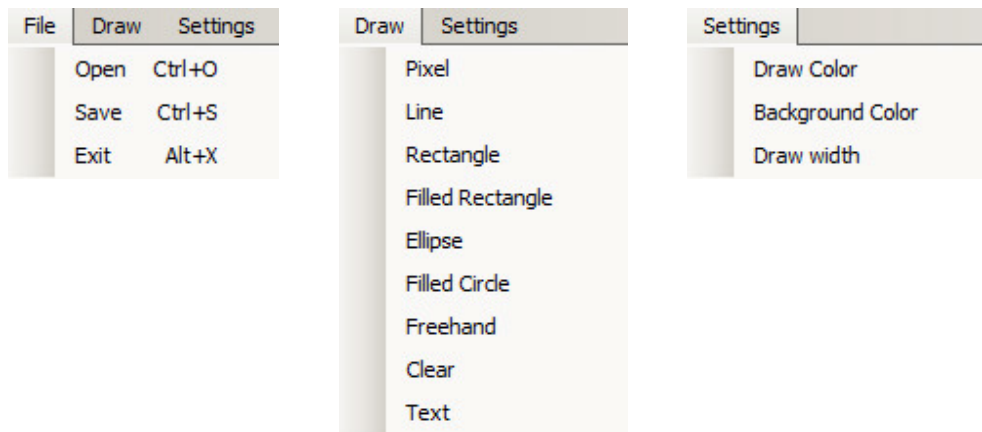


Figura 3. Meniul aplicației

Se va completa bara de unelte cu butoanele care se văd în Figura 1. Toate obiectele de pe controlul ToolStrip vor fi de tipul Button (Figura 4).

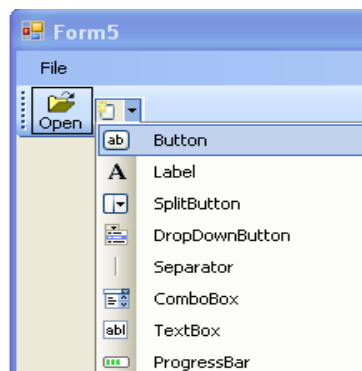


Figura 4. Adăugarea de butoane pe ToolStrip

Iată câteva proprietăți utile ale obiectelor de tip ToolStripButton:

- *DisplayStyle* – permite setarea aspectului unui control de tip ToolStripButton (Text, Image sau ImageAndText)
- *TextImageRelation* – permite stabilirea poziției textului în raport cu imaginea (Overlay, ImageAboveText, ImageBeforeText, TextAboveImage, TextBeforeImage)
- *ToolTipText* – textul care apare atunci când cursorul de mouse este poziționat deasupra butonului.

În continuare se setează proprietatea *DisplayStyle* la valoarea *ImageAndText*, proprietatea *TextImageRelation* la valoarea *ImageAboveText*.

Desenarea interfeței grafice a aplicației fiind încheiate, vom continua cu indicații privitoare la codul care trebuie scris pentru ca aplicația să funcționeze corect. Originea sistemului de coordonate asociat controlului PictureBox se află în colțul stânga-sus al acestuia. Aplicația poate fi dezvoltată utilizând metodele grafice ale unui obiect de tip Graphics. Un obiect de tip Graphics poate fi asociat cu diverse suprafețe de desenare, de exemplu, suprafața controlului PictureBox. În exemplul următor se scrie cod pentru trasarea unei linii din colțul stânga sus până în colțul dreapta jos al controlului de tip PictureBox:

```
Dim myGraphics As Graphics = PictureBox1.CreateGraphics()
myGraphics.DrawLine(Pens.Black, 0, 0, PictureBox1.Width, PictureBox1.Height)
```

Grafica reprezentată prin codul de mai sus este considerată a fi grafică nepersistentă (elementele grafice nu sunt redesenate, de exemplu în cazul obturării parțiale sau totale a controlului de tip PictureBox). Pentru a evita astfel de situații vom implementa facilități de grafică persistentă, lucrând cu un obiect de tip Bitmap.

```
Dim myBitmap As Bitmap
```

În mod uzual un obiect de tipul PictureBox este folosit pentru a afișa conținutul unor fișiere grafice.

Obiectele de tip Bitmap sunt de regulă utilizate pentru stocarea în memorie, pixel cu pixel, a unor imagini grafice.

Clasa Graphics oferă numeroase metode grafice pentru desenarea de obiecte grafice (linii, elipse, dreptunghiuri, curbe Bezier etc.).

Pentru ca grafica realizată cu ajutorul aplicației noastre să fie persistentă, exemplul cu linia trasată pe diagonala principală a controlului PictureBox se poate rescrie astfel:

```
'evenimentul Load al formularului...
'crearea unui bitmap avand aceleasi dimensiuni cu controlul PictureBox
myBitmap = New Bitmap(PictureBox1.Width, PictureBox1.Height)
myGraphics = Graphics.FromImage(myBitmap)

...

'desenarea unei linii in obiectul de tip Bitmap (asociat obiectului
myGraphics)
myGraphics.DrawLine(Pens.Black, 0, 0, PictureBox1.Width, PictureBox1.Height)
'asocierea imaginii de fundal a controlului PictureBox cu obiectul Bitmap
PictureBox1.Image = myBitmap
```

Modificarea culorii de fundal

Pentru alegerea unei culori se va folosi controlul `ColorDialog`. Acest control permite afișarea unei ferestre de dialog standard pentru alegerea unei culori. Pentru afișarea dialogului trebuie apelată metoda `ShowDialog()`.



Figura 5. Fereastra standard pentru alegerea culorii

Fereastra de dialog va fi afișată modal, iar culoarea aleasă va fi returnată de către proprietatea `Color`.

Pentru modificarea culorii de fundal a suprafeței de desenare va fi utilizată metoda `Clear` a obiectului `myGraphics`. Rezultatul apelării acestei metode va fi ștergerea tuturor elementelor grafice și umplerea fundalului cu o culoare uniformă. Aceeași metodă va fi utilizată și la curățarea suprafeței de desenare.

```
ColorDialog1.ShowDialog()
myGraphics.Clear(ColorDialog1.Color)
PictureBox1.Image = myBitmap
```

Modificarea culorii de desenare și a grosimii desenului

Pentru modificarea culorii de desenare va fi afișată fereastra controlului `ColorDialog`. Pentru specificarea culorii de desenare precum și a grosimii desenului este necesară utilizarea unui obiect de tipul `Pen`.

```
Dim myPen As Pen
'crearea unui creion de culoare albastra si grosime 2 pixeli.
myPen = New Pen(Color.Blue, 2)
```

Pentru modificarea grosimii desenului va fi trebui afișată o fereastră ca cea din Figura 6. La crearea acesteia, fereastra va fi aleasă ca fiind de tipul `Dialog`.

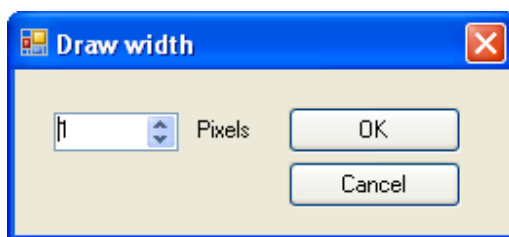


Figura 6. Alegerea grosimii de desenare

Fereastra va conține un control de tipul `NumericUpDown`. Acesta permite introducerea sau modificarea unei valori numerice întregi prin introducerea de la tastatură a unui număr sau alegerea lui prin apăsarea săgeților sus/jos asociate. Valoarea introdusă este dată de proprietatea *Value* a acestuia.

Pentru apelarea din cod a ferestrei dialog se va utiliza următoarea secvență de cod:

```
Dim f As New Dialog1
If f.ShowDialog() = Windows.Forms.DialogResult.OK Then
    ...
    'apelarea membrilor obiectului f
End If
```

Metoda *ShowDialog* va returna o constantă care corespunde butonului apăsat de utilizator (în cazul nostru, *OK* sau *Cancel*).

Desenarea de puncte

Utilizatorul va putea să deseneze puncte atunci când va apăsa butonul stânga al mouse-ului deasupra suprafeței de desenare. În consecință, se va identifica evenimentul corespunzător acestei operații. Acesta este *MouseDown*.

Spre deosebire de Visual Basic 6, în Visual Studio 2008, nu exista o metoda dedicata pentru de desenarea de puncte. Pentru a desena un pixel se va trasa o elipsă cu lățimea și înălțimea de un pixel.

```
myGraphics.DrawEllipse(myPen, x, y, 1, 1)
PictureBox1.Image = myBitmap
```

Desenarea de linii

Utilizatorul va putea desena segmente de dreaptă definind prima pereche de coordonate (x,y) la apăsarea butonului stânga a mouse-ului, iar a doua pereche la eliberarea butonului. Între apăsarea și eliberarea butonului de mouse, utilizatorul va deplasa cursorul de mouse între cele două puncte care vor defini segmentul de dreaptă dorit. Metoda grafică ce permite desenare de linii este *DrawLine*.

Sfaturi utile



Cele mai importante evenimente de mouse asociate controlului PictureBox care permit desenarea prin definirea corectă a coordonatelor unei figuri sunt:

- MouseDown: definirea primei perechi de coordonate $(x1, y1)$
- MouseUp: definirea celei de a doua perechi de coordonate $(x2, y2)$. Tot aici are loc desenarea figurii între perechile de coordonate $(x1, y1)$ și $(x2, y2)$.

Desenarea de dreptunghiuri

Desenarea de dreptunghiuri se va efectua precizând coordonatele coltului stânga sus al dreptunghiului, a lățimii și înălțimii lui. Metoda grafică ce permite desenarea de dreptunghiuri este *DrawRectangle*.

Desenarea de elipse

Desenarea unei elipse se va putea efectua prin precizarea colțului stânga-sus (apăsarea butonului stânga al mouse-ului) și a lățimii și înălțimii dreptunghiului circumscris elipsei. Metoda grafică ce permite desenarea de elipse este *DrawEllipse*.

Unde greșesc studenții cel mai des?



Cei mai mulți studenți greșesc la desenarea dreptunghiurilor și elipselor în sensul că, pentru trasarea corectă a acestora, trebuie ținut cont de următoarele:

- precizarea coordonatelor punctului stânga-sus al dreptunghiului, respectiv al dreptunghiului circumscris elipsei, se face folosind funcția *Min* a clasei *Math* (se aplică, de exemplu, atunci când primul colț definit cu ajutorul mouse-ului este colțul dreapta-jos, iar apoi colțul stânga-sus).
- precizarea lățimii și înălțimii dreptunghiului se face folosind funcția *Abs* a clasei *Math* (în caz contrar putând fi obținute segmente de dreaptă de lungime negativă).

Desenarea liberă

Desenarea liberă va consta din desenarea cu ajutorul cursorului de mouse într-un mod similar celui în care se desenează pe o hârtie cu ajutorul creionului. Desenarea liberă va începe odată cu apăsarea butonului stânga al mouse-ului și se va sfârși odată cu eliberarea acestuia. Între aceste două evenimente utilizatorul poate mișca cursorul de mouse, traiectoria descrisă de acesta fiind afișată pe suprafața de desenare ca o dâră continuă. În acest caz apare necesitatea folosirii evenimentului *MouseMove* (acesta apare atunci când deplasăm cursorul de mouse deasupra unui control).

Sfaturi utile



Pentru memorarea coordonatelor unui punct de pe PictureBox, utilizați un obiect de tipul Point.

Salvarea desenului ca imagine

Pentru salvarea conținutului suprafeței de desenare se va utiliza un obiect de tipul SaveFileDialog.

```
Dim save As New SaveFileDialog
```

Câteva proprietăți importante ale obiectului de tip SaveFileDialog sunt:

- *ShowDialog* – lansează fereastra standard pentru salvarea de fișiere.

```
If save.ShowDialog() = Windows.Forms.DialogResult.OK Then
    '...
End If
```

- *InitialDirectory* – permite specificarea directorului implicit

```
save.InitialDirectory = "c:\"
```

- *Filter* – permite precizarea extensiilor posibile ale fișierelor care vor fi salvate

```
save.Filter = "JPG files (*.jpg)|*.jpg|Bitmaps (*.bmp)|*.bmp|Gif (*.gif)|*.gif"
```

- *FilterIndex* – folosită pentru a specifica indexul extensiei implicite din toate extensiile posibile.

```
save.FilterIndex = 1
```

Salvarea propriu-zisă se face folosind metoda *Save* a obiectului Bitmap:

```
myBitmap.Save(save.FileName)
```

Încărcarea unei imagini

Pentru încărcarea unei imagini în PictureBox se va utiliza un obiect de tipul OpenFileDialog.

```
Dim open As New OpenFileDialog
```

Cele mai importante proprietăți ale unui Obiect de tipul OpenFileDialog sunt: *ShowDialog*, *InitialDirectory*, *Filter*, *FilterIndex*.

Pentru a încărca imaginea deschisă în controlul PictureBox și pentru a o putea modifica ulterior, sunt necesare următoarele:

```
myBitmap = System.Drawing.Image.FromFile(open.FileName)
myGraphics = Graphics.FromImage(myBitmap)
PictureBox1.Image = myBitmap
```

Inserarea de text în imagine

În continuare se dorește ca la un click de mouse pe controlul PictureBox să se deschidă o fereastră care să permită introducerea de text, precum cea de mai jos:

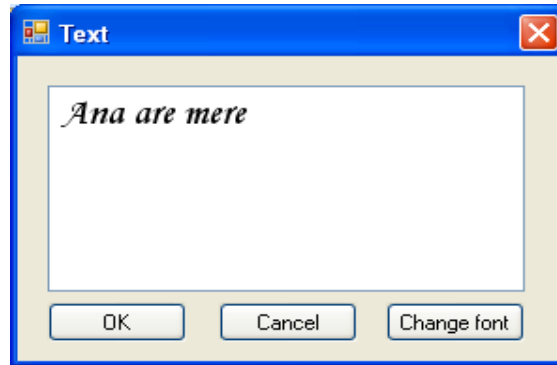


Figura 7. Modificarea fontului

Când se va apăsa comanda „Ok”, textul introdus va apărea pe PictureBox începând cu poziția în care s-a făcut click. Comanda „Change Font” va deschide fereastra standard de editare a fontului. Se va declara un obiect de tipul FontDialog:

```
Dim fnt As New FontDialog
```

Deschiderea ferestrei de specificare a fontului se face în felul următor:

```
If fnt.ShowDialog() = Windows.Forms.DialogResult.OK Then
    '...
End If
```

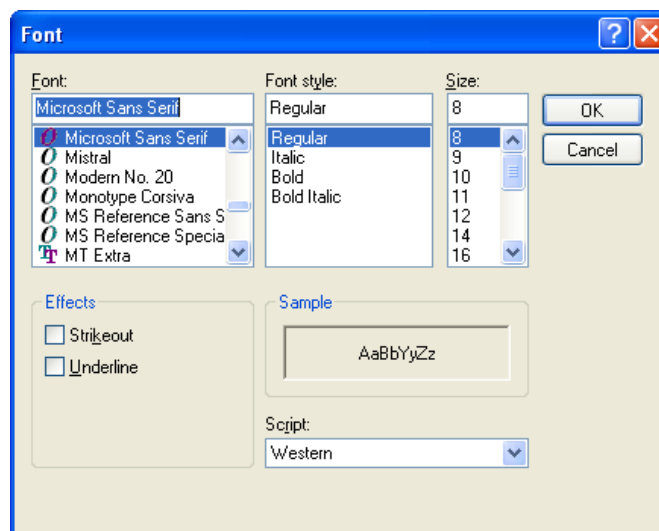


Figura 8. Fereastra standard pentru alegerea fontului

Setarea fontului selectat se face în felul următor:

```
TextBox1.Font = fnt.Font
```

Cu ce ne-am ales?



Prin aplicația dezvoltată în cadrul laboratorului de astăzi am învățat să utilizăm în scopuri grafice controlul PictureBox și evenimentele MouseDown, MouseUp și MouseMove asociate acestuia, ne-am familiarizat cu noțiunea de grafică persistentă utilizând obiectele de tip Graphics și Bitmap, și am învățat să lucrăm cu metodele grafice de bază.

Bibliografie



[1] <http://msdn.microsoft.com/en-us/vbasic/default.aspx>