

گرافیک

- هدف‌های رفتاری: پس از آموزش این فصل، هنرجو می‌تواند:
- از دستورات و توابع گرافیکی در برنامه‌های خود استفاده کند؛
- شکل‌های مختلفی را با نوشتن کدهای خاصی، ترسیم نماید.

استفاده از هنر گرافیک در پروژه‌ها سبب می‌شود که فرم‌ها گویاتر و جذاب‌تر شوند. علاوه بر این، کاربران نیز می‌توانند با استفاده از گرافیک‌ها عملیات خاصی را به سادگی انجام دهند.

۱-۷- اضافه کردن گرافیک به فرم

با استفاده از کنترل‌های Picture Box و Image می‌توان تصویری را به فرم اضافه کرد. با وجود این که در فصل اول با این دو کنترل و تفاوت‌های آن‌ها آشنا شدید ولی در این جا این تفاوت‌ها را به گونه دیگری مورد بررسی قرار می‌دهیم:

الف - Picture Box ، برای گروه‌بندی کنترل‌های دیگر می‌تواند استفاده شود ولی Image این قابلیت را ندارد.

ب - Picture Box ، در صورت بزرگ بودن تصویر، قسمتی از آن را که به اندازه کادرش است نشان می‌دهد و مشخصه‌ای به نام AutoSize دارد که در صورت True بودن، می‌تواند کادر Picture Box را به اندازه واقعی تصویر بزرگ یا کوچک کند ولی کنترل Image در صورت بزرگ بودن تصویر، به‌طور پیش‌فرض کادرش را به اندازه تصویر بزرگ می‌کند و دارای مشخصه‌ای به نام Stretch است که در صورت True بودن، قبل از قرار دادن تصویر، می‌تواند تصویر را به اندازه کادر خود تبدیل کند. درج تصویر در هر دو کنترل، مشابه بوده و با یکی از روش‌های صفحه بعد انجام می‌شود:

● در زمان طراحی فرم : با استفاده از مشخصه Picture می‌توان تصویر دلخواه را در این کنترل‌ها درج کرد.

● در زمان کدنویسی : با استفاده از تابع LoadPicture() می‌توان تصویر را در این کنترل‌ها قرار داد که شکل کلی آن به صورت زیر است :

```
ControlName.Picture = LoadPicture(StrFilePath)
```

● ControlName نام کنترل تصویری است.

● LoadPicture نام تابع است.

● StrFilePath رشته‌ای است که مسیر و نام پرونده گرافیکی موردنظر روی حافظه جانبی را تعیین می‌کند.



اگر به جای StrFilePath رشته خالی قرار دهید، تصویر جاری حذف

می‌شود.

در زمان کدنویسی می‌توان تصویر یک کنترل را به کنترل دیگری نسبت داد. به عنوان مثال، کد زیر تصویر Image2 را به Image1 نیز نسبت می‌دهد :

```
Image1.Picture = Image2.Picture
```

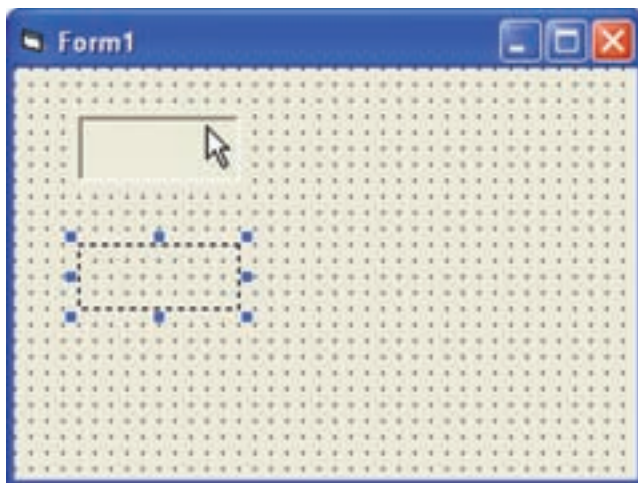


روش‌های درج تصویر فوق، فقط مختص این دو کنترل نیستند و می‌توان در تمام کنترل‌هایی که مشخصه Picture دارند (مانند Form ، Command Button و ...) از این روش‌ها برای درج تصویر استفاده کرد.

مثال ۱-۷

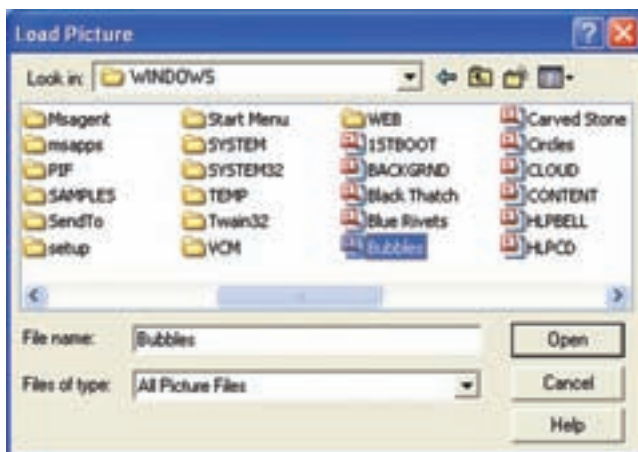
۱- پروژه Standard EXE را باز کرده و آن را smp1Grfx نام‌گذاری کنید. نام پیش فرض فرم را به frmMain تغییر دهید و مشخصه Caption آن را با Simple Graphics مقداردهی کنید.

- ۲- یک کنترل Picture Box روی فرم اضافه کنید. نام این کنترل را picMain قرار دهید.
- ۳- یک کنترل Image به فرم اضافه کنید و نام آن را imgMain قرار دهید.
- ۴- کنترل‌های فوق را مطابق شکل ۷-۱ تغییر اندازه داده و جابه‌جا کنید.



شکل ۷-۱- مقدار پیش‌فرض مشخصه **BorderStyle** مربوط به کنترل **PictureBox** برابر **1-Single Fixed** و کنترل **Image** برابر با **None** است.

- ۵- کادر تصویر picMain را انتخاب کنید. از پنجره Properties در مقابل مشخصه Picture روی علامت سه نقطه کلیک کنید تا کادر محاوره‌ای Load Picture باز شود (شکل ۷-۲).

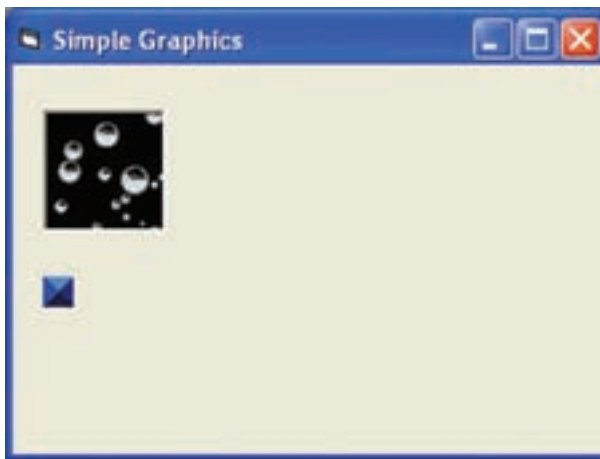


شکل ۷-۲- پوشه Windows موجود در سیستم، انواع مختلفی از فایل‌های گرافیکی را ارائه می‌کند.

نگاه

Bubbles.bmp یک تصویر نقش بیتی است که همراه ویندوز ارائه می‌شود. اگر به هر دلیلی این پرونده در مسیر مورد نظر نبود، می‌توانید هر پرونده نقش بیتی یا نشانه، metafile، JPEG و GIF را انتخاب کنید. همه این قالب‌ها را می‌توان در PictureBox قرار داد.

- ۶- از پوشه Windows فایل Bubbles.bmp را انتخاب کنید. هنگامی که روی دکمه Open کلیک کنید، این تصویر نقش بیتی در کادر تصویر ظاهر خواهد شد.
- ۷- مقدار مشخصه AutoSize کادر تصویر picMain را با True مقداردهی کنید. انجام این کار، سبب می‌شود که کادر تصویر به اندازه تصویر، بزرگ شود.
- ۸- کنترل Image را انتخاب کرده و مقدار مشخصه picture آن را پرونده تصویری Triangles.bmp قرار دهید (این فایل نیز در پوشه Windows قرار دارد). کنترل Image مشخصه AutoSize ندارد.
- ۹- پروژه را ذخیره کرده و اجرا کنید (شکل ۳-۷).



شکل ۳-۷- کنترل Image از مشخصه AutoSize پشتیبانی نمی‌کند و به‌طور خودکار ابعاد این کنترل مطابق با تصویر، تغییر اندازه پیدا می‌کند.

پژوهش: بررسی کنید که چه کنترل‌هایی برای گروه‌بندی کنترل‌ها استفاده می‌شود.

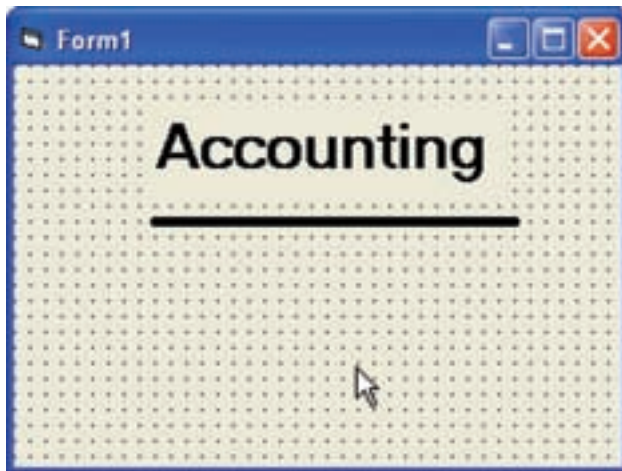


۷-۲- کنترل‌های ترسیمی

جعبه ابزار Visual Basic دارای دو کنترل ترسیمی است :

- کنترل خط، که بین دو نقطه یک خط راست رسم می‌کند.
- کنترل شکل، که روی فرم اشکال هندسی رسم می‌کند.

۷-۲-۱- ترسیم خط : خط راست یکی از اشکالی است که کاربردهای فراوانی می‌تواند داشته باشد، حتی اگر برنامه دارای جنبه‌های گرافیکی هم نباشد. در شکل ۷-۴ یک فرم را که در آن از خط استفاده شده ملاحظه می‌کنید؛ این فرم می‌تواند فرم معرفی یک برنامه باشد.

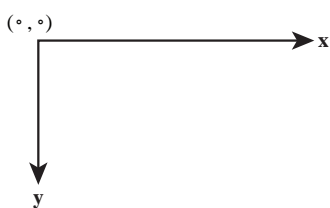


شکل ۷-۴- با خط می‌توان روی قسمت‌های مهم فرم تأکید کرد.

اگر روی کنترل خط (Line) در جعبه ابزار دوبار کلیک کنید، Visual Basic یک خط راست در وسط فرم قرار خواهد داد. این خط دارای دو دستگیره تغییر اندازه در دو انتهای خود است که با آن‌ها می‌توانید اندازه و جهت خط را تغییر دهید. مهم‌ترین خواص خط عبارتند از :

- Border Color : رنگ خط
- Border Style : شکل ظاهری خط (به جدول ۷-۱ و شکل ۷-۵ نگاه کنید).
- Border Width : پهنای خط بر حسب نقطه ($\frac{1}{72}$ اینچ)
- X1, Y1, X2, Y2 : مختصات دوسر خط (یک خط با دو نقطه انتهایی آن تعریف

می‌شود.)



محورهای مختصات به صورت روبه‌رو است. مبدأ مختصات گوشه بالا و سمت چپ فرم می‌باشد.

نکته

اگر `BorderWidth` بزرگ‌تر از ۱ شود. `BorderStyle` فقط حالت خط پیوسته خواهد داشت.

جدول ۱-۷. مقادیر خاصیت `Border Style` کنترل خط

مفهوم	ثابت نام‌دار
رنگ زمینه فرم از ورای خط دیده خواهد شد.	0 - Transparent
خط پیوسته	1 - Solid
خط چین	2 - Dash
نقطه چین	3 - Dot
خط - نقطه - خط	4 - Dash - Dosh - Dash
خط - نقطه - نقطه - خط	5 - Dash - Dot - Dot - Dash



شکل ۵-۷. شکل ظاهری خط را خاصیت `Border Style` تعیین می‌کند.

۲-۲-۷- ترسیم شکل : با کنترل شکل (Shape) می‌توانید اشکال مختلف هندسی روی فرم خود رسم کنید. شکلی که رسم می‌شود توسط خاصیت Shape کنترل معین می‌شود :

● 0-Rectangle : مستطیل

● 1-Square : مربع

● 2-Oval : بیضی

● 3-Circle : دایره

● 4-Rounded Rectangle : مستطیل با گوشه‌های گرد

● 5-Rounded Square : مربع با گوشه‌های گرد

کنترل Shape، علاوه بر خاصیت Shape، چند خاصیت مهم دیگر هم دارد که آن‌ها را در جدول ۲-۷ ملاحظه می‌کنید.

جدول ۲-۷- خواص کنترل Shape

مفهوم	خاصیت
نوع شکل ترسیمی را مشخص می‌کند.	Shape
(0/1) شفافیت شکل رسم شده را معین می‌کند.	Back Style
رنگ خط دور شکل	Border Color
ظاهر خط دور شکل	Border Style
ضخامت حاشیه شکل (برحسب twips)	Border Width
رنگ طرح داخلی شکل (به خاصیت Fill Style نگاه کنید.)	Fill Color
طرح داخل شکل (شکل ۶-۷ هشت طرح ممکن را نشان می‌دهد.)	Fill Style
ارتفاع شکل	Height
عرض شکل	Width

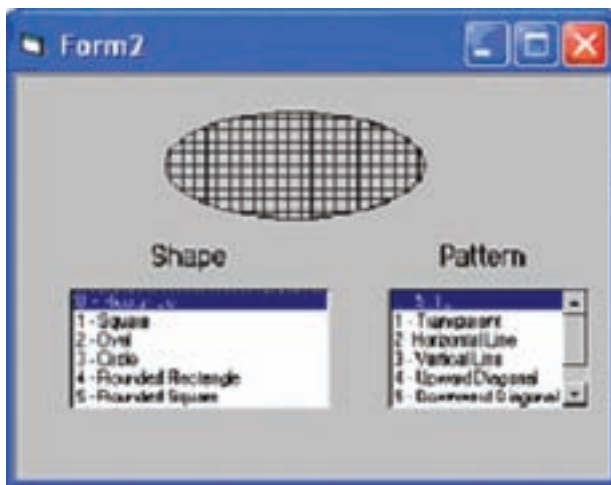


شکل ۶-۷. هشت طرح موجود برای خاصیت Fill Style کنترل Shape

مثال ۲-۷

فرمی طراحی کنید که با استفاده از دو کادر لیست، نوع شکل و شیوه هاشور داخل شکل را انتخاب کنید.

شکل ۷-۷ اجرای این برنامه را نشان می دهد. با انتخاب گزینه ای از دو کادر لیست شکل و طرح می توانید ظاهر شکل موجود روی فرم را عوض کنید.



شکل ۷-۷. با انتخاب شکل و طرح نتیجه کار را ببینید.

مراحل زیر را دنبال کنید :

۱- یک پروژه جدید باز کنید.

۲- کنترل‌های موجود در شکل ۷-۷ را روی فرم قرار دهید.

۳- کد زیر را به برنامه اضافه کنید :

```
Private Sub Frm_Load()
```

اضافه کردن عناصر لیست کنترل IstShape ، که شامل انواع شکل هاست.

```
IstShape.AddItem "0 - Rectangle"
```

```
IstShape.AddItem "1 - Square"
```

```
IstShape.AddItem "2 - Ovel"
```

```
IstShape.AddItem "3 - Circle"
```

```
IstShape.AddItem "4 - Rounded Rectangle"
```

```
IstShape.AddItem "5 - Rounded Square"
```

اضافه کردن عناصر لیست کنترل Istpattern ، که شامل انواع Style های پرکردن درون

شکل است.

```
IstPattern.AddItem "0 - Solid"
```

```
IstPattern.AddItem "1 - Transparent"
```

```
IstPattern.AddItem "2 - Horizontal Line"
```

```
IstPattern.AddItem "3 - Vertical Line"
```

```
IstPattern.AddItem "4 - Upward Diagonal"
```

```
IstPattern.AddItem "5 - Downward Diagonal"
```

```
IstPattern.AddItem "6 - Cross"
```

```
IstPattern.AddItem "7 - Diagonal Cross"
```

```
IstShape.ListIndex = 0
```

انتخاب اولین عنصر لیست Istshape ،

```
IstPattern.ListIndex = 0
```

انتخاب اولین عنصر لیست Istpattern ،

```
End Sub
```

```
Private Sub Istpattern_Click()
```

shpSample.FillStyle = lstPattern.ListIndex پرکردن شکل با Style انتخاب شده
End Sub

Private Sub lstshap_Click()
shpSample.Shape = lstShape.ListIndex تغییر شکل براساس شکل انتخاب شده
End Sub

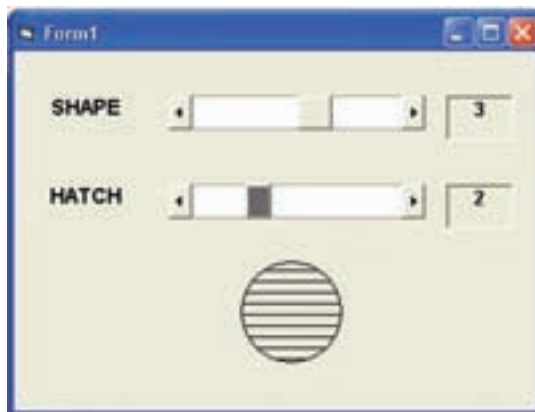
Private Sub mnuFileExit_Click()
End
End Sub

۴- برنامه را اجرا کنید. گزینه‌هایی را در دو جعبه لیست Shape و Pattern انتخاب کرده و نتیجه را مشاهده کنید.

پژوهش: کنترل‌های Shape و Line چه رویدادهایی دارند؟ 

مثال ۳-۷

فرمی به شکل زیر طراحی کنید که دارای دو نوار لغزان برای تغییر شکل و هاشور آن باشد.



شکل ۸-۷

از Hscroll1 برای تغییر شکل استفاده می‌شود لذا مقدار آن از ۰ تا ۵ تغییر می‌کند.
 از Hscroll2 برای تعیین شیوه پرکردن داخل شکل استفاده می‌شود و مقدار آن
 از ۰ تا ۷ تغییر می‌کند.

```
Private Sub Form_Load()
```

```
    Hscroll1.Min = 0
```

```
    Hscroll1.Max = 5
```

```
    Hscroll2.Min = 0
```

```
    Hscroll2.Max = 7
```

```
    Shapel.Shape = 0
```

شکل مستطیل با رنگ سیاه پر شده است

```
    Shapel.FillStyle = 0
```

```
End Sub ()
```

```
Private Sub Hscroll1_Change()
```

```
    Shapel.FillStyle = Hscroll1.Value
```

مقدار Value جدید شیوه پرکردن داخل شکل را تعیین می‌کند.

```
End Sub
```

```
Private Sub Hscroll2_Change()
```

```
    Shapel.Shape = Hscroll2.Value
```

مقدار Value جدید شکل جدید را تعیین می‌کند.

```
End Sub
```

۳-۷- متدهای ترسیمی

گرافیک به خط و اشکال هندسی محدود نیست. با Visual Basic قادرید نقطه به نقطه صفحه نمایش را در اختیار خود گرفته و کنترل کنید.

۱-۳-۷- متد Pset: متدی که با آن می‌توانید پیکسل‌ها را دستکاری کنید Pset نام دارد.

با متدهای ترسیمی، دیگر نیازی به کنترل‌های گرافیکی ندارید. این متدهای ترسیمی را روی کنترل جعبه تصویر هم می‌توانید اعمال کنید.

شکل کلی متد Pset چنین است:

```
frmName.Pset [Step] (intX, intY), [color]
```

● (intX, intY) مختصات نقطه‌ای است که می‌خواهیم روشن شود.

مختصات این نقطه را می‌توان به دو روش تعیین کرد.

۱- به صورت مطلق بدون Step

۲- به صورت نسبی با استفاده از Step که مختصات نقطه را با توجه به فاصله آن از موقعیت

فعلی مکان نما تعیین می‌کند.

مثال: frmDraw.Pset (100, 200)

frmDraw.Psetstep (20, 20)

دستور اول برای روشن کردن پیکسل (۱۰۰, ۲۰۰) است و دستور دوم روشن کردن پیکسل

(۱۲۰, ۲۲۰) که (۱۰۰+۲۰, ۲۰۰+۲۰) می‌باشد.

● color، کد رنگی است که با آن نقطه مورد نظر روشن می‌شود.

نقطه (۰, ۰) در متد Pset گوشه چپ بالای فرم است (البته این وضعیت را با خواص ScaleX

و ScaleY می‌توانید تغییر دهید.) دستور زیر پیکسل (۱۰۰, ۲۰۰) را روشن می‌کند:

frmDraw.Pset (100, 200) "Turn on a Pixel"

رنگ این پیکسل همان رنگ ForeColor فرم خواهد بود. با تعیین یک عدد هگزادسیمال یا یکی از ثابت‌های

Visual Basic (از قبیل vbRed, vbBlue, vbGreen, vbYellow, vbMagenta, vbCyan یا

vbWhite) برای آرگومان Color متد Pset می‌توانید رنگ نقطه را هم تنظیم کنید. دستور زیر نقطه (۱۰۰, ۲۰۰)

را به رنگ زمینه فرم درمی‌آورد:

frmDraw.Pset (100,200), frmDraw.BackColor "Turn off a Pixel"

همان‌طور که گفته شد، آرگومان Step باعث تغییر در تعیین مکان پیکسل‌ها خواهد شد. به کمک

Step می‌توانید مکان یک پیکسل را نسبت به نقطه قبلی تنظیم کنید. به دستور زیر توجه کنید:

frmDraw.Pset Step (300,350) "Relative Pixel Location"

با این دستور پیکسل (۴۰۰, ۵۵۰) روشن خواهد شد (۳۵۰+۲۰۰, ۳۰۰+۱۰۰). توجه کنید

که (۱۰۰, ۲۰۰) مختصات نقطه قبلی است.

با متد Pset خط هم می‌توان رسم کرد؛ به مثال زیر توجه کنید:

For intX = 1 To 100

frmDraw.Pset (intX, 250)

Next intX

رسم خطی افقی که شامل نقاطی با $y=250$ است و x آن‌ها از ۱ تا 1000 تغییر می‌کند.
۲-۳-۷-متد Line: برای رسم خط، متد مستقلی وجود دارد که آن را در زیر ملاحظه

می‌کنید:

`frmName.Line [Step] (intX1, intY1)-[Step] (intX2, intY2), [Color], [B | BF]`

● در این دستور دو زوج، نقطه مختصات ابتدا و انتهای خط هستند. با آرگومان Step می‌توانید مکان نقطه‌ها را نسبت به نقطه‌های قبلی محاسبه کنید. دستور زیر یک خط از پیکسل (۱۰۰، ۱۰۰) به پیکسل (۱۵۰، ۱۵۰) رسم می‌کند.

`FrmDraw.Line (100,100) - (150, 150)`

● آرگومان Color رنگ خط را مشخص می‌کند (اگر میل ندارید خط با رنگ ForeColor

فرم رسم شود).

● برای رسم مستطیل می‌توانید از گزینه B در متد Line استفاده کنید.

`frmDraw.Line (100,100) - (150, 150), B`

در این دستور، زوج نقاط مختصات گوشه چپ - بالا و راست - پایین مستطیل خواهند بود: کامای اضافه را فراموش نکنید! زیرا محل آرگومان رنگ در دستور line را تعیین می‌کند و اگر فراموش شود B به عنوان رنگ در نظر گرفته می‌شود.

● برای رنگ کردن داخل مستطیل رسم شده می‌توانید از گزینه BF استفاده کنید:

`frmDraw.Line (35,40) - (150, 175), vbGreen, BF 'A Green Box`

دستور فوق یک مستطیل سبزرنگ رسم خواهد کرد. اگر یکی از گوشه‌های مستطیل خارج از محدوده فرم قرار گیرد، Visual Basic آن را فقط تا لبه‌های فرم رسم خواهد کرد، و حتی اگر بعد از رسم مستطیل فرم را بزرگ کنید باز هم قسمت پنهان مستطیل را نخواهید دید مگر آن که مشخصه Auto Redraw آن را با True مقدار دهی کنید.



در کد صفحه بعد، روالی را مشاهده می‌کنید که چند مستطیل از سمت چپ - بالای فرم به سمت

گوشه راست - پایین آن رسم می‌کند؛ به شکل ۷-۹ نگاه کنید.

1: Private Sub cmdBoxes_Click()

2: Dim intStartX As Integer

3: Dim intStartY As Integer

4: Dim intLastX As Integer

5: Dim intLastY As Integer

6: Dim intCtr as Integer

7: intStartX = 0

8: intStartY = 0

9: intLastX = 1000

10: intLastY = 800

11: رسم ۲۰ مستطیل

12: For intCtr = 1 To 20

13: frmBoxes.Line (intStart X, intStartY) – (intLastX, intLastY), B

رسم مستطیل

14:

15: 'Prepare for next set of boxes

تعیین مختصات مستطیل بعدی

16: intStartX = intStartX + 400

17: intStartY = intStartY + 400

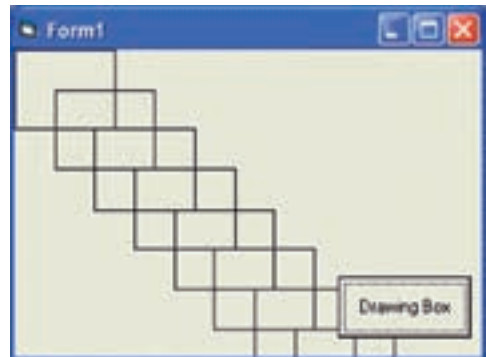
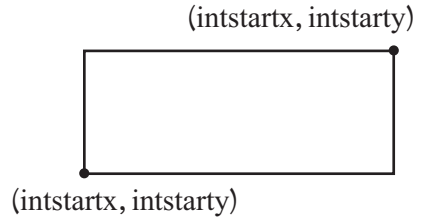
18: intLastX = intLastX + 400

19: intLastY = intLastY + 400

20: Next intCtr

21:

22: End Sub



شکل ۹-۷- رسم مستطیل با متد Line

۳-۳-۷-متد Circle : متد Circle متدی است که می‌توانید به کمک آن دایره و بیضی

رسم کنید :

`frmDraw.Circle [Step] [intX, intY), sngRadius, [Color], , , sngAspect`

(کاماهای اضافی مربوط به آرگومان‌های پیشرفته‌تر هستند که در این کتاب آن‌ها را مورد بحث

قرار نمی‌دهیم؛ فقط اشاره می‌کنیم که حتماً باید آن‌ها را قبل از آرگومان `sngAspect` قید کنید.)

● `intX` و `intY` مختصات مرکز دایره هستند.

● `sngRadius` شعاع دایره (برحسب پیکسل) است.

● آرگومان `Step` باعث تعیین مکان نسبی مرکز دایره خواهد شد.

دستور زیر دایره‌ای به مرکز (۲۰۰، ۳۰۰) شعاع ۱۰۰ پیکسل رسم خواهد کرد :

`frmDraw.Circle (300,200), 100`

اگر رنگی در آرگومان `Color` قید نشود، این متد از رنگ `ForeColor` فرم استفاده خواهد

کرد. برای رسم بیضی می‌توانید

● از آرگومان `sngAspect` (که نسبت قطر عمودی به قطر افقی آن است) استفاده کنید. اگر این

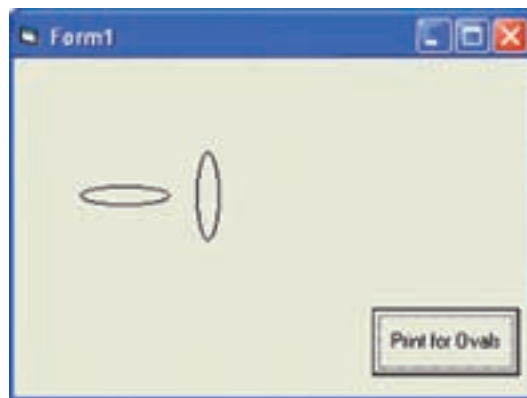
آرگومان ۱ باشد (حالت پیش‌فرض) یک دایره رسم خواهد شد، اما اگر عددی کوچک‌تر یا بزرگ‌تر از

۱ باشد دایره (به ترتیب) به صورت بیضی افقی یا عمودی تغییر خواهد کرد. در شکل ۱-۷ بیضی‌هایی

را که دو دستور زیر رسم می‌کنند، مشاهده می‌کنید :

`frmDraw.Circle (1000, 1250), 400, , , , 0.2`

`frmDraw.Circle (1750, 1250), 400, , , , 4`



شکل ۱-۷- شکل بیضی را آرگومان `SngAspect` تعیین می‌کند.



۴-۲- رنگ آمیزی

همان‌طور که در کار با اغلب کنترل‌ها مشاهده کردید، می‌توان رنگ رویه و زمینه آن‌ها را تغییر داد. برای تعیین رنگ، روش‌های متفاوتی وجود دارد که از جمله عبارت است از:

- استفاده از تابع **QBColor**: به کمک این تابع می‌توان رنگ‌های مورد استفاده در QBASIC را به کار برد. شکل کلی این تابع به صورت $QBColor(n)$ است. مقدار متغیر n می‌تواند از صفر تا ۱۵ باشد.



دایره‌ای به شعاع 100 و با رنگ قرمز ترسیم کنید.

`Circle(100,200), 100, QBColor(4)`

- استفاده از ثابت‌های نام‌دار و یژوال بیسیک: این ثابت‌ها معمولاً با پیشوند `vb` و سپس نام رنگ ظاهر می‌شوند. مانند `vbGreen`, `vbRed` و ...



نقطه‌ای در مختصات $(100, 200)$ را با رنگ قرمز ترسیم کنید:

`Pset (100,200), vbRed`

- استفاده از تابع **RGB**: این تابع با ترکیب سه رنگ قرمز، سبز و آبی، رنگ‌های جدیدی را تولید می‌کند. شکل کلی تابع به صورت زیر است:

`RGB(ColorRed, ColorGreen, ColorBlue)`

مقدار هر سه پارامتر تابع فوق، از صفر تا ۲۵۵ می‌تواند باشد.

مثال ۷-۷

خطی به طول 100° و به صورت عمودی با رنگ قرمز ترسیم کنید :

Line (100,200) – (200,200), RGB (255,0,0)

این تابع به دلیل این که می تواند $256 \times 256 \times 256$ رنگ تولید کند، نسبت به روش های قبلی کامل تر است.

● استفاده از کدهای رنگی که در پنجره Properties در مقابل مشخصه های رنگی وجود دارد.

مثال ۷-۸

نقطه ای با رنگ قرمز در مختصات (100° و 200°) ترسیم کنید :

Pset (100,200), & HFF &

این روش نیز از نظر تعداد رنگ هایی که تولید می کند، شبیه تابع RGB است.

مثال ۷-۹

فرمی به شکل زیر طراحی کنید که با درگ کردن روی کادر تصویر بتوان هر چیزی نوشت (شبییه تخته سیاه).



شکل ۷-۱۱

۱- منویی با ساختار زیر ایجاد کنید :

File

```
  _ New _  
  Exit
```

Caption	Name
& File	mnuFile
& New	itmNew
	sepone
̄&Exit	itmExit

۲- یک کادر تصویر به نام PicBoard روی فرم قرار دهید.

۳- کد زیر را تایپ کرده و پروژه را اجرا کنید :

```
Private Sub Form_Load()
```

```
    PicBoard.ForeColor = QBColor(15)    تعیین رنگ قلم
```

```
    PicBoard.BackColor = QBColor(0)     تعیین رنگ زمینه
```

```
    PicBoard.DrawWidth = 3              تعیین ضخامت قلم
```

```
End Sub
```

```
Private Sub PicBoard_MouseMove (button As Integer, Shift As Integer, x  
As Single, y As Single)
```

```
    If Button = 1 Then
```

```
        PicBoard.Pset (x,y)             رسم نقطه در محل حرکت ماوس
```

```
    End If
```

```
End Sub
```

```
Private Sub itmNew_Click()
```

```
    PicBoard.Cls                         پاک کردن تخته
```

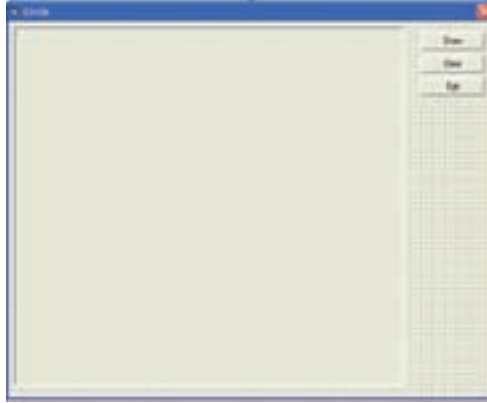
```
End Sub
```

```
Private Sub itmExit_Click()
```

```
    End
```

```
End Sub
```

ترسیم دایره‌های تصادفی



شکل ۱۲-۷

```
Private Sub CmdClear_Click()
```

```
    Pic1.Cls      پاک کردن محتوای کنترل Pic1
```

```
End Sub
```

```
Private Sub CmdDraw_Click()
```

```
    Dim Intx As Integer, Inty As Integer, IntR As Integer, Intc As Integer
```

```
    به منظور تولید اعداد تصادفی متفاوت در هر بار اجرای برنامه
```

```
    For i = 1 To 50
```

```
        Intx = Int(Rnd* Pic1.ScaleWidth)
```

```
        تولید عدد تصادفی بین ۰ و پهنای کنترل Pic1 برای x
```

```
        Inty = Int(Rnd* Pic1.ScaleHeight)
```

```
        تولید عدد تصادفی بین ۰ و ارتفاع کنترل Pic1 برای y
```

```
        IntR = Int(Rnd* 40)
```

```
        تولید عدد تصادفی بین ۰ و ۴۰ برای شعاع دایره
```

```
        Intc = Int(Rnd* 16)
```

```
        تولید عدد تصادفی بین ۰ و ۱۶ برای رنگ
```

```
        Pic1.Circle (Intx, Inty), IntR, QBColor (Intc)
```

```
        رسم دایره
```

```
Next
```

```
End Sub
```

```
Private Sub CmdExit_Click()
```

```
Unload Me
```

```
End Sub
```

اگر بخواهیم در رویداد Form-load از متدهای ترسیمی استفاده کنیم باید مشخصه AutoRedraw فرم را True کنیم این مشخصه سبب می شود که پس از قرارگرفتن فرم در حافظه تمام ترسیمات دوباره انجام شود. در حالت پیش فرض این مشخصه مساوی False است و پس از load شدن فرم چیزی روی فرم رسم نشده و تمام ترسیمات باید دوباره رسم شود.

```
Private Sub Form_Load()
```

```
Pic1. AutoRedraw = True
```

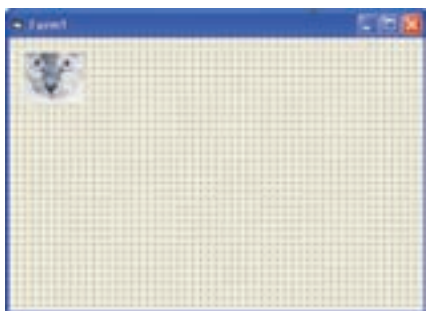
```
Pic1. BackColor = QBColor(15)
```

```
Pic1. ScaleMode = vbPixel
```

```
Pic1. TabStop = False
```

```
End Sub
```

از کلید Tab می توان برای انتقال فوکوس به کنترل های روی فرم استفاده کرد. اگر بخواهیم از انتقال فوکوس به کنترل با کلید Tab جلوگیری کنیم خصوصیت Tab Stop آن کنترل را مساوی False قرار می دهیم (پیش فرض این خصوصیت True است).



شکل ۷-۱۳

مثال ۷-۱۱

چیدن کاشی وار یک تصویر روی فرم

۱- پروژه جدیدی به صورت زیر ایجاد کنید(شکل

۷-۱۳) و تصویر موردنظر را برای کنترل Picture تعیین

کنید.

۲- کد مربوطه را وارد کنید :

```
Option Explicit
```

```
Private Sub Form_Paint()
```

Dim wid As Single

Dim hgt As Single

Dim x As Single

Dim y As Single

wid = Picture1.ScaleWidth

به دست آوردن پهنای Picture1

hgt = Picture1.ScaleHeig

به دست آوردن ارتفاع Picture1

y = 0

Do While y < Me.ScaleHeight تا زمانی که y از ارتفاع فرم کوچکتر است

x = 0

رسم یک سطر از Picture ها

Do While x < Me.ScaleWidth تا زمانی که x از پهنای فرم کوچکتر است

رسم Picture جدید در مکان تعیین شده PaintPicture Picture1.Picture,x,y, wid, hgt

x = x + wid

تعیین x بعدی

Loop

تمام شدن رسم یک سطر

y = y + hgt

تعیین y بعدی

Loop

End Sub

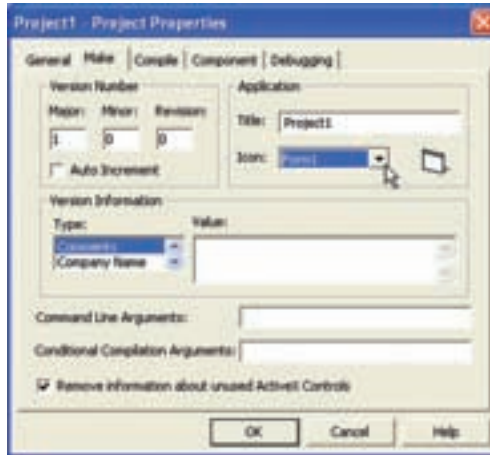
۳- پروژه را اجرا کنید (شکل ۷-۱۴).



شکل ۷-۱۴

۷-۵- ایجاد نشانه فرم

برنامه‌های ویندوز معمولاً دارای آیکن الصاق شده‌ای در داخل پرونده اجرایی هستند تا برنامه را به‌طور گرافیکی، نمایان کنند. هنگامی که برنامه‌ای را ایجاد می‌کنید، VB نشانه فرم را به آن اضافه می‌کند. مگر این که بخواهید این نشانه را تغییر دهید (شکل ۷-۱۵).



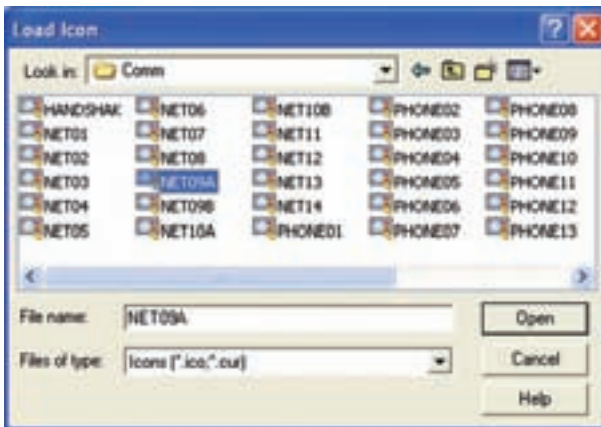
شکل ۷-۱۵- مشخصه‌های منحصر به فرد برنامه را در کادر
محواره‌ای Project Properties تعیین کنید.

با تعیین نشانه‌ای برای مشخصه Icon یک فرم، می‌توان نشانه سفارشی را به پرونده اجرایی برنامه اختصاص داد. ویروال بیسیک شامل چندین نشانه است که در پوشهٔ VB\Graphic\Icons قرار دارند. مراحل زیر، چگونگی تغییر

نشانه فرم را بیان می‌کنند:

۱- مشخصه Icon فرم را انتخاب کنید.

۲- روی دکمه سه نقطه کلیک کنید تا کادر محاوره‌ای Load Icon باز شود، نشانه موردنظر را انتخاب کنید (شکل ۷-۱۶).



شکل ۷-۱۶- در ویندوز، پرونده‌های نشانه، در همهٔ نماها تصاویر خودشان را نشان می‌دهند.

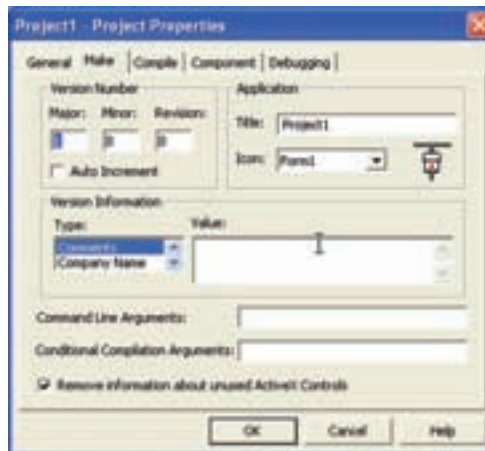
۳- روی دکمه Open کلیک کنید تا نشانه به فرم اضافه شود.
 اگر پروژه فقط دارای یک فرم باشد، اضافه کردن نشانه سفارشی به مشخصه Icon فرم، به طور خودکار همان نشانه را برای برنامه تعیین می‌کند. اگر پروژه دارای چندین فرم باشد، باید نشانه برنامه کاربردی را از کادر محاوره‌ای Project Properties تعیین کنید.

نکته

نمی‌توان نشانه‌ای را در VB یا Paint ویندوز ایجاد کرد. ولی CD ویژوال بیسیک دارای برنامه‌ای برای ایجاد نشانه است. این برنامه Microsoft Image Editor (imagedit.exe) نام دارد که در مسیر Common\Tools\Imagedit قرار دارد. اگر این برنامه را نمی‌توانید پیدا کنید، احتمالاً آن را نصب نکرده‌اید، برنامه نصب ویژوال بیسیک را دوباره اجرا کرده و آن را نصب کنید.

تعیین نشانه برای برنامه کاربردی

- ۱- نشانه‌ای را برای هر فرم در برنامه کاربردی تعیین کنید.
- ۲- کادر محاوره‌ای Project Properties را باز کرده و زبانه Make را انتخاب کنید.
- ۳- از لیست بازشوی Icon، فرمی را که دارای نشانه مورد نظر برای استفاده به عنوان نشانه برنامه است انتخاب کنید (شکل ۱۷-۷).



شکل ۱۷-۷- می‌توان اغلب مشخصه‌های پروژه را در کادر محاوره‌ای Project Properties تعیین کنید.

- ۱- برنامه‌ای بنویسید که 70° نقطه در محل‌های تصادفی و با رنگ و پهنای تصادفی ترسیم کند.
- ۲- برنامه‌ای بنویسید که مجموعه‌ای از خطوط با طول‌ها و رنگ‌های متفاوت ترسیم نماید.
- ۳- برنامه‌ای بنویسید که مجموعه‌ای از هشت دایره متحال‌المركز با فاصله 10° پیکسل از یکدیگر ترسیم نماید.
- ۴- برنامه‌ای بنویسید که یک شبکه توری (GRID) بر روی تصویری در یک Picture box ترسیم نماید.
- ۵- برنامه‌ای بنویسید که به کاربر امکان جابه‌جا کردن خط با مشخص کردن مقادیر $X1, Y1, X2, Y2$ را بدهد.
- ۶- برنامه‌ای بنویسید که کاربر بتواند با درگ کردن روی فرم، خطی را رسم کند و نقطه شروع و پایان خط ضخیم‌تر از بقیه نقاط باشد.
- ۷- برنامه‌ای بنویسید که تصویری را وسط فرم قرار دهد و سپس بارش برف را روی آن طراحی کنید.