

توانایی استفاده از جلوه‌های گرافیکی و چاپ در ویژگی‌های بیسیک

هدف‌های رفتاری

پس از مطالعه این واحد کار از فراگیر انتظار می‌رود که:

- ۱- مفاهیم مربوط به سیستم مختصات را بدانند.
- ۲- توانایی تغییر مختصات را به منظور انجام ترسیمات داشته باشد.
- ۳- توانایی به‌کارگیری متدهای گرافیکی زیر را داشته باشد:
PSet , Line , Circle , Point , Cls , Print , TextHeight , TextWidth
- ۴- توانایی استفاده از خصوصیات گرافیکی زیر را داشته باشد:
CurrentX , CurrentY , AutoRedraw , DrawMode , DrawStyle , DrawWidth , FillStyle
- ۵- توانایی استفاده از توابع QBColor و RGB را داشته باشد.
- ۶- توانایی استفاده از امکانات چاپ، خصوصیات و متدهای آن را در برنامه‌ها داشته باشد.
- ۷- توانایی استفاده از امکانات چندرسانه‌ای و کنترل MCI و MAM
- ۸- توانایی استفاده از شیء Picture و رویه‌های SavePicture و LoadPicture

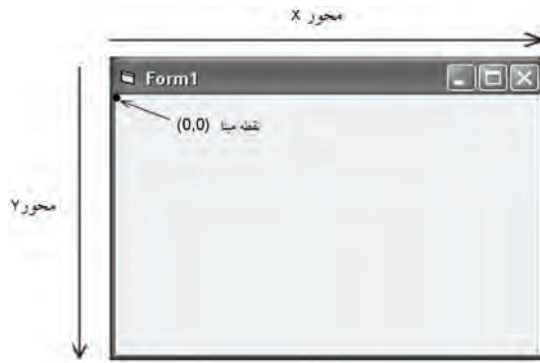
کلیات

تاکنون چگونگی طراحی و ساخت رابط‌های گرافیکی را با استفاده از فرم‌ها و بعضی از کنترل‌ها آموختید؛ اما گاهی اوقات وجود فرم‌ها و کنترل‌ها بدون جلوه‌های گرافیکی محیط نرم‌افزار را خسته‌کننده و غیر قابل استفاده می‌کند. با استفاده از جلوه‌های گرافیکی نظیر رنگ، نمودارهای گرافیکی و تصاویر متحرک می‌توانید به کاربر در مشاهده و درک بهتر گزارشات و نتیجه محاسبات کمک کنید.

ویژوال بیسیک قابلیت بالایی در استفاده از جلوه‌های گرافیکی در اختیار شما قرار می‌دهد و می‌توانید این جلوه‌های گرافیکی را با دو روش ایجاد کنید: کنترل‌های گرافیکی مثل Line, Shape و متدهای گرافیکی مثل Circle, PSet, Line و ...

۱-۱۲ مفهوم سیستم مختصات در ویژوال بیسیک

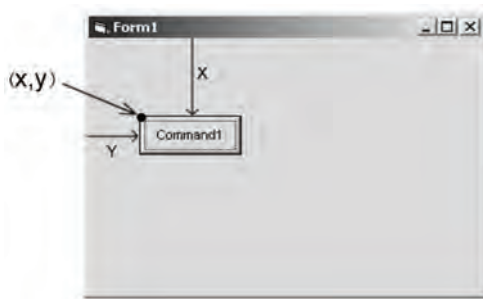
در ویژوال بیسیک نیز مانند ریاضیات و هندسه برای انجام هر نوع ترسیمی از سیستم مختصات استفاده می‌شود، هم‌چنین از دو بعد یا محور مختصات برای تعیین موقعیت ترسیمات استفاده می‌شود (شکل ۱-۱۲).



شکل ۱-۱۲

مختصات هر نقطه به صورت (X,Y) تعیین می‌شود مقدار X موقعیت نقطه را در طول محور X و مقدار Y موقعیت نقطه را در طول محور Y بیان می‌کند که مقدار شروع در هر یک از محورها صفر است. دقت داشته باشید که محورهای مختصات در ویژوال بیسیک با محورهای مختصات در ریاضیات متفاوت است. در شکل ۱-۱۲ نحوه قرار گرفتن محورها بر نقطه مبنا قابل مشاهده است. مختصات نقطه مبنا $(0,0)$ است که در گوشه بالا و چپ شیء مربوطه که معمولاً فرم است قرار دارد.

وقتی کنترل را جابه‌جا کرده یا اندازه آن را تغییر می‌دهید، از سیستم مختصات فرمی که کنترل در آن قرار دارد، استفاده می‌کنید. در واقع کنترل‌ها و هرگونه ترسیمات گرافیکی از سیستم مختصات شیء که در آن رسم می‌شوند، تبعیت می‌کنند. اگر کنترلی در روی فرم قرار گیرد موقعیت و اندازه کنترل از سیستم مختصات فرم پیروی می‌کند و اگر خطی در یک جعبه تصویر PictureBox رسم شود، موقعیت و اندازه خط از سیستم مختصات کنترل جعبه تصویر استفاده می‌کند.



شکل ۲-۱۲ تعیین موقعیت یک کنترل به وسیله سیستم مختصات

برای تعریف موقعیت ترسیمات گرافیکی به وسیله محورهای واحدهای اندازه‌گیری استفاده می‌شود که به آن مقیاس می‌گویند. در ویژوال بیسیک هر محور در سیستم مختصات می‌تواند مقیاس خاص خود را داشته باشد.

۲-۱۲ تغییر سیستم مختصات

می‌توانید سیستم مختصات یک شیء خاص مثل فرم را به وسیله خصوصیت Scale یا متد Scale روی مقادیر مورد نظرتان تنظیم کنید. برای انجام این کار می‌توانید از مقیاس پیش‌فرض استفاده کرده یا یکی از مقیاس‌های استاندارد را انتخاب کنید یا این که یک مقیاس جدید را ایجاد کنید. با تغییر مقیاس سیستم مختصات می‌توانید اندازه و موقعیت ترسیمات گرافیکی را روی فرم با توجه به نیازتان به آسانی تنظیم کنید.

هر فرم یا بعضی از کنترل‌ها مانند PictureBox چندین خصوصیت Scale، نظیر ScaleMode، ScaleWidth، ScaleHeight، ScaleTop، ScaleLeft و یک متد Scale دارند که به وسیله آن‌ها می‌توانید سیستم مختصات خود را تعریف کنید. مقیاس پیش‌فرض twip است.

همان طور که قبلاً هم اشاره کردیم هر ۵۶۷ twip برابر با یک سانتی متر است. برای انتخاب یک مقیاس استاندارد می توانید یکی از مقادیر موجود در جدول ۱-۱۲ را برای خصوصیت ScaleMode فرم یا کنترل مورد نظر خود در نظر بگیرید.

جدول ۱-۱۲ مقادیری که خصوصیت ScaleMode کسب می کند.

ثابت رشته‌ای	ثابت عددی	توضیح (نوع مقیاس)
vbUser	۰	مقادیر تعریفی کاربر در خصوصیات ScaleWidth, ScaleHeight, ScaleLeft, ScaleTop استفاده می شوند.
vbTwips	۱	twip
vbPoints	۲	Point (۱ Inch = ۷۲Point)
vbPixels	۳	Pixel (یک pixel کوچک ترین واحد نمایشی در صفحه نمایش یا چاپگر است و تعداد آن ها در هر اینچ به مقدار وضوح تصویر بستگی دارد.)
vbCharacters	۴	Character
vbInches	۵	Inch
vbMillimeters	۶	Milimeter
vbCentimeters	۷	Centimeter

اگر بخواهید مختصات نقطه مبنا را تغییر دهید یا مقیاس جدید را در یک کنترل یا فرم ایجاد کنید می توانید از خصوصیات ScaleWidth, ScaleHeight, ScaleLeft و ScaleTop استفاده کنید. خصوصیات ScaleLeft و ScaleTop با دریافت مقادیر عددی، مختصات نقطه مبنا را در کنترل و فرم معین می کنند. مقدار پیش فرض برای این دو خصوصیت صفر است. مقدار این خصوصیات را می توانید از طریق پنجره خصوصیات تغییر دهید و یا با استفاده از نوشتن کد در رویدادها و رویه های مورد نظر این کار را انجام دهید. شکل کلی استفاده از این خصوصیات به صورت زیر است:

object.ScaleLeft = value

object.ScaleTop = value

منظور از object نام یک فرم یا کنترل است و استفاده از آن اختیاری است و در صورتی که از آن استفاده نشود فرمی که فوکوس دارد در نظر گرفته خواهد شد. value یک مقدار عددی است که مختصات نقطه مبنا را تعیین می کند در صورت

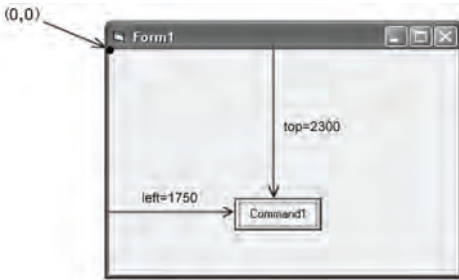
عدم استفاده از value می‌توانید مقدار فعلی مختصات نقطه مبنا را به دست آورید. به عنوان مثال فرض کنید در یک فرم همراه با یک کنترل دکمه فرمان این مقادیر تنظیم شده است:

```
Form1.ScaleMode = 1
Command1.Top = 2300
Command1.Left = 1750
```

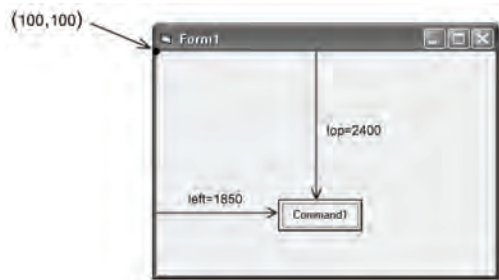
اگر بخواهیم در این فرم مختصات نقطه مبنا را (100, 100) قرار دهیم در این صورت خصوصیات مربوطه را به صورت زیر تنظیم می‌کنیم:

```
Form1.ScaleTop = 100
Form1.ScaleLeft = 100
```

در این صورت مقدار خصوصیت ScaleMode در Form1 به طور خودکار روی مقدار صفر قرار می‌گیرد و در ضمن خصوصیت Top و Left در دکمه فرمان به طور خودکار به ترتیب روی مقادیر 2400° و 1850° تنظیم می‌شوند و همان طور که از مقایسه مقادیر در دو حالت مشاهده می‌کنید مقادیر (x,y) برای کنترل دکمه فرمان با توجه به نقطه مبنای جدید (1000, 1000) تنظیم می‌شود. این دو حالت را می‌توانید در شکل‌های ۳-۱۲ و ۴-۱۲ مشاهده کنید.



شکل ۳-۱۲



شکل ۴-۱۲

به علاوه به وسیله این دو خصوصیت می‌توانید از مقدار مختصات نقطه مبنا مطلع شوید در رویه بعد، پس از تغییر مقدار این دو خصوصیت، مقادیر جدید در یک کادر پیغام نمایش داده می‌شوند.

```
Private Sub cmdshow_Click ()
    Form1.ScaleLeft = 150
    Form1.ScaleTop = 180
    MsgBox " ScaleLeft = " + Str ( ScaleLeft ) + " ScaleTop = " + Str ( ScaleTop )
End Sub
```



نکته در صورت عدم استفاده از بخش object، فرمی که فوکوس دارد در نظر گرفته خواهد شد.

علاوه بر تغییر مختصات نقطه مبنا، می‌توانید مقیاس سیستم مختصات را نیز تغییر دهید. خصوصیات ScaleWidth و ScaleHeight واحد اندازه‌گیری در محورهای X و Y را تعیین می‌کنند. مقدار این خصوصیات را می‌توانید از طریق پنجره خصوصیات یا با نوشتن کد مناسب تغییر دهید. شکل کلی استفاده از این خصوصیات به صورت زیر است:

```
object.ScaleHeight = value
```

```
object.ScaleWidth = value
```

منظور از object نام یک فرم یا کنترل است و value یک مقدار عددی است که واحد اندازه‌گیری را در محورها تعیین می‌کند. در صورتی که از object استفاده نشود، فرمی که فوکوس دارد در نظر گرفته خواهد شد و در صورت عدم استفاده از بخش value می‌توانید مقادیر ذخیره شده در این دو خصوصیت را به دست آورید.
به عنوان مثال به رویه زیر توجه کنید:

```
Private Sub cmdscale_Click()
```

```
Form1.ScaleWidth = 1000
```

```
Form1.ScaleHeight = 500
```

```
End Sub
```

با اجرای رویه فوق معیار اندازه‌گیری در محور افقی (X)، یک هزارم ($\frac{1}{1000}$) عرض داخلی (Width) فرم و در محور عمودی (Y)، یک پانصدم ($\frac{1}{500}$) ارتفاع داخلی (Height) فرم خواهد بود.



نکته خصوصیات ScaleWidth و ScaleHeight واحدها را با توجه به ابعاد داخلی فرم یا شیء مربوطه تعیین می‌کنند. این ابعاد شامل حاشیه‌ها یا منوها یا نوار عنوان نمی‌شوند. این دو خصوصیت همواره در رابطه با بخش قابل ترسیم داخل فرم یا شیء مربوطه تعریف می‌شوند.

به عنوان مثال یک فرم همراه با یک دکمه فرمان با مشخصات زیر را ایجاد کنید:

```
Form1.BorderStyle = None
```

Form1.Height = 3000

Form1.Width = 4200

Form1.ScaleMode = twip

Command1.Height = 400

Command1.Width = 1200

اگر تنظیمات فوق را به ترتیب انجام دهید سپس مقدار خصوصیات ScaleHeight و ScaleWidth را ملاحظه کنید خواهید دید که مقدار این دو خصوصیت مانند مقدار ارتفاع و عرض فرم است زیرا فرم شما بدون حاشیه و نوار عنوان است و تمام فضای فرم، فضای قابل دسترس محسوب می‌شود. اما اگر مقدار خصوصیت BorderStyle را Sizeable قرار دهید مقدار دو خصوصیت ScaleWidth و ScaleHeight مقادیر کمتری خواهند بود زیرا بخشی از ابعاد فرم به نوار عنوان و حاشیه‌ها داده شده است که جزء فضاهای قابل دسترس نیستند. همچنین معیار اندازه‌گیری در محورها توسط خصوصیت ScaleMode تعیین می‌شود که از نوع twip است.

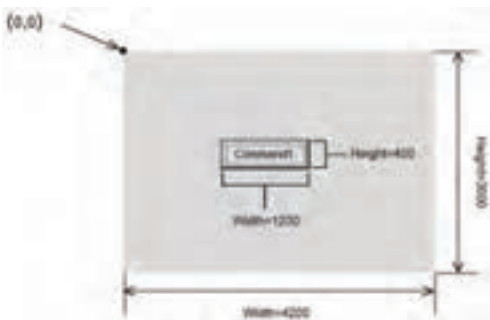
اکنون خصوصیات ScaleHeight و ScaleWidth فرم را روی ۵۰۰ و ۱۰۰۰ تنظیم کنید با تغییر یکی از این خصوصیات، خصوصیت ScaleMode به‌طور خودکار روی مقدار 0-User تنظیم می‌شود پس از تغییر دو خصوصیت فوق، خصوصیات Height و Width کنترل Command1 را مشاهده کنید. همان‌طور که می‌بینید مقدار آن‌ها به ترتیب به ۶۶/۶۶۷ و ۲۸۵/۷۱۴ تغییر کرده است. در واقع برای محاسبه هر یک از این مقادیر به این صورت عمل شده است (شکل ۵-۱۲ و ۶-۱۲).

$$400 \times \frac{500}{3000} = 66/667$$

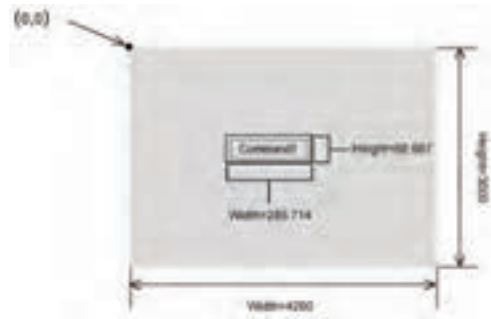
مقدار خصوصیت Command1.Height

$$1200 \times \frac{1000}{4200} = 285/714$$

مقدار خصوصیت Command1.Width



شکل ۵-۱۲



شکل ۶-۱۲



نکته • تنظیم هر یک از خصوصیات Scale، مقدار خصوصیت ScaleMode را به طور خودکار روی ۰ تنظیم می کند.

• انتخاب مقیاس بزرگ تر از صفر برای خصوصیت ScaleMode مقدار خصوصیات ScaleHeight و ScaleWidth را به طور خودکار روی مقادیر جدید تنظیم می کند و مقدار خصوصیات ScaleTop و ScaleLeft را روی صفر تنظیم می کند.

هر چهار خصوصیت Scale می توانند مقادیر اعشاری و حتی منفی داشته باشند. در صورت استفاده از اعداد منفی برای خصوصیات ScaleWidth و ScaleHeight جهت محورها در سیستم مختصات تغییر می کند، مثلاً برای فرم و کنترلی با مشخصات زیر فرمی مطابق شکل ۷-۱۲ خواهید داشت:

```
Form1.ScaleWidth = - 1000
```

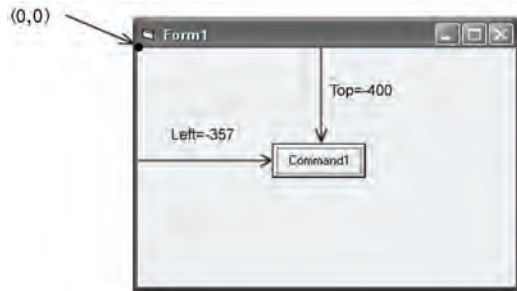
```
Form1.ScaleHeight = - 1000
```

```
Form1.ScaleLeft = 0
```

```
Form1.ScaleTop = 0
```

```
Command1.Left = - 357
```

```
Command1.Top = - 400
```



شکل ۷-۱۲

به عنوان آخرین مثال در رابطه با دو خصوصیت ScaleWidth و ScaleHeight به این رویه توجه کنید:

```
Private Sub cmdscale_Click()
```

```
Form1.ScaleMode = 1
```

```
Form1.Width = 4200
```

```
Form1.height = 3000
```

```
Print "Form1.Width = " Form1.Width, "Form1.Height = " Form1.Height
```

```
Print "Form1.ScaleWidth = " Form1.ScaleWidth
```

```
Print "Form1.ScaleHeight = " Form1.ScaleHeight
```

```
Print Form1.ScaleWidth = 1000
```

```
Print Form1.ScaleHeight = 1500
```



```
Print "Form1.Width = " Form1.Width, "Form1.Height = " Form1.Height;
```

```
Print "Form1.ScaleWidth = " Form1.ScaleWidth
```

```
Print "Form1.ScaleHeight = " Form1.ScaleHeight
```

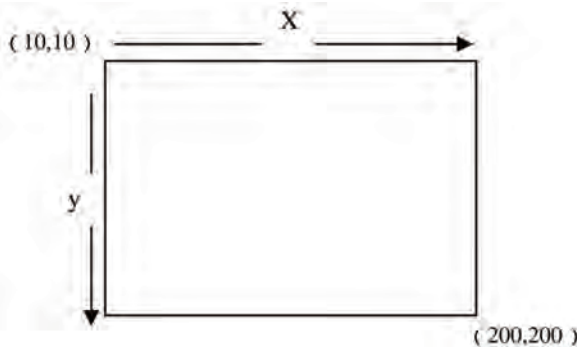
```
End Sub
```

در این رویه رویداد ابتدا نوع مقیاس اندازه‌گیری twip در نظر گرفته شده است سپس عرض و ارتفاع فرم مقداردهی می‌شود. اولین دستور Print ابتدا عرض و ارتفاع فرم یعنی 4200 و 3000 و دو فرمان بعدی نیز مقادیر 4080 و 2595 را برای خصوصیات ScaleWidth و ScaleHeight نمایش می‌دهد. در واقع این دو خصوصیت ابعاد فرم را بدون در نظر گرفتن حاشیه و نوار عنوان فرم نمایش می‌دهند. اما در خطوط بعدی مقدار دو خصوصیت ScaleWidth و ScaleHeight روی 1000 و 1500 تنظیم می‌شود در این حالت خصوصیت ScaleMode نیز به‌طور خودکار روی صفر تنظیم می‌شود. پس از مقداردهی دو خصوصیت ScaleWidth و ScaleHeight، سه دستور Print دیگر اجرا می‌شوند اولین Print همان ابعاد قبلی یعنی 4200 و 3000 را برای ابعاد فرم نمایش می‌دهند اما دو دستور Print بعد مقادیر جدید دو خصوصیت ScaleWidth و ScaleHeight یعنی 1000 و 1500 را نمایش خواهند داد.

روش دیگری که در رابطه با تغییر سیستم مختصات می‌توان به‌کار برد استفاده از متد Scale است. در واقع متد Scale راه‌حل مناسب و آسان‌تری برای تنظیم سیستم مختصات می‌باشد. شکل کلی متد Scale به‌صورت زیر است:

```
object. Scale (x1,y1) - (x2,y2)
```

مقادیر عددی $x1,y1$ مختصات گوشه بالایی و سمت چپ شیء و مقادیر عدد $x2,y2$ مختصات گوشه پایینی و سمت راست شیء را مشخص می‌کنند.



شکل ۸-۱۲

object در واقع نام شیء است که می‌خواهید سیستم مختصات آن را تعیین کنید و در صورت عدم استفاده از این قسمت، فرمی که فوکوس دارد به‌عنوان شیء مربوطه در نظر گرفته می‌شود. به‌عنوان مثال دستور زیر سیستم مختصات را در فرم به‌صورت شکل ۸-۱۲ درمی‌آورد.

```
Form1. Scale (10,10)-(200,200)
```

در واقع فرمان فوق چهار خصوصیت Scale را به این صورت تنظیم می‌کند:

ScaleWidth = 190

ScaleHeight = 190

ScaleTop = 10

ScaleLeft = 10

۳-۱۲ خصوصیات و متدهای گرافیکی

در این جا لازم است تا چگونگی انجام انواع ترسیمات گرافیکی را بیاموزید. تاکنون این کار را با استفاده از کنترل‌های گرافیکی انجام می‌دادید اما کنترل‌های معرفی شده همواره نیازهای گرافیکی را برطرف نمی‌کنند و در پاره‌ای از مواقع نیز کار را با مشکلات متعدد روبه‌رو می‌سازند. استفاده از متدها و تنظیم خصوصیات گرافیکی نیاز گرافیکی شما را در پروژه‌های برنامه‌نویسی برآورده می‌کند.

۳-۱۲-۱ متد PSet

به وسیله این متد می‌توانید نقاط مورد نظر خود را در مکان‌های مناسب قرار دهید شکل کلی این متد به صورت زیر است:

object. PSet Step (x,y) , color

منظور از object، شیئی است که نقطه روی آن رسم می‌شود. اگر از ذکر آن خودداری کنید فرمی که فوکوس را در اختیار دارد مکان رسم نقطه است. کلمه کلیدی Step نیز اختیاری است و در صورت استفاده از آن هنگام رسم نقطه، مکان ترسیم با توجه به موقعیت جاری در شیئی که در آن ترسیم انجام شده، انتخاب می‌شود.

x, y مقادیر عددی از نوع Single هستند که مختصات محل ترسیم نقطه را مشخص می‌کنند. علاوه بر این می‌توانید رنگ نقطه مورد نظر را به وسیله بخش color تعیین کنید در صورت عدم استفاده از این قسمت، رنگی که در خصوصیت ForeColor شیئی که نقطه در آن رسم می‌شود (به عنوان مثال فرم) در نظر گرفته خواهد شد. جدول مربوط به مقادیر رنگ‌ها در جدول ۲-۱۲ آورده شده است. می‌توانید از ثابت‌های رشته‌ای یا از ثابت‌های عددی در مبنای ۱۶ استفاده کنید.

جدول ۲-۱۲ مقادیر رنگ در ویژوال بیسیک

ثابت رشته‌ای	ثابت عددی (مبنای ۱۶)	توضیح
vbBlack	&H0	سیاه
vbRed	&HFF	قرمز
vbGreen	&HFF00	سبز
vbYellow	&HFFFF	زرد
vbBlue	&HFF0000	آبی
vbMagenta	&HFF00FF	بنفش
vbCyan	&HFFFFFF00	فیروزه‌ای
vbWhite	&8HFFFFFF	سفید

به عنوان مثال این دستورات یک نقطه به رنگ آبی روی فرم نمایش می‌دهد:

```
Form1.ForeColor = vbBlue
```

```
PSet ( 1000, 200 )
```

اکنون دستور زیر را در نظر بگیرید:

```
PSet ( 500 , 700 ) , vbCyan
```

این دستور نقطه‌ای را با رنگ فیروزه‌ای در مختصات ۵۰۰ و ۷۰۰ رسم می‌کند حال اگر بلافاصله دستور زیر اجرا شود:

```
PSet Step ( 500 , 700 ) , vbGreen
```

مختصات نقطه مربوطه با توجه به مختصات نقطه رسم شده قبلی محاسبه می‌شود بنابراین دو نقطه روی یکدیگر قرار نخواهند گرفت.

مثال ۱: می‌خواهیم پروژه‌ای طراحی کنیم که ده نقطه به صورت تصادفی روی یک



فرم ترسیم کند.

به این منظور عملیات زیر را به ترتیب انجام دهید:

۱ - برنامه ویژوال بیسیک را اجرا کرده و یک

پروژه از نوع Standard EXE ایجاد کنید.

۲ - نام فرم frmgraphic و عنوان آنرا روی

Geraphic تنظیم کنید سپس یک دکمه فرمان با نام

cmdpset و عنوان &Pset مطابق شکل ۹-۱۲ روی فرم

قرار دهید.



شکل ۹-۱۲

۳ - رویداد click دکمه PSet را به صورت زیر تنظیم کنید:

```
Private Sub cmdpset_Click()
```

```
Dim i As Integer
```

```
Randomize
```

```
For i = 1 To 10
```

```
    PSet (Int(Rnd * 1000 ), Int(Rnd * 2000 ))
```

```
Next i
```

```
End Sub
```

- ۴ - پروژه و فرم را با نام Point ذخیره کنید سپس برنامه را اجرا و روی دکمه PSet کلیک کنید تا ۱۰ نقطه به صورت تصادفی روی فرم ترسیم شوند (شکل ۹-۱۲).
- ۵ - از برنامه خارج شوید و به پنجره ویژوال بیسیک بازگردید.

تمرین:



پروژه Point را به گونه‌ای تغییر دهید که نقاط با رنگ‌های مختلف به طور پیوسته روی فرم نمایش داده شوند.

۲-۳-۱۲ متد Line

به وسیله متد Line می‌توانید انواع خطوط و مستطیل‌های توپر و توخالی را رسم کنید. شکل کلی این متد به صورت زیر است:

```
object. Line Step (x1,y1) - ( x2,y2 ), color, B F
```

عملکرد گزینه‌های object و Step مانند متد PSet است. مقادیر x_1 و y_1 مختصات نقطه ابتدای خط و مقادیر x_2 و y_2 مختصات نقطه انتهایی را در خط تعیین می‌کنند. به وسیله بخش Color نیز می‌توانید رنگ خط را مشخص کنید.

استفاده از حرف B یک مستطیل خالی و استفاده از حرف F به همراه حرف B (BF) یک مستطیل توپر ایجاد می‌کند، البته استفاده از این دو کاراکتر اختیاری است. به عنوان مثال به دستورات زیر توجه کنید:

```
Line (500,800) - (1500 , 1500 )
```

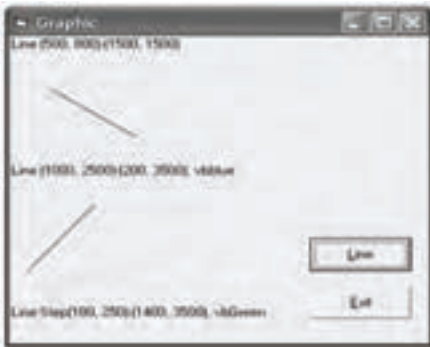
```
Line (1000 , 2500) - ( 200,3500 ),vbBlue
```

Line Step (100,250) – (1400,3500) , vbGreen

Line (410,870) – (1500,1800), vbBlue, B

Line (410,2500) – (2000,3700), vbRed, BF

در این مجموعه دستورات، دستور اول یک خط با رنگی که در خصوصیت ForeColor فرم تعیین شده است ترسیم می‌کند، دستور دوم یک خط با رنگ آبی و دستور سوم نیز یک خط با رنگ سبز ترسیم می‌کنند اما در دستور سوم به دلیل استفاده از Step نقطه شروع از انتهای نقطه‌ای که خط دوم تعیین می‌کند استفاده می‌شود. دستور چهارم و پنجم نیز به ترتیب یک مستطیل توخالی با رنگ آبی و یک مستطیل توپر با رنگ قرمز رسم می‌کنند. در شکل ۱۰-۱۲ و ۱۱-۱۲ نتیجه اجرای این دستورات را مشاهده می‌کنید.



شکل ۱۰-۱۲



شکل ۱۱-۱۲

۱۲-۳-۳ مند Circle

به وسیله این متد می‌توانید انواع دایره، بیضی و کمان را رسم کنید. شکل کلی این متد به صورت زیر است: `object. Circle step (x,y), radius , color , start , end , aspect`
 بخش `object` و `step` و `color` مانند توضیحات ارائه شده در متد `PSet` است. مقادیر عددی `x` و `y` از نوع `Single` بوده و مختصات مرکز دایره یا بیضی را با توجه به مقدار `ScaleMode` تعیین می‌کند.

مقدار عددی `radius` نیز از نوع `Single` است و مقدار شعاع دایره را براساس مقدار `ScaleMode` معین می‌کند. مقادیر عددی `start` و `end` از نوع `Single` و اختیاری بوده و موقعیت شروع و خاتمه کمان را برای ترسیم معین می‌کند. مقدار مجاز برای این دو مقدار از -2π رادیان تا 2π رادیان است. در صورت عدم استفاده از این دو مقدار، کمان ترسیمی از صفر تا 2π رادیان در نظر گرفته می‌شود.



نکته • جهت ترسیم کمان، خلاف جهت حرکت عقربه‌های ساعت است. مقدار عددی aspect از نوع Single است و نسبت دو قطر عمودی و افقی را در بیضی معین می‌کند اگر این مقدار ۱ باشد دایره و در غیر این صورت برای مقادیر بزرگ‌تر از ۱، بیضی‌های عمودی و برای مقادیر کوچک‌تر از ۱، بیضی‌های افقی ایجاد می‌شود.

- منظور از عدد π ، ثابت 3.14159265358979 است.
- برای رسم قطاع‌های بیضی یا دایره همراه با خطوط شعاع آن‌ها از مقادیر منفی استفاده کنید.

به عنوان مثال به دستورات زیر توجه کنید:

Circle (700 , 1000),500, vbRed

Circle (600,2000), 400, Pi/2,3* Pi/2

Circle (600,3500), 400, -Pi/6, -Pi/3

Circle (1700,1200), 500, vbRed,,2

Circle (1700,2500), 500, vbRed,,0.2

Circle (1000,4000), 1500,, -0.001, Pi/2,0.4

نتیجه اجرای این دستورات را در شکل‌های ۱۲-۱۲ و ۱۲-۱۳ مشاهده می‌کنید.



شکل ۱۲-۱۲



شکل ۱۲-۱۳



شکل ۱۲-۱۴

تمرین:



پروژه‌ای طراحی کنید که شکل ۱۲-۱۴ را روی یک فرم ترسیم کند.

۴-۳-۱۲ متد Point

این متد با دریافت مختصات یک نقطه، شماره رنگ آن را به صورت یک عدد صحیح از نوع Long بازمی‌گرداند. شکل کلی این متد به صورت زیر است:

object. Point (x,y)

به عنوان مثال فرض کنید که با استفاده از متد Line، یک مستطیل توپر با رنگ قرمز رسم شده است:

```
Line (500,500)-(2000,2500) , vbRed , BF
```

```
Print . Point (700,800)
```

در دستور دوم مختصات نقطه‌ای را که در مستطیل قرمز قرار دارد، به متد Point می‌دهد و در نتیجه مقدار ۲۵۵ که بیانگر رنگ قرمز است توسط متد Point بازگشت می‌یابد و نمایش داده می‌شود.

۵-۳-۱۲ خصوصیات CurrentX و CurrentY

به وسیله این دو خصوصیت می‌توانید موقعیت جاری مکان نما در صفحه ترسیمات را تغییر دهید. خصوصیت CurrentX مختصات مکان‌نما را در جهت محور X و خصوصیت CurrentY مختصات مکان‌نما را در جهت محور Y تعیین می‌کنند. شکل کلی نحوه استفاده از این دو خصوصیت به صورت زیر است:

```
object. CurrentX = x
```

```
object. CurrentY = y
```

مقادیر x و y مقادیر عددی هستند که موقعیت مکان‌نما را در سطح شیء که ترسیمات گرافیکی روی آن انجام می‌گیرد مشخص می‌کنند. اگر مقادیر x و y استفاده نشود می‌توان مختصات فعلی مکان‌نما را به دست آورد. به دستورات زیر توجه کنید:

```
Form1.CurrentX = 2000
```

```
Form1.CurrentY = 1000
```

```
PSet (CurrentX, CurrentY)
```

در این دستورات ابتدا مختصات مکان‌نما به نقطه (۱۰۰۰ و ۲۰۰۰) تغییر پیدا می‌کند سپس به وسیله مقدار این دو خصوصیت نقطه‌ای در همین مختصات رسم می‌شود. به عنوان مثالی دیگر به رویه زیر توجه کنید:

Private Sub cmdshow_Click()

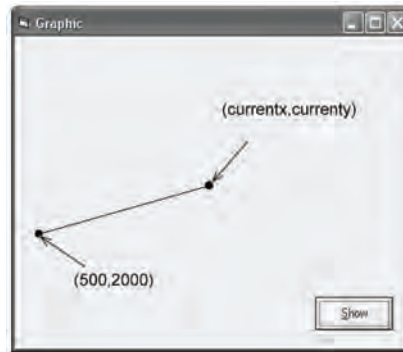
Form1.CurrentX = ScaleWidth / 2

Form1.CurrentY = ScaleHeight / 2

Line (CurrentX, CurrentY)-(500, 2000)

End Sub

با استفاده از خصوصیات ScaleWidth و ScaleHeight ابعاد فرم به دست آمده و با تقسیم آن‌ها بر عدد ۲ مختصات نقطه میانی در فرم محاسبه شده است و فرمان Line، خطی را از این نقطه تا نقطه (۵۰۰، ۲۰۰۰) رسم می‌کند (شکل ۱۵-۱۲).



شکل ۱۵-۱۲

تمرین:



پروژه‌ای طراحی کنید که مختصات مرکز یک دایره و شعاع و رنگ آن را دریافت کرده و دایره موردنظر را رسم کند.

۶-۳-۱۲ متد Cls

شکل کلی این متد به صورت زیر است:

object. Cls

این متد صفحه را کاملاً پاک کرده و مکان نما را به مختصات (۰ و ۰) انتقال می‌دهد. قبلاً بانحوه استفاده از این متد آشنا شده‌اید. به عنوان مثال در رویه زیر ابتدا یک مستطیل توپر با رنگ بنفش و یک دایره با رنگ قرمز رسم می‌شود و بعد از رسم آن‌ها یک کادر پیغام (MessageBox) با جمله Choose ok to clear this background نمایش داده می‌شود که در صورت فشردن دکمه

فرمان OK در کادر پیغام، سطح فرم پاک می‌شود و مقدار صفر برای دو خصوصیت CurrentX و CurrentY در کادر پیغام دیگری به نمایش درمی‌آید که نشان‌دهنده عملکرد متد Cls در تغییر موقعیت جاری مکان‌نماست.

```
Private Sub cmdcls_Click()
```

```
Dim pi, msg
```

```
pi = 3.14159265358979
```

```
Line (200, 150)-(850, 600), vbMagenta, BF
```

```
Circle (1600, 1800), 400, vbRed
```


```
msg = "Choose OK to clear this background"
```

```
MsgBox msg
```

```
Cls
```

```
MsgBox "CurrentX=" + Str(CurrentX) + "CurrentY=" + Str(CurrentY)
```

```
End Sub
```

 **مثال ۲:** پروژه‌ای طراحی کنید که به وسیله آن کاربر بتواند اشکال هندسی مانند نقطه، خط و انواع مستطیل را رسم کند. به این منظور مراحل بعد را انجام دهید:

۱- یک پروژه از نوع Standard EXE ایجاد کنید که شامل یک فرم با عرض و ارتفاع

۸۰۰۰ و ۶۰۰۰ باشد. سپس نام و عنوان آن را به ترتیب روی FrmDrawing و Drawing

تنظیم کنید و خصوصیت Appearance آن را روی Flat- تنظیم کنید. (شکل ۱۶-۱۲).

۲- در این مرحله بخش تعاریف فرم و رویدادهای ماوس را به صورت زیر تنظیم کنید:

```
Option Explicit
```

```
Dim intx As Integer, inty As Integer
```

```
Private Sub Form_MouseDown(Button As Integer, Shift As _
```

```
Integer, X As Single, Y As Single)
```

```
If Button = vbLeftButton Then
```

Pset (X,Y)

intx = X

inty = Y

End If

End Sub

Private Sub Form_MouseUp(Button As Integer,Shift As Integer , X As Single, Y As Single)

If Button = vbLeftButton Then

Select Case Shift

Case vbCtrlMask: Line (intx, inty)-(X,Y)

Case vbShiftMask: Line (intx, inty)-(X,Y), , B

Case vbAltMask: Line (intx, inty)-(X, Y), , BF

End Select

End If

End Sub

Private Sub Form_MouseMove(Button As Integer,Shift As_
Integer , X As Single, Y As Single)

If Button = vbRightButton Then

PSet (X,Y)

End If

End Sub



شکل ۱۶-۱۲

در رویداد MouseDown با استفاده از دستور If مقدار آرگومان Button بررسی می‌شود. اگر این مقدار برابر با vbLeftButton باشد به این معنی است که کلید سمت چپ ماوس فشرده شده است و با استفاده از دستور PSet یک نقطه در موقعیتی که اشاره‌گر ماوس قرار دارد، ترسیم می‌شود و همین مقادیر یعنی X و Y در دو متغیر عمومی ذخیره می‌شوند تا برای رویداد MouseUp مورد استفاده قرار گیرند. متد PSet می‌تواند یک نقطه در مختصات x و y با رنگ color ایجاد کند، color می‌تواند یکی از ثابت‌های رنگ ویژوال بیسیک باشد. شکل کلی استفاده از این دستور به صورت $PSet(x,y,color)$ می‌باشد. استفاده از آرگومان color اختیاری است.

از رویداد MouseUp برای ترسیم انواع خطوط و مستطیل‌های توپر و توخالی استفاده شده است. در این رویداد ابتدا رها شدن کلید سمت چپ ماوس به وسیله دستور If کنترل می‌شود، سپس با استفاده از یک دستور Select Case فشرده شدن کلیدهای Ctrl، Shift و Alt در زمان رها شدن کلید چپ ماوس بررسی می‌شود و اگر در زمان رها شدن کلید سمت چپ ماوس، کلید Ctrl نگه داشته شود، متد $(X,Y)-(Line(intx,inty))$ در اولین Case اجرا شده و یک خط بین نقطه با مختصات $(intx,inty)$ یعنی محل اشاره‌گر در زمان فشرده شدن کلید چپ ماوس و نقطه با مختصات (X,Y) یعنی محل اشاره‌گر در زمان رها شدن کلید چپ ماوس رسم می‌شود. اگر در زمان رها شدن کلید چپ ماوس، کلید Shift نگه داشته شود، متد $(X,Y,B)-(Line(intx,inty))$ در Case دوم اجرا شده و یک مستطیل بین نقاط $(intx,inty)$ و (X,Y) رسم می‌کند. پارامتر B در متد Line سبب رسم یک مستطیل

خواهد شد. اما اگر در زمان رها شدن کلید چپ ماوس، کلید Alt نگه داشته شود، متد $Line(x1,y1)-(x2,y2),color,BF$ در Case آخر اجرا شده و یک مستطیل توپر بین نقاط مربوطه رسم خواهد شد.

متد Line می تواند یک خط را با دریافت موقعیت نقاط ابتدا و انتهای آن ترسیم کند. شکل کلی این متد به صورت $Line(x1,y1)-(x2,y2),color$ است که $x2,x1$ فاصله نقاط ابتدا و انتهای خط از سمت چپ فرم و $y1$ و $y2$ فاصله نقاط ابتدا و انتهای خط از بالای فرم می باشد، آرگومان color رنگ خط ترسیمی را مشخص می کند. استفاده از این آرگومان اختیاری است.

برای رسم یک مستطیل توخالی از کاراکتر B در متد Line به صورت $Line(x1,y1)-(x2,y2),color,B$ استفاده کنید و برای رسم یک مستطیل توپر از عبارت BF در متد Line به صورت $Line(x1,y1)-(x2,y2),color,BF$ استفاده کنید.

آخرین رویداد در ماژول فرم، رویداد MouseMove فرم است که در این رویداد با بررسی آرگومان Button فشرده شدن دکمه راست ماوس بررسی می شود. اگر در زمان حرکت اشاره گر ماوس کلید راست ماوس نگه داشته شود (Right Drag) متد PSet اجرا شده و یک نقطه در محل اشاره گر ترسیم خواهد کرد.

۳ - پروژه را با نام drawing ذخیره کرده و سپس آن را اجرا کنید.

۴ - روی فرم، عمل کلیک انجام دهید. مشاهده می کنید که نقاط کوچکی با رنگ سیاه روی فرم ایجاد خواهد شد.

۵ - روی فرم عمل درگ انجام دهید و هم زمان کلید Ctrl را پایین نگه دارید، سپس کلید چپ ماوس را رها کنید. مشاهده می کنید که یک خط با رنگ سیاه ترسیم می شود. همین عمل را با کلیدهای Shift و Alt انجام دهید و نتیجه را بررسی کنید.

۶ - در این مرحله با کلید راست ماوس عمل درگ را انجام دهید، می بینید که با حرکت ماوس نقاط ترسیم می شوند.

۷ - از برنامه خارج شده و به پنجره ویژوال بیسیک بازگردید.

۷-۳-۱۲ متد Print

تاکنون بارها از این متد استفاده کرده‌اید. این متد می‌تواند هرگونه اطلاعات اعم از متن، مقادیر عددی، مقادیر مربوط به خصوصیات و نظایر آن‌ها را روی فرم نمایش دهد. شکل کلی این متد به صورتی است که در ادامه می‌آید:

```
object.Print outputlist
```

object نام شیئی است که اطلاعات روی آن نمایش داده می‌شوند. outputlist اطلاعاتی است که توسط متد Print روی شیء object نمایش داده می‌شود. استفاده از outputlist و object اختیاری است و در صورتی که outputlist استفاده نشود یک خط خالی نمایش داده خواهد شد.



نکته:

- استفاده از کاراکتر " ; " در متد Print سبب می‌شود تا اطلاعات به صورت چسبیده به هم و بدون فاصله از هم نمایش داده شوند.
- استفاده از کاراکتر " , " در متد Print سبب می‌شود تا اطلاعات با فاصله ۱۴ ستون از یکدیگر نمایش داده شوند.

به‌عنوان مثال، رویداد زیر عبارت Visual Basic 6 را به شکل‌های مختلف نمایش می‌دهند:

```
Private Sub cmdprint_Click( )
```

```
Dim str1 As String
```

```
Dim str2 As String
```

```
Dim str3 As String
```

```
str1 = "Visual"
```

```
str2 = "Basic"
```

```
str3 = "6"
```

```
Print str1; str2; str3
```

```
Print str1, str2, str3
```

```
Print str1;
```

```
Print str2,
```

```
Print str3
```

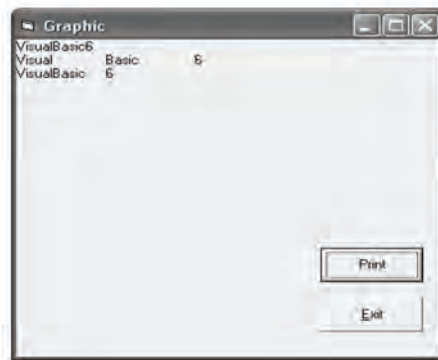
```
End Sub
```

نتیجه اجرای این دستورات به صورت زیر خواهد بود:

```
Visual Basic 6
```

```
Visual Basic 6
```

```
VisualBasic 6
```



شکل ۱۷-۱۲

برای نمایش فضاهای خالی توسط متد Print، می‌توانید از تابع Space استفاده کنید. مثلاً دستور زیر قبل از نمایش کلمه BASIC، ۵ فضای خالی ایجاد می‌کند.

```
Print Space (5); " BASIC "
```

در ضمن می‌توانید به وسیله تابع Tab اطلاعات خود را در ستون‌های مورد نظر نمایش دهید. شکل کلی تابع Tab به این صورت است:

```
Tab (n)
```


که در آن n یک مقدار عددی است که شماره ستون مورد نظر را معین می‌کند. مثلاً فرمان زیر، کلمه VISUAL را از ستون دوم به بعد و کلمه BASIC را از ستون دهم به بعد نمایش می‌دهد.

```
Print Tab (2) ; " VISUAL " ; Tab (10) ; " BASIC"
```

اما اگر این فرمان به صورت زیر استفاده شود:

```
Print Tab (2) ; " VISUAL " ; Tab (5) ; " BASIC "
```

کلمه VISUAL از ستون دوم خط جاری نمایش داده می‌شود اما چون پس از نمایش کلمه VISUAL، مکان نما در ستون هشتم قرار می‌گیرد و Tab دوم به ستون پنجم اشاره می‌کند بنابراین کلمه BASIC در ستون پنجم خط بعد نمایش داده خواهد شد.

 **نکته** در صورتی که مقدار n در تابع Tab عددی منفی باشد، ستون شماره ۱ در نظر گرفته خواهد شد.

۸-۳-۱۲ متدهای TextWidth و TextHeight

متد TextWidth عرض متن مورد نظر و متد TextHeight ارتفاع متن مورد نظر برای نمایش را معین می‌کنند. شکل کلی این دو متد به صورت زیر است:

```
object. TextWidth ( string )
```

```
object. TextHeight ( string )
```

string یک عبارت رشته‌ای است. هر دو متد یک مقدار عددی از نوع Single را بازمی‌گردانند مثلاً برای نمایش عبارت VISUAL BASIC در وسط صفحه می‌توانید به این صورت عمل کنید:

```
Private Sub cmdtext_Click ( )
```

```
CurrentX = (ScaleWidth - TextWidth ("VISUAL BASIC")) / 2
```

```
CurrentY = (ScaleHeight - TextHeight ("VISUAL BASIC")) / 2
```

```
Print "VISUAL BASIC"
```

```
End Sub
```

نتیجه اجرای این رویه را در شکل ۱۸-۱۲ مشاهده کنید:

در این رویه به وسیله دو متد TextWidth و TextHeight و محاسبه عرض و ارتفاع متن و با کم کردن این مقادیر از عرض و ارتفاع فرم که به وسیله خصوصیات ScaleWidth و ScaleHeight به دست می‌آیند و تقسیم مقدار به دست آمده بر عدد ۲، محل نمایش متن را به گونه‌ای که در وسط صفحه قرار گیرد محاسبه شده‌اند. مزیت این کار این است که حتی اگر ابعاد فرم تغییر یابد و رویه مجدداً اجرا شود متن مورد نظر با توجه به ابعاد جدید باز

شده و در وسط صفحه قرار می‌گیرد.



شکل ۱۸-۱۲

۹-۳-۱۲ خصوصیت AutoRedraw

شاید تاکنون در هنگام استفاده از دستورات گرافیکی به این نکته دقت کرده‌اید که در بعضی از مواقع ترسیمات انجام شده در روی فرم از بین می‌رود یا در اثر قرار گرفتن پنجره‌های دیگر روی پنجره‌ای که ترسیمات در آن انجام شده است ترسیمات شما مخدوش می‌شود. به‌عنوان مثال یک فرم با یک دکمه فرمان به گونه‌ای طراحی کنید که با فشردن دکمه فرمان یک مستطیل توپر در روی فرم رسم شود سپس برنامه را اجرا کرده و روی دکمه فرمان کلیک کنید و بعد پنجره برنامه را به حداقل اندازه برسانید (به‌وسیله دکمه‌کنترلی Minimize) و مجدداً روی آیکن پنجره در نوار وظیفه کلیک کنید تا پنجره برنامه به حالت قبل بازگردد. در این صورت اثری از شکل رسم شده مشاهده نخواهد شد.

فرم‌ها و بعضی از کنترل‌ها مانند PictureBox خصوصیتی به نام AutoRedraw دارند که در حالت پیش‌فرض مقدار این خصوصیت False است اگر این خصوصیت روی مقدار True تنظیم شود در هنگام تغییر مکان یا تغییر اندازه فرم و نظایر آن، ترسیمات موجود از بین نمی‌روند و با نمایش مجدد فرم یا کنترل نمایش داده می‌شوند.

در مثالی که در این قسمت انجام دادید خصوصیت AutoRedraw را برای فرم برنامه روی مقدار True تنظیم کنید و برنامه را مجدداً اجرا کنید و روی دکمه فرمان کلیک کنید. فرم را به حالت Minimize درآورده و دوباره آن را روی دسکتاپ نمایش دهید. مشاهده می‌کنید در این حالت مشکل قبلی از بین رفته است.

شکل کلی استفاده از این خصوصیت به این صورت است:

`object.AutoRedraw = Boolean`

object یک مقدار اختیاری است که می‌تواند نام فرم یا کنترل باشد و در صورت عدم استفاده از آن، نام فرمی که فوکوس دارد استفاده می‌شود. مقدار Boolean یک مقدار منطقی است که می‌تواند True یا False باشد. در صورت عدم استفاده از مقدار Boolean، مقدار خصوصیت بازگشت داده می‌شود. مقدار پیش‌فرض این خصوصیت False است.

تمرین:



از خصوصیت AutoRedraw در پروژه‌هایی که تاکنون طراحی کرده‌اید استفاده کرده و نتیجه را با حالت قبل از استفاده این خصوصیت بررسی کنید.

۱۰-۳-۱۲ خصوصیت DrawMode

به وسیله این خصوصیت می‌توان شکل ظاهری ترسیماتی (از نظر رنگ) را که به وسیله متدهای گرافیکی نظیر Circle، Line، PSet و غیره انجام می‌شوند تعیین کرد. شکل کلی نحوه استفاده از این خصوصیت به صورت زیر است:

`object.DrawMode = value`

مقدار value نیز اختیاری بوده و شکل ظاهری ترسیمات را مشخص می‌کند. در صورت عدم استفاده از این مقدار، مقدار خصوصیت بازگشت داده می‌شود. مقادیری که بخش value می‌تواند کسب کند در جدول ۳-۱۲ آورده شده‌اند.

توجه داشته باشید که در جدول ۳-۱۲ منظور از رنگ pen، رنگی است که متد گرافیکی برای رسم شکل گرافیکی از آن استفاده می‌کند. در واقع این خصوصیت به ویژوال بیسیک می‌گوید که چگونه رنگ یک pixel نمایشی روی صفحه را تعیین کند به عبارت دیگر ویژوال بیسیک بر اساس رنگ فعلی pixel و رنگ نقطه‌ای که در این pixel به وسیله متد گرافیکی ایجاد می‌شود، ترکیب رنگی مناسب را با توجه به مقدار خصوصیت ScaleMode تعیین می‌کند. به عنوان مثال دستورات زیر نحوه استفاده از این خصوصیت را نشان می‌دهند:

Form1.DrawMode = vbInvert

Form1.BackColor = vbRed

Line (500 , 500) - (2000 , 2500) vbGreen, BF

در حالت عادی این دستورات یک مستطیل توپر به وسیله متد Line با رنگ سبز رسم می کنند اما چون مقدار خصوصیت DrawMode روی vbInvert تنظیم شده، از رنگ معکوس رنگ قرمز (یعنی رنگ فعلی نقاطی که مستطیل به وسیله آن‌ها نمایش داده می شود) استفاده می شود و مستطیل با این رنگ دیده خواهد شد. بنابراین اگر رنگ دیگری نیز در دستور Line استفاده شود تغییری در رنگ مستطیل ایجاد نمی شود.

جدول ۳-۱۲ مقادیر مربوط به خصوصیت DrawMode

ثابت رشته‌ای	ثابت عددی	توضیح
vbBlackness	۱	رنگ سیاه
vbNotMergePen	۲	عکس حالت 15-vbMergePen
vbMaskNotPen	۳	ترکیب رنگ بر اساس رنگ زمینه و رنگ معکوس pen
vbNotCopyPen	۴	عکس حالت 13-CopyPen
vbMaskPenNot	۵	ترکیب رنگ‌های عمومی با رنگ pen و رنگ معکوس، رنگی که نمایش داده شده است.
vbInvert	۶	رنگ معکوس، رنگی که نمایش داده شده است.
vbXorPen	۷	ترکیب رنگ‌ها بر اساس رنگ pen یا رنگی که نمایش داده شده است.
vbNotMaskPen	۸	عکس حالت 9-MaskPen
vbMaskPen	۹	ترکیب رنگ‌های عمومی با رنگ pen و رنگی که نمایش داده شد.
vbNotXorPen	۱۰	عکس حالت 7-vbXorPen
vbNop	۱۱	ترسیمی انجام نمی‌شود.
vbMergeNotePen	۱۲	ترکیب رنگ معکوس، رنگ pen و رنگی که نمایش داده شده است.
vbCopyPen	۱۳	به‌طور پیش‌فرض رنگ pen و اگر رنگ pen تعیین نشود رنگ ForeColor استفاده می‌شود.
vbMergePenNot	۱۴	ترکیب رنگ pen و رنگ معکوس، رنگی که نمایش داده شده است.
vbMergePen	۱۵	ترکیب رنگ pen و رنگی که نمایش داده شده است.
vbWhiteness	۱۶	رنگ سفید



نکته مقدارپیش فرض خصوصیت DrawMode، 13-vbCopyPen است که سبب می‌شود از رنگی که در متد گرافیکی تعیین می‌شود استفاده شود و در صورتی که رنگ توسط متد تعیین نشود از رنگی که در خصوصیت ForeColor تعیین شده استفاده شود.

۱۱-۳-۱۲ خصوصیت DrawStyle

به وسیله این خصوصیت می‌توان نوع و حالت خطوط را در ترسیمات گرافیکی تعیین کرد. شکل کلی نحوه استفاده از این خصوصیت به صورت زیر است:

object. DrawStyle = value

مقدار value اختیاری بوده و نوع و حالت ترسیمات را مشخص می‌کند. در صورت عدم استفاده از این مقدار، مقدار خصوصیت بازگشت داده می‌شود. مقادیری که value می‌تواند کسب کند در جدول ۴-۱۲ ارایه شده است.

جدول ۴-۱۲ مقادیر مربوط به خصوصیت DrawStyle

ثابت رشته‌ای	ثابت عددی	حالت نمایشی
vbSolid	۰	_____
vbDash	۱	— — — —
vbDot	۲	· · · · ·
vbDashDot	۳	— · — · —
vbDashDotDot	۴	— · · — · · —
vbInvisible	۵	خط نامرئی
vbInsideSolid	۶	لبه بیرونی حاشیه بر لبه بیرونی شکل منطبق است.

رویداد زیر، حالت‌های مختلف ترسیم را برای مقادیر صفر تا ۴ مورد استفاده قرار می‌دهد.

```
Private Sub cmdshow_Click( )
    DrawStyle = vbSolid
    Line (500, 200)-(5000, 200), vbBlue
    DrawStyle = vbDash
    Circle (900,1000), 400, vbRed
```

```
DrawStyle = vbDot
```

```
Line (2000, 500)-(5000, 1500), vbBlue, B
```

```
DrawStyle = vbDashDot
```

```
Circle (900, 2500),400, vbRed
```

```
DrawStyle = vbDashDotDot
```

```
Line (2000 ,2000)-( 5000-3000), vbBlue, B
```

```
End Sub
```



شکل ۱۹-۱۲

۱۲-۳-۱۲ خصوصیت DrawWidth

به وسیله این خصوصیت می‌توانید ضخامت خطوط را برای ترسیمات گرافیکی معین کنید. شکل کلی نحوه استفاده از این خصوصیت به صورت زیر است:

object.DrawWidth = size

مقدار size اختیاری بوده و یک عبارت عددی است که اندازه قلم را برای ترسیمات معین می‌کند و در صورت عدم استفاده از این مقدار، مقدار خصوصیت بازگشت داده می‌شود. مقدار size می‌تواند مقداری بین ۱ تا ۳۲۷۶۷ باشد.

در این رویداد حالت‌های مختلف ترسیم را با اندازه‌های مختلف قلم مشاهده می‌کنید (شکل ۲۰-۱۲).

```
Private Sub cmdshow_Click( )
```

```
DrawStyle = vbDot
```

```
DrawWidth = 1
```

```
Line (500, 150)-(5000, 150), vbBlue
```

```
DrawWidth = 2
```

Circle (900, 1000), 400, vbRed

DrawWidth = 3

Line (2000, 500)-(5000, 1500), vbBlue, B

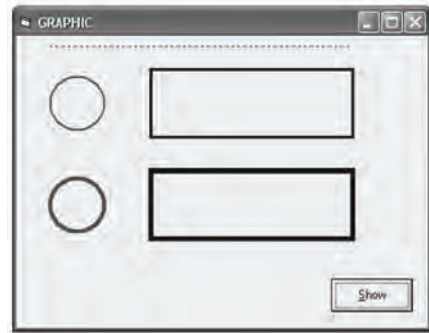
DrawWidth = 4

Circle (900, 2500), 400, vbRed


DrawWidth = 5

Line (2000, 2000)-(5000, 3000), vbBlue, B

End Sub



شکل ۲۰-۱۲

 **نکته** در صورتی که خصوصیت DrawWidth روی عدد بزرگ‌تر از یک تنظیم شود مقادیر ۱ تا ۴ برای خصوصیت DrawStyle در زمان اجرای متدهای گرافیکی عملکردی از خود نشان نمی‌دهند و مانند مقدار vbSolid عمل می‌کنند.

۱۳-۳-۱۲ خصوصیت FillStyle

به وسیله این خصوصیت می‌توانید ترسیماتی نظیر دایره، بیضی و مستطیل را با حالت‌های مختلف پر کنید. شکل کلی نحوه استفاده از این خصوصیت به صورت زیر است:

object.FillStyle = number

مقدار number اختیاری بوده و عددی صحیح است که حالت موردنظر را برای پر کردن ترسیمات معین می‌کند. در صورت عدم استفاده از این بخش، مقدار خصوصیت بازگشت داده می‌شود. مقادیر مجاز برای number در جدول ۵-۱۲ ارایه شده است:

جدول ۵-۱۲ مقادیر مربوط به خصوصیت FillStyle

ثابت رشته‌ای	ثابت عددی	توضیح
vbFsWithSolid	۰	داخل شکل با رنگی که در خصوصیت FillColor تعیین شده پر می‌شود.
vbFsWithTransparent	۱	داخل شکل با رنگ زمینه پر می‌شود.
vbFsWithHorizontalLine	۲	داخل شکل با خطوط افقی پر می‌شود.
vbFsWithVerticalLine	۳	داخل شکل با خطوط عمودی پر می‌شود.
vbFsWithUpwardDiagonal	۴	داخل شکل با خطوط مایل پر می‌شود (از چپ به راست).
vbFsWithDownwardDiagonal	۵	داخل شکل با خطوط مایل پر می‌شود (از راست به چپ).
vbFsWithCross	۶	داخل شکل با خطوط عمودی و افقی پر می‌شود (حالت شطرنجی).
vbFsWithDiagonalCross	۷	داخل شکل با خطوط عمودی و افقی مایل پر می‌شود (حالت شطرنجی مایل).

در این رویداد حالت‌های مختلف ترسیم را برای مقادیر متفاوتی از خصوصیت FillStyle نشان می‌دهد:

```
Private Sub cmdshow_Click()
    DrawStyle = vbSolid
    DrawWidth = 1
    FillColor = vbGreen
    FillStyle = 0
    Circle (900, 1000), 400, vbRed
    FillColor = vbBlack
    FillStyle = 1
    Line (2000, 500)-(5000, 1500), vbBlue, B
    FillStyle = 2
    Circle (900, 2200), 400, vbRed
    FillStyle = 3
    Line (2000, 1700)-(5000, 2700), vbBlue, B
    FillStyle = 4
    Circle (900, 3500), 400, vbRed
    FillStyle = 5
```

Line (2000, 3000)-(5000, 4000), vbBlue, B

FillStyle = 6

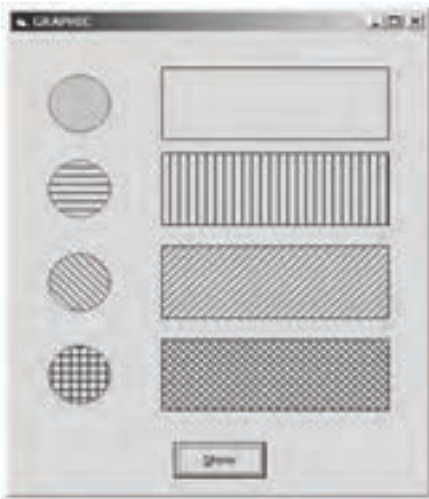
Circle (900, (4800, 400, vbRed

FillStyle = 7

Line (2000, 4300)-(5000, 5300), vbBlue, B

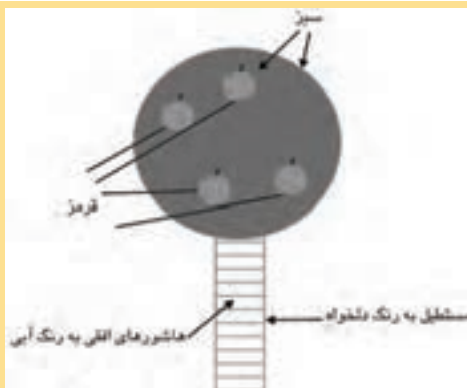
End Sub

در این رویداد پس از تعیین مقادیر مورد نظر برای خصوصیات DrawStyle و DrawWidth، مقدار خصوصیت FillColor روی سبز تنظیم شده است. خصوصیت FillColor رنگ قلم را برای ترسیماتی که به وسیله خصوصیت FillStyle ایجاد می‌شود معین می‌کند.



شکل ۲۱-۱۲

به عنوان مثال در دایره اول چون مقدار خصوصیت FillStyle صفر است دایره با رنگ FillColor یعنی سبز پر می‌شود. در مورد سایر مقادیر FillStyle نیز خصوصیت FillColor رنگ خطوط عمودی، افقی، مایل و غیره که شکل را پر می‌کند تعیین می‌کند. نتیجه اجرای این رویداد را می‌توانید در شکل ۲۱-۱۲ مشاهده کنید.



شکل ۲۲-۱۲

تمرین: 

پروژه‌ای طراحی کنید که این شکل را روی یک فرم نمایش دهد:

۴-۱۲ تابع QBcolor

این تابع با دریافت یک عدد بین صفر و ۱۵، یک عدد از نوع Long را که بیانگر رنگ معادل عدد دریافتی است بازمی‌گرداند. شکل کلی این تابع به صورت زیر است:

QBColor (color)

آرگومان color یک عدد از نوع صحیح است که براساس جدول ۶-۱۲ قابل استفاده خواهند بود.

جدول ۶-۱۲ مقادیر قابل استفاده برای آرگومان color

مقدار عددی	رنگ	مقدار عددی	رنگ
۰	سیاه	۸	خاکستری
۱	آبی	۹	آبی روشن
۲	سبز	۱۰	سبز روشن
۳	فیروزه‌ای	۱۱	فیروزه‌ای روشن
۴	قرمز	۱۲	قرمز روشن
۵	بنفش	۱۳	بنفش روشن
۶	زرد	۱۴	زرد روشن
۷	سفید	۱۵	سفید روشن

به عنوان مثال در دستورات زیر با استفاده از تابع QBColor، ۵۰۰۰ نقطه با رنگ‌های متفاوت نمایش داده خواهد شد.

```
Dim i As Integer
```

```
Dim XPos As Single, YPos As Single
```

```
DrawWidth = 4
```

```
Randomize
```

```
For i = 1 To 5000
```

```
    XPos = Rnd * ScaleWidth
```

```
    YPos = Rnd * ScaleHeight
```

```
    PSet (XPos, YPos), QBColor(Rnd * 15)
```

```
Next i
```


۵-۱۲ تابع RGB

در ویژوال بیسیک تابع دیگری به نام RGB وجود دارد که می‌تواند ترکیبات رنگی را با توجه به نیاز ایجاد کند. این تابع می‌تواند با دریافت سه مقدار عددی برای سه رنگ اصلی تمام ترکیبات مورد نظر را ایجاد کند. شکل کلی این تابع به صورت زیر است:

RGB (red , green , blue)

این تابع سه آرگومان اجباری دارد که می‌توانند اعداد صحیح از صفر تا ۲۵۵ را کسب کنند. آرگومان red مقدار رنگ قرمز، آرگومان green مقدار رنگ سبز و آرگومان blue مقدار رنگ آبی را معین می‌کنند. مقدار بازگشتی این تابع یک عدد از نوع Long است که بیانگر ترکیب رنگی درخواستی است. مقادیر سه آرگومان فوق برای رنگ‌های استاندارد در جدول ۷-۱۲ ارائه شده است.

جدول ۷-۱۲ مقادیر سه رنگ اصلی برای رنگ‌های استاندارد

رنگ	مقدار آرگومان red	مقدار آرگومان green	مقدار آرگومان blue
سیاه	۰	۰	۰
آبی	۰	۰	۲۵۵
سبز	۰	۲۵۵	۰
فیروزه‌ای	۰	۲۵۵	۲۵۵
قرمز	۲۵۵	۰	۰
بنفش	۲۵۵	۰	۲۵۵
زرد	۲۵۵	۲۵۵	۰
سفید	۲۵۵	۲۵۵	۲۵۵

به عنوان مثال دستورات زیر پانصد دایره با ابعاد و مختصات و رنگ‌های تصادفی ایجاد می‌کند.

```
Dim i As Integer
```

```
Dim XPos As Single, YPos As Single
```

```
DrawWidth = 4
```

```
Randomize
For i = 1 To 500
  XPos = Rnd * ScaleWidth
  YPos = Rnd * ScaleHeight
  (Circle (XPos, YPos), Rnd * 800, RGB(Rnd * _255, Rnd * 255, Rnd * 255)
Next i
```

۶-۱۲ شی چاپگر (Printer Object)

تاکنون کلیه عملیاتی که انجام داده‌اید روی فرم و صفحه نمایش انجام شده است اما گاهی اوقات لازم است تا اطلاعات مورد نیاز خود را به وسیله چاپگر روی کاغذ چاپ کنید. ویژوال بیسیک در این زمینه نیز امکانات لازم را مهیا کرده است. شما با استفاده از شی چاپگر، علاوه بر انجام عملیات چاپ می‌توانید به وسیله خصوصیات این شی عملیات چاپ را به نحو مناسبی مدیریت کنید. در این بخش به معرفی متدهای چاپ و معرفی خصوصیات شی چاپگر می‌پردازیم.

۱-۶-۱۲ متدهای چاپ

تاکنون متدهای مختلفی را برای نمایش اطلاعات و ترسیمات آموختید. در این متدها می‌توانید به جای پارامتر object از شی چاپگر استفاده کنید؛ بنابراین به آسانی می‌توانید از متدهای Circle، Line، PSet، Scale، TextHeight، TextWidth و Print استفاده کنید. فقط کافی است برای معرفی شی چاپگر از کلمه Printer استفاده کنید. به‌عنوان مثال نتایج حاصل از اجرای دستورات زیر روی کاغذ چاپ خواهند شد.

```
Printer.Print " IN THE NAME OF GOD. "
```

```
(Printer.Line (100,100) - (800,800
```

```
Printer.Circle (150,150), 500
```

متد EndDoc

این متد سبب می‌شود تا عملیات چاپ متوقف شده و تا زمانی که چاپگر آماده چاپ شود اطلاعات مربوط به چاپ، روی دیسک یا حافظه کامپیوتر ذخیره می‌شود. شکل کلی

نحوه استفاده از این متد به صورت زیر است:

Printer.EndDoc

متد KillDoc

این متد می‌تواند در زمان چاپ اطلاعات، عملیات چاپ را خاتمه دهد. شکل کلی نحوه استفاده از این متد به صورت زیر است:

Printer.KillDoc

متد NewPage

متد Newpage می‌تواند عملیات چاپ صفحه جاری را خاتمه داده و چاپگر، چاپ را از صفحه بعدی انجام دهد. شکل کلی نحوه استفاده از این متد در ادامه می‌آید:

Printer.NewPage

۲-۶-۱۲ خصوصیات شیء چاپگر

تنظیمات مربوط به چاپگرها نیز مانند اشیای دیگر به وسیله تعدادی از خصوصیات قابل دستیابی و تغییر هستند. در این بخش مهم‌ترین خصوصیات شیء چاپگر را مورد بررسی قرار می‌دهیم. بعضی از خصوصیات نیز قبلاً توضیح داده شده‌اند مانند: DrawMode, Style, CurrentY, CurrentX, FillStyle, FillColor, DrawWidth, خصوصیات مربوط به قلم‌ها (Fonts) و ...

خصوصیت ColorMode

به وسیله این خصوصیت می‌توان نوع چاپگر را از نظر چاپ رنگی یا سیاه سفید تعیین کرد. شکل کلی نحوه استفاده از این خصوصیت به صورت زیر است:

Printer.ColorMode = value

value یک ثابت عددی یا رشته‌ای است که نوع چاپ را معین می‌کند. این مقدار می‌تواند یکی از مقادیر موجود در جدول ۸-۱۲ باشد. در صورت عدم استفاده از بخش value، مقدار فعلی خصوصیت بازگشت داده خواهد شد.

جدول ۸-۱۲ مقادیر مربوط به خصوصیت ColorMode

توضیح	ثابت عددی	ثابت رشته‌ای
چاپ سیاه سفید	۱	vbPRCMMonochrome
چاپ رنگی	۲	vbPRCMColor



نکته استفاده از چاپ رنگی یا سیاه سفید به امکانات چاپگر بستگی دارد.

خصوصیت Copies

به وسیله این خصوصیت می‌توانید تعداد نسخه‌هایی که چاپگر چاپ می‌گیرد تعیین کنید. شکل کلی نحوه استفاده از این خصوصیت به صورت زیر است:

Printer.Copies = number

Number یک عبارت عددی از نوع صحیح است که تعداد نسخه‌ها را برای چاپ معین می‌کند و در صورت عدم استفاده از آن، مقدار فعلی خصوصیت بازگشت داده خواهد شد.

خصوصیت DeviceName

این خصوصیت نام دستگاه چاپگر پیش فرض را باز می‌گرداند. نام چاپگرها در زمان نصب آن‌ها از طریق برنامه Control Panel توسط کاربر تعیین می‌شود. مثلاً اگر یک چاپگر EPSON LQ 300 را با نام myprinter و به صورت پیش فرض نصب کرده باشید فرمان زیر نام چاپگر یعنی myprinter را نمایش می‌دهد.

Print Printer.DeviceName

خصوصیت DriverName

این خصوصیت نام راه‌انداز (driver) دستگاه چاپگر پیش فرض را باز می‌گرداند. به عنوان مثال اگر چاپگر EPSON LQ 500 را با نام myprinter نصب کرده باشید فرمان زیر نام راه‌انداز نصب شده یعنی EPSON LQ 500 را نمایش خواهد داد.

Print Printer.DriverName

خصوصیت Orientation

به وسیله این خصوصیت می‌توانید جهت انجام عملیات چاپ را روی صفحه کاغذ تعیین کنید. عملیات چاپ می‌تواند به صورت portrait یا landscape باشد. در شکل ۲۳-۱۲ تفاوت این دو حالت نمایش داده شده است.



شکل ۲۳-۱۲

شکل کلی نحوه استفاده از این خصوصیت به صورت زیر است:

Printer.Orientation = value

value یک ثابت عددی یا رشته‌ای است که جهت چاپ اطلاعات را روی کاغذ معین می‌کند. این مقدار می‌تواند یکی از مقادیر موجود در جدول ۹-۱۲ باشد. در صورت عدم استفاده از مقدار value، مقدار فعلی خصوصیت بازگشت داده خواهد شد.

جدول ۹-۱۲ مقادیر مربوط به خصوصیت Orientation

ثابت رشته‌ای	ثابت عددی	توضیح
vbPRORPortrait	۱	چاپ به صورت portrait
vbPRORLandscape	۲	چاپ به صورت landscape

خصوصیت Page

این خصوصیت شماره صفحه در حال چاپ را در اختیار برنامه قرار می‌دهد. شکل کلی نحوه استفاده از این خصوصیت به این صورت است:

Printer.Page

خصوصیت PaperSize

به وسیله این خصوصیت می‌توانید نوع و ابعاد کاغذ چاپ را تنظیم کنید. شکل کلی نحوه استفاده از این خصوصیت به صورت زیر است:

Printer.PaperSize = value

value یک ثابت عددی یا رشته‌ای است که ابعاد کاغذ را معین می‌کند. این مقدار می‌تواند یکی از مقادیر موجود در جدول ۱۰-۱۲ باشد. در صورت عدم استفاده از مقدار value، مقدار فعلی خصوصیت بازگشت داده خواهد شد.

جدول ۱۰-۱۲ مقادیر مربوط به اندازه کاغذ در خصوصیت PageSize

ثابت رشته‌ای	ثابت عددی	ابعاد کاغذ
vbPRPSLetter	۱	$8 \frac{1}{2} \times 11$ Inch
vbPRPSLetterSmall	۲	$8 \frac{1}{4} \times 11$ Inch (اندازه کوچک)
vbPRPSTabloid	۳	11×17 Inch
vbPRPSLedger	۴	17×11 Inch
vbPRPSLegal	۵	$8 \frac{1}{2} \times 14$ Inch
vbPRPSStatement	۶	$5 \frac{1}{2} \times 8$ Inch
vbPRPSExecutive	۷	$7 \frac{1}{2} \times$ Inch
vbPRPSA3	۸	A3 (۲۹۷× mm)
vbPRPSA4	۹	A4 (×۲۹۷ mm)
vbPRPSA۴Small	۱۰	A4 (اندازه کوچک)
vbPRPSA5	۱۱	A5 (۱۴۸×mm)

خصوصیت Port

به وسیله این خصوصیت می‌توان نام پورت مربوط به چاپگری را که چاپ به آن ارسال می‌شود به دست آورد. شکل کلی نحوه استفاده از این خصوصیت به این صورت است:

Printer.Port

چاپگرها معمولاً از پورت‌های موازی LPT1 و LPT2 استفاده می‌کنند. 

خصوصیت PrintQuality

این خصوصیت کیفیت وضوح چاپ را در چاپگر معین می‌کند. شکل کلی نحوه استفاده از این خصوصیت به صورت زیر است:

Printer.PrintQuality = value

value یک ثابت عددی یا رشته‌ای است که میزان وضوح چاپ را معین می‌کند در صورت عدم استفاده از مقدار value مقدار فعلی خصوصیت بازگشت داده خواهد شد.

مقدار value می‌تواند یکی از مقادیر موجود در جدول ۱۱-۱۲ باشد.

جدول ۱۱-۱۲ مقادیر مربوط به کیفیت چاپ در خصوصیت PrintQuality

ثابت رشته‌ای	ثابت عددی	توضیح
vbPRPQDraft	-۱	چاپ با کیفیت Draft
vbPRPQLow	-۲	چاپ با کیفیت پایین
vbPRPQMedium	-۳	چاپ با کیفیت متوسط
vbPRPQHigh	-۴	چاپ با کیفیت بالا

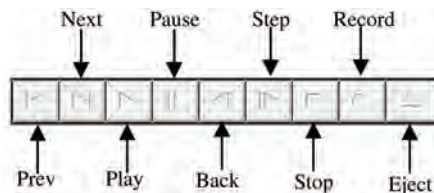
۱۲-۷ چند رسانه‌ای (Multimedia)

یکی از ویژگی‌های دیگر زبان برنامه‌نویسی ویژوال بیسیک توانایی استفاده از امکانات صوتی و تصویری در آن است. با استفاده از کنترل‌های ارائه شده در زبان برنامه‌نویسی ویژوال بیسیک می‌توان انواع فایل‌های ویدئویی، صوتی و موسیقی را مورد استفاده قرار داد.

۱۲-۷-۱ کنترل MCI (Media Control Interface)

با استفاده از این کنترل می‌توان انواع فایل‌های صدا، موسیقی، ویدئویی و سی‌دی‌های صوتی را پخش کرد.

این کنترل به‌طور پیش‌فرض در جعبه ابزار مشاهده نمی‌شود. برای اضافه کردن این کنترل گزینه Components را از منوی Project انتخاب کنید سپس در کادر محاوره Components که نمایش داده می‌شود کادر علامت Microsoft Multimedia Control 6.0 را انتخاب کنید و روی دکمه OK کلیک کنید. در این مرحله کنترل MCI به کنترل جعبه ابزار اضافه می‌شود. کنترل MCI شامل ۹ دکمه است که وظایف مختلفی را به عهده دارند. عملکرد این دکمه‌ها در شکل ۱۲-۲۴ و مطابق جدول ۱۲-۱۴ قابل مشاهده است.



شکل ۱۲-۲۴



مثال ۳: می‌خواهیم پروژه‌ای طراحی کنیم که به وسیله آن بتوان فایل‌های ویدئویی

از نوع avi را مشاهده کرد برای این کار عملیات بعد را به ترتیب انجام دهید:

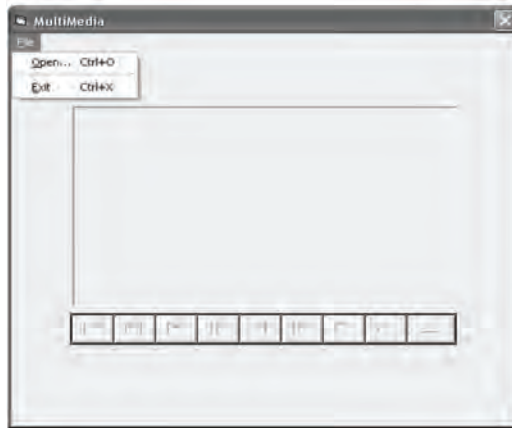
۱- یک پروژه از نوع StandardEXE ایجاد کنید سپس خصوصیت‌های فرم را مطابق جدول

۱۲-۱۲ تنظیم کنید.

جدول ۱۲-۱۲ خصوصیات فرم

مقدار	خصوصیت
frmmultimedia	Name
Multimedia	Caption

۲- یک کنترل منو و کادر تصویر با نام PicVideo مطابق شکل ۱۲-۲۵ روی فرم قرار دهید.



شکل ۱۲-۲۵

۳- کنترل MCI را به جعبه ابزار اضافه کنید. در جعبه ابزار روی آیکن MMControl

دابل کلیک کنید تا کنترل MCI روی فرم قرار گیرد، سپس خصوصیت Name آن را

روی مقدار mcivideo و اندازه و موقعیت آن را مطابق شکل ۱۲-۲۵ تنظیم نمایید.

۴- به ماژول فرم بروید و دستورات زیر را در رویداد Load فرم تایپ کنید.

```
mcivideo.DeviceType = "aviVideo"
```

```
mcivideo.hWndDisplay = picvideo.hWnd
```


دستور اول با استفاده از خصوصیت DeviceType کنترل MCI را برای نمایش فایل‌ها با قالب‌بندی از نوع AVI آماده می‌کند. با استفاده از این خصوصیت می‌توان انواع رسانه‌های مالتی‌مدیا را برای استفاده تنظیم کرد. این خصوصیت می‌تواند مقادیر ارائه‌شده در جدول ۱۲-۱۳ را کسب کند.

جدول ۱۲-۱۳

مقدار	توضیح
AVIVideo	پخش فایل‌های صوتی تصویری AVI
CDAudio	پخش سی‌دی صوتی
DigitalVideo	پخش فایل‌های ویدئویی دیجیتال
Videodisc	پخش دیسک‌های ویدئویی
WaveAudio	پخش فایل‌های صوتی از نوع WAV

دستور دوم در این رویداد نیز سبب می‌شود تصویر فایل ویدئویی در کنترل جعبه تصویر Picvideo نمایش داده شود.

۵ - رویداد کلیک گزینه mnuopen را به صورت زیر تنظیم کنید:

```
Private Sub mnuopen_Click()
    Dim strpath As String
    strpath = InputBox("Enter Path and Filename :", "Input Data")
    mcivideo.FileName = strpath
    mcivideo.Command = "Open"
End Sub
```

در این رویداد ابتدا با استفاده از یک کادر ورود داده، مسیر و نام فایل ویدئویی دریافت شده و در متغیر strpath ذخیره می‌شود سپس محتویات strpath در خصوصیت Filename کنترل MCI قرار می‌گیرد تا کنترل MCI بتواند به فایل ویدئویی دسترسی پیدا کند سپس خصوصیت Command کنترل MCI روی مقدار Open تنظیم می‌شود تا کنترل آماده پخش فایل ویدئویی شود. این خصوصیت می‌تواند فرمان‌های مختلفی را برای اجرا در کنترل مشخص کند. این فرمان‌ها در جدول ۱۴-۱۲ قابل مشاهده است.

جدول ۱۴-۱۲

مقدار	توضیح
Open	ابزار مورد نیاز برای کنترل MCI را باز می کند.
Close	ابزار مورد نیاز برای کنترل MCI را می بندد.
Play	فایل ویدئویی، صدا یا موسیقی را پخش می کند.
Pause	عملیات پخش یا ضبط را موقتاً قطع می کند.
Stop	عملیات ضبط یا پخش را قطع می کند.
Back	از موقعیت جاری کمی به عقب حرکت می کند.
Step	از موقعیت جاری کمی به جلو حرکت می کند.
Prev	به ابتدای شیار (Track) جاری حرکت می کند. اگر ۳ ثانیه از شیار جاری پخش شده باشد به ابتدای شیار قبلی باز می گردد.
Next	یک ترک به جلو حرکت می کند.
Record	عملیات ضبط را آغاز می کند.
Eject	سی دی را به درایو وارد یا از آن خارج می کند.

۶ - پروژه و فرم را با نام playvideo ذخیره کرده سپس برنامه را اجرا کنید.

۷ - روی گزینه Open از منوی File کلیک کرده، نام و مسیر فایل ویدئویی خود را که دارای پسوند avi است در کادر ورود داده تایپ کنید و روی دکمه OK کلیک نمایید تا به فرم برنامه بازگردید.

۸ - در این مرحله بعضی از دکمه های کنترل MCI فعال می شوند روی دکمه پخش (Play) کلیک کنید تا فایل ویدئویی مورد نظرتان در کنترل جعبه تصویر نمایش داده شود سپس عملکرد سایر دکمه های کنترل MCI را بررسی نمایید.

۹ - از برنامه خارج شده و به پنجره ویژوال بیسیک بازگردید.


تمرین:

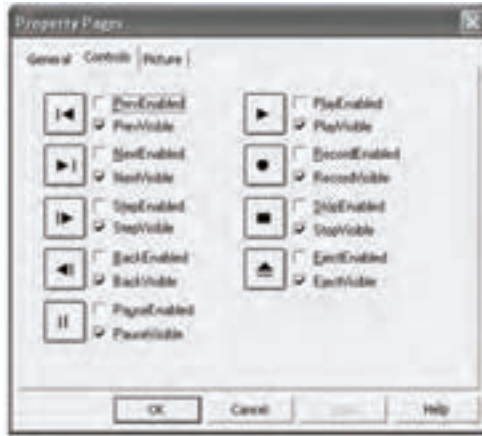


پروژه playvideo را به گونه ای تغییر دهید که به وسیله آن بتوان فایل های صوتی با پسوند WAV را پخش کرد.

نکته در کنترل MCI با استفاده از خصوصیت Orientation می توان کنترل را به صورت افقی (mciOrientHorz) یا عمودی (mciOrientVert) تنظیم نمود.



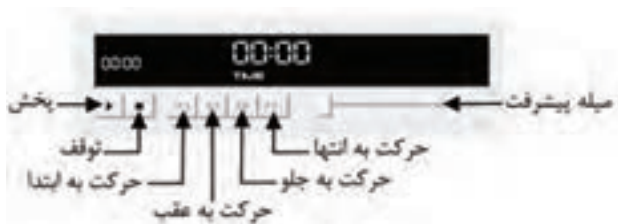
نکته  کلیه دکمه‌های موجود در کنترل MCI را می‌توان با استفاده از صفحه‌های خصوصیت (Property Pages) مرئی یا مخفی، فعال یا غیرفعال نمود. به منظور دسترسی به صفحه‌های خصوصیت در پنجره خصوصیات روی دکمه **...** روبه‌روی خصوصیت Custom کلیک کنید سپس زبانه Controls را انتخاب کنید (شکل ۱۲-۲۶).



شکل ۱۲-۲۶

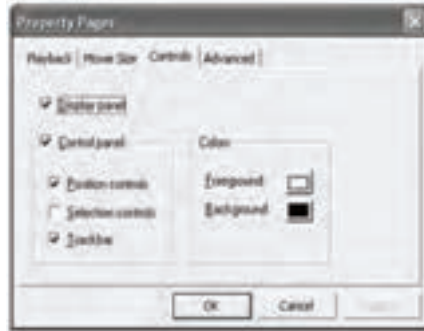
۱۲-۷-۲ کنترل Microsoft Active Movie (MAM)

با استفاده از این کنترل می‌توان انواع فایل‌های صوتی و موسیقی مانند WAV و MIDI را پخش کرد و فایل‌های تصویری مانند AVI، MPEG و MOV را نمایش داد. این کنترل به طور عادی در جعبه ابزار مشاهده نمی‌شود. برای اضافه کردن این کنترل گزینه Components را از منوی Project انتخاب کنید سپس در کادر محاوره Components که نمایش داده می‌شود کادر علامت Microsoft Active Movie Control را انتخاب کنید و روی دکمه OK کلیک کنید تا کنترل MAM به جعبه ابزار اضافه شود. این کنترل نیز مانند کنترل MCI دکمه‌های متعددی را شامل می‌شود که وظایف متفاوتی را بر عهده دارند (شکل ۱۲-۲۷).



شکل ۱۲-۲۷

در این کنترل به طور پیش فرض فقط دکمه‌های پخش (Play) و توقف (Stop) قابل استفاده می‌باشد. به منظور دسترسی به سایر دکمه‌ها می‌توانید پس از انتخاب کنترل، در پنجره خصوصیات روی دکمه **...** روبه‌روی خصوصیت Custom کلیک کنید. در کادر محاوره Property Pages روی زبانه Controls کلیک کنید (شکل ۲۸-۱۲). با استفاده از کادرهای علامت موجود در این بخش می‌توان دکمه‌ها و سایر بخش‌های کنترل را مخفی یا قابل مشاهده کرد.




شکل ۲۸-۱۲

مثال ۴: می‌خواهیم پروژه‌ای طراحی کنیم که به وسیله آن بتوان فایل‌های صوتی، موسیقی و ویدئویی را پخش کرد. برای این کار عملیات زیر را به ترتیب انجام دهید:

- ۱ - یک پروژه از نوع Standard EXE ایجاد کنید. خصوصیت‌های فرم را مطابق جدول ۱۵-۱۲ تنظیم کنید.

جدول ۱۵-۱۲

مقدار	خصوصیت
frmmpeg	Name
Movie	Caption

- ۲ - یک کنترل منو مطابق شکل ۲۹-۱۲ روی فرم قرار دهید سپس کنترل MAM را به جعبه ابزار اضافه کنید. در جعبه ابزار روی آیکن Active Movie  دابل کلیک کنید تا کنترل روی فرم قرار گیرد سپس خصوصیت Name آن را روی mammovie تنظیم کنید.
- ۳ - کنترل mammovie را انتخاب کرده و خصوصیت AutoStart آن را روی مقدار True تنظیم کنید تا در زمانی که فایل صوتی یا تصویری باز می‌شود کنترل MAM آن را پخش کند.



شکل ۲۹-۱۲

- ۴ - خصوصیت ShowPositionControls را در کنترل mammovie روی مقدار True تنظیم کنید تا چهار دکمه دیگر نیز قابل مشاهده و استفاده شوند.
- ۵ - رویداد Click گزینه Open را به صورت زیر تنظیم کنید:

```
Private Sub mnuopen_Click()
```

```
Dim strpath As String
```

```
strpath = InputBox("Enter Path and Filename :", "Input Data")
```

```
mammovie.FileName = strpath
```

```
End Sub
```

در این رویداد پس از دریافت نام و مسیر فایل موسیقی یا ویدئویی و ذخیره‌سازی آن در متغیر strpath، محتویات این متغیر در دستور سوم محتویات این متغیر در خصوصیت FileName کنترل mammovie قرار داده می‌شود تا آن را پخش کند.

۶ - پروژه و فرم را با نام activemovie ذخیره کرده سپس برنامه را اجرا کنید.

۷ - با استفاده از گزینه Open در منوی File، مسیر و نام یک فایل mpg را تعیین کنید. سپس در هنگام پخش فایل مورد نظرتان، عملکرد دکمه‌ها و سایر بخش‌ها در کنترل mammovie را بررسی نمایید.

۸ - از برنامه خارج شده و به پنجره ویژوال بیسیک بازگردید.

تمرین:



پروژه activemovie را برای انواع دیگر فایل‌ها آزمایش کنید.

۸-۱۲ شی تصویر (Picture)

با استفاده از شی تصویر نیز می‌توانید یک تصویر را روی فرم یا در کنترل کادر تصویر نمایش دهید. به منظور ایجاد یک شی تصویر از نوع داده Picture استفاده می‌شود. با شی تصویر می‌توان انواع فایل‌های گرافیکی مانند BMP، GIF، ICO و JPG را نمایش داد.



مثال ۵: می‌خواهیم رویه‌ای بنویسیم که بتواند یک تصویر را با استفاده از شی تصویر نمایش دهد.

به این منظور می‌توان از یک شی تصویر همراه با تابع LoadPicture استفاده کرد به این ترتیب رویه به صورت زیر تعریف می‌شود:

```
Public Sub showpicture(strpath As String)
```

```
Dim objpicture As Picture
```

```
Set objpicture = LoadPicture(strpath)
```

```
Set picshow.Picture = objpicture
```

```
End Sub
```

این رویه با یک آرگومان (strpath) برای دریافت مسیر و نام فایل تعریف شده است سپس داخل رویه، یک شی از نوع Picture با نام objpicture با دستور Dim تعریف شده است در مرحله بعد با استفاده از دستور Set و تابع LoadPicture تصویر موردنظر در شی تصویر objpicture بارگذاری می‌شود و در پایان با دستور شی تصویر در کنترل کادر تصویر picshow نمایش داده می‌شود.

برای فراخوانی رویه ShowPicture می‌توان به صورت زیر عمل نمود:

```
Call showpicture (" C:\WinXP\Coffee Bean.Bmp")
```

اگر لازم باشد تصویر روی فرم نمایش داده شود می‌توان از متد PaintPicture استفاده کرد. به عنوان مثال دستور سوم در رویه showpicture را می‌توان به این صورت تغییر داد تا تصویر روی فرم نمایش داده شود.

```
PaintPicture objpic, 20, 50, 3000, 4000
```

این دستور شی تصویر را در مختصات $X = 20$ و $Y = 50$ با عرض 3000 و ارتفاع 4000 روی فرم نمایش می‌دهد.



• **نکته** • از متد PaintPicture می‌توان در کنترل کادر تصویر و شیء چاپگر نیز استفاده کرد.
 • متد LoadPicture از انواع فایل‌های گرافیکی مانند BMP، ICO، GIF، JPG و ... پشتیبانی می‌کند.

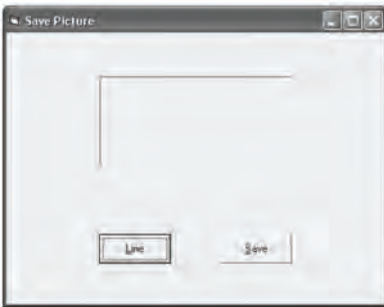
تمرین:



پروژه‌ای طراحی کنید که با استفاده از آن بتوان هر تصویر دلخواهی را با استفاده از یک کادر محاوره Open مشاهده کرد به علاوه امکان بزرگ یا کوچک کردن تصویر موردنظر نیز وجود داشته باشد.



مثال ۶: می‌خواهیم پروژه‌ای طراحی کنیم که به وسیله آن بتوان ترسیمات رسم شده روی فرم را در یک فایل ذخیره کرد. برای این کار عملیات زیر را به ترتیب انجام دهید:
 ۱ - یک پروژه از نوع Standard EXE ایجاد کنید و فرم و کنترل‌های آن را مطابق شکل ۱۲-۳۰ و جدول ۱۲-۱۶ و ۱۲-۱۷ تنظیم کنید.



شکل ۱۲-۳۰

جدول ۱۲-۱۶ خصوصیات فرم

خصوصیت	مقدار
Name	frmdrawing
Caption	Save Picture

جدول ۱۲-۱۷ خصوصیات کنترل‌ها

کنترل / خصوصیت	Command Button	Command Button	Picture Box
Name	cmddrawline	cmdsave	picdrawing
Caption	&Line	&Save	_____

۲ - رویداد Click دکمه Line را به صورت زیر تنظیم کنید.

Private Sub cmddrawline_Click()

ScaleMode = vbPixels

AutoRedraw = True

Line (10, 10)-(200, 200), vbRed

End Sub

در این رویداد عملیات رسم یک خط انجام می‌گیرد.
۳ - دستور زیر را در رویداد Click دکمه Save تایپ کنید:

SavePicture Image, "C:\drawing.bmp"

رویه فرعی SavePicture می‌تواند ترسیمات انجام شده روی فرم یا کنترل کادر تصویر را در یک فایل گرافیکی از نوع BMP ذخیره کند. آرگومان اول در این رویه فرعی خصوصیت Image فرم یا کنترل کادر تصویری است که ترسیمات روی آن انجام شده است و آرگومان دوم مسیر و نام فایلی است که ترسیمات در آن ذخیره می‌شود.

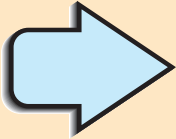
۴ - پروژه و فرم را با نام savepicture ذخیره کرده سپس آنرا اجرا کنید و ابتدا روی دکمه Line و سپس Save کلیک کنید. در پایان به مسیر: C بروید و فایل drawing.bmp را با برنامه نقاشی ویندوز باز کرده و محتویات آنرا بررسی کنید.

۵ - از برنامه خارج شوید و به پنجره ویژوال بیسیک بازگردید.

تمرین:



پروژه‌ای طراحی کنید که به وسیله آن بتوان هر نوع فایل گرافیکی مانند GIF، JPG، CUR، ICO و ... را در فایل جدیدی با قالب بندی BMP ذخیره کرد.



Learn in English

Circle Method

Draws a circle, ellipse, or arc on an object.

Syntax

object.**Circle** [*Step*] (x, y), radius, [*color, start, end, aspect*]

The **Circle** method syntax has the following object qualifier and parts:

Part	Description
object	Optional. Object expression that evaluates to an object in the Applies To list. If object is omitted, the Form with the focus is assumed to be object.
Step	Optional. Keyword specifying that the center of the circle, ellipse, or arc is relative to the current coordinates given by the CurrentX and CurrentY properties of object.
(x, y)	Required. Single values indicating the coordinates for the center point of the circle, ellipse, or arc. The ScaleMode property of object determines the units of measure used.
radius	Required. Single value indicating the radius of the circle, ellipse, or arc. The ScaleMode property of object determines the unit of measure used.
color	Optional. Long integer value indicating the RGB color of the circle's outline. If omitted, the value of the ForeColor property is used. You can use the RGB function or QBColor function to specify the color.
start, end	Optional. Single-precision values. When an arc or a partial circle or ellipse is drawn, start and end specify (in radians) the beginning and end positions of the arc. The range for both is -2π radians to 2π radians. The default value for start is \cdot radians; the default for end is $2 * \pi$ radians.
aspect	Optional. Single-precision value indicating the aspect ratio of the circle. The default value is 1.0, which yields a perfect (non-elliptical) circle on any screen.

واژه نامه	
Circle	دایره
Component	جزء
Current	جاری
Draw	رسم کردن، کشیدن
Line	خط
Movie	فیلم
Multimedia	چند رسانه‌ای
Orientation	جهت
Point	نقطه
Quality	کیفیت
Radius	شعاع دایره
Scale	مقیاس
Style	سبک، روش

خلاصه مطالب

- ترسیمات گرافیکی را می‌توان روی فرم یا کنترل PictureBox یا چاپگر ایجاد کرد.
- به وسیله خصوصیت ScaleMode می‌توان مقیاس را در محورهای مختصات تنظیم کرد.
- به وسیله خصوصیت ScaleLeft و ScaleTop می‌توان مختصات نقطه مبنا را تنظیم کرد.
- خصوصیات ScaleHeight و ScaleWidth مقیاس را می‌توان در سیستم مختصات فرم یا کنترل PictureBox تنظیم کرد.
- به وسیله خصوصیت Scale نیز می‌توان مختصات نقطه مبنا و مقیاس سیستم مختصات را در فرم یا کنترل مورد نظر تنظیم کرد.
- به وسیله متد PSet می‌توان هر نقطه دلخواهی را در موقعیت مورد نظر ترسیم کرد.
- به وسیله متد Line می‌توان انواع خطوط مستطیل و مستطیل توپر را در مکان مورد نظر رسم کرد.
- متد Circle می‌تواند انواع دایره، بیضی یا کمانی از دایره و یا بیضی را رسم کند.
- به وسیله متد Point می‌توان رنگ یک نقطه را به دست آورد. این متد رنگ نقطه مورد نظر را به صورت یک عدد از نوع Long باز می‌گرداند.
- متدهای CurrentX و CurrentY موقعیت جاری مکان‌نما را در صفحه ترسیمات معین می‌کنند. خصوصیت CurrentX موقعیت جاری مکان‌نما را در محور X و CurrentY موقعیت جاری مکان‌نما را در محور Y معین می‌کنند.

- متد `Cls` می‌تواند صفحه ترسیمات را پاک کند.
- به وسیله متد `Print` هر نوع عبارت، می‌توان مقدار متغیرها و خصوصیات را نمایش داد.
- با استفاده از تابع `SpC` در متد `Print` می‌توان به تعداد مورد نظر فضای خالی ایجاد کرد.
- با استفاده از تابع `Tab` در متد `Print` می‌توان اطلاعات نمایشی را در ستون مورد نظر نمایش داد.
- متدهای `TextWidth` و `TextHeight` می‌توانند به ترتیب عرض و ارتفاع عبارت رشته‌ای را که دریافت می‌کنند معین کنند.
- به وسیله خصوصیت `AutoRedraw` می‌توان از حذف ترسیمات موجود در یک پنجره جلوگیری به عمل آورد.
- خصوصیت `DrawMode` می‌تواند رنگ ترسیمات گرافیکی را تعیین کند.
- به وسیله خصوصیت `DrawStyle` می‌توان نوع خطوط را در متدهای گرافیکی نظیر `Line` و `Circle` تعیین کرد.
- با استفاده از خصوصیت `DrawWidth` می‌توان ضخامت خطوط و نقاط را در متد `Line`، `Circle` و `PSet` مشخص کرد.
- خصوصیت `FillStyle`، نوع و حالت خطوطی که سطح یک شکل گرافیکی نظیر مستطیل، دایره یا بیضی را پر می‌کند تعیین می‌کند.
- به وسیله خصوصیت `FillColor` می‌توان رنگ مورد نظر را برای پوشاندن سطح یک شکل گرافیکی نظیر مستطیل، دایره یا بیضی تعیین کرد.
- تابع `QBColor` می‌تواند با دریافت یک عدد صحیح، رنگ متناظر آن را به صورت یک عدد از نوع `Long` تعیین کند.
- به وسیله تابع `RGB` می‌توانید ترکیبات رنگی مورد نظرتان را بر اساس مقدار سه رنگ اصلی آبی، قرمز و سبز ایجاد کنید.
- با استفاده از شیء `Printer` و خصوصیات و متدهای این شیء، می‌توان اطلاعات مورد نظر را به وسیله چاپگر روی کاغذ چاپ کرده و بر نحوه انجام عملیات چاپ نظارت کرد.
- به وسیله کنترل `MCI` و `MAM` می‌توان انواع فایل‌های صدا، موسیقی و ویدئویی را پخش کرد.
- به وسیله رویه‌های `LoadPicture` و `PaintPicture` می‌توان فایل‌های گرافیکی را روی کنترل کادر تصویر یا فرم نمایش داد.
- به وسیله رویه `SavePicture` می‌توان ترسیمات روی فرم یا کنترل کادر تصویر را روی دیسک ذخیره کرد.

آزمون نظری

۱ - به وسیله کدام خصوصیت می توان یکی از مقیاس های استاندارد را در ویژوال بیسیک انتخاب کرد؟

الف - ScaleMode ب - thgScaleHei ج - ScaleWidth د - Scale

۲ - کدام گزینه برای رسم یک مستطیل توپر توسط متد Line مناسب است؟

الف - B ب - BF ج - F د - FB

۳ - کدام تابع در متد Print می تواند اطلاعات نمایشی را در ستون مشخصی نمایش دهد؟

الف - Spc ب - Point ج - Tab د - Spcb

۴ - خروجی فرمان,,, 2 1000 (150,200) Circle به صورت..... است.

الف - بیضی افقی ب - بیضی عمودی

ج - نیمدایره د - ربع دایره

۵ - به وسیله کدام خصوصیت می توان ضخامت ترسیمات را تعیین کرد؟

الف - DrawMode ب - DrawStyle

ج - DrawWidth د - FillStyle

۶ - کدام متد می تواند عملیات چاپ را خاتمه دهد؟

الف - EndDoc ب - Port ج - NewPage د - KillDoc

۷ - کدام تابع می تواند ترکیبات رنگی را بر اساس رنگ های اصلی ایجاد کند؟

الف - QBColor ب - BackColor ج - RGB د - ColorMode

۸ - کدام خصوصیت، رنگ مورد نظر برای پر کردن یک مستطیل یا دایره را معین می کند؟

الف - FillStyle ب - FillColor

ج - ColorMode د - ForeColor

۹ - به وسیله کدام خصوصیت موقعیت جاری مکان نما در صفحه ترسیمات در جهت محور

افقی معین می شود؟

الف - CurrentX ب - CurrentY

ج - Point د - گزینه های الف و ب صحیح هستند.

۱۰ - واحد اندازه گیری پیش فرض در سیستم مختصات ویژوال بیسیک عبارت است

از:

الف - cm ب - inch ج - pixel د - twip

۱۱ - رویه SavePicture ترسیمات را به صورت فایل‌های گرافیکی از نوع ذخیره می‌کند.

الف - JPG ب - BMP ج - ICO د - CUR

۱۲ - کدام یک از کنترل‌ها امکان پخش فایل‌های ویدئویی با پسوند MPG را دارند؟

الف - MCI ب - Image ج - MAM د - Picture

۱۳ - Which of the following answer can be used in circle method to draw an ellipse?

a- radius b- start c- aspect d- end

۱۴ - سیستم مختصات در ویژوال بیسیک را توضیح داده و نحوه تنظیم آن را بیان کنید.

۱۵ - نحوه ساخت و استفاده از شیء تصویر را توضیح دهید.

۱۶ - نحوه استفاده از شیء چاپگر را بیان کرده و خصوصیت‌ها و متدهای آن را توضیح دهید.

۱۷ - کاربرد متدهای Line، PSet، Circle و Point را با ذکر مثال توضیح دهید.

۱۸ - کنترل‌های چند رسانه‌ای را در ویژوال بیسیک نام برده و نحوه استفاده از آن‌ها

را برای پخش فایل‌های موسیقی و نمایش فیلم توضیح دهید.

۱۹ - خصوصیت‌های CurrentY، CurrentX، DrawMode، DrawStyle و DrawWidth را

توضیح دهید.

۲۰ - نحوه ذخیره‌سازی ترسیمات گرافیکی را با استفاده از شیء تصویر توضیح دهید.

آزمون عملی

۱ - پروژه‌ای طراحی کنید که شکل زیر را نمایش دهد.



- ۲ - پروژه‌ای طراحی کنید که کاربر بتواند انواع بیضی، دایره و کمان را با توجه به مقادیر دلخواهش مشاهده کند.
- ۳ - پروژه‌ای طراحی کنید که با دریافت معدل دانش‌آموزان ۵ کلاس ۲۰ نفره میانگین معدل هر کلاس را محاسبه کند، سپس نتایج را به صورت نمودار میله‌ای نمایش دهد.
- ۴ - پروژه‌ای طراحی کنید که به طور پیوسته دایره‌ها با اندازه و رنگ‌های متفاوت به صورت تصادفی ایجاد و نمایش داده شوند.
- ۵ - پروژه‌ای طراحی کنید که مختصات سه رأس یک مثلث را دریافت نموده، سپس آن را ترسیم کند.
- ۶ - پروژه‌ای طراحی کنید که حرکت یک اتومبیل را شبیه‌سازی کند.
- ۷ - پروژه‌ای طراحی کنید که به وسیله آن بتوان هر تصویر دلخواه را با اندازه مورد نظر مشاهده کرد و در صورت تمایل آن را به صورت فایل گرافیکی از نوع BMP ذخیره نمود.