به نام حق

برنامه سازی (۲)

فصل اول

گرافیک و تصویر در برنامه به کمک Image و PictureBox

خصوصیات PictureBox:

- Picture: انتخاب تصویر
- Autosize: اگر Trueباشد، کادر به اندازه تصویر در می آید.
- اگر تصویر بزرگتر از کادر باشد فقط بخشی که جا می شود، قابل دیدن است.
- Name: نام کادر

خصوصیات Image:

- Picture: انتخاب تصویر
- Autosize: اگر Trueباشد، کادر به اندازه تصویر در می آید.
- Stretch : اگر True باشد و تصویر بزرگتر از کادر باشد تصویر را به اندازه کادر در می آورد
- Name: نام کادر

به کمک دستور loadPicture می توان تصویری از مسیر دلخواه روی کامپیوتر در برنامه در کادر مربوطه نمایش داد.

```
نام کادر تصویر.picture= loadPicture("نام و مسیر فایل تصویر شامل نام، پسوند، پوشه یا درایو ...")

مثال→ Picture1. picture= loadPicture("E:\VB1\pic\b.bmp ")
```

دستورات دیگر پردازش تصویر

- دستور PaintPicture : باعث می شود که تصویر طبق الگوی خاصی نمایش یابد

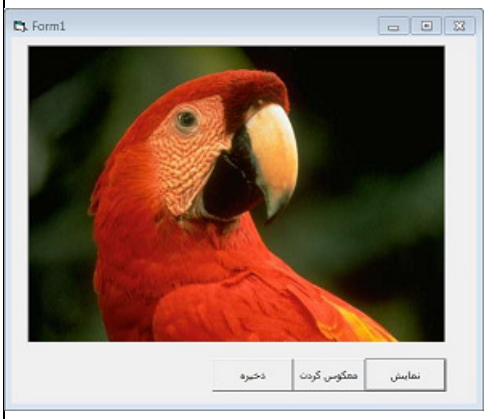
مثال: نمایش تصویر با معکوس رنگ

```
Picture1.PaintPicture  Picture1.Picture, 0, 0, , , , , , , vbDstInvert
```

- دستور savePicture : تصویری که در کنترل روی فرم دیده می شود در جایی دیگر ذخیره کنیم.

```
SavePicture  Picture1.Picture,"مسیر و نام فایل برای ذخیره"
SavePicture  Picture1.Picture,"d:\a.jpg"
```

مثال ۱: بارگذاری و معکوس کردن و ذخیره تصویر
توضیح: app.path مسیر ذخیره پروژه را بر می گرداند.

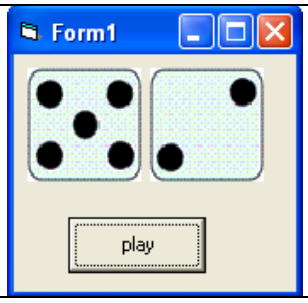



```
Private Sub Command1_Click()  ' نمایش تصویر ذخیره شده در مسیر برنامه
    Picture1.Picture = LoadPicture(App.Path & "\images\tuti.jpg")

End Sub

Private Sub Command2_Click() ' نمایش تصویر بصورت معکوس رنگها
    Picture1.PaintPicture Picture1.Picture, 0, 0, , , , , , , vbDstInvert
End Sub

Private Sub Command3_Click()  ' ذخیره تصویر نمایش داده شده با نامی دیگر
    SavePicture Picture1, App.Path & "\images\tuti2.jpg"
End Sub
```

مثال ۲: برنامه ای طراحی نمایید که با انتخاب دو عدد تصادفی بین ۱ تا ٦ انداختن دو تاس ٦ وجهی را با انتخاب تصاویر مربوطه نشان دهد.
توضیح: app.path مسیر ذخیره پروژه را بر می گرداند.

نخست فایلهای با نام 1.gif تا 6.gif را در پوشه Images از مسیر ذخیره برنامه ذخیره کنید

```
Dim d1 As Integer, d2 As Integer
   d1 = 1 + Int(6 * Rnd())
   d2 = 1 + Int(6 * Rnd())
   Image1.Picture = LoadPicture(App.Path & "\images\" & d1 & ".gif")
   Image2.Picture = LoadPicture(App.Path & "\images\" & d2 & ".gif")
```



رویدادهای ماوس و صفحه کلید

انواع رویداد ماوس که در اثر کلیک ها و حالات مختلف بوجود می آیند:

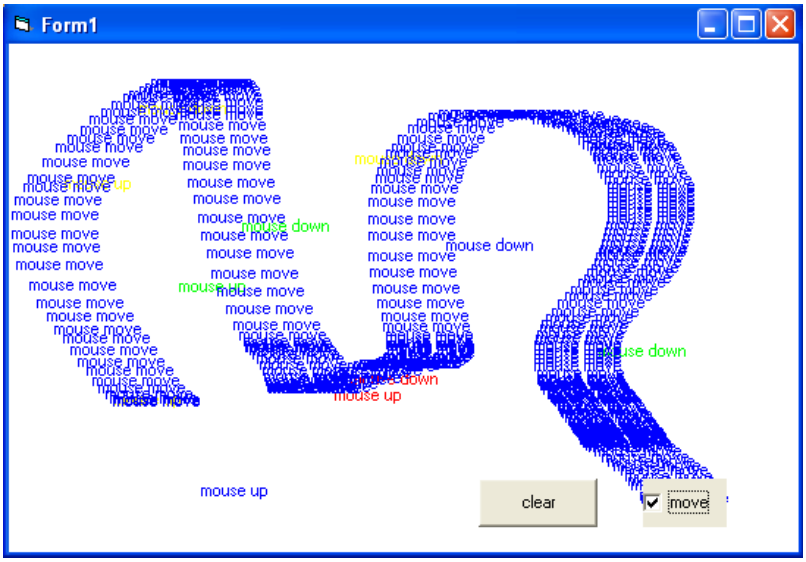
- جابجایی ماوس MouseMove
- کلیک(چپ) Click
- دابل کلیک double-Click(DblClick)
- کلیک راست right click
- کشیدن و رها کردن (Drag & Drop)
- فشار دادن ماوس MouseDown
- رها کردن ماوس MouseUp

هنگامی که ماوس فشار داده می شود می توان برای آن رویدادی را تعریف کرد در این رویداد چهار پارامتر دیده می شود:

- Button: کدام دکمه فشار داده شده است(۱: چپ، ۲: راست، ٤: وسط)
- Shift: فشار داده شدن کلیدهای تبدیل صفحه کلید همراه با کلیک(alt: 4, Ctrl:2, shift:1)
- X, Y: موقعیت ماوس روی صفحه

مثال رویداد های مهم ماوس

```
Private Sub Command1_Click()→  Clear  رویداد دکمه
    Cls  →پاک کردن فرم
End Sub

Private Sub Form_Click()  → تنظیم
رنگ نوشته ها با یک رنگ تصادفی
    Dim a As Integer
    Randomize
    a = 1 + Int(Rnd() * 4)
    Select Case a
        Case 1
            ForeColor = vbRed
        Case 2
            ForeColor = vbBlue
        Case 3
            ForeColor = vbGreen
        Case 4
            ForeColor = vbYellow
    End Select
End Sub
```




```
Private Sub Form_DblClick()→ تنظیم رنگ پس زمینه با یک رنگ تصادفی
    Dim a As Integer
    Randomize
    a = 1 + Int(Rnd() * 4)
    Select Case  a
        Case 1
            BackColor = vbWhite
        Case 2
            BackColor = vbYellow
        Case 3
            BackColor = vbCyan
        Case 4
            BackColor = vbBlack
    End Select
End Sub
```

```
Private Sub Form_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
    CurrentX = X
    CurrentY = Y
    Print "mouse down"
End Sub

Private Sub Form_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    If Check1.Value = 1 Then
        CurrentX = X
        CurrentY = Y
        Print "mouse move"
    End If
End Sub

Private Sub Form_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
    CurrentX = Y
    CurrentY = X
    Print "mouse up"
End Sub
```

کشیدن و رها کردن کنترلها روی فرم

۱- روش خودکار: در این حالت خصوصیت DragMode کنترلی که می خواهیم حرکت دهیم به مقدار Automatic تغییر می دهیم. سپس در رویداد *DragDrop از فرم/* *دستورات زیر نوشته می شود:*

Source.move x,y

مثال: برنامه ای برای حرکت دادن کنترل Label1 طراحی کنید.

Label1→ DragMode: Automatic

در رویداد Form_DragDrop دستور زیر را بنویسید

Source.Move X, Y

```
Private Sub Form_DragDrop(Source As Control, X As Single, Y As Single)
Source.Move X, Y
End Sub

Private Sub Form_Load()
Label1.DragMode = vbAutomatic
End Sub
```

۲- روش دوم جابجا کردن:

- در این حالت خصوصیت DragMode کنترلی که می خواهیم حرکت دهیم به مقدار Manual تغییر می دهیم
- در قسمت general دو متغیر زیر برای ذخیره محل قبلی کنترل تعریف کنید:
  Dim X0 As Single, Y0 As Single
- در رویداد MouseDown مربوط به کنترل(روی فرم) دستور زیر را می نویسیم

نام کنترل. Drag
X0 = X
Y0 = Y

- در رویداد Form_DragDrop دستور زیر نوشته می شود:

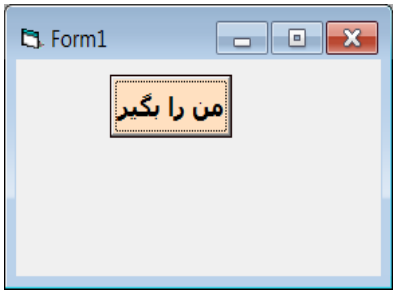
Source.Left = X - X0
Source.Top = Y - Y0

```
Dim X0 As Single, Y0 As Single
Private Sub Command1_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
   Command1.Drag
   X0 = X
   Y0 = Y
End Sub

Private Sub Form_DragDrop(Source As Control, X As Single, Y As Single)
   Source.Left = X - X0
   Source.Top = Y - Y0
End Sub
```

مثال: برنامه ای طراحی کنید که با حرکت ماوس روی یک دکمه، موقعیت دکمه طوری عوض شود که قابل کلیک نباشد(از ماوس فرار کند). cmd1 نام دکمه است.

```
Private Sub cmd1_Click()
    MsgBox "آفرین که توانستی!"
    End
End Sub
```

```
Private Sub cmd1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    Dim a As Long, b As Long
    Randomize
    a = 10 + Rnd() * ScaleWidth
    b = 10 + Rnd() * ScaleHeight
    cmd1.Left = a
    cmd1.Top = b

    If cmd1.Left > Width Then cmd1.Left = Width - cmd1.Width
    If cmd1.Top > Height Then cmd1.Top = Height - cmd1.Height
End Sub
```



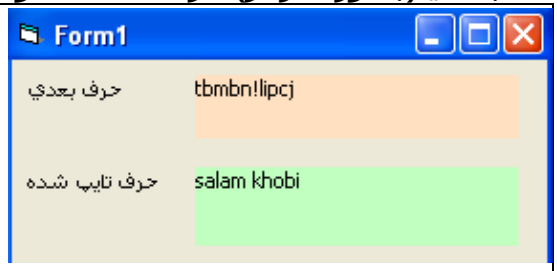
رویداد های صفحه کلید: با فشار دادن یک کلید چه اتفاقی بیافتد.

- KeyPress: زمانی که کلیدی فشار داده شود(برای تشخیص حروف بزرگ و کوچک، اعداد، علائم نقطه گذاری، کلیدهای Tab,Enter,Space کاربرد دارد و برای برخی کلیدهای تبدیلی مثل Shift , alt ,Ctrl بکار نمی رود)
- KeyUp: زمانی که کلیدی فشار داده شده و رها شود
- KeyDown : زمانی که کلیدی فشار داده شود

KeyUp, KeyDown برای تشخیص هر کلید در صفحه کلید کاربرد دارند

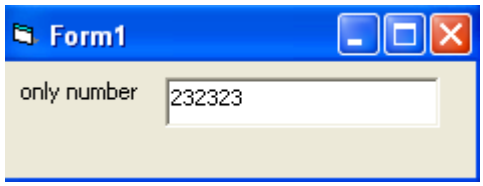
مثال: برنامه بنویسید که با تایپ هر حرف، آن حرف در Label2 و حرف بعدي(بصورت رمز) در Label1 نوشته شود

```
Private Sub Form_KeyPress(KeyAscii As Integer)
Dim s1 as string, s2 as string
S1= Label1.Caption
S2= Label2.Caption
Select Case KeyAscii
Case 65 To 90, 32, 97 To 122  → A..Z,  a..z , space
    Label1.Caption = s1 & Chr(KeyAscii + 1)→ رمزدار کردن
    Label2.Caption = s2 & Chr(KeyAscii)
Case 8    → backspace
   Label1.Caption = Mid(s1, 1, Len(s1) - 1) → حرف آخر حذف شود
   Label2.Caption = Mid(s2, 1, Len(s2) - 1)
End Select
End sub
```



مثال ۲: تایپ فقط عدد در کادر متن

```
Private Sub Text1_KeyDown(KeyCode As Integer, Shift As Integer)
        Dim s As String
        s = Text1.Text
        Select Case KeyCode
        Case vbKey0 To vbKey9 → کلید عددی
            Text1.Text = s & Chr(KeyCode)
        Case vbKeyBack
            Text1.Text = Mid(s, 1, Len(s) – 1)    → پاک کردن آخرین رقم
        End Select
End Sub

Private Sub Text1_KeyPress(KeyAscii As Integer)
   KeyAscii = 0   → خنثی کردن بقیه کلیدها
End Sub
```

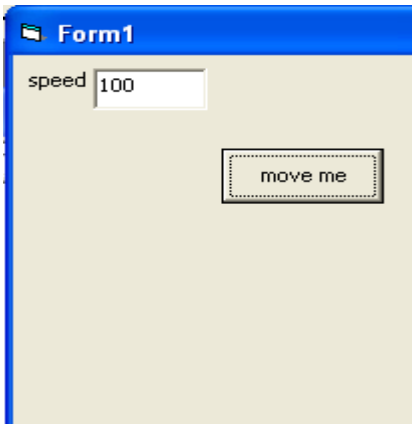


مشخصه KeyPreview: برای اینکه رویداد های صفحه کلید توسط فرم(Form) دریافت گردند، باید این مشخصه با True تنظیم گردد.

مثال ۳: می خواهیم به کمک کلیدهای جهت دار(در قسمت کلیدهای عددی راست ۸↑، ۲↓، ٤←، ٦→) یک کنترل Command را در صفحه جابجا کنیم.(کلید numLock باید روشن باشد)

- در یک پروژه جدید روی فرم یک کنترل Command (با نام cmdMove)قرار می دهیم
- در رویداد Form_KeyDown دستورات زیر را می نویسیم

```
Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
   Dim speed As Integer
   speed = Text1.Text   → اندازه سرعت حرکت
   Select Case KeyCode
      Case vbKeyNumpad8  → حرکت به بالا
         cmdMove.Top = cmdMove.Top – speed
      Case vbKeyNumpad2  → حرکت به پایین
         cmdMove.Top = cmdMove.Top + speed
      Case vbKeyNumpad6  → حرکت به راست
         cmdMove.Left = cmdMove.Left + speed
      Case vbKeyNumpad4  → حرکت به چپ
         cmdMove.Left = cmdMove.Left – speed
   End Select
End Sub

Private Sub Form_Load()
   KeyPreview = True  → تا اینکه فرم رویداد کلید را دریافت کند
End Sub
```



تمرین: برنامه حرکت Command را طوری اصلاح کنید که با نگه داشتن کلید shift سرعت دوبرابر، کلید Ctrl سرعت نصف و کلید alt سرعت ۱/٤ شود

ارسال ضربات کلید به برنامه

بدون صفحه کلید با ماوس صفحه کلید بسازیم
دستور زیر را استفاده می کنیم:

Sendkeys "نام کلیدها"

قوانین:

- برای دریافت ضربات کلیدها برای یک کنترل باید دستور زیر را برای آن نوشت:
  آماده نوشتن در آن → SetFocus.نام کنترل
- برای ارسال کارکترهای خاص(غیر از الفبا و ارقام) باید نام آنرا داخل {} نوشت.

sendKeys "7 {+} 6" → 7 + 6
SendKeys "{home}" →home فشار داده شدن کلید

- ارسال کلیدهای تبدیل: + برای shift، ^ برای Ctrl و % برای Alt
- برای ارسال چند کلید همراه با کلیدهای تبدیل: داخل پرانتز

SendKeys "+(AB)" → shift , A , B
SendKeys "+%H" → shift, Alt, H
SendKeys "%^{Delete}" → alt , Ctrl ,Delete

- برای ارسال یک کلید به تعداد معین(تکرار) باید از قالب *{تعداد تکرار کلید}* استفاده کرد و بین کلید و تعداد فاصله باشد:

sendKeys “{A 5}” →A ارسال ۵ مرتبه حرف

توجه: در برخی سیستم عاملها مثل ویندوز ۷ ممکن است دستور Sendkeys عمل نکند. برای اجرای برنامه، باید برنامه را در مسیر دلخواه ذخیره کرده و سپس اجرایی(*.exe) آنرا در همان مسیر ساخته و سپس فایل اجرایی را با دابل کلیک روی نام آن اجرا نمایید.
نحوه ایجاد فایل اجرایی: (ProectName نام پروژه ذخیره شده است.)

File → Make ProectName.Exe

مثال ۱: در یک پروژه یک Text و چندین دکمه(با عناوین کلیدهای دلخواه) قرار دهید که با زدن هر دکمه کلید مورد نظر به Text ارسال شود.

مثال۲: به کمک ضربات کلید با دستور SendKeys یک ماشین حساب بدون کمک از صفحه کلید طراحی کنید.

۱. به تعداد لازم کنترل Command قرار داده و Caption هریک را با مقدار مورد نظر تنظیم کنید. برای مدیریت بهتر باید نام آنها (Name) را به Cmd1 الی آخر تغییر داد.
۲. یک Text قرار می دهیم
۳. در رویداد هر دکمه دستور زیر را می نویسیم:

نام کنترل متن.SetFocus
SendKeys "عنوان کلید"

مثل:

```
Text1.SetFocus
SendKeys "1"
```

حل:

۱- در اول برنامه

Dim v1 As Single, v2 As Single, r As Single → متغیرهای عددی
*Dim op As String * 1 → عملگر (+, -, *, /)*

۲- در دکمه عملگرها دستور زیر:

v1 = Val(Text1.Text) → مقدار اول
op = "+" → عملگر
Text1.Text = "" → خالی کردن برای مقدار دوم

۳- رویداد دکمه =

```
v2 = Val(Text1.Text) → مقدار دوم
Select Case op → انجام عمل مورد نظر
Case "+"
   r = v1 + v2
Case "-"
   r = v1 - v2
Case "*"
   r = v1 * v2
Case "/"
   r = v1 / v2
End Select
Text1.Text = CStr(r) → نمایش نتیجه در کادر متن
```

۴- دکمه c برای پاک کردن یک رقم:

```
Text1.SetFocus
SendKeys "{backspace}"
```

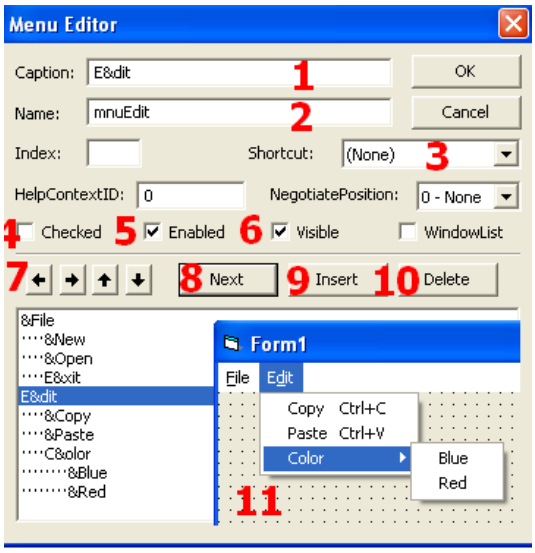
۵- دکمه Clear برای پاک کردن همه:

```
Text1.SetFocus
SendKeys "{home}+{end}{delete}"      یا Text1.Text=""
```

فصل دوم - منو ها(Menu)

منو امکانی است برای تعریف فرمانهای مختلف جهت اجرای دستورات خاص در برنامه
به دو شیوه منو را می توان ایجاد کرد:

- بصورت خودکار با Application Wizard و تنظیمات پنجره های ظاهر شده
- بصورت دستی؛ که به روشهای زیر میتوان تعریف منو را انجام داد:
  - Ctrl+E
  - Tools→Menu Editor



شرح تعریف منوها

۱- Caption: عنوان ظاهری منو که با قرار دادن علامت & جلو یک حرف میتوان(بدون ماوس) با زدن کلید Alt و حرف مورد نظر، آن منو را فعال کرد: مثل E&dit
۲- Name: نامی غیر تکراری که به هر فرمان منو داده می شود(بهتر است نام منو با Mnu آغاز شود: مثل mnuEdit)
۳- Shortcut: کلیدهای میانبر برای اجرا
۴- checked: حالت انتخاب یا عدم انتخاب فرمان منو
۵- enabled: فعال بودن یا نبودن فرمان منو
۶- visible: ظهور یا مخفی شدن فرمان منو
۷- ↑↓→←: دکمه های جابجایی منوها
۸- next: حرکت بین منوها
۹- insert: ایجاد فرمان منوی جدید
۱۰- delete: حذف فرمان منوی مورد نظر
۱۱- نمونه منو ایجاد شده روی فرم

توجه: برای قرار دادن خط جداساز بین بخشهای منو، فرمان جدیدی ایجاد کرده و در Caption آن علامت خط تیره(-) نوشته و حتماً نامی(Name) مثل sep1 برای آن تعیین کنید

برای نوشتن کد مربوط به فرمان منو، بعد از بستن پنجره فوق(با زدن OK) روی فرمان مورد نظر کلیک کنید تا محل کد ظاهر شود. مثلاً با زدن فرمان منوی Blue :

```
Private Sub mnuBlue_Click()
    BackColor = vbBlue → دستورات مورد نظر
End Sub
```

تمرین: منویی به شکل زیر ایجاد کنید. و در فرمان هر یک دستوراتی بنویسید که رنگ پس زمینه و پیش زمینه یک Label روی فرم تغییر کند.

کلید میانبر	عنوان منو	
	Backcolor	Color
Ctrl+K	Black	رنگ پس زمینه
Ctrl+W	White	
Ctrl+Y	Yellow	
	ForeColor	
Ctrl+G	Green	رنگ پیش زمینه
Ctrl+R	Red	
Ctrl+B	Blue	

منوی بازشو (Popup)

منویی غیر ثابت که معمولاً در اثر کلیک راست فعال می شود. برای ایجاد چنین منویی، نخست منو را در Menu Editor ایجاد کرده و سپس با دستور زیر فعال می شود:
تعیین موقعیت نمایش منو اختیاری است و اگر ننویسیم در محل کلیک ظاهر می شود.

```
popupMenu    نام منو , , x , y
PopupMenu    mnuColor  , , 1000,2000
```

```
Private Sub Form_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
        If  Button = 2   then   PopupMenu   mnuColor   → با کلیک راست روی فرم منوی رنگ باز شود
End Sub
```

مثال: برنامه ای طراحی کنید که منوهای تمرین قبل را داشته باشد و با کلیک چپ روی فرم، منوی رنگ پس زمینه و با کلیک راست منوی رنگ پیش زمینه ظاهر شود.

```
Private Sub Form_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
   If Button = 1 Then  → کلیک چپ
      PopupMenu mnuBack
   ElseIf Button = 2 Then    → کلیک راست
      PopupMenu mnuFore
   End If
End Sub

Private Sub mnuBalck_Click()
   Label1.BackColor = vbBlack  → تغییر رنگ پس زمینه برچسب
End Sub

Private Sub mnuBlue_Click()
   Label1.ForeColor = vbBlue   → تغییر رنگ پیش زمینه برچسب
End Sub

Private Sub mnuRed_Click()
   Label1.ForeColor = vbRed
End Sub

Private Sub mnuWhite_Click()
   Label1.BackColor = vbWhite
End Sub

Private Sub mnuYellow_Click()
   Label1.ForeColor = vbYellow
End Sub
```

استفاده از Clip Board

Clip Board ، مکانی است که اطلاعات و اشیاء کپی شده در آن قرار می گیرند تا بعداً در جای دیگری استفاده شوند.

برای اینکه بتوان متنی را از یک Textbox به کمک Clip Board کپی کنیم مراحل زیر را انجام می دهیم:

۱- قرار دادن متن موجود از کادر اول در Clip Board

```
ClipBoard.SetText   Text1.Text
```

۲- انتقال متن به کادر دوم

```
Text2.Text= ClipBoard.GetText
```

برای پاک کردن محتویات clip board از دستور زیر استفاده می شود:

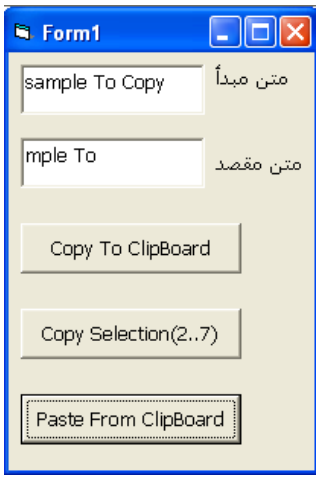
```
ClipBoard.Clear
```

برای انتخاب بخشی از متن مبدأ، باید از دستورات زیر کمک گرفت(اولین کارکتر دارای شماره صفر است)

```
Text1.SelStart= ۲   → شماره کارکتر شروع
Text1.SelLength=۷   → تعداد کارکتر انتخابی
ClipBoard.SetText   Text1.SelText → کپی متن انتخاب شده
```

مثال: متن موجود در کادر مبدأ را به کمک clip board به کادر مقصد کپی کنید. همچنین در دکمه ای دیگر بخشی از متن(مثلاً از کارکتر ۲ تا ۷) را کپی نمایید.

```
Private Sub cmdCopy_Click()
   Clipboard.SetText    Text1.Text  →clipboard کپی متن به
End Sub

Private Sub cmdPaste_Click()
   Text2.Text = Clipboard.GetText  → clipboard انتقال متن از
End Sub

Private Sub cmdSelCopy_Click()
   Text1.SelStart = 2
   Text1.SelLength = 7
   Clipboard.SetText    Text1.SelText  → clipboard کپی بخش انتخاب شده متن به
End Sub
```



تمرین:

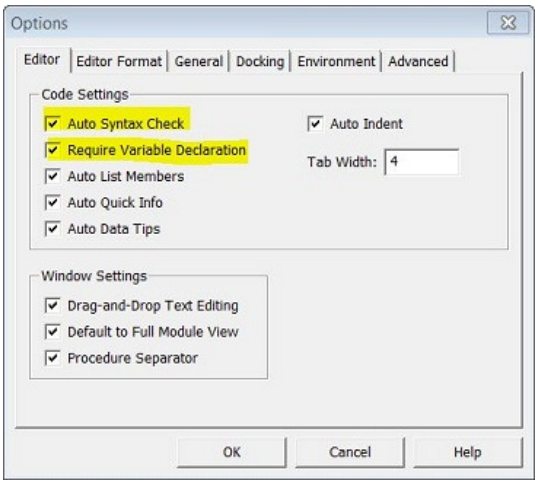
- مثال منوی بازشو
- مثال clip board
- خود آزمایی ۷ صفحه ۶۹

فصل سوم -اشکال زدایی برنامه(Debug)

انواع خطاها در برنامه

۱. **خطای نحوی(Syntax Error)**: که در اثر رعایت نکردن دستورات و قالب زبان برنامه نویسی روی می دهد(قبل از اجرا و توسط مترجم اعلام می گردد)

- عدم تعریف متغیرها و استفاده از آنها در برنامه --> جلوگیری توسط Option Explicit در اول برنامه یا تنظیم مسیر زیر(تیک دار کردن) تا در صورت تعریف نشدن متغیر قبل از اجرا هشدار دهد:



Tools→ Options→ Require Variable Declaration

از این به بعد همیشه خودبخود عبارت Option Explicit در اول برنامه نوشته میشود تا در صورت تعریف نشدن متغیری برای جلوگیری از اشکال برنامه، هشدار دهد.

- اشتباه در نوشتن قالب دستورات مانند دستور if بدون then یا مثلاً نوشتن کلمه nxet به جای next در حلقه for : --> اعلام خطا توسط تیک دار کردن گزینه زیر

Tools→ Options→ Auto syntax check

۲. **خطای زمان اجرا(Run Time Error)**: در حین اجرای برنامه روی داده و از قبل از آن آگاه نیستیم. مثل خطای تقسیم بر صفر(division by zero) یا سرریز(overflow) در محاسبات

```
Dim A as Integer, B as Integer, C as Integer , D as Integer
D = 0
A= 5۰۰۰
B = A / D→ خطای تقسیم بر صفر که هنگام اجرا روی می دهد
C = A *1000  → خطای سرریز : مقدار ۵۰۰۰۰۰۰ در متغیر جا نمی شود
```

---> رفع از طریق مدیریت خطاها(در ادامه می خوانیم)

۳. **خطای منطقی(Logical Error)**: برنامه اجرا می شود ولی نتیجه مورد انتظار تولید نمیشود که به علت عدم دقت برنامه نویس روی می دهند. مانند مثال زیر:

```
dim num as integer, nim as integer
 Num = 5+7
Print  nim   →  خروجی عدد صفر است نه ۱۲
```

--> رفع توسط اشکالزدایی و اجرای خط به خط

اشکالزدایی و اجرای خط به خط:

خط به خط برنامه را اجرا کرده و مقادیر متغیرها را در هر لحظه ببینیم.

۱. می توان تا خط خاصی از برنامه، عادی اجرا کنیم و بعد خط به خط اجرا نماییم این کار با ایجاد نقطه توقف(break Point) انجام می شود به سه روش:
   - کلیک روی یک خط دستور و زدن F9
   - کلیک روی حاشیه پنجره کد
   - از منوی Debug گزینه Toggle....

۲. نمایش مقدار متغیرها به روشهای زیر:
   - نگه داشتن ماوس و مکث روی متغیر

- انتخاب کلمه متغیر و کلیک راست و گزینه Add Watch تا در لیست قرار گیرد

۳. اجرای خط به خط برنامه
- F8 (debug→ step into): خط به خط اجرا را میبینیم و اگر به فراخوانی یک زیر برنامه (روال یا تابع) برسیم، به داخل آن نیز رفته و خط به خط ادامه می دهد.
- shift +F8 (debug→ step over): خط به خط اجرا را میبینیم و اگر به فراخوانی یک زیر برنامه(روال یا تابع) برسیم، آنرا کامل اجرا می کند و به داخل آن *نمی رود*.
- Ctrl +F8 (Debug→ run to cursor) :اجرا تا محل مکان نمای ماوس

پنجره های دیگر که از منوی View فعال می شوند:
Locals: برای مشاهده مقادیر متغیرهای محلی در دسترس
Watches: نمایش مقدار لحظه ای متغیرها
Immediate: آزمایش خطوطی از کد بدون اجرای برنامه با دستور print
CallStack: ترتیب فراخوانی روالها و توابع را نشان می دهد.

تمرین: برنامه محاسبه فاکتوریل را بصورت قدم به قدم اجرا کرده و مقادیر متغیرها را در هر مرحله بررسی نمایید و در جدولی بنویسید.
- ✓ روی خط چهارم، نقطه توقف(break point) قرار دهید
- ✓ مقدار همه متغیرها را به پنجره watch اضافه کنید
- ✓ با F8 اجرا را دنبال کنید.

```
Dim n As Integer, k  As Integer, P As Long
n = Val (Text1.Text)
P = 1
 For   k  =  1 To  n
     P = P  *  k
Next   k
Print   P
```

توجه مهم: دستور Option Explicit را در اول همه برنامه ها بنویسید. تا تعریف متغیر اجباری شود و خطای کمتری روی دهد.

مدیریت خطا در خطاهای زمان اجرا: در صورت خطای زمان اجرا مانند تقسیم بر صفر یا سرریز(مقداری در متغیر جا نشود)، برنامه چه کار کند؟

۱. قبل از دستورات مشکوک دستور زیر را می نویسیم:

On Error goto یک برچسب

۲. دستورات مشکوک را می نویسیم و از روال خارج می شویم(با دستور exit sub)
۳. نام برچسب را نوشته و آخر آن علامت **(:)** قرار داده و دستورات مدیریت و واکنش برنامه به خطا را در خطوط بعدی می نویسیم

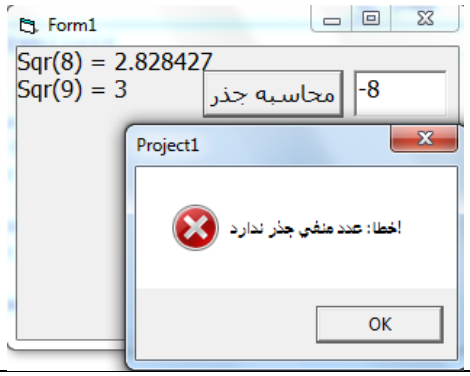
توجه: شیئ Err در برنامه حاوی اطلاعاتی در مورد خطای رخ داده می باشد:
Err.Number → شماره خطا
Err.Description → شرح خطا

مثال: می خواهیم برنامه ای برای تقسیم نوشته و خطاها را مدیریت کنیم. پیامی حاوی شماره و نوع خطا با دکمه های زیر نمایش داده و نتیجه انتخاب کاربر را مدیریت می نماییم:
- Abort: خروج از برنامه با دستور end
- Retry: سعی مجدد با اصلاح متغیر تقسیم، پیگیری و انجام مجدد دستور تقسیم (resume)
- Ignore: صرف نظر از انجام تقسیم و اجرای دستور بعدی (resume next)

```
Dim n As Integer                          → General متغیر سراسری در
Private Sub CmdDivide_Click()             → برنامه انجام تقسیم در یک دکمه
   Dim h As Integer                       → متغیر برای ذخیره نتیجه کادر پیام که چه دکمه ای زده است
   On Error GoTo TestError                → TestError اگر خطای زمان اجرا روی داد برو به دستورات محل برچسب
      Print 2500 / n                      → دستورات مشکوک برنامه
      Print "دستور بعدی"                  → پیام موفقیت اجرا
    Exit Sub                              → خروج از روال
TestError :                               → برچسب
   →Abort, Retry, Ignore نمایش پیغامی حاوی شماره و نوع خطا و نیز دکمه های
   h = MsgBox("خطای شماره " & CStr(Err.Number) & " = " & Err.Description, vbAbortRetryIgnore)
   If (h = vbIgnore) Then Resume Next     → از خطا صرف نظر کن و برو به دستور بعدی
   If (h = vbAbort) Then End              → از برنامه خارج شو
   If (h = vbRetry) Then                  → دوباره با مقدار دیگری دستورات را اجرا کن(سعی مجدد)
      n = 10                              → تغییر و اصلاح متغیر تقسیم
      Resume                              → پیگیری مجدد دستورات
   End If
End Sub
```

مثال ۲: خطای زمان اجرا برای ریشه عدد منفی:

```
در رویداد دکمه محاسبه جذر
Dim n As Integer, r As Single
n = Val(Text1.Text)          →ورود عدد
On Error GoTo khata          →khata اگر خطا روی داد برو به بخش
    r = Sqr(n)               → دستور محاسبه ریشه
    Print "Sqr(" & CStr(n) & ") = " & CStr(r) → چاپ نتیجه
   Exit Sub    → خروج از روال
khata:
    MsgBox "خطا: عدد منفي جذر ندارد! " , vbCritical  → پیغام خطا
```



تمرین عملی: برای هر یک از موارد زیر تعیین کنید چه نوع خطایی وجود دارد، دلیل بروز خطا چیست و برای رفع آن چکار باید کرد؟ هریک از آنها را در رویداد دکمه ای با عنوان شماره آن بنویسید.

ردیف	برنامه	رفع خطا	خطا و دلیل آن
۱	Dim a as integer, b as single A= 5*300 Print b	چاپ a به جای b	خطای منطقی(عدم دقت برنامه نویس)
۲	Dim a as integer A= 5 If a < 2 then Print a End if	دستور print a در خط بعد نوشته شود یا end if حذف شود	خطای نحوی(عدم رعایت دستور زبان)
۳	Dim a as integer, b as integer A= 1050 B= 1000 Print sqr(b-a)	• اصلاح تفریق که عدد منفی بدست نیاید • کنترل خطا با دستور on Error	خطای زمان اجرا(ریشه عدد منفی)
۴	Dim a as integer, b as integer A= 64 B= a mod 2 Print a/b	• اصلاح دستورات که تقسیم بر صفر انجام نشود • کنترل خطا با دستور on Error	خطای زمان اجرا(تقسیم بر صفر) Division by zero
۵	Dim a as integer For a = 1 To 20 Print a Nexta	دستور nexta بصورت next a نوشته شود	خطای نحوی(عدم رعایت دستور زبان)
۶	Dim a As Integer, b As Integer, c As Integer b = 6 c = 1 For a = 1 To 20 c = c * b Print c Next a	• اصلاح دستورات که متغیر c با ظرفیت مورد نظر تعریف شود($۶^{۲۰}$) • کنترل خطا با دستور on Error	خطای زمان اجرا(سرریز) overflow

فصل چهارم - آرایه ها

متغیر: مکانی نامدار از حافظه برای ذخیره موقت اطلاعات
آرایه: مجموعه ای از متغیرهای همنام و هم نوع که بصورت متوالی در حافظه ذخیره می شوند.

نحوه تعریف آرایه:

Dim نام آرایه(تعداد عناصر) as نوع داده

تعداد	....	۲	۱	۰
	...			

مثال: ذخیره نمرات(نوع داده اعشاری) درس برنامه سازی همه دانش آموزان(۳۱ نفر)

```
Dim Nomre(30) as  single
```

حال ۳۱ متغیر داریم که از شماره صفر تا ۳۰ پشت سرهم قرار گرفته اند:

۳۰	....	۲	۱	۰
	...			

دسترسی به اعضای آرایه

از نام آرایه و شماره محل آن استفاده می کنیم: (شماره محل)نام آرایه

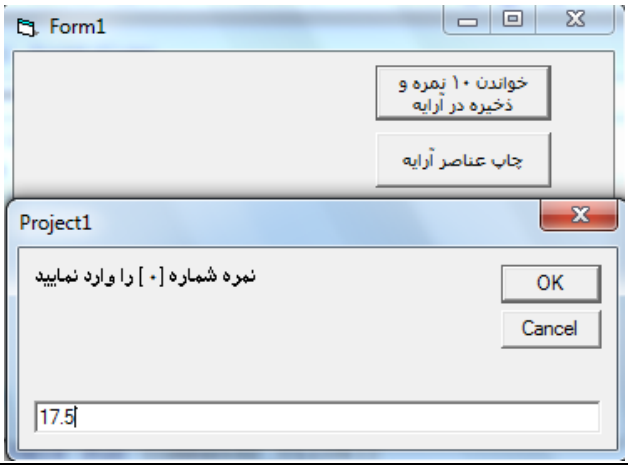
nomre(15) → نمره دانش آموز شماره ۱۵

می توان تک تک آنها را با دستور انتساب مقدار داد (تغییر داد) یا از ورودی خواند یا چاپ کرد:

```
nomre(15)=17.5                  → تغییر نمره دانش آموز شماره ۱۵ به ۱۷/۵
nomre(10)= inputbox("ورود نمره")  → خواندن نمره دانش آموز شماره ۱۰
print   nomre(6)                → چاپ نمره دانش آموز شماره ۶
```

نکته ۱: برای پردازش همه عناصر آرایه معمولاً از حلقه for استفاده می شود.
مثال1 : دستوراتی بنویسید که ۱۰ نمره را خوانده، در آرایه ذخیره نموده و سپس آنها را چاپ کند.

```
Option Explicit
Dim nomre(9) As Single  → آرایه برای ذخیره ۱۰ نمره(شماره ۰ تا ۹)
Private Sub Command1_Click() → دکمه خواندن نمرات
   Dim i As Integer            → متغیر حلقه
   Dim payam As String         → متغیر نمایش پیام
   For i = 0 To 9              → حلقه پردازش آرایه
      payam = " نمره شماره [" & CStr(i) & "] را وارد نمایید:"
      nomre(i)  = Val(InputBox(payam))  → خواندن هر نمره
   Next i
End Sub

Private Sub Command2_Click()→ دکمه چاپ نمرات
   Dim i As Integer
   For i = 0 To 9       → حلقه پردازش آرایه
      Print nomre(i);   → چاپ تک تک نمرات
   Next i
End Sub
```



تمرین ۱ : در دکمه جدید، دستوراتی بنویسید که حاصلجمع و میانگین نمرات را نیز چاپ کند

```
Private Sub Command3_Click() → دکمه محاسبه مجموع و میانگین
   Dim i As Integer, sum As Single, mean As Single
   sum = 0   → متغیر مجموع
   For i = 0 To 9
      sum = sum + nomre(i)  → جمع کردن نمرات
   Next i
   mean = sum / 10  → محاسبه میانگین
   Print
   Print "Sum = " & CStr(sum)& ", Mean = " & CStr(mean)
End Sub
```

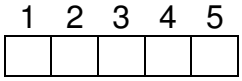
تمرین ۲ : در دکمه جدید، دستوراتی بنویسید که بزرگترین و کوچکترین نمره را پیدا و چاپ کند.

```
Private Sub Command۴_Click()  → دکمه پیدا کردن بیشترین و کمترین
   Dim i As Integer, Max As Single, Min As Single
   Max = nomre(۰)   →فرض کنیم اولین نمره از همه بیشتر است
   Min = nomre(۰)   →  فرض کنیم اولین نمره از همه کمتر است
   For i = ۰ To ۹
      If (nomre(i) > Max) Then  → اگر نمره بیشتری پیدا شد
         Max = nomre(i)          → ذخیره بیشترین در متغیر مربوطه
      ElseIf (nomre(i) < Min) Then  → اگر نمره کمتری پیدا شد
         Min = nomre(i)             → ذخیره کمترین در متغیر مربوطه
      End If
   Next i
   Print   → چاپ در خط بعدی
   Print "بیشترین = " & CStr(Max); ", کمترین = " & CStr(Min)
End Sub
```

نکته ۲: اگر بخواهیم اندیس شروع آرایه به جای صفر از یک شروع شود، در اولین خط برنامه (قسمت General)دستور زیر را می نویسیم:

Option Base 1

Dim A(5) as integer →

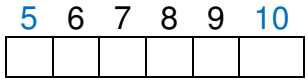


مثال:

نکته ۳: می توان اندیس های شروع و پایان آرایه را بصورت دلخواه نیز تنظیم کرد. برای این کار داخل پرانتز هنگام تعریف آرایه با کلمه **to** آنها را مشخص می کنیم:

Dim نام آرایه(شروع to پایان) as نوع داده

Dim A(5 to 10) as integer →



مثال:

نکته ۴: توسط دو تابع زیر میتوان اندیس شروع و پایان آرایه را بدست آورد:

Dim A(2 to 10) as Integer

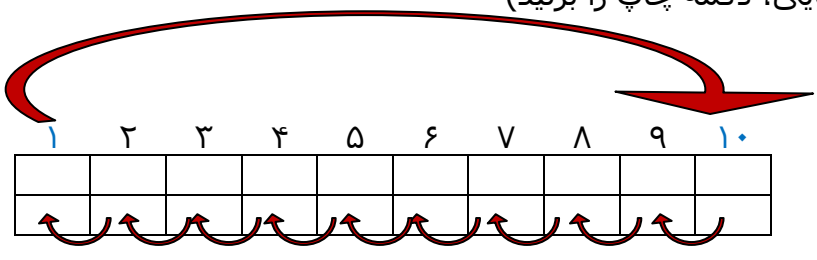
LBound(A) → اندیس شروع ۲

UBound(A) → اندیس پایان ۱۰

نکته 5: تعداد عناصر آرایه از فرمول زیر بدست می آید:

تعداد عناصر= شماره پایان - شماره شروع + ۱ → ۱۰ - ۵ + ۱=۶

مثال: دستوراتی بنویسید که اعداد ۱ تا ۱۰ را به ترتیب در آرایه ذخیره کرده و سپس محتویات آنرا یک خانه به سمت چپ بصورت چرخشی جابجا نماید.
(برای مشاهده نتیجه،بعد از هر بار جابجایی، دکمه چاپ را بزنید)



```
Option Explicit
Option Base 1 → اندیس آرایه از ۱ شروع شود
Dim A(10) As Integer → تعریف آرایه ۱۰ عنصری

Private Sub Command1_Click()→ ذخیره اعداد ۱ تا ۱۰ در آرایه
   Dim i As Integer
   For i = 1 To 10
      A(i) = i
   Next i
End Sub

Private Sub Command2_Click()→ چاپ آرایه
   Dim i As Integer
   Print
   For i = 1 To 10
      Print A(i);
   Next i
End Sub

Private Sub Command3_Click()→جابجایی عناصر آرایه
   Dim i As Integer, t As Integer
   t = A(1) → ذخیره عنصر اول در یک متغیر کمکی
   For i = 2 To 10 → حلقه پیمایش آرایه
      A(i - 1) = A(i) → انتقال هر عنصر به محل قبل از خود(چپ)
   Next i
   A(10) = t → ذخیره عنصر اول در محل آخر
End Sub
```

تمرین: در دکمه ای دیگر عناصر را به صورت چرخشی به سمت راست جابجا نمایید.

آرایه با طول متغیر

در حالت معمولی بعد از تعریف آرایه، تغییر تعداد عناصر آن ممکن نیست. برای اینکه در برنامه بتوانیم بر حسب نیاز طول آرایه را تغییر دهیم، نیاز به شیوه دیگری داریم:

۱- آرایه را در شروع بصورت زیر تعریف می کنیم که تعداد عناصر نامعلوم است:

Dim نام آرایه() As نوع

۲- در هر جای برنامه که لازم شد، آرایه را با طول جدید بصورت زیر تعریف می نماییم:

Redim نام آرایه(تعداد مورد نظر) {شیوه تعریف}

توجه مهم: اگر در قسمت {شیوه تعریف}، کلمه Preserve را بنویسیم محتویات قبلی نیز حفظ می شود و اگر این قسمت خالی باشد، محتویات قبلی آرایه پاک می شود.

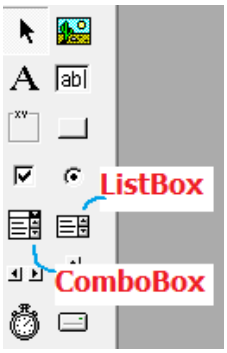
مثال: به کمک آرایه با طول متغیر، نخست آرایه ای با اندازه ۵ تعریف نموده، سپس طول آنرا به ۱۰ افزایش داده و در دو حالت استفاده از Preserve و عدم استفاده از آن، نتیجه را نمایش دهید

```
Option Explicit
Option Base 1 → اندیس آرایه از ۱ شروع شود
Dim A() As Integer → تعریف آرایه بدون تعیین تعداد عناصر

Private Sub Command1_Click()→ تعریف آرایه با ۵ عنصر
   ReDim A(5) As Integer → تعریف مجدد آرایه با ۵ عنصر
   Dim i As Integer
   For i = 1 To 5 → ذخیره اعداد ۱ تا ۵ در آرایه
      A(i) = i
   Next i
End Sub

Private Sub Command2_Click()→ چاپ آرایه
   Dim i As Integer
   Print
   پیمایش آرایه از اندیس شروع تا پایان
```

```
   For i = LBound(A) To UBound(A)
      Print A(i);
   Next i
End Sub

Private Sub Command3_Click()→preserve  تعریف آرایه با ۱۰ عنصر با
   ReDim Preserve A(10)→ تعریف مجدد آرایه با ۱۰ عنصر و حفظ مقادیر قبلی
   Dim i As Integer
   For i = 6 To 10  → ذخیره اعداد ۶ تا ۱۰ در بقیه خانه های اضافه شده به آرایه
      A(i) = i
   Next i
End Sub
Private Sub Command۴_Click()→preserve  تعریف آرایه با ۱۰ عنصر بدون
   ReDim A(10)→ تعریف مجدد آرایه با ۱۰ عنصر و بدون حفظ مقادیر قبلی
   Dim i As Integer
   For i = 6 To 10  → ذخیره اعداد ۶ تا ۱۰ در بقیه خانه های اضافه شده به آرایه
      A(i) = i
   Next i
End Sub
```

آشنایی با کنترلهای جدید

- **کادر لیست (Listbox)**
- **کادر لیست بازشو یا ترکیبی(Combobox)؛** که ترکیبی از Listbox و textbox است



هر دو برای گروه بندی و لیست بندی اطلاعات بکار می روند. اطلاعات آنها در یک آرایه رشته ای به نام List ذخیره می گردد.

تفاوت آنها

- در کادر لیست، لیست عناصر معلوم است ولی در لیست باز شو، فقط یک عنصر دیده می شود.
- در کادر لیست، امکان نمایش چند ستونی وجود دارد، اما در لیست بازشو چنین نیست.

مشخصه های مشترک

- List: آرایه ای رشته ای برای ذخیره اطلاعات
- ListCount: تعداد عناصری که در لیست ذخیره شده است
- ListIndex: شماره عنصر انتخاب شده(از شماره صفر تا (ListCount - 1))
- Sorted: اگر true باشد، لیست داده ها را مرتب می کند.

روالهای مشترک

- Clear: پاک کردن همه محتویات
- AddItem: افزودن رشته جدید
- RemoveItem: حذف سطر انتخاب شده
- ...

اضافه کردن داده ها به لیست: به دو روش امکان پذیر است:

۱. هنگام طراحی فرم، داده ها را در مشخصه List بصورت دستی وارد کنیم. بعد از وارد کردن هر داده، Ctrl+enter را بزنیم

۲. در دستورات برنامه از ساختار زیر استفاده نماییم:

نام کنترل . AddItem داده مورد نظر
List1.AddItem "ali" مثلاً

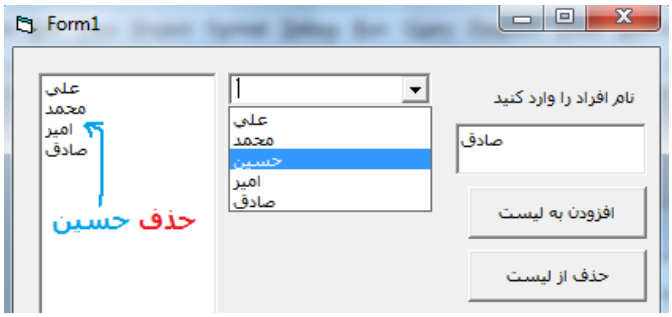
حذف کردن داده ها از لیست: از دستور زیر استفاده نماییم:

نام کنترل . RemoveItem شماره عنصر مورد نظر

مثلاً: حذف عنصر سطر انتخاب شده List1. RemoveItem (List1. ListIndex)

مثال ۳: در یک کادر متن نام افراد را وارد کرده و در کنترلهای لیست و لیست بازشو ذخیره نمایید و با کلیک روی دکمه حذف، عنصر انتخاب شده لیست حذف گردد.

```
Private Sub Command1_Click()→ افزودن به لیست
   List1.AddItem Text1.Text
   Combo1.AddItem Text1.Text
End Sub

Private Sub Command۲_Click()→ حذف از لیست
   List1.RemoveItem List1.ListIndex
End Sub
```



شیوه های Combobox : سه حالت دارد که توسط مشخصه style تعیین می شود:

۱- حالت صفر(۰): لیست بازشو و امکان اضافه کردن با تایپ

۲- حالت یک(۱): از حالت بازشو خارج شده و شبیه یک TextBox و listbox می شود. می توان از آن انتخاب کرد یا در آن تایپ نمود

۳- حالت دو(۲): فقط خواندنی؛ امکان اضافه کردن با تایپ نداریم ولی انتخاب ممکن است.

یادآوری: در زمان طراحی، در مشخصه List مقادیر را وارد کنید و بعد از هر مورد Ctrl+Enter بزنید

تمرین۱: مثال صفحه ۱۰۲

مقایسه رشته ها(یادآوری برنامه سازی ۱)؛ ترتیب الفبایی متن با دستور :

متغیر عددی=strComp(رشته اول , رشته دوم , نوع مقایسه)

نوع مقایسه:

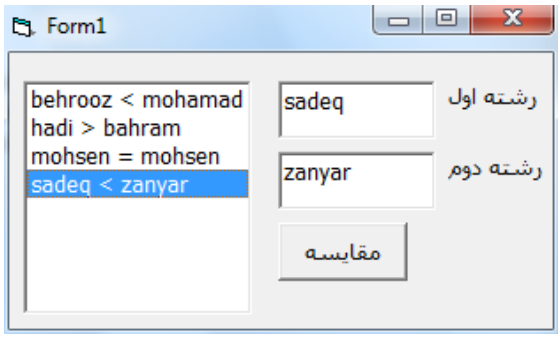
- حالت صفر: به حرف کوچک و بزرگ انگلیسی حساس است(مثلاً ali با Ali متفاوت است)
- در حالت یک: حساس نیست. اگر نوع مقایسه مشخص نشود پیش فرض حالت یک است.

متغیر عددی: جواب مقایسه :

- (۰)صفر: مساوی بودن دو رشته؛ رشته دوم = رشته اول
- **۱ :** رشته دوم > رشته اول ؛ ترتیب الفبایی رشته اول بعد از دومی است
- ۱- : رشته دوم < رشته اول؛ ترتیب الفبایی رشته اول قبل از دومی است

مثال: مقایسه دو رشته و نمایش نتیجه در یک Listbox

```
Private Sub Command1_Click()→ دکمه مقایسه
   Dim c As Integer
   c = StrComp(Text1.Text , Text2.Text ,1)→ انجام مقایسه
   Select  Case   c        → بررسی نتیجه مقایسه
      Case 0:  List1.AddItem Text1.Text & " = " & Text2.Text
      Case 1:  List1.AddItem Text1.Text & " > " & Text2.Text
      Case -1: List1.AddItem Text1.Text & " < " & Text2.Text
   End Select
End Sub
```



آرایه های کنترلی

اگر چندین کنترل مختلف ولی همنام و هم نوع روی فرم داشته باشیم، به آنها آرایه کنترلی می گویند. این کنترلها توسط مشخصه index از هم تفکیک می گردند.

ایجاد آرایه های کنترلی

روش اول: ایجاد آنها در زمان طراحی

۱. طراحی کنترلها:
   - قرار دادن کنترل روی فرم و تعیین نام آن
   - کپی گرفتن از آن با copy
   - Paste کردن کنترل و زدن Yes در کادر پیام ظاهر شده

مثلاً اگر دکمه ای به نام Cmd ایجاد کرده باشیم و مراحل فوق را انجام دهیم، دو کنترل به نامهای Cmd(0), Cmd(1) ایجاد خواهند شد.

اگر بخواهیم کد رویداد آنها را بنویسیم با کلیک روی یکی از آنها رویداد بصورت زیر خواهد شد:

```
Private Sub Cmd_Click(Index as Integer)→ رویداد دکمه ها بصورت مشترک
                                        ➔ Index: پارامتری که شماره دکمه را در خود دارد
    Cmd(Index).Caption = Index  → مثلاً تنظیم عنوان هر دکمه با شماره آن
End Sub
```

مثال: طراحی یک برنامه ماشین حساب به کمک آرایه کنترلی

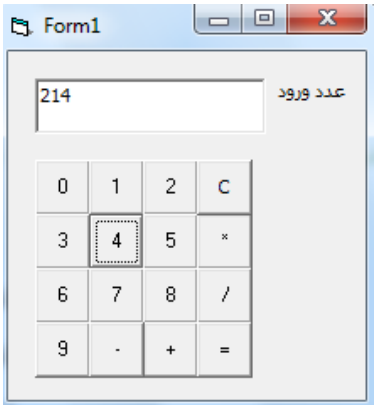
مراحل کار:

۱. ایجاد پروژه جدید و ذخیره در پوشه calc(نام پروژه: calcPrj و نام فرم CalcFrm)
۲. قرار دادن یک دکمه با نام Cmd
۳. کپی گرفتن از آن(copy)
۴. Paste کردن کنترل و زدن Yes در کادر پیام(۱۴ بار paste)
   - شماره ۰ تا ۹ برای رقم ها
   - ۱۰: + ، ۱۱: - ، ۱۲: * ، ۱۳: / ، ۱۴: =
۵. تنظیم caption هر دکمه
۶. قرار دادن یک textbox به نام txtValue و یک برچسب Label با عنوان عدد ورودی
۷. نوشتن برنامه در سه رویداد:
   - تعریف متغیرهای سراسری General
   - رویداد کلیک یک دکمه شامل اعداد، عملگر و مساوی
۸. افزودن دکمه حذف یک رقم به نام DelDigit و با عنوان C
۹. دکمه خروج

```
Option Explicit
Dim Value1 As Long →مقدار اول
Dim Value2 As Long →مقدار دوم
Dim Result As Single  →نتیجه
Dim Operator As String * 1 →عملگر

Private Sub Cmd_Click(Index As Integer)→ رویداد کلیک هر دکمه
 Select Case Index
    Case Is < 10
      txtValue = txtValue & Index
   Case 10
         Operator = "+"
      Value1 = Val(txtValue.Text)
      txtValue.Text = ""
   Case 11
      Operator = "-"
      Value1 = Val(txtValue.Text)
      txtValue.Text = ""
   Case 12
      Operator = "*"
```

```
Private Sub DelDigit_Click() →C  دکمه
   Dim a As String
    a = txtValue.Text
   If Len(a) > 0 Then
      a = Mid(a, 1, Len(a) - 1)
   End If
   txtValue.Text = a
End Sub
```

```
    Value1 = Val(txtValue.Text)
    txtValue.Text = ""
  Case 13
    Operator = "/"
    Value1 = Val(txtValue.Text)
    txtValue.Text = ""
  Case 14
    Value2 = Val(txtValue.Text)
    txtValue.Text = ""
    Select Case Operator
    Case "+"
      Result = Value1 + Value2
    Case "-"
      Result = Value1 - Value2
    Case "*"
      Result = Value1 * Value2
    Case "/"
      Result = Value1 / Value2
    End Select
    txtValue.Text = Result
 End Select
End Sub
```



روش دوم: ایجاد آرایه های کنترلی در زمان اجرا

می خواهیم کنترلهایی را از روی یک نمونه موجود روی فرم ، مثلاً با زدن یک دکمه، ایجاد کنیم.

با دستور:

Load نام کنترل(شماره بعدی) → ایجاد نمونه دیگر از کنترل با شماره بعدی

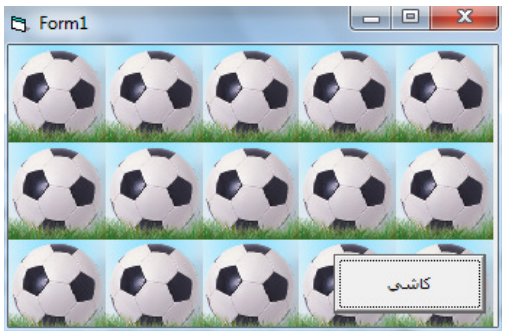
توجه کنید که برای کنترل موجود باید نخست مشخصه Index را صفر کنید و سپس دستور بالا را روی آن اجرا نمایید. همچنین باید حتماً محل آنرا توسط مشخصه های Top و Left مشخص نموده و برای مشاهده آن مشخصه visible را برابر True کنید.

مثال: فرض کنید می خواهید کل صفحه را با یک تصویر شبیه کاشی پر کنید.

مراحل کار:

۱. قرار دادن کنترل تصویر(Image) روی فرم و با نام Img و تنظیم تصویر آن
۲. تغییر مشخصه Index کنترل تصویر به صفر
۳. تعریف یک متغیر برای شمارش در قسمت General بصورت : Dim n As Integer و در رویداد Form_load() می نویسیم: n=0
۴. در رویداد یک دکمه دستورات را می نویسیم:

```
Option Explicit
Dim n As Integer

Private Sub Form_Load()
   n = 0
   Img(0).Left = 0
   Img(0).Top = 0
End Sub

Private Sub Command1_Click()→ دکمه کاشی
 n = n + 1                    → شماره بعدی
Load Img(n)                   → ایجاد کنترل بعدی
Img(n).Visible = True         → نمایش آن
Img(n).Left = Img(n - 1).Left + Img(n - 1).Width   →تعیین فاصله از چپ کنترل جدید بعد از کنترل قبلی
Img(n).Top = Img(n - 1).Top                        → تعیین فاصله از بالا، همانند کنترل قبلی
If (Img(n).Left + Img(n).Width > Form1.Width) Then →اگر کنترل جدید از حاشیه فرم خارج شد
    Img(n).Top = Img(n).Top + Img(n).Height        →در خط بعد نمایش بده
   Img(n).Left = 0  →اول خط
End If
End Sub
```



کنترلهای نوار لغزان(Scroll bar): برای جابجایی و تغییر داده ها یا نمایش محدوده ها
- HScrollBar: افقی
- VScrollBar: عمودی

مشخصه های مهم:
- Min کمترین مقدار
- Max بیشترین مقدار
- Value مقدار فعلی
- Largechange میزان جابجایی با کلیک روی نوار
- SmallChange میزان جابجایی وقتی روی فلش ها کلیک کنیم.

تمرین: مثال صفحه ۱۱۲و۱۱۴

مرتب سازی داده ها(Sort): هدف تغییر مکان داده ها در یک لیست(آرایه) است طوریکه دارای نظم خاصی شوند. دو نوع نظم داریم:
- **صعودی**: از کوچک به بزرگ، مثل: 5,7,12,20
- **نزولی**: از بزرگ به کوچک، مثل: 20,12,7,5

اعمال اصلی مرتب سازی عبارتند از:
۱. **مقایسه**: مقایسه عناصر لیست با هم تا کوچکتر یا بزرگتر تعیین شود
۲. **جابجایی**: جابجایی و تعویض محل آنها در لیست طبق روش خاص

روش مرتب سازی حبابی(bubble sort) : فرض بر این است که میخواهیم لیست را به صورت صعودی مرتب کنیم ،مراحل کار بصورت زیر است:

۱. همه عناصر لیست را از ابتدا تا انتها بررسی کرده و آنها را دو به دو مقایسه می کنیم و اگر عنصر سمت چپ بزرگتر بود آنها را جابجا میکنیم. با یک بار پیمایش لیست بزرگترین عنصر هر کجا باشد خود را به آخر لیست می رساند.
۲. مرحله یک را تکرار میکنیم، اما تا یکی مانده به آخر، مقایسه ها و جابجایی ها را انجام می دهیم.
۳. مراحل فوق هر بار تا یکی مانده به آخر بقیه لیست ادامه دارد تا وقتی که آرایه تمام شود یعنی یک خانه آرایه باقی بماند.

مراحل	۵	۱۵	۴	۱	۸
۱	۵	۴	۱۵	۱	۸
	۵	۴	۱	۱۵	۸
	۵	۴	۱	۸	۱۵
۲	۴	۵	۱	۸	۱۵
	۴	۱	۵	۸	۱۵
	۴	۱	۵	۸	۱۵
۳	۱	۴	۵	۸	۱۵
	۱	۴	۵	۸	۱۵
۴	۱	۴	۵	۸	۱۵

برآورد تعداد مقایسه ها : با فرض اینکه طول آرایه n باشد در مرحله اول n-1 مقایسه، در مرحله دوم n-2 مقایسه، در مرحله سوم n-3 مقایسه و ... انجام می دهیم.
اگر همه مقایسه ها را با هم جمع کنیم:
تعداد کل مقایسه ها = (n-1) + (n-2) + (n-3) + …+ 2 + 1
→ تعداد کل مقایسه ها = n*(n-1)/2

برآورد تعداد جابه جایی ها : قابل محاسبه نیست چون بستگی به نحوه چیدمان عناصر در آرایه اولیه دارد ولی قابل شمارش است.
مثال:
مثلاً: n=5 → تعداد مقایسه ها = 5*(5-1)/2 = 20/2=10 → محاسبه با فرمول
با شمارش آنها → تعداد جابجایی = 6

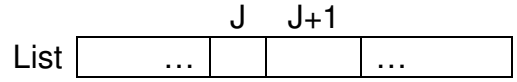
یادآوری: توسط دستورات زیر می توان محدوده یک آرایه را بدست آورد:
LBound(نام آرایه)→ حد پایین آرایه
UBound(نام آرایه)→ حد بالای آرایه

مثال:
Dim A(4 to 29) as integer → LBound(A) = 4, UBound(A) = 29

توسط دستور زیر می توان محتویات یک آرایه را پاک نمود:
Erase نام آرایه → مثال: Erase A تمام عناصر را برابر صفر قرار می دهد.

برنامه پیاده سازی مرتب سازی حبابی:

Dim List(10) As Integer → تعریف آرایه در اول برنامه



بعد از مقایسه دو محل J,J+1 ، برای جابجایی این دو محل از دستورات زیر استفاده می شود:

```
If List(J) > List(J + 1)  Then          → مقایسه هر عنصر با بعدی خود
        T = List(J)                     → دستورات جابجایی به کمک ظرف کمکی T
        List(J) = List(J + 1)
        List(J + 1) = T
End If
```

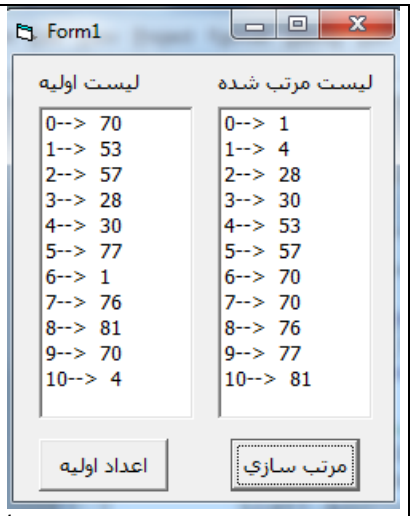
پس برنامه کلی مرتب سازی به شکل زیر خواهد بود:

```
Dim I As Integer, J As Integer, T As Integer    →  متغیرهای لازم
  For I = UBound(List) To LBound(List) Step -1  →  کنترل تعداد مراحل
    For J = LBound(List) To  I – 1              → بررسی عناصر از اول تا یکی مانده به آخر
      If List(J) > List(J + 1)  Then            → مقایسه دو به دو هر عنصر با بعدی خود
        T = List(J)                             → دستورات جابجایی به کمک ظرف کمکی T
        List(J) = List(J + 1)
        List(J + 1) = T
      End If
    Next J
  Next I
```

مثال عملی:

۱- پروژه جدید و قرار دادن دو کنترل ListBox و دو دکمه به نامهای cmdGen و CmdSort روی فرم
۲- تعریف آرایه در بخش General در اول برنامه
۲- نوشتن دستورات پرکردن آرایه با عناصر تصادفی در دکمه cmdGen:
۳- نوشتن دستورات برنامه مرتب سازی به روش حبابی در دکمه CmdSort:

```
Dim List(10) As Integer  → تعریف آرایه در اول برنامه

Private Sub cmdGen _Click()
   Dim I As Integer
   List1.Clear
   Erase List  →  پاک کردن آرایه
   For I = LBound(List) To UBound(List)
      List(I) = Int(Rnd * 100)   → پرکردن آرایه با عناصر تصادفی
      List1.AddItem CStr(I) & "-->  " & CStr(List(I)) →List1  اضافه کردن عناصر به
   Next
   List2.Clear
End sub

Private Sub CmdSort _Click()
   Dim I As Integer, J As Integer, T As Integer      →   متغیرهای لازم
   For I = UBound(List) To LBound(List) Step -1   →   کنترل تعداد مراحل
      For J = LBound(List) To   I – 1                 →بررسی عناصر از اول تا یکی مانده به آخر
         If List(J) > List(J + 1)  Then               → مقایسه دو به دو هر عنصر با بعدی خود
            T = List(J)                               → دستورات جابجایی به کمک ظرف کمکی T
            List(J) = List(J + 1)
            List(J + 1) = T
         End If
      Next J
   Next I
   List2.Clear
   For I = LBound(List) To UBound(List)
     →List2  ذخیره عناصر آرایه مرتب شده در
      List2.AddItem    CStr(I) & "-->  " & CStr(List(I))
   Next I
End sub
```



جستجو در آرایه :در پیدا کردن محل یک عنصر در آرایه یا لیست کاربرد دارد.

دو روش وجود دارد:

۱. **خطی**: ابتدایی ترین روش جستجو بوده و در صورت مرتب نبودن لیست از آن استفاده می شود.(البته اگر لیست هم مرتب باشد قابل استفاده است) .

۲. **دودویی**: فقط بر روی لیست های مرتب کاربرد دارد.

شرح روش خطی: ایده کار به این صورت است که از اول تا آخر لیست به دنبال عنصر گمشده می گردیم . اگر طول لیست N باشد:

- بهترین حالت : در اولین مقایسه پیدا شود(اولین عنصر باشد) : ۱ مقایسه
- بدترین حالت : عنصر گمشده در لیست نباشد یا آخرین عنصر باشد: N مقایسه
- متوسط: در اواسط لیست باشد: N/2 مقایسه

مثال:

۱- در پروژه جدید، یک ListBox، ، یک TextBox و دو دکمه CmdGen, CmdSearch قرار دهید.

۲- در قسمت General: Dim List(10) As Integer

۳- در دکمه CmdGen دستوراتی را بنویسید که لیست را با عناصر تصادفی پر می کند.

۴- در دکمه CmdSearch دستورات جستجو را بنویسید:

```
Private Sub cmdGen _Click()
   Dim I As Integer
   List1.Clear
   Erase List                                  → پاک کردن آرایه
   For I = LBound(List) To UBound(List)
      List(I) = Int(Rnd * 100)                 → پرکردن آرایه با عناصر تصادفی
      List1.AddItem Cstr(i) & " :  " & cstr(List(I))   → List1 اضافه کردن عناصر به
   Next
End sub

Private Sub CmdSearch _Click()
    Dim Key As Integer, I As Integer, Index As Integer
   Dim Found As Boolean                        → متغیر برای کنترل ادامه جستجو
   Key = Val(Text1.Text)                       → خواندن مقدار کلید جستجو
   Found = False                               → فعلاً یافت نشده است
   For I = LBound(List) To UBound(List)        → پیمایش همه لیست
      If (List (I) = Key) Then                 → مقایسه هر عنصر با کلید جستجو
         Found = True                          → پیدا شد
         Index = I                             → ذخیره محل پیدا شدن
         Exit For                              → خروج از حلقه
      End If
   Next
   If Found = True Then
      MsgBox "یافت شد در محل : " & CStr(Index)
   Else
      MsgBox "یافت می نشود!"
   End If
End sub
```

روش جستجوی دودویی: با فرض اینکه داده ها به صورت صعودی مرتب شده اند، ایده کار به صورت زیر است . ابتدا عنصر وسط را پیدا میکنیم
مقایسه عنصر وسط با کلید جستجو که سه حالت پیش می آید :

۱. عضو وسط = کلید جستجو : عنصر پیدا شد و کار تمام است.
۲. عضو وسط > کلید جستجو : در این حالت عنصر مورد نظر در سمت راست نیست و باید در سمت چپ آنرا جستجو کنیم

نیمه چپ	وسط	نیمه راست

۳. عضو وسط < کلید جستجو : عنصر در سمت چپ نیست و باید در سمت راست آنرا جستجو کنیم

نیمه راست	وسط	نیمه چپ

در بدترین حالت تعداد مقایسه های لازم از فرمول زیر بدست می آید: اگر n تعداد عناصر باشد:
تعداد مقایسه = $\log_2 n$ → مثلاً n=1000 , $\log_2 1000 = 10$

مثال : در آرایه موجود عدد ۳۰ را پیدا کنید .

عناصر آرایه										محدوده شروع و پایان		محاسبه وسط(M)	مقایسه با عنصر وسط
۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰	از(L)	تا(R)		
۳	۵	۱۰	۱۲	۱۵	۳۰	۳۳	۳۷	۴۲	۴۶	۱	۱۰	M=(1+10)/2 = 5	۱۵<۳۰
					۳۰	۳۳	۳۷	۴۲	۴۶	۶	۱۰	M=(6+10)/2 = 8	۳۷>۳۰
					۳۰	۳۳				۶	۷	M=(6+7)/2 = 6	۳۰=۳۰

مثال پیاده سازی:

در یک پروژه جدید،

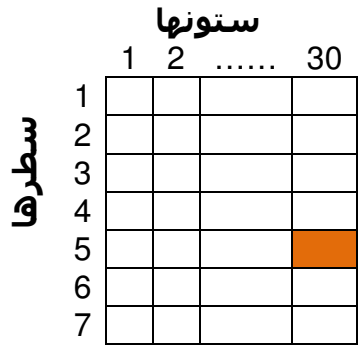
- یک ListBox و یک Textقرار دهید
- دکمه ای با نام cmdGen برای ایجاد داده ها(دستورات مشابه روش جستجوی خطی)، دکمه ای به نام CmdSort برای مرتب سازی(با دستورات مثال قبلی مرتب سازی) و دکمه با نام cmdSearch قرار دهید.

```
Private Sub CmdSearch _Click()
   Dim Key As Integer, L  As Integer, R As Integer, M  As Integer
   Dim Found As Boolean
   Key = Val(Text1.Text)          → عدد مورد جستجو
   Found = False                  →  فعلاً پیدا نشده
  L = LBound(List)                → محدوده شروع
  R = UBound(List)                → محدوده پایان
      →تا زمانیکه پیدا نشده و لیست تمام نشده است ادامه بده
   Do While (L <= R) And (Not Found)
      M = (L + R) \ 2             →محل عنصر وسط
      If (Key < List(M)) Then     → اگر عدد از  عنصر وسط کوچکتر بود
         R = M - 1
      ElseIf (Key > List(M)) Then  →اگر عدد از  عنصر وسط بزرگتر بود
         L = M + 1
      Else                        →اگر عدد با  عنصر وسط برابر بود
         Found = True             → یافت شد
      End If
   Loop

   If Found = True Then
      MsgBox "یافت شد در محل : " & CStr(M)
   Else
      MsgBox "یافت می نشود!"
   End If
End sub
```

آرایه های چند بعدی

تا کنون با آرایه های خطی یا یک بعدی(یک سطری) آشنا شدیم. گاهی لازم است برای ذخیره داده ها از آرایه های دو بعدی، سه بعدی و ... نیز استفاده کنیم. مثلاً برای ذخیره نمرات دانش آموزان یک کلاس، آرایه یک بعدی کافی است اما اگر بخواهیم نمرات دانش آموزان همه کلاسها را ذخیره کنیم، به آرایه دو بعدی (شبیه ماتریس)نیاز داریم که از تعدادی سطر و هر سطر از تعدادی ستون تشکیل شده است.

- سطرها: تعداد کلاسها، مثلاً ۷ کلاس
- ستونها: تعداد دانش آموزان هر کلاس مثلاً ۳۰ نفر

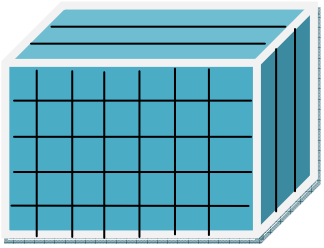


برای تعریف آرایه دو بعدی به شکل زیر عمل می کنیم:

Dim نام آرایه(تعداد ستون و تعداد سطر) as نوع داده ها

Dim nomre(7,30) as single → آرایه دو بعدی برای ذخیره نمرات دانش آموزان همه کلاسها

Print nomre(2,25) → مثال: چاپ نمره دانش آموز شماره ۲۵ از کلاس شماره ۲



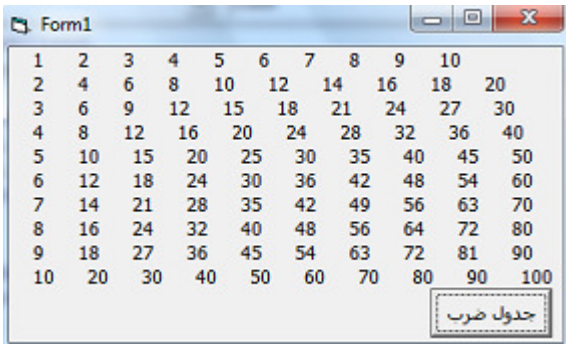
برای تعریف آرایه های سه بعدی و بیشتر هم بدین صورت عمل می کنیم. آرایه سه بعدی را می توان تعدادی صفحه در نظر گرفت که هر صفحه خود از سطرها و ستونها تشکیل شده است:

Dim نام آرایه(تعداد ستون و تعداد سطر و تعداد صفحه) as نوع داده ها

آرایه سه بعدی با ۵ صفحه، هر صفحه نیز ۷ سطر و ۳ ستون دارد → Dim A(5,7,3) as integer

مثال: جدول ضرب را در آرایه ای ذخیره کرده و روی فرم چاپ کنید

```
Option Explicit
Dim jadval(10, 10) As Integer

Private Sub Command1_Click → دکمه جدول ضرب
Dim i As Integer, j As Integer
For i = 1 To 10
   For j = 1 To 10
      jadval(i, j) = i * j → جدول ضرب در آرایه دو بعدی
     Print Spc(2); jadval(i, j); → چاپ عناصر آرایه با دو فاصله
  Next j
     Print
 Next i
End Sub
```

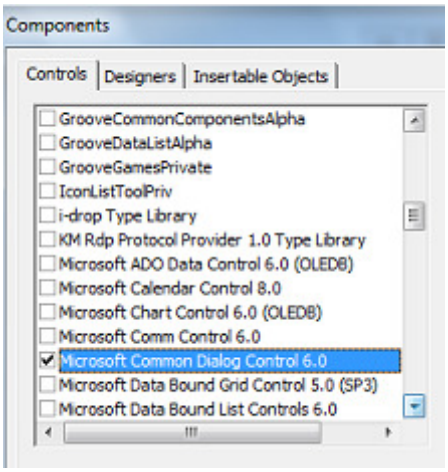


فصل پنجم-فرمهای آماده

Project→ Add Form→ انتخاب نوع فرم

انواع فرم آماده:
- About: درباره برنامه حاوی اطلاعات برنامه مانند نسخه ، تاریخ، تولید کننده و ...
- Login: دریافت نام کاربری و رمز جهت ورود به برنامه
- Splash: ظاهر شدن پیامی در اول اجرای برنامه(بدون نوار عنوان)
- ...

نکته: برای اینکه فرمی در اول اجرای برنامه ظاهر شود در قسمت زیر باید فرم مورد نظر انتخاب شود:
Project→ properties→startup Object



کنترل Common Dialog

این کنترل، کادر محاوره ای برای موارد زیر فراهم می کند.
- ذخیره(Save) و باز کردن فایل(Open)
- چاپ(print)
- انتخاب رنگ(Color)
- انتخاب فونت (Font)

نخست از مسیر زیر این کنترل را به جعبه ابزار اضافه نمایید:
Project→ Components→ Microsoft Common Dialog Control 6.0→ Apply, OK

سپس آنرا به فرم اضافه نمایید. مشخصات و ویژگیهای آن با کلیک راست و انتخاب گزینه properties یا زدن دکمه موجود در قسمت properties تنظیم می گردد.
خلاصه دستورات و تنظیمات بصورت زیر است:

۱. ذخیره فایل توسط دستور ShowSave و بازکردن فایل توسط دستور ShowOpen
با تنظیم مشخصات زیر:
- Filter: برای تنظیم فیلتر نوع فایل بصورت زیر:
" نوع فیلتر پسوند فایل|توضیحات" = Filter.نام کنترل
- FileName: نام فایلی که کاربر انتخاب کرده است
- InitDir: مسیر پوشه پیش فرض
- DialogTitle: عنوان کادر محاوره ای

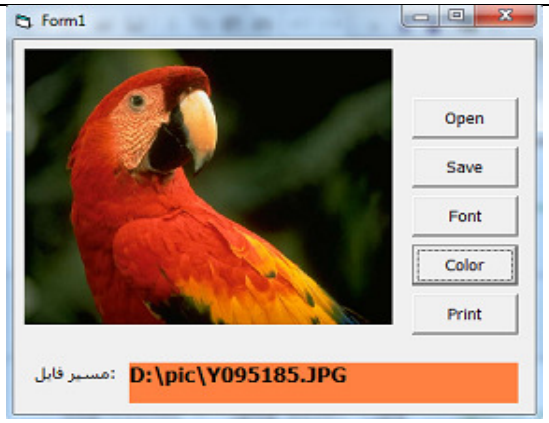
۲. انتخاب قلم و فونت توسط دستور ShowFont و مشخصه های زیر
- Flags: دارای مقادیر ۱(صفحه نمایش)، ۲(چاپگر)و ۳(هردو)
- FontName: نام فونت انتخاب شده
- FontSize: اندازه فونت انتخاب شده
- FontBold: حالت Bold انتخاب شده یا نه
- FontItalic: حالت Italic انتخاب شده یا نه
- FontUnderline: حالت UnderLine انتخاب شده یا نه

۳. انتخاب رنگ توسط دستور ShowColor و مشخصه های زیر
- Flags: دارای مقدار : CDLCCRGBInit
- Color: رنگ انتخاب شده

۴. انتخاب چاپگر و تنظیمات آن توسط دستور ShowPrinter و مشخصه های زیر

- Flags: نیاز به تنظیم ندارد
- Copies: تعداد کپی چاپ
- FromPage: از صفحه
- ToPage: تا صفحه

مثال:
در یک پروژه یک کنترل image(stretch = true) و یک کنترل Dialog به نام Dlg ، یک Label و پنج دکمه(با نامگذاری مطابق عملکرد آنها) اضافه نمایید.
در رویداد هر دکمه مطابق کاربرد دستورات آنرا بنویسید:




```
Private Sub CmdColor_Click()→رنگ
   Dlg.Flags =  CDLCCRGBInit →تنظیمات برای انتخاب رنگ
   Dlg.ShowColor              → ظاهر شدن پنجره انتخاب رنگ
   Label1.BackColor = Dlg.Color → Label  تنظیم رنگ پس زمینه
End Sub
Private Sub CmdFont_Click()→فونت
   Dlg.Flags = 1              → تنظیمات برای انتخاب فونت
   Dlg.ShowFont               → ظاهر شدن پنجره انتخاب فونت
   Label1.FontName = Dlg.FontName  → تنظیمات فونت برچسب
   Label1.FontSize = Dlg.FontSize
   Label1.FontBold = Dlg.FontBold
End Sub
Private Sub CmdOpen_Click()→بازکردن تصویر
    Dlg.Filter="jpeg File|*.jpg|bmp Files|*.bmp"   → محدود کردن نوع فایلها
   Dlg.InitDir="D:\pic "                          → مسیر پیش فرض
   Dlg.showOpen                                   →open باز کردن کادر
   If Dlg.FileName <> " "  then                    → اگر فایلی انتخاب شده بود
         Imag1.picture=loadPicture(Dlg.FileName)  → نمایش تصویر
   Else
         Imag1.picture=loadPicture("")            → پاک کردن تصویر
   End if
  Label1.Caption = Dlg.FileName                   →نمایش مسیر فایل تصویری در برچسب
End Sub
Private Sub CmdPrint_Click()→چاپگر
   Dlg.ShowPrinter   → نمایش تنظیمات چاپ
End Sub
Private Sub CmdSave_Click()→ذخیره
   Dlg.Filter = "jpeg File|*.jpg"    →  محدود کردن نوع فایلها
   Dlg.InitDir = "D:\pic"            → مسیر پیش فرض
   Dlg.ShowSave                      →save باز کردن کادر
   If Dlg.FileName <> "" Then    → اگر نام فایلی تعیین شده بود
      SavePicture Image1.Picture, Dlg.FileName →ذخیره تصویر  موجود در محل دلخواه انتخابی
   End If
   Label1.Caption = Dlg.FileName
End Sub
```

تمرین۱: در یک پروژه، فرمهای آماده About,splash,login را اضافه کرده و با اجرای فرمان هر دکمه آنها را نمایش دهید. می توانید هریک از فرمها را به دلخواه دستکاری نمایید.
تمرین۲: پیاده سازی مثال فوق

برنامه های چند سندی ساده Multiple Document Interface(MDI)
برنامه ای است که همه پنجره های آن داخل یک پنجره اصلی باز می شوند.
مراحل ایجاد:
۱. ایجاد پروژه جدید
۲. ذخیره فرم با نام FrmChild و تنظیم مشخصه MDIChild = True
۳. افزودن فرم MDI جدید: Project→ Add MDI Form
۴. تنظیم فرم جدید به عنوان فرم اصلی اجرای برنامه

Project→properties→ startup→MDIForm1

۵. ایجاد منوهای زیر

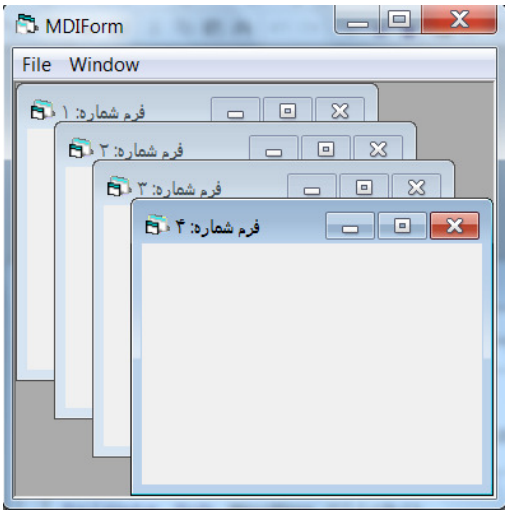
File → new , Exit
Window→cascade, horizontal, Vertical

۶. نوشتن دستورات ایجاد فرمهای جدید در منوی new
۷. توسط دستور Arrange از فرم MDI اصلی نحوه چیدمان فرمها را تنظیم می کنیم هر یک در منوی مربوطه Window:
   - *vbCascade : چیدمان آبشاری*
   - *vbTileHorizontal : چیدمان افقی*
   - *vbTileVertical : چیدمان عمودی*

نکته: خاصیت AutoShowChildren از فرم اصلی MDI می تواند دارای دو مقدار باشد

- False: در این صورت فرم های داخل آن خود بخود دیده نمی شوند و باید با دستور Show آنها را نمایش داد.
- True : نیازی به دستور Show برای نمایش فرم های داخلی نیست.

```
Option Explicit

Private Sub MnuNew_Click()
  Dim newForm As New frmChild   → تعریف فرم جدید
  Dim fn As Integer             →برای شمارش فرمها
  Load newForm                  → بارگذاری فرم جدید
  fn = Forms.Count – 1          → بدست آوردن تعداد کل فرمهای پروژه
 newForm.Caption = "فرم شماره: " & fn → تنظیم عنوان فرم با شماره آن
End Sub

Private Sub mnuCascade_Click()
   MDIForm1.Arrange   vbCascade
End Sub

Private Sub mnuHorizontal_Click()
   MDIForm1.Arrange   vbTileHorizontal
End Sub

Private Sub mnuVertical_Click()
   MDIForm1.Arrange   vbTileVertical
End Sub

Private Sub mnuExit_Click()
   End
End Sub
```



توجه: فقط کنترلهای Common Dialog,Picture box, Timer را میتوان روی فرمMDI قرار داد. برای استفاده از کنترلهای دیگر نخست باید کنترل Picture را به فرم افزوده و سپس کنترلهای دیگر را روی آن قرار داد.

فصل ششم -زمان و تاریخ

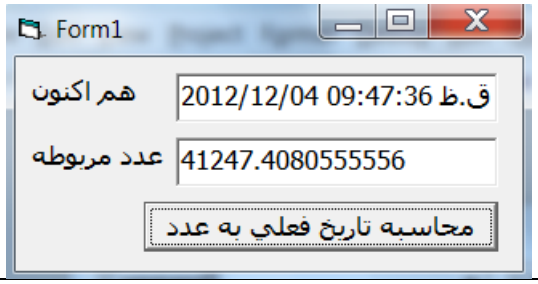
واحد زمان و تاریخ در VB بر حسب روز است. و در قالب عددی اعشاری بیان می شود. قسمت صحیح تاریخ و قسمت اعشاری زمان است.

زمان و تاریخ	عدد تاریخ
یک روز	۱
یک هفته	۷
یک ساعت	۱/۲۴
یک ثانیه	۱/۸۶۴۰۰
۱۸۹۹/۱۲/۳۱ نیمه شب	۰
ساعت ۱۲ ظهر ۱۹۰۰/۰۱/۱۵	۱۵.۵
۰۴/۱۲/۲۰۱۲ ۰۹:۱۹:۴۵ قبل از ظهر	۴۱۲۴۷.۳۸۸۷۱۵۲۷۷۸

نکته: مقادیر تاریخ را می توان در بین علامت # در متغیر از نوع Date ذخیره نمود:

```
Dim  d  as date
D= # December 4,2012  9:52 am#
```

```
Private Sub MnuNew_Click()
   Dim d As Date
   d = Now()  → تاریخ و زمان فعلی
   Text1.Text = CStr(d)
   Text2.Text = Format(d, "General number")
End Sub
```



کنترل Timer

برای اندازه گیری فواصل زمانی اجرای دستورات؛ یعنی هر چند وقت یکبار دستورات اجرا شوند.

خصوصیات مهم:

- **Interval**: اندازه گیری فواصل زمانی بر حسب میلی ثانیه(هزارم ثانیه) مثلاً ۱۰۰۰=۱ ثانیه
  مقادیر : ۵۵ الی ۶۵۵۳۵
- **Enabled**: فعال(true) و غیر فعال(False)

رویداد : با کلیک روی آن ،Timer() حاوی دستورات برنامه

نکته: توسط مشخصه Enabled می توان کار کردن و کار نکردن Timer را کنترل کنیم:

Timer1.Enabled = False → دستورات داخل آن اجرا نمی شوند

Timer1.Enabled = True → دستورات داخل آن اجرا می شوند

مثال صفحه ۱۵۵: حرکت دادن یک Label به کمک Timer

کنترل Timer را با Interval دلخواه غیر صفر مقدار دهی کنید و در رویداد Timer() آن دستورات زیر را بنویسید.

تمرین: صفحه ۱۵۵

بدست آوردن زمان و تاریخ سیستم

- زمان: با دستور Time()
- تاریخ: با دستور Date()
- هردو با هم: با دستور Now()

مثال: صفحه ۱۵۶و ۱۵۷

توضیح: توسط خصوصیت WindowState از فرم می توان وضعیت فرم را بررسی نمود:

- VbNormal: حالت عادی
- VbMinimized: حالت کمینه
- VbMaximized: حالت بیشینه

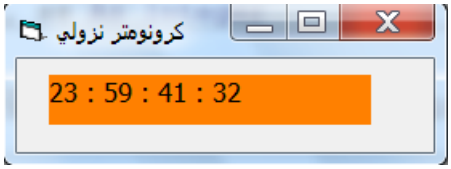
مثال: پروژه اي طراحي كنيد كه متني دلخواه درعنوان فرم بصورت متحرک از چپ به راست حرکت نماید. یک Timer با interval=100

```
Option Explicit
Dim MyText As String
Private Sub Form_Load()
   MyText = "دانه دانه خال دانه دانه"
End Sub

Private Sub Timer1_Timer()
   MyText = Right(MyText, 1) & Left(MyText, Len(MyText) - 1)
   Caption = MyText
End Sub
```

تمرین: چراغ راهنمایی را شبیه سازی کنید. سه کنترل Label قرار دهید که رنگ زمینه آنها نشان دهنده رنگ چراغ باشد. چراغ سبز ۲۰ ثانیه، بعد چراغ زرد ۴ ثانیه و سپس چراغ قرمز ۱۵ ثانیه روشن بماند و دوباره این روال تکرار شود.

مثال: طراحی کرونومتر نزولی

```
Option Explicit
Dim h As Integer, m As Integer, s As Integer, ss As Integer

Private Sub Form_Load()
   h = 23 :   m = 59 :   s = 59:    ss = 100
End Sub

Private Sub Timer1_Timer()
   ss = ss - 1
   If (ss = 0) Then
      ss = 100
      s = s - 1
   End If
   If (s = 0) Then
      s = 59
      m = m - 1
   End If
   If (m = 0) Then
      m = 59
      h = h - 1
   End If
   If (h = 0) Then h = 23
   Label1.Caption = h & " : " & m & " : " & s & " : " & ss
End Sub
```



برخی توابع

- تابع Format: برای قالب بندی داده ها طبق فرمت خاص که تنظیمات آن در جدولهای صفحه ۱۶۱ کتاب آمده است:

رشته خروجی = Format(عبارت, "قالب یا الگوی رشته")
مثال S= format(1234,"standard") → s= 1,234.00
S= format(date,"short Date") → s= 2010/12/06

- تابع DateDiff برای محاسبه اختلاف دو تاریخ: تنظیمات در جدول صفحه ۱۶۲ کتاب

عدد صحیح (long) = DateDiff (تاریخ پایان, تاریخ شروع, بر حسب)
مثال a= DateDiff("d", "2010/10/25","2010/12/09")→ a= 45 روز
a= DateDiff("h", "2010/10/25","2010/12/09")→ a= 1080 ساعت

جدول محاسبه اختلاف بر حسب مقادیر مختلف

سال	فصل	ماه	روز سال	روز	روزهفته	هفته	ساعت	دقیقه	ثانیه
yyyy	q	m	y	d	w	ww	h	n	S

- تابع IsDate برای بررسی درستی تاریخ

IsDate("2010/12/09") → True تاریخ درست

IsDate("shorbaw") → False مقدار تاریخ نیست

مثال: تاریخ تولد خود به میلادی را وارد کرده و سن خود را برحسب سال، روز ، ساعت و ثانیه در text ها نمایش دهید.

```
Private Sub cmdDate_Click()
   Dim bDate  As Date  →متغیر تاریخ
   If IsDate(Text1.Text) Then  →بررسی درستی تاریخ وارد شده
    bDate = CDate( Text1.Text)  →ذخیره تاریخ
   Else
    MsgBox "خطا: تاریخ نامعتبر است!", vbCritical, vbOKOnly
    Exit Sub
   End If
   →محاسبه اختلاف تاریخ امروز با تاریخ تولد
    Text2.Text = CStr(DateDiff("yyyy", bDate, Date))
    Text3.Text = CStr(DateDiff("d", bDate, Date))
    Text4.Text = CStr(DateDiff("h", bDate, Date))
    Text5.Text = CStr(DateDiff("s", bDate, Date))
End Sub
```

انواع متغیرها

۱. **سراسری:** در قسمت general تعریف می شوند و در کل برنامه قابل دسترسی هستند
۲. **محلی:** داخل رویدادها یا زیر برنامه ها تعریف شده و فقط داخل زیر برنامه ای که در آن وجود دارند قابل دسترسی و استفاده هستند. این متغیرها با اتمام کار زیر برنامه خود بخود از بین می روند.
۳. **محلی ایستا(Static):** مشابه متغیرهای محلی هستند ولی این متغیرها با اتمام کار زیر برنامه مقدار حاصل از اجرای قبلی را حفظ می کنند.

نحوه تعریف متغیرهای ایستا: به جای **dim** از کلمه **static** استفاده می شود:

```
Static  n  as  integer
```

مثال: برنامه ای بنویسید که به کمک متغیر ایستا، تعداد کلیک های روی یک دکمه را بشمارد.
یک دکمه روی فرم قرار دهید و دستورات زیر را در آن بنویسید:

```
Static n as integer
N= n+1
Command1.caption = " شما این تعداد دفعه کلیک کرده اید: " & n
```

تمرین: تعداد ۱۶ دکمه(Command) در چهار سطر و ستون بچینید. یک بازی شبیه MineSweeper ویندوز طراحی کنید که باید همه دکمه ها را فقط یک بار کلیک نمایید. در صورتیکه دکمه ای بیش از یکبار کلیک شود بازنده هستید و اگر همه را فقط یک بار کلیک نمایید برنده.
راهنمایی: متغیر ایستا از نوع Boolean که کلیک شدن را حفظ کند و متغیر سراسری برای شمارش دکمه های کلیک شده

فصل هفتم - گرافیک

دو کنترل زیر برای ذخیره تصویر بکار می روند:

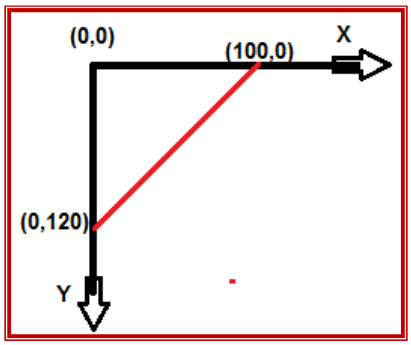
- **Image**
  ۱. خصوصیت stretch اگر true باشد تصویر را به اندازه کادر خود تبدیل می کند در غیر این صورت کادر خود را تغییر اندازه می دهد.
- **Picurebox**
  ۱. برای گروه بندی دیگر کنترلها هم بکار می رود ولی Image این قابلیت را ندارد
  ۲. خصوصیت autosize اگر true باشد کادر خود را به اندازه تصویر تبدیل می کند در غیر این صورت فقط تصویر را به اندازه کادر نمایش می دهد.

*خصوصیت مشترک در هر دو picture می باشد که تصویر را ذخیره می کند توسط دستور زیر:
نام کنترل.picture = loadpicture(مسیر و نام فایل تصویر)

مثال

Image1.picture = loadpicture("d:\pic\myfile.jpg") →
انتخاب عکس موجود(myfile.jpg) در پوشه pic و در درایو D:
Image1.picture = loadpicture("") → پاک کردن عکس داخل آن
Image1.Picture = image2.picture → کپی تصویر از یک کنترل به دیگری

کنترلهای رسم اشکال گرافیکی



۱. **خط(Line) :** بین دو نقطه با مشخصه های زیر:
- **BorderColor:** رنگ خط
- **BorderStyle:** شکل ظاهری خط(جدول صفحه ۱۷۵)
- **BorderWidth:** پهنای خط بر حسب پیکسل

۲. **شکل(Shape):** یک شکل هندسی که توسط مشخصه Shape تعیین می گردد:
- **0-Rectangle:** مستطیل
- **1-Square:** مربع
- **2-Oval:** بیضی
- **3-Circle:** دایره
- **4-Rounded Rectangle:** مستطیل با گوشه گرد
- **5-Rounded Square:** مربع با گوشه گرد

✓ بقیه مشخصه های شکل هندسی در جدول صفحه ۱۷۶

تمرین: اجرای مثالهای صفحه ۱۷۷(lstPattern: list2, LstShape:List1, shpShape:Shape1) و صفحه ۱۷۹

دستورات رسم اشکال:

در کنترل فرم و کادر تصویر(picture Box) می توان شکل رسم کرد

۱- **رسم نقطه:**

نام کنترل.Pset [step] (X,Y), رنگ

این دستور نقطه (X,Y) را به رنگ داده شده رسم می کند. اگر رنگ را ننویسیم به رنگ پیش زمینه فرم رسم می کند. کلمه step اختیاری است و نقطه جدید را به نسبت نقطه قبلی رسم می کند:

مثال:

Form1.Pset (100,200),vbRed → رسم نقطه(۲۰۰و۱۰۰) به رنگ قرمز
Form1.Pset step (50,70),vbBlue → رسم نقطه(۲۷۰و۱۵۰) به رنگ آبی (به نسبت نقطه قبلی)
Form1.Pset (300,400) → رسم نقطه(۴۰۰و۳۰۰) به رنگ پیش زمینه فرم

Form1.Pset (500,500),Form1.BackColor →
رسم نقطه(۵۰۰و۵۰۰) به رنگ پس زمینه فرم = پاک کردن نقطه

رسم خط به کمک رسم نقطه و حلقه تکرار:

```
For  n = 1 to 100
   Form1.Pset (n , 250), vbGreen  → رسم خط افقی از نقطه (۲۵۰و۱) تا (۲۵۰و۱۰۰)
Next   n
```

مثال: **تخته سیاه**

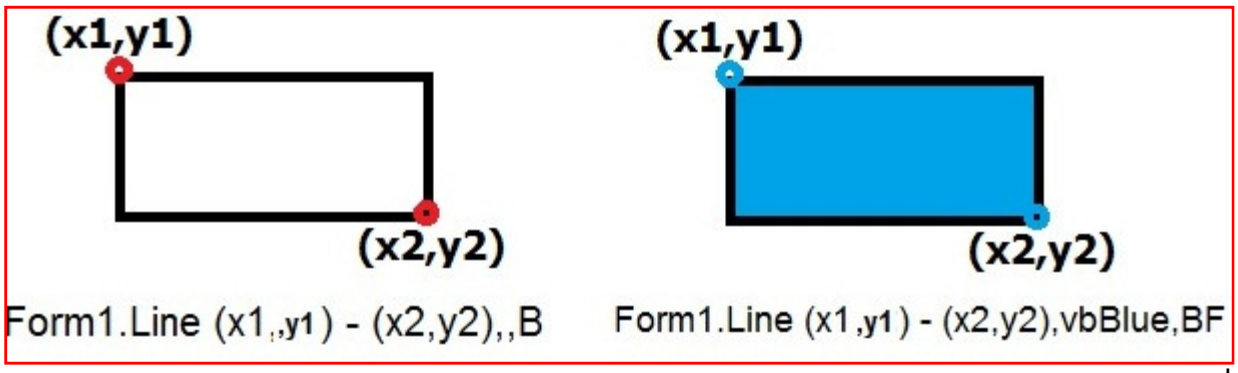
- یک pictureBox به نام PicBox روی فرم قرار دهید و رنگ پس زمینه آنرا سیاه کنید.
- در رویداد PicBox_MouseMove دستورات را بنویسید:

```
If Button = 1 Then
   PicBox.PSet (X, Y), vbYellow
End If
```

۲- **رسم خط:**

نام کنترل. Line [step] (X1,Y1)-(X2,Y2), رنگ , B یا BF

این دستور نقطه (X1,Y1) را به (X2,Y2) وصل کرده و خط به رنگ داده شده رسم می شود. کاربرد step و رنگ مشابه رسم نقطه است.اگر B را بنویسیم مستطیل رسم خواهد کرد که (X1,Y1) نقطه بالا و سمت چپ و (X2,Y2) مختصات نقطه پایین و راست است.BF مستطیل توپر رسم می کند.



|

مثال:

Form1.Line (100,200) - (500,600) ,vbRed → رسم خط بین دو نقطه داده شده به رنگ قرمز
Form1.Line (200,300) - (600,700) ,vbBlue, B → رسم مستطیل به رنگ آبی
Form1.Line (200,300) - (600,700) , , BF → رسم مستطیل توپر به رنگ زمینه فرم(جای رنگ خالی مانده)

۳- **رسم دایره:**

نام کنترل.Circle [step] (X,Y), اندازه شعاع, رنگ , , , نسبت

این دستور دایره ای به مرکز (X,Y) با رنگ و شعاع داده شده رسم می کند. کاربرد step و رنگ مشابه رسم نقطه است. نسبت عددی است که نسبت قطر عمودی به افقی را نشان می دهد. عدد یک دایره، عدد کوچکتر از یک، بیضی افقی و عدد بزرگتر از یک، بیضی عمودی رسم می کند.

مثال:

Form1.Circle (100,200) , 50, vbRed → رسم دایره قرمز به مرکز داده شده و شعاع ۵۰ پیکسل

Form1.circle (100,200) , 50, vbBlue, , , 0.5 →
رسم بیضی افقی با رنگ آبی به مرکز داده شده و شعاع ۵۰ پیکسل
Form1.circle (100,200) , 50, vbGreen, , , 1.5 →
رسم بیضی عمودی با رنگ سبز به مرکز داده شده و شعاع ۵۰ پیکسل

مثال: دایره های تصادفی

- یک pictureBox به نام PicBox روی فرم قرار دهید و رنگ پس زمینه آنرا سیاه کنید.
- در رویداد PicBox_MouseDown دستورات زیر را بنویسید:

```
Dim c As Integer
Dim r As Integer
If Button = 1 Then → دکمه چپ ماوس
   c = 1 + Int (15 * Rnd) → رنگ تصادفی بین ۱ تا ۱۵
   r = 1 + Int (500 * Rnd)→ شعاع تصادفی بین ۱ تا ۵۰۰
   PicBox.circle (X,Y), r, QBColor(c)
End If
```

در یک دکمه

```
Dim c As Integer,  r As Integer, i as long
For i = 1 to 10000
   c = 1 + Int (15 * Rnd) → رنگ تصادفی بین ۱ تا ۱۵
   r = 1 + Int (500 * Rnd)→ شعاع تصادفی بین ۱ تا ۵۰۰
   PicBox.circle (500,500), r, QBColor(c)
Next  i
```

یک picturebox به نام PicBox و مربعی تنظیمات زیر:
Scaleheight=1000 , Scalewidth=1000 و زمینه سیاه

روشهای انتخاب رنگ

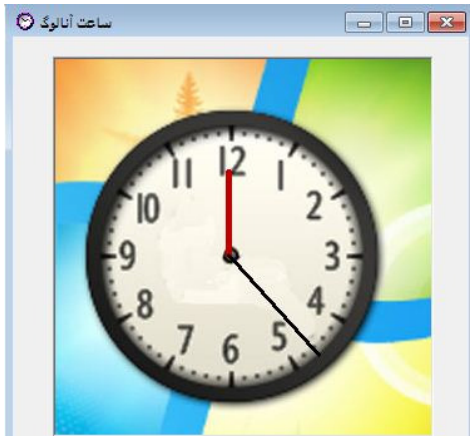
۱- **ثابتهای نامدار در VB** مانند: vbRed, vbBlue, ... که حدود ۸ رنگ وجود دارد.
۲- **استفاده از تابع QBColor(n)** که n کد رنگ بوده و عددی بین ۰ تا ۱۵ است. مثلاً کد ۴، رنگ قرمز است. این تابع ۱۶ رنگ مختلف می سازد
رسم نقطه (۲۰۰و۱۰۰) به رنگ قرمز → Form1.Pset (100,200), QBColor (4)

۳- **استفاده از تابع RGB(R, G, B)** که حدود ۱۶ میلیون رنگ مختلف می سازد
- R: درجه رنگ قرمز از ۰ تا ۲۵۵
- G: درجه رنگ سبز از ۰ تا ۲۵۵
- B: درجه رنگ آبی از ۰ تا ۲۵۵

رسم نقطه(۲۰۰و۱۰۰) به رنگ ترکیبی → Form1.Pset (100,200),RGB(100,200,150)

مثال: ساعت آنالوگ
یک PictureBoxو دو خط را روی آن قرار دهید طوری که ابتدای(x1,y1) و انتهای خطوط(x2,y2) در تصویر ساعت به ترتیب روی مرکز و عدد ۱۲ قرار گیرند. Timer1.Inerval = 1000

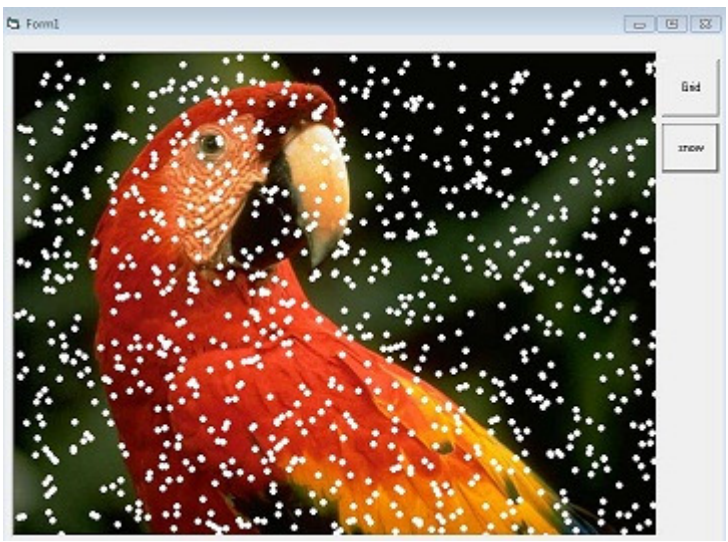
```
Option Explicit

Const p = 3.14159
Dim t1 As Single,  t2 As Single , r1 As Single , r2 As Single

Private Sub Form_Load()
        t1 = -p / 2
        t2 = -p / 2
        r1 = Sqr((Line1.X2 - Line1.X1) ^ 2 + (Line1.Y2 - Line1.Y1) ^ 2)
        r2 = Sqr((Line2.X2 - Line2.X1) ^ 2 + (Line2.Y2 - Line2.Y1) ^ 2)
End Sub

Private Sub Timer1_Timer()
        Dim x3 As Single
        Dim y3, x As Single
        Static m As Integer

        t1 = t1 + p / 30
        Line1.X2 = Line1.X1 + r1 * Cos(t1)
        Line1.Y2 = Line1.Y1 + r1 * Sin(t1)

        m = m + 1
        If (m = 60) Then
          m = 0
          t2 = t2 + p / 30
          Line2.X2 = Line2.X1 + r2 * Cos(t2)
          Line2.Y2 = Line2.Y1 + r2 * Sin(t2)
        End If
End Sub
```



مثال: ایجاد شبکه توری و بارش برف روی یک تصویر

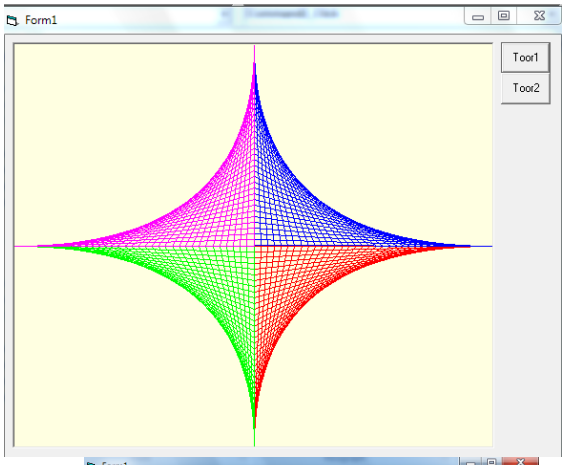
یک picturebox (با نام *PicBox*)را با تصویری تنظیم کنید و یک Shape دایره با نام Snow با رنگ زمینه و داخل آن بصورت سفید روی تصویر قرار دهید. دقت نمایید که خصوصیت Index آنرا صفر کنید.

```
Option Explicit
Dim w As Integer, h As Integer, n As Long

Private Sub Form_Load()
   w = PicBox.ScaleWidth
   h = PicBox.ScaleHeight
End Sub

Private Sub Command1_Click()   →Grid  ایجاد شبکه توری یا
        PicBox.Cls
        For n = 0 To w Step 20
              PicBox.Line (n, 0)-(n, h)
        Next
        For n = 0 To h Step 20
              PicBox.Line (0, n)-(w, n)
        Next
End Sub

Private Sub Command2_Click()
→ شبیه سازی بارش برف
        PicBox.Cls
        For n = 1 To 1000
              Load Snow(n)
              Snow(n).Visible = True
              Snow(n).Left = Rnd() * w
              Snow(n).Top = Rnd() * h
        Next
End Sub
```

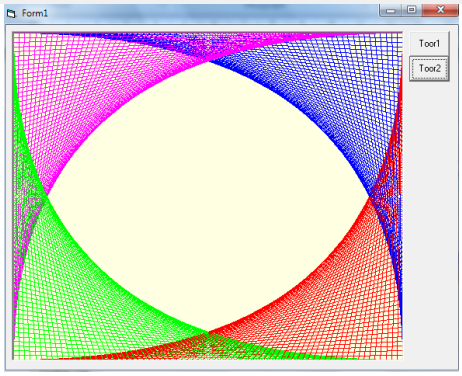


مثال: ایجاد رسم ریاضی
یک picturebox (با نام *PicBox*) و دو دکمه را روی فرم قرار دهید.

```
Option Explicit
Const n = 10
Dim x0 As Single,  y0 As Single ,  a As Integer
Private Sub Command1_Click()   → ایجاد شبکه توری در وسط
PicBox.Cls
X0= PicBox.ScaleWidth / 2
Y0= PicBox.ScaleHeight / 2
For a = 0 To x0 Step n
    PicBox.Line (a, y0)-(x0, y0 + a), vbGreen
    PicBox.Line (a, y0)-(x0, y0 - a), vbMagenta
    PicBox.Line (x0 + a, y0)-(x0, 2 * y0 - a), vbRed
    PicBox.Line (x0 + a, y0)-(x0, a), vbBlue
 Next
End Sub
```




```
Private Sub Command2_Click()→  ایجاد شبکه توری در گوشه ها
PicBox.Cls
X0= PicBox.ScaleWidth
Y0= PicBox.ScaleHeight
For a = 0 To x0 Step n
  PicBox.Line (a, y0)-(x0, y0 - a), vbRed
  PicBox.Line (0, y0 - a)-(x0 - a, y0), vbGreen
  PicBox.Line (a, 0)-(0, y0 - a), vbMagenta
  PicBox.Line (a, 0)-(x0, a), vbBlue
Next
End Sub
```



فصل هشتم – تولید بسته برنامه setup

اطلاعات برنامه و پرژه از طریق منوی زیر در بخشهای مختلف پنجره قابل تنظیم است:
Project → properties
همچنین با برنامه نویسی می توان اطلاعات تنظیم شده آنرا به شکل زیر مانند مثال نمونه بدست آورد:

```
MsgBox "نام فایل اجرایی برنامه = " & App.EXEName
MsgBox "نام شرکت = " & App.CompanyName
MsgBox "شماره نسخه اصلی= " & App.major
و غیره ...
```

ترجمه برنامه : به دو شیوه قابل انجام است که در مسیر قابل تنظیم است:
Project → properties→compile

- Native-Code: اندازه فایل اجرایی بزرگتر است و سرعت اجرا بالاتر است. و به کد کارآمدتر تبدیل شده و از قابلیتهای پردازنده بهتر استفاده می کند ولی به DLL های زمان اجرا نیاز دارد
- P-Code : اندازه فایل اجرایی کوچکتر است و سرعت اجرا کمتر است

تولید فایل اجرایی برنامه و ذخیره در محل ذخیره پروژه:

File → make project name

برای اجرا روی آن کلیک کنید که البته روی هر کامپیوتری اجرا نمی شود. برای اجرا در هر کامپیوتری باید setup بسازید و نصب کنید.قبل از هر کاری پروژه را در پوشه ای ذخیره کنید.

۱. برنامه Package & Development Wizard را از مسیر زیر پیدا کنید

Start → all programs → Microsoft VB Tools

۲. با زدن Browse فایل پروژه را انتخاب کنید و روی package کلیک کنید.
۳. در مراحل بعدی کافی است دکمه next را زده و تنظیمات پیش فرض را تأیید نمایید.
۴. برنامه setup در پوشه ای به نام package در مسیر ذخیره پروژه قرار می گیرد.