

```

18 `concatenating the contents onto a string
19 `buffer variable.
20 Do While ts.AtEndOfStream<>True
21 StrData = StrData & ts.Read(1)
22 Loop
23
24 `Close the TextSteam
25 ts. Close
26
27 `Assign the contents of the string buffer
28 `to the Text property of aText Box.
29 txtData.Text=strData

```

می‌توان از متد CreateTextFile() شیء FileSystem Object برای ایجاد یک فایل استفاده کرد. سپس می‌توان متد Write شیء TextStream را برای نوشتن داده‌ها در آن فایل به کار برد. کد زیر، چگونگی انجام این عمل را نشان می‌دهد. به متد Write در خط ۱۵ توجه کنید.

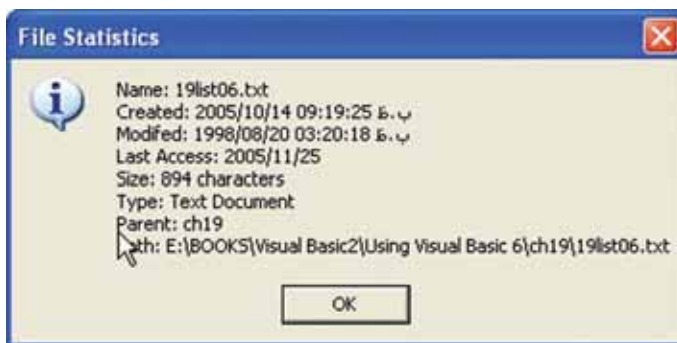
```

01 `Set a common dialog filter for text
02 cdlgMain.Filter = "Text (*.txt)*.txt"
03 `Open a save dialog
04 cdlgMain.ShowSave
05 `Get a filename for saving the data
06 gf_strOpenFile = cdlgMain.FileName
07 `Create a TextStream objet to the
08 `filename
09 Set ts = fso.CreateTextFile (gf_strOpenFile, True)
11 `Get the text to write to the file from
12 `the Text property of the TextBox
13 strData = txtData.Text
14 `Do the write
15 ts.Write(strData)
16 `Close the TextSteam
17 ts.Close

```

علاوه بر استفاده از شیء‌های سیستم فایل برای کارکردن با فایل‌ها و داده‌ها می‌توان مشخصه‌های

شیء‌ها را برای به‌دست‌آوردن اطلاعات نیز به‌کار برد. کد زیر، چگونگی استفاده از مشخصه‌های مختلف شیء File برای گزارش آماری درباره‌ی فایل را نشان می‌دهد. شکل ۲۷-۴ نتیجه‌ی این کد را نشان می‌دهد.

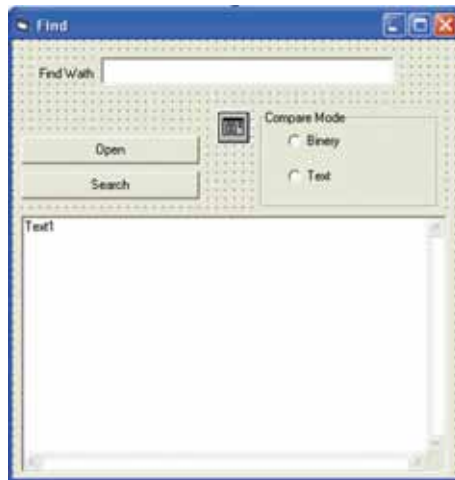


شکل ۲۷-۴_ مشخصه‌های مختلف شیء File را برای به‌دست‌آوردن اطلاعاتی درباره‌ی فایل، به‌کار ببرید.

- 01 Dim fso As New FileSystemObject
- 02 Dim fil As File
- 03 Dim strData As String
- 04
- 05 `Get a file by using the global variable filename
- 06 Set fil = fso.GetFile (gf_strOpenFile)
- 07 `Get the filename
- 08 strData = "Name:" & fil.Name & vbCrLf
- 09 `Get the date created
- 10 strData = strData & "Created:" & fil.DateCreated & vbCrLf
- 11 `Get the date last modified
- 12 strData = strData & "Modified:" & fil.DateLastModified & vbCrLf
- 13 `Get the date last accessed
- 14 strData = strData & "Last Access:" & fil.DateLast Accessed & vbCrLf
- 15 `Get the size in bytes
- 16 strData = strData & "Size:" & CStr (fil.Size)& "characters" & vbCrLf
- 17 "Get the file type
- 18 strData = strData & "Type:" & fil.Type & vbCrLf
- 19 `Get the parent folder
- 20 strData = strData & "Parent:" & file.Parent Folder.Name & vbCrLf

- 21 `Get the file path
- 22 strData = strData & "Path: & fil." Path & vbCrLf
- 23 `Report the findings
- 24 MsgBox strData, vbOKCancel, "File Statistics"

مثال ۴-۱۱- جستجو در یک فایل متنی



شکل ۴-۲۸

کد این برنامه به صورت زیر است :

```
Private Sub CmdOpen_Click ()
    Dim Fsol As New File System Object
    Dim Stream1 As TextStream
    Dim StrX As String, Cnt As Integer, N As Integer
    Cmdlg1.Filter = "Text File|*.txt|All File|*.*"
    Cmdlg1.ShowOpen
    If Cmdlg1.FileName <> "" Then
        Set Stream1.Fsol.Open TextFile (Cmdlg1.FileName, ForReading)
        Txt1=""
        Do While Not Stream1.AtEndOfStream
            Str X=Stream1.ReadLine
            Txt1= Txt1 + StrX+vbNewLine
        Loop
        Stream1.Close
    End If
End Sub
```

```

End Sub
Private Sub CmdSearch_Click ()
Dim CompMode As Byte
\vbBinaryCompare= 0
\vbTextcompare=1
If OptBin.Value = True Then
    CompMode = 0
Else
    CompMode=1
End If
Cnt=0
N= InStr (1, Txt1, TxtFind, CompMode)
Do While N>0
    Cnt = Cnt+1
    N=InStr (N+1, Txt1, TxtFind, CompMode)
Loop
MsgBox "" & TxtFind & "" " & Cnt & " Time Repeated"
End Sub
Private Sub Form_Load ()
    OptText.Value = True
End Sub

```

مثال ۴-۱۲:

جستجو و جایگزینی در یک فایل متنی:

کد این برنامه به صورت زیر است:



شکل ۴-۲۹

```

Private Sub CmdOpen_Click ()
    Dim Fso1 As New FileSystem Object
    Dim Stream1 As TextStream
    Dim StrX As String, LngCnt As Integer, IntN As Integer
    Cmd1.Filter = "Text File|*.txt| All File|*. *"
    Cmd1.ShowOpen
    If Cmd1.FileName <>"Then"
        Set Stream1= Fso1.OpenText File(Cmd1.FileName, ForReading)
        Txt1= ....
        Do While Not Stream1. AtEndOfStream
            StrX = Stream1. ReadLine
            Txt 1=Txt1+StrX+vbNewLine
        Loop
        Stream1. Close
    End If
End Sub

Private Sub CmdReplace_Click ()
    Dim CompMode As Byte, BytYN As Byte, IntN As Integer
    Dim str 1 As String, str2 As String, LngCnt As Long
    \vbBinaryCompare =0
    \vbTextCompare = 1
    If OptBin.Value = True Then
        CompMode =0
    Else
        CompMode=1
    End If
    If TxtFind="" Then Exit Sub
    LngCnt=0
    IntN = InStr (1, Txt1, TxtFind, CompMode)
    Do While IntN>0
        BytrYN=MsgBox ("Do you want replace", vbQuestion+ vbYesNo)
        If BytYN= vbYes Then
            str1= Left (Tctl, IntN -1)
            str2=Mid (Tctl, IntN+Len(TxtFind))
            Txt1= str1+ TxtReplace+str2
        End If
    End While

```

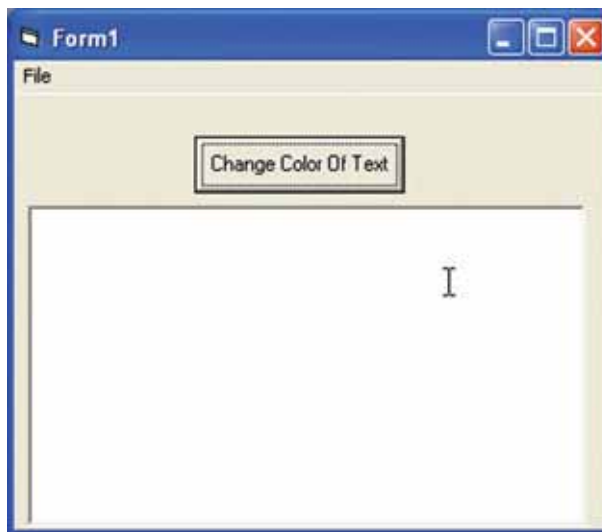
```

    LngCnt=LngCnt+1
    IntN=InStr(IntN+1,Txt1, TxtFind, CompMode)
Loop
MsgBox"""" &TxtFind &"""" "&LngCnt &" Time Repeated"
End Sub
Private Sub Form_ Load ()
    OptText.Value = True
En Sub

```

مثال ۱۳-۴: این مثال، چگونگی خواندن فایل‌های ترتیبی را نمایش می‌دهد. در این برنامه از کنترل کادر محاوره‌ای هم استفاده خواهیم کرد. این پروژه فقط برای نمایش محتویات فایل‌ها است ولی ویژگی‌های دیگری هم دارد:

- انتخاب فایل در کادر محاوره‌ای بازکردن فایل
 - تغییر رنگ زمینه‌ی فایل با کادر محاوره‌ای رنگ
 - تصحیح مکان کنترل‌های روی فرم بعد از تغییر اندازه‌ی فرم
 - مقابله با خطا از طریق کد نوشته شده برای کادر محاوره‌ای بازکردن فایل
 - قابلیت خواندن فایل‌های BAT و TXT. حداکثر تا ۴۰۹۶ بایت
- شکل ۳-۴ اجرای این برنامه را نشان می‌دهد.



شکل ۳-۴ برنامه‌ای برای مشاهده‌ی فایل‌های متنی و دستوری

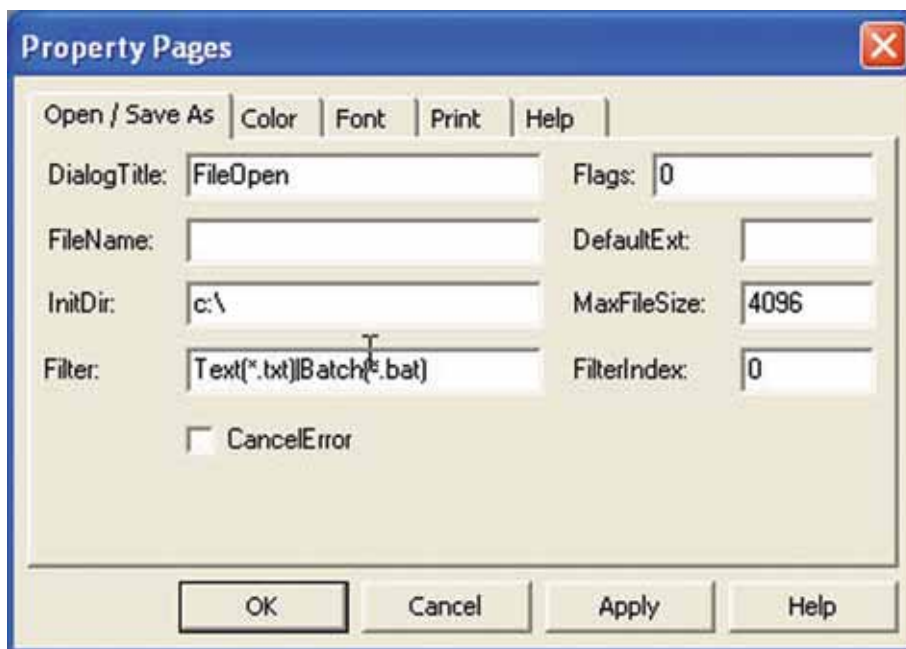
ایجاد فرم برنامه

فرم برنامه را با توجه به جدول ۴-۴ بسازید. ذکر یک نکته در مورد منوی برنامه را لازم می‌دانیم: تمام گزینه‌های منو، به جز گزینه‌ی اول باید یک سطح جلوتر باشند تا در زیر منوی File ظاهر شوند.

جدول ۴-۴ - کنترل‌های برنامه‌ی نمایش فایل

مشخصه / نام کنترل	مقدار مشخصه
Form Name	FrmFile
Form Caption	File Viewer
Form Height	4620
Form Width	6570
List box Name	lstFile
List box Height	2205
List box Left	720
List box Top	1320
List box Width	8415
Command button Name	cmdColor
Command button Caption	& Change the File Color
Command button Height	495
Command button Left	2760
Command button Top	480
Menu item # 1 Caption	&File
Menu item # 1 Name	mnuFile
Menu item # 1 Caption	&Open
Menu item # 2 Name	mnuFileOpen
Menu item # 3 Caption	
Menu item # 3 Name	mnuFileSep1
Menu item # 4 Caption	E&xit
Menu item # 4 Name	MnuFileExit
Common dialog box Name	ComFile
Common dialog box Dialog Title	File Open
Common dialog box InitDir	C:\
Common dialog box Filter	Text (*.txt)*. txt Batch (*.bat)*. bat
Common dialog box CancelError	True
Common dialog box MaxFileSize	4096

یکی از ساده‌ترین روش‌های تنظیم کادر محاوره‌ای استفاده از مشخصه‌ی Custom است (شکل ۳۱-۴).



شکل ۳۱-۴- برای تنظیم کادر محاوره‌ای، از مشخصه‌ی Custom آن استفاده کنید.

کادر محاوره‌ای انتخاب رنگ نیازی به هیچ‌گونه تنظیم قبل از اجرای برنامه ندارد.
نوشتن کد برنامه

```

1: Private Sub cmdColor_Click ()
2:     'Use the common dialog box to
3:     'let the user change the background
4:     'color of the list box.
5:     comFile.ShowColor
6:     lstFile.BackColor = comFile.Color
7: End Sub
8:
9: Private Sub Form_Resize ()
10: Dim intMsg As Integer 'For MsgBox ()
11: ' Change the size of the list box
12: ' if thd user resizes the form

```



```

13: ‘
14: ‘ This event procedure executes when the
15: ‘ form first loads also
16: ‘
17: ‘Make sure the form is not sized so small
18: ‘ that the list box cannot display
19:   If (frmFile.Width <400) Or (frmFile.Height < 3500) Then
20:   ‘Hide list box
21:   ‘and warn user
22:   lstFile.Visible = False
23:   intMsg = MsgBox (“The form is too small to display the file”, vbCritical)
24: Else
25:   ‘Turn on list box display in case it
26:   ‘was turned off previously
27:   lstFile.Visible=True
28:   ‘Adjust the size of the list box to the form
29:   ‘Adjust the command button’s location to the form
30:   lstFile.Width= frmFile.Width – 1440
31:   lstFile.Height= frmFile.Height – 2500
32:   cmdColor.Left = (frmFile.Width / 2) – 500
33: End If
34: End Sub
35:
36: Private Sub mnuFileExit_Click ()
37: ‘Program termination option
38: End
39: End Sub
40:
41: Private Sub mnuFile Open_Click ()
42: Dim strFileLine As String
43: ‘Set up for Cancel click
44: On Error GoTo comErrorHandler
45:
46: ‘Display the File Open dialog box
47: comFile.ShowOpen
48: ‘Code either continues if User clicks OK

```

```

49: 'or skips to error handler if Cancel clicked
50:
51: 'Open the file the user selected
52: Open comFile.FileName For Input As #1
53: ' Make room for new file
54: lstFileClear
55:
56: 'Read one complete line of the file
57: Line Input #1, strFileLine
58: lstFile.AddItem strFileLine
59:
60: 'Keep reading and adding to the list
61: 'box until the end of file is reached
62: Do Until (EOF (1))
63:   Line Input # 1, strFileLine
64: lstFile.AddItem strFileLine
65: Loop
66: `Close open file
67: Close
68: comErrorHandler:
69: ' Do nothing if the user clicks Cancel
70: End Sub

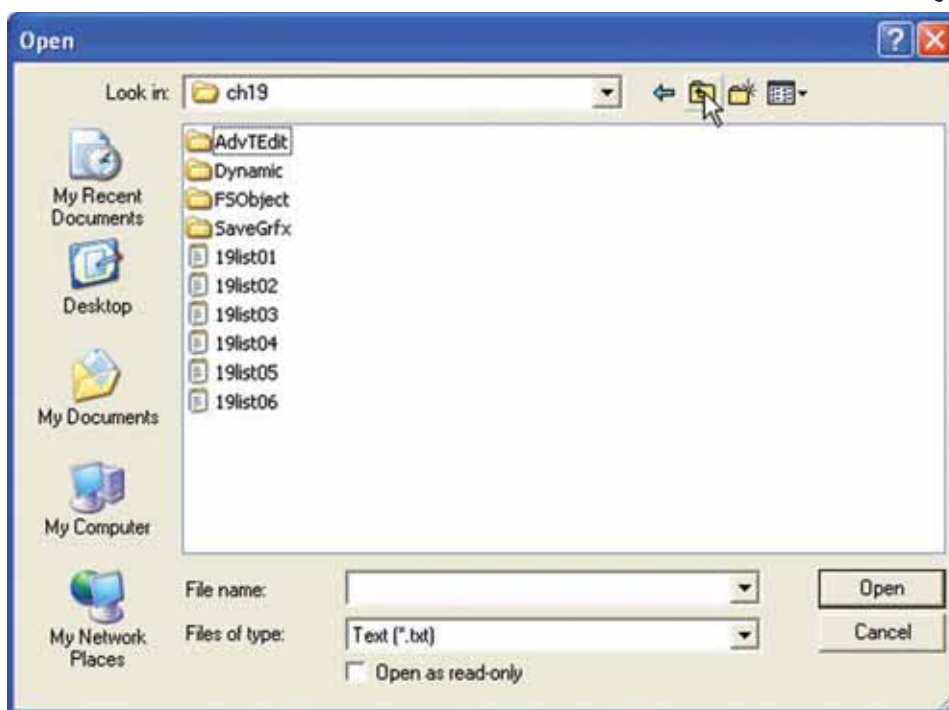
```

تحلیل کد برنامه

اولین روال برنامه (خطوط ۱ تا ۷) کادر محاوره‌ای رنگ را نمایش داده و رنگ زمینه‌ی کنترل کادر لیست را به رنگ انتخاب شده در می‌آورد (خط ۶).

روال Form_Resize () (یکی از طولانی‌ترین روال‌های برنامه) از خط ۹ شروع می‌شود. این روال رویداد، هرگاه کاربر اندازه‌ی فرم را تغییر دهد (یا در شروع اجرای برنامه) اجرا خواهد شد. این روال اندازه‌ی کادر لیست و مکان دکمه فرمان را با توجه به ابعاد فرم تغییر خواهد داد. اگر اندازه‌ی فرم خیلی کوچک شود، دستور If خط ۱۹ ضمن هشدار به کاربر، کنترل کادر لیست را پنهان خواهد کرد. اما در غیر این صورت، خط ۲۷ با تنظیم ابعاد کادر لیست و مکان دکمه‌ی فرمان، تناسب بصری فرم برنامه را حفظ خواهد کرد.


در خط ۴۱ طولانی‌ترین روال برنامه شروع می‌شود: روال بازکردن و خواندن فایل انتخاب‌شده. هنگامی که کاربر در منوی File گزینه‌ی Open را کلیک کند، این روال ابتدا کادر محاوره‌ای بازکردن فایل را ظاهر می‌کند (شکل ۴-۳۲). به خط ۴۷ و متد ShowOpen توجه کنید. (کادر محاوره‌ای بازکردن فایل امکان مشاهده‌ی فایل‌های Text یا Bat را به کاربران می‌دهد). خط ۴۴ برای مقابله با کلیک‌شدن دکمه‌ی Cancel تعبیه‌شده است. البته این روتین در مقابله با خطا هیچ کاری انجام نمی‌دهد و فقط روال را پایان می‌دهد (چون در واقع نیاز به هیچ اقدام خاصی هم وجود ندارد). هنگامی که اجرای برنامه به خط ۵۱ می‌رسد، کاربر فایل مورد نظرش را انتخاب کرده است و خط ۵۲ این فایل را باز خواهد کرد. دستور Open خط ۵۲ این فایل را در حالت ترتیبی باز خواهد کرد. چون این فایل برای خروجی (Output) باز شده است، برنامه‌ی ما قادر به دستکاری آن نخواهد بود.



شکل ۴-۳۲ کادر محاوره‌ای بازکردن فایل به کاربر امکان انتخاب فایل مورد نظر را می‌دهد.

در خط ۵۴ کادر لیست پاک می‌شود. خطوط ۵۷ و ۵۸ اولین خط فایل را خوانده و آن را (با متد AddItem) در کادر لیست می‌نویسند.

بعد از خواندن اولین خط فایل، حلقه‌ای که از خط ۶۲ شروع می‌شود بقیه‌ی فایل را می‌خواند. برای تشخیص پایان فایل از تابع EOF() استفاده شده است، چون برنامه از قبل نمی‌تواند بداند که چند خط در فایل باز شده وجود دارد. تابع EOF() با رسیدن به پایان فایل مقدار True را برمی‌گرداند و در این حالت حلقه متوقف شده و برنامه از خط ۶۶ ادامه خواهد یافت. فایل باز شده در خط ۶۷ بسته شده و برنامه در انتظار کاربر می‌ماند تا فایل دیگری را باز کند.

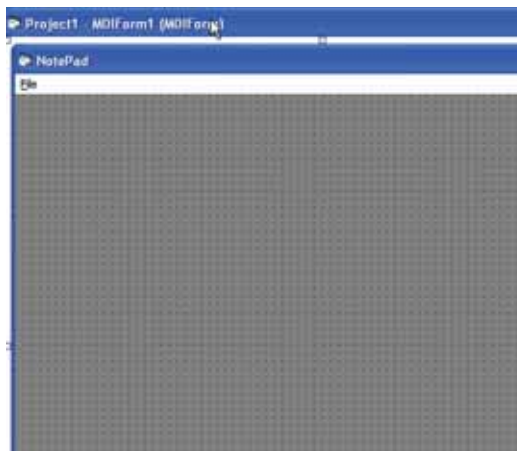
 **تحقیق:** چرا در خط ۶۲ از دستور زیر استفاده نکرده‌ایم؟

Do Until (EOF(1)= True)

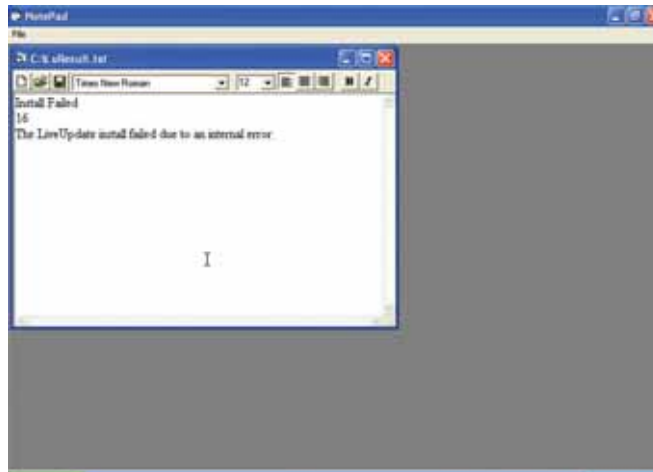
مثال ۴-۱۴ - NotePad کامل: از توضیح چگونگی ایجاد این مثال پرهیز می‌کنیم و آن را به عنوان تمرین به عهده‌ی هنرجو واگذار می‌کنیم. فقط ذکر این نکته ضروری است که در این مثال، از دو کنترل جدید ImageList و Toolbar استفاده کرده‌ایم.



شکل ۴-۳۳



شکل ۴-۳۴



شکل ۳۵-۴ - اجرای برنامه

```

'MDI Form Example
Dim IsChange As Boolean
Dim FileName As String
Private Sub Command1_Click ()
    MDIForm1.Show
End Sub
Private Sub Combo1_Click()
    Text1.FontName = Combo1.Text
End Sub
Private Sub Combo2_Change ()
    Text1.FontSize = Val (Combo2.Text)
End Sub

Private Sub Combo2_Click ()
    Text1.FontSize = Val (Combo2.Text)
End Sub

Private Sub Combo2_KeyPress (KeyAscii As Integer)
    If KeyAscii < 48 Or KeyAscii > 57 Then KeyAscii =0
End Sub
Private Sub Form_Activate()
    Text1.Width=Me.Width - 90
  
```

```
Text1.Height=Me.Height - 400 - Toolbar1.Height - 30
Text1.Top =Toolbar1.Height
End Sub
```

```
Private Sub Form_Load ()
IsChange = False
For i = 0 To Screen.FontCount -1
    Combo1.AddItem Screen.Fonts (i)
Next i
For i= 0 To Screen.FontCount -1
    If Combo1.List (i)= "Times New Roman" Then
        n=i
    End If
Next i
Combo1.ListIndex=n
Text1.FontName = Combo1.Text
For i= 8 To 72 Step 2
    Combo 2.AddItem i
Next
Combo 2.ListIndex =2
Text1.FontSize = Val(Combo2. Text)
End Sub
```

```
Private Sub Form_Resize()
Text1.Width = Me.Width -90
Text1.Height = Me.Height - 400 - Toolbar1. Height-30
Text1.Top = Toolbar1.Height
End Sub
```

```
Private Sub Form_Unload (Cancel As Integer)
If IsChange= True Then
    yn = MsgBox ("Do You Want Save The Change?",vYes No)
    If yn = vbYes Then MnuSave_Click
End If
End Sub
```

```
Private Sub mnunew_Click()
```

```
Dim frm1 As Form1
Set frm1 = New Form1
frm1. Show
End Sub
```

```
Private Sub MnuOpen_Click()
Dim fsol As New FileSystemObject
Dim stream1 As TextStream
Cmsl1.Filter="Text File|*.txt"
Cmdl1.ShowOpen
Filellame = Cmdl1.FileName
If FileName <>"" Then
    Set stream1 = fsol.OpenTextFile (FileName, ForReading)
    Me.Caption = FileName
    Me.Text1 = ""
    Me.Text1= stream1.ReadAll
    stream1.Close
    IsChange=False
End If
End Sub
```

```
Private Sub MnuSave_Click()
Dim fsol As New FileSystemObject
Dim stream1 As TextStream
If FileName <>"" Then
    Set stream1 = fsol.OpenTextFile (FileName, ForWriting, True)
    stream1.Write Me.Text1
    stream1.Close
    IsChange=False
Else
    MnuSaveAs_Click
End If
End Sub
Private Sub mnuExit_Click()
End
End Sub
```

```

Private Sub MnuSaveAs_Click()
Dim fso1 As New FileSystem Object
Dim stream1 As Text Stream
Cmdl1.Filter = "Text File|*.txt"
Cmdl1.ShowSave
If Cmdl1.FileName<>""Then
    Set stream1=fso1.OpenTextFile(Cmdl1.FileName, ForWriting, True)
    stream1.Write Me. Text 1
    stream1.Close
    IsChange =False
End If
End Sub

Private Sub Text1_Change()
    IsChange = True
End Sub

Private Sub Toolbar1_ButtonClick (ByVal Button As MSComctlLib.Button)
SelectCase Button.Key
Case"tblnew"
    mnunew_Click
Case "tblopen"
    MnuOpen_Click
Case "tblsave"
    MnuSave_Click
Case "tblleft"
    Text1.Alignment =0
Case "tblcenter"
    Text1.Alignment = 2
Case "tblright"
    Text1.Alignment = 1
Case "tblbold"
    Text 1.FontBold = Not Text1.FontBold
Case "tblitalic"
    Text1.FontItalic = Not Text1.FontItalic
End Select
End Sub

```


خودآزمایی و تحقیق

- ۱- چرا تغییر نام یک کلاس به نامی که مورد استفاده قرار نگرفته است، اهمیت دارد؟
 - ۲- در یک ماژول کلاس، چند کپی از داده ماژول می‌تواند در هر زمان موجود باشد؟
 - ۳- مدت زمان طول عمر داده در یک ماژول استاندارد، چقدر است؟
 - ۴- چگونه می‌توان یک ماژول کلاس به پروژه‌ی VB اضافه کرد؟
 - ۵- در یک ماژول استاندارد، چند کپی از داده‌ی ماژول در هر زمان امکان وجود خواهد داشت؟
 - ۶- در یک ماژول کلاس، طول عمر داده‌ی ماژول، چقدر است؟
 - ۷- هنگامی که یک ماژول کلاس استاندارد ایجاد می‌شود، کدام روال وظیفه‌ی مقداردهی اولیه‌ی شیء کلاس را به انجام می‌رساند؟
 - ۸- دو روش برای امتحان کلاس یک شیء نام ببرید.
 - ۹- فرق بین پیوند دادن و پنهان کردن چیست؟
 - ۱۰- آیا در جایی که لازم است به شیء‌های درون روال‌ها دسترسی داشته باشید، باید آن‌ها را منتقل نمایید؟
 - ۱۱- یک شکل، نقطه و دایره را با استفاده از توارث ایجاد کنید.
- توضیح: ابتدا یک واسط کلاس به نام IShape ایجاد می‌کنیم. واسط IShape شامل سه متد عمومی به نام‌های Area، Name و ToString است. همچنین کلاس‌های CPoint و CCircle که جزئی از واسط IShape هستند، ایجاد می‌شوند.
۱۲. برنامه‌ای بنویسید که شامل یک تعریف ساده برای کلاس CTime باشد. این کلاس دارای سه متغیر از نوع صحیح به نام‌های mSecond، mMinute، mHour است. این متغیرها در سطح متغیرهای مدول هستند.
- متغیرهای نمونه صحیح mSecond، mMinute، mHour به صورت Private اعلان شده‌اند. به این ترتیب دسترسی به آن‌ها از طریق متدها و کلاس ممکن است.
- کلاس CTime شامل متدهای عمومی SetTime، ToUniversalTime و ToStandardTime است. این متدهای عمومی یا سرویس‌دهنده‌های عمومی کلاس در برگزیده‌ی واسط عمومی هستند.

این متدها توسط سرویس گیرنده‌های کلاس به منظور دستکاری کردن داده‌های ذخیره شده در شیء‌های کلاس، مورد استفاده قرار می‌گیرند.

متد `Set Time` سه آرگومان صحیح می‌گیرد و از آن‌ها برای تنظیم زمان استفاده می‌کند. هر آرگومان محدوده‌ای را بررسی خواهد کرد. برای مثال، مقدار `mHour` باید بزرگتر یا مساوی و کمتر از ۲۴ باشد. زیرا ما از قالب متداول زمان استفاده می‌کنیم. هر مقداری خارج از این محدوده، مقدار غیرقابل قبول بوده و با صفر تنظیم خواهد شد.

متد `ToUniversalTime` آرگومانی دریافت نمی‌کند و یک قالب رشته‌ای شامل سه رقم را برگشت می‌دهد دو رقم برای `mHour`، دو رقم برای `mMinute` و دو رقم برای `mSecond`. به‌عنوان مثال، `083033` نشان‌دهنده زمان `8:30:33AM` است. متد `ToStandardTime` آرگومانی دریافت نمی‌کند و یک رشته برگشت می‌دهد. این متد یک قالب رشته‌ای استاندارد برای زمان شامل `mHour` و `mMinute` به همراه جداکننده‌های کولن و `AM` یا `PM` است.

۱۳- یک برنامه ساده بنویسید که دارای کنترل `OLE` باشد. در این کنترل، شیء `Paint` را پنهان کنید. منوی طراحی کنید که دارای گزینه‌های ذخیره و بازکردن تصویر باشد. برنامه را اجرا کرده و با دابل کلیک روی کنترل، تصویری را ترسیم کنید و سپس تصویر را ذخیره و از برنامه خارج شوید و دوباره برنامه را اجرا کنید و تصویر ذخیره‌شده را بارگذاری نمایید.

مدیریت پایگاه داده

هدف‌های رفتاری: فراگیر پس از پایان این فصل، خواهد توانست :

۱- با کنترل‌های داده کار کند.

۲- با تکنیک‌های DAO و ADO بتواند بانک‌های اطلاعاتی مورد نیاز را

ایجاد و مدیریت کند.

در فصل اول با نحوه‌ی پردازش فایل‌ها آشنا شده‌اید. وژوال بیسیک توانایی

قابل توجهی در پردازش فایل‌ها دارد.

مشکلی که پردازش فایل‌ها دارند، این است که روش ساده‌ای برای دسترسی به داده‌ها را ارایه می‌کنند و هیچ طرح و نقشه‌ی مناسبی در ارایه پرس و جو از داده‌ها ندارند. سیستم‌های پایگاه داده علاوه بر توانایی پردازش فایل، می‌توانند داده‌ها را به گونه‌ای سازماندهی کنند که مشکل و پیچیدگی پرس و جو ساده‌تر شود. اکثر سیستم‌های پایگاه داده که مورد استفاده وژوال بیسیک هستند (در کامپیوترهای شخصی) از نوع پایگاه داده رابطه‌ای هستند. زبان ساخت یافته پرس و جو^۱ (SQL) مورد استفاده سیستم‌های پایگاه داده رابطه‌ای است و از آن برای ایجاد پرس و جوها استفاده می‌شود. وژوال بیسیک برنامه‌نویسان را قادر می‌سازد تا کدهایی بنویسند تا با استفاده از پرس و جوهای SQL به داده‌های موجود در سیستم‌های پایگاه داده رابطه‌ای دسترسی پیدا کنند. برخی از نرم‌افزارهای پایگاه داده رابطه‌ای که از محبوبیت بیشتری برخوردار هستند، عبارتند از: Microsoft Access، Oracle و SQL Server.

با مفهوم، مزایا و ویژگی‌های پایگاه داده‌ها در درس بانک اطلاعاتی آشنا شده‌اید. بنابراین، از شرح مفاهیم اولیه پایگاه داده (انواع، جدول، رکورد، فیلد، پرس و جو، مجموعه رکورد^۲، شاخص و ...) و انواع داده‌هایی که در فیلدها می‌توان ذخیره کرد، پرهیز کرده و به شرح چگونگی کارکردن با آن‌ها در وژوال بیسیک می‌پردازیم.

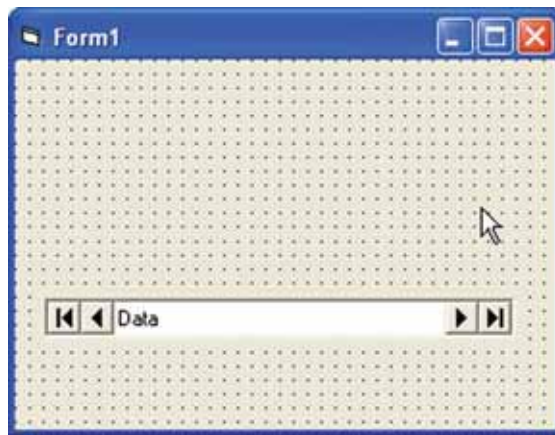
۱- Structured Query Language

۲- Recordset

۱-۵- کارکردن با کنترل Data

کنترل Data پیوندی بین اطلاعات موجود در بانک اطلاعاتی و کنترل ویژوال بیسیک است که برای نمایش اطلاعات به کار می‌برید. همزمان با این که مشخصه‌های کنترل Data را تعیین می‌کنید، مشخص می‌شود که کدام بانک اطلاعاتی و چه بخشی از آن قابل دسترس باشد. به طور پیش فرض، کنترل Data یک مجموعه رکورد پویایی (dynaset - type) را از یک یا چند جدول بانک اطلاعاتی ایجاد می‌کند. این بدین معنی است که مجموعه رکورد پویاست، بنابراین هنگامی که داده‌های جدول اصلی تغییر می‌کنند، رکوردهای موجود در مجموعه رکورد نیز به هنگام می‌شوند.

همچنین کنترل Data وظایف پیمایش که برنامه کاربردی برای سوئیچ کردن بین رکوردها نیاز دارد را نیز ارائه می‌کند. با استفاده از این دکمه‌ها کاربران می‌توانند به اولین یا آخرین رکورد یا به رکورد قبلی یا بعدی در مجموعه رکورد رجوع کنند (شکل ۱-۵).

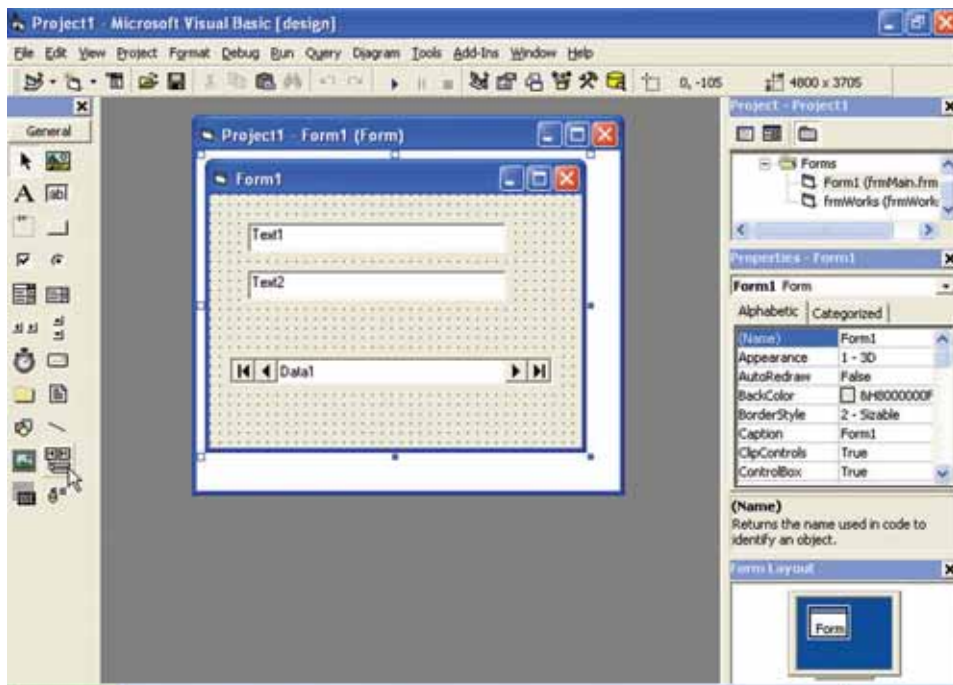


شکل ۱-۵- کنترل Data مقدار مشخصی Caption آن‌ها را بین دکمه‌های پیمایش، نمایش می‌دهد

مثال ۱-۵- کاربرد کنترل Data

- ۱- پروژه‌ی جدیدی ایجاد کرده و آن را DataProj نامگذاری کنید. فرم پیش فرض را frmMain نامگذاری کنید.
- ۲- کنترل Data را از جعبه ابزار انتخاب کرده و کنترل Data را روی فرم ترسیم کنید.
- ۳- نام پیش فرض کنترل Data را تغییر ندهید.
- ۴- دو کنترل کادر متن به فرم اضافه کنید و نام آن‌ها را txtFirst و txtLast قرار دهید.

کنترل‌های Data و کادرهای متن را مطابق شکل ۵-۲ روی فرم قرار دهید.
 ۵- پروژه را با نام DataProj-vbp ذخیره کنید.



شکل ۵-۲. از کنترل Data برای دسترسی به بانک اطلاعاتی و برای مقید کردن کنترل‌های روی فرم به فیلدهای بانک اطلاعاتی، استفاده کنید.

۱-۱-۵. ارتباط کنترل Data با بانک اطلاعاتی

بعد از اینکه کنترل Data را روی فرم قرار دادید، نیاز به ایجاد اتصالی بین این کنترل و اطلاعات بانک اطلاعاتی دارید. این کار با تنظیم مشخصه‌های کنترل Data انجام می‌شود.

نکته: مشخصه‌ی DatabaseName با مشخصه‌ی Name یکسان نیست. مشخصه‌ی Name نام شیء کنترل Data را تعیین می‌کند و برای رجوع به شیء در کد مورد استفاده قرار می‌گیرد. مشخصه‌ی DatabaseName نام فایل بانک اطلاعاتی که کنترل Data به آن دسترسی دارد را تعیین می‌کند.

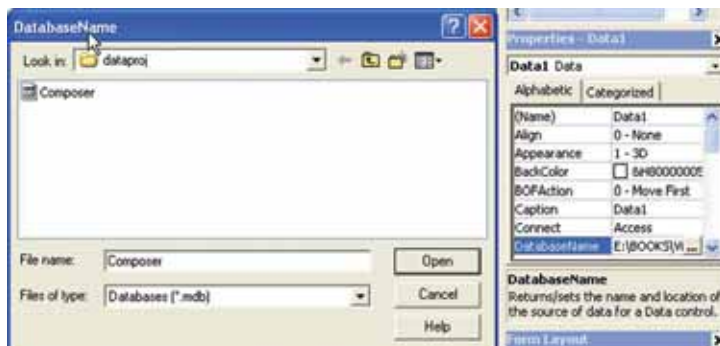
اگرچه چندین مشخصه می‌توانند روی روشی که کنترل Data با بانک اطلاعاتی ارتباط برقرار می‌کند، تأثیر بگذارند، ولی فقط دو مشخصه برای برقراری ارتباط با موتور بانک اطلاعاتی Jet مورد نیاز هستند: DatabaseName و RecordSource.

تعیین DatabaseName کنترل Data را به یک بانک اطلاعاتی خاص مرتبط می‌کند و RecordSource جدولی را در داخل این بانک اطلاعاتی مشخص می‌کند. بعد از مقداردهی این دو مشخصه، کنترل Data آماده‌ی بازیابی، ایجاد و ویرایش اطلاعات است.

نکته: فرض ما در این فصل این است که شما می‌توانید بانک‌های اطلاعاتی مورد نیاز را در Access ایجاد کنید. در مثال زیر، از بانک اطلاعاتی Composer که قبلاً ایجاد شده است، استفاده می‌کنیم. همچنین می‌توانید از بانک‌های اطلاعاتی Biblio.mdb و Nwind.mdb که به همراه ویژوال بیسیک ارایه می‌شوند، استفاده کنید.

مثال ۲-۵ اتصال کنترل Data به بانک اطلاعاتی و جدول:

- ۱- کنترل Data روی فرم frmMain از پروژه‌ی DataProj.vbp را که در مثال قبل ایجاد کردید انتخاب کنید.
- ۲- مشخصه‌ی DatabaseName را در پنجره‌ی Properties انتخاب کنید (شکل ۳-۵).
- ۳- از کادر محاوره‌ای Database Name بانک اطلاعاتی Composer.mdb را انتخاب کنید.
- ۴- جدول tblComposers را از لیست بازشوی مشخصه‌ی RecordSource انتخاب کنید (شکل ۴-۵).



شکل ۳-۵ در مشخصه‌ی DatabaseName روی سه نقطه کلیک کنید تا کادر محاوره‌ای مربوطه باز شود.

شکل ۴-۵ نام فایل بانک اطلاعاتی را در مشخصه‌ی DatabaseName و نام مجموعه رکورد را از لیست بازشوی مشخصه‌ی RecordSource که شامل نام تمام جدول‌ها و پرس و جوهای بانک اطلاعاتی تعیین شده است. انتخاب کنید.



نکته: قبل از انتخاب مقداری برای مشخصه‌ی DataField کنترل دیگری، مطمئن باشید که برای مشخصه‌ی RecordSource کنترل Data جدولی را تعیین کرده‌اید. اگر جدولی تعیین نکرده باشید، خطایی را دریافت خواهید کرد.

۵- کادر متن txtFirst را روی فرم انتخاب کنید.

۶- برای مشخصه‌ی DataSource کادر متن، در پنجره‌ی Properties مقدار Data 1 را

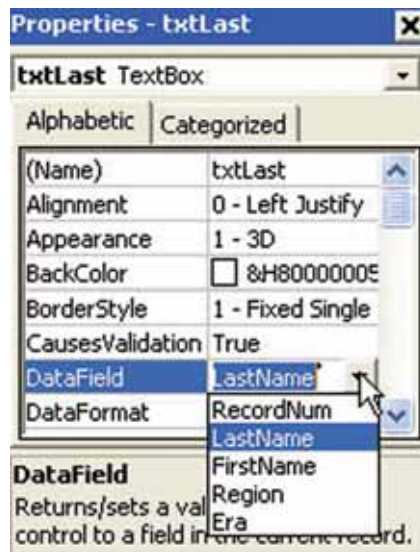
انتخاب کنید (شکل ۵-۵).



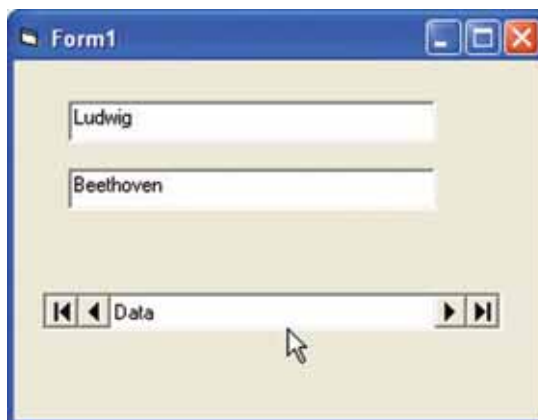
شکل ۵-۵ مشخصه‌ی DataSource تمام کنترل‌های Data بر روی فرم را فهرست می‌کند.

شکل ۵-۶ تمام فیلدهای جدول تعیین شده برای مشخصه‌ی RecordSource کنترل Data در لیست بازشوی DataField نمایش داده می‌شوند.

- ۷- برای مشخصه‌ی DataField کادر متن، مقدار FirstName را انتخاب کنید.
- ۸- مشخصه‌ی DataSource کادر متن txtLastName را نیز با Data 1 مقداردهی کنید.
- ۹- در پنجره‌ی Properties، فیلد LastName را برای مشخصه‌ی DataField کادر متن txtLastName انتخاب کنید (شکل ۵-۷).
- ۱۰- کد را ذخیره و اجرا کنید (شکل ۵-۸).



شکل ۵-۷ - هنگامی که فیلد را از جدول برای مشخصه‌ی DataField کادر متن تعیین می‌کنید، محتوای آن فیلد در کادر متن به صورت رکورد به رکورد، ظاهر می‌شود.



شکل ۵-۸ - می‌توان با استفاده از دکمه‌های پیمایش کنترل Data رکوردهای جدول را مشاهده کرد.