

۲- فیلدها را با مقدار پر کنید. (حداقل دو فیلد Name و Phone)

۳- روی Update کلیک کنید تا محتوا ذخیره شود.

به کمک دکمه‌های Delete و Find می‌توان رکوردی را حذف یا جستجو نمود. با مفهوم و عملکرد بقیه‌ی دکمه‌ها آشنا خواهید شد. در مثال زیر، می‌خواهیم یک سیستم مدیریت پایگاه داده (DBMS) ساده‌ای را برای انجام عملیاتی نظیر افزودن، حذف و ذخیره‌ی رکورد ایجاد کنیم.

دفترچه تلفن - مدیریت پایگاه داده

۱- یک کپی از فایل پایگاه داده‌ی تلفن ایجاد کنید تا در صورت خرابی احتمالی، کپی سالمی از آن را داشته باشید. می‌خواهیم یک DBMS ساده برای این پایگاه داده ایجاد کنیم تا رکوردهای خاصی را نمایش داده و امکان ویرایش آن‌ها را فراهم کند و عملیات درج، حذف و ذخیره را نیز انجام دهد.

۲- فرمی به شکل ۱۲-۷ ایجاد کنید. مشخصه‌های کنترل‌ها را به صورت زیر تنظیم نمایید :

frmBooks

Caption Phone List
Name frmPhone

dtaTitles

Caption Phone Numbers
ConnectionString PhoneList
RecordSource SELECT*FROM PhoneList ORDER BY Name
Name dtaPhone
LockType adLockOptimistic

Label1:

Caption Description

Label2:

Caption Phone

Label3:

Caption Name

txtAuthor

DataSource dtaPhone
DataField Name
Locked False
Name txtName
MaxLength 40

TabIndex	1
txtISBN	
DataSource	dtaPhone
DataField	Phone
Locked	False
Name	txtPhone
MaxLength	15
Tab Index	3
txtTitle	
DataSource	dtaPhone
DataField	Description
Locked	False
Name	txtDesc
MaxLength	40
TabIndex	2
Command1:	
Caption	&Add
Name	cmdAdd
Command2:	
Caption	&Save
Enabled	False
Name	cmdSave
Command3:	
Caption	&Delete
Name	cmdDelete

بعد از تنظیم مشخصه‌ها فرم شبیه شکل شکل ۷-۱۲ خواهد بود :

شکل ۷-۱۲

در این لحظه می‌توانید برنامه را اجرا کنید و با استفاده از کنترل داده‌ها، بانک اطلاعاتی را پیمایش کنید. گزینه‌های دیگر را امتحان نکنید، زیرا نیاز به کدنویسی دارند.

۳- کد زیر را برای این برنامه‌ی کاربردی، تایپ کنید:

```
Private Sub cmdLetter_Click(Index As Integer)
Dim BookMark1 As Variant
'Mark your place in case no match is found
BookMark1 = dtaPhone.Recordset.Bookmark
dtaPhone.Recordset.MoveFirst
dtaPhone.Recordset.Find "Name> = " + cmdLetter(Index).Caption + ""
If dtaPhone.Recordset.EOF= True Then
    dtaPhone.Recordset.Bookmark = BookMark1
End If
txtName.SetFocus
End Sub
Private Sub cmdAdd_Click()
cmdAdd.Enabled = False
cmdSave.Enabled = True
cmdDelete.Enabled = False
dtaPhone.Recordset.AddNew
txtName.SetFocus
End Sub
Private Sub cmdSave_Click()
dtaPhone.Recordset.Update
dtaPhone.Refresh
cmdAdd.Enabled = True
cmdSave.Enabled = False
cmdDelete.Enabled = True
txtName.SetFocus
End Sub
Private Sub cmdDelete_Click()
dtaPhone.Recordset.Delete
dtaPhone.Recordset.MoveNext
If dtaPhone.Recordset.EOF = True Then
    dtaPhone.Refresh
    If dtaPhone.Recordset.BOF = True Then
```

```

MsgBox "You must add a record.", vbOKOnly + vbInformation, "Empty file"
Call cmdAdd_Click
Else
    dtaPhone.Recordset.MoveFirst
End If
End If
txtName.SetFocus
EndSub

```

۴- برنامه‌ی کاربردی را ذخیره کنید. آن را اجرا کنید. رکوردهایی را اضافه، حذف و ویرایش کنید.

تمرین: دکمه‌ای به فرم فوق اضافه کنید تا از طریق مودم، شماره تلفن رکورد موردنظر را شماره‌گیری نماید.

مثال ۳-۷- دفترچه تلفن - ایجاد گزارش داده‌ها:

ایجاد Data Environment

۱- پروژه‌ی Standard EXE جدیدی را شروع کنید.

۲- در منوی Project روی Add Data Environment کلیک کنید. اگر این گزینه در منو نیست، روی Components کلیک کنید. از زبان‌های Designers، گزینه Data Environment را انتخاب کرده و روی OK کلیک کنید تا به منو اضافه شود.

۳- نیاز به اشاره به بانک اطلاعاتی داریم. در پنجره‌ی Data Environment روی زبان‌های Connection1 کلیک راست کرده و Properties را انتخاب کنید. در کادر محاوره‌ای Data Link Properties، گزینه‌ی Microsoft Jet 3.51 OLE DB Provider را انتخاب نمایید. روی Next کلیک کنید تا زبان‌های Connection باز شود. روی دکمه‌ی بیضی شکل کلیک کنید. فایل بانک اطلاعاتی تلفن (mdb) را پیدا کنید. روی OK کلیک کنید تا کادر محاوره‌ای بسته شود.

۴- روی زبان‌های Connection1 کلیک راست کرده و با انتخاب Rename نام آن را به Phone تغییر دهید. دوباره روی همین زبان‌ها کلیک راست کنید و Add Command را انتخاب کنید تا زبان‌های Command1 ایجاد شود. روی این زبان‌ها کلیک راست کرده و Properties را انتخاب کنید. مشخصه‌های زیر را تعیین کنید:

Command Name	Phonelist
Connection	Phone
DataBase Object	Table

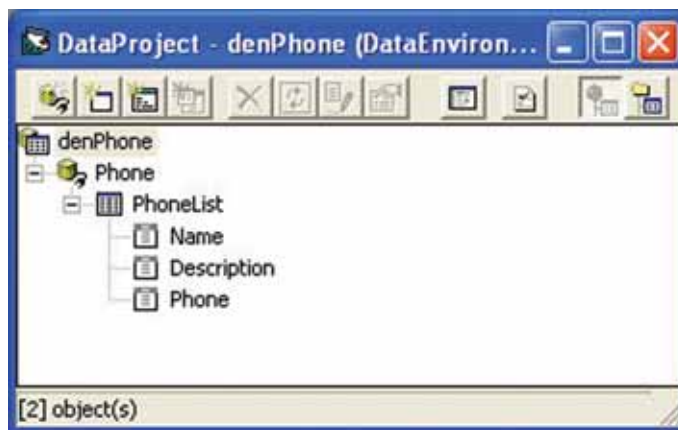
ObjectName

PhoneList

۵- روی OK کلیک کنید. محیط به پایگاه داده متصل می‌شود.

۶- پنجره‌ی Properties را نمایش داده و مشخصه‌ی نام محیط را denPhone قرار دهید.

از منوی File گزینه‌ی Save denPhone As را انتخاب کنید. محیط را در پوشه‌ی مناسبی ذخیره کنید. پنجره‌ی محیط داده‌ها شبیه شکل ۷-۱۳ خواهد بود.



شکل ۷-۱۳

ایجاد گزارش داده‌ای (Data Report)

بعد از ایجاد Data Environment، می‌توان یک Data Report ایجاد کرد. پنجره‌ی Data Environment را باز نگه دارید تا شیء‌ها را از آن به فرمی که برای Data Report ایجاد خواهیم کرد، درگ نمایید.

۱- از منوی Project روی Add Data Report کلیک کنید.

در صورتی که این گزینه نبود، گزینه‌ی Components و سپس زبانه‌ی Designers را انتخاب

کنید و سپس Data Report را انتخاب نموده و روی OK کلیک کنید.

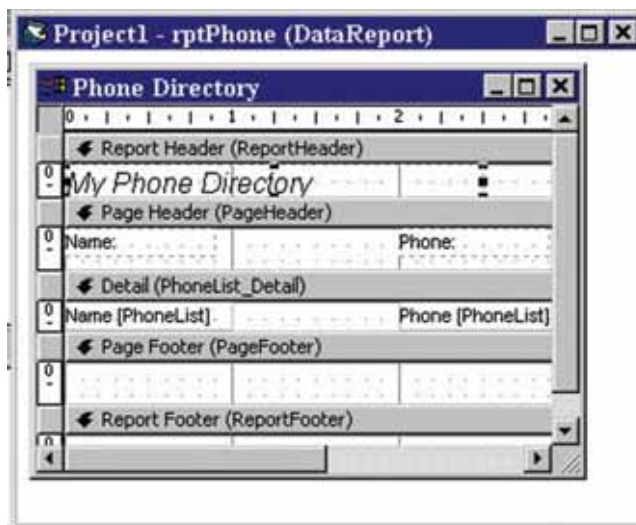
۲- مشخصه‌های زیر را برای گزارش، تنظیم کنید :

Name	rptPhone
Caption	Phone Directory
DataSouce	denphone (انتخاب کنید، تایپ نکنید)
DataMember	PhoneList (تایپ نکنید، انتخاب کنید - نام جدول)

۳- روی Data Report کلیک راست کرده و Retrieve Structure را انتخاب کنید. این عمل یک قالب گزارش مبتنی بر Data Environment ایجاد می کند.

۴- گزارش داده ها پنج بخش دارد : Report Header، Page Header، Detail، Page Footer و Report Footer، سر صفحه ها و پاورقی ها شامل اطلاعاتی است که می خواهید در هر صفحه از گزارش چاپ شود. برای قراردادن اطلاعات روی این بخش ها، روی آن ها کلیک راست کرده و Add Control را انتخاب کنید. سپس کنترل مورد نظر را انتخاب نمایید. این کنترل ها را data report controls (کنترل های گزارش داده) می نامند و مشخصه های آن ها مشابه کنترل های معمولی، مقداردهی می شوند.

۵- بخش Detail برای طرح بندی اطلاعاتی است که می خواهید برای هر رکورد بانک اطلاعاتی، چاپ شود. در این مثال، دو فیلد Name و Phone را قرار خواهیم داد. روی زبانه ی Name در پنجره ی Data Environment کلیک کرده و آن را به بخش Datail از Data Report درگ کنید. دو عنصر ظاهر خواهد شد : کادر متن Name و کادر متن [phoneList]. اولین کادر متن، اطلاعات تیتراست. این کادر متن را به بخش Page Header منتقل کنید. دومین کادر متن، مقدار Name را از جدول PhoneList ارایه می کند. اکنون زبانه ی phone را نیز مانند Name درگ کرده و همان عملیات را تکرار کنید. فرم شبیه شکل ۷-۱۴ خواهد بود.



شکل ۷-۱۴

۶- از منوی File گزینه‌ی Save rptphone As را انتخاب کنید. محیط را در پوشه‌ی مناسبی ذخیره کنید.

دسترسی به گزارش داده‌ها

۱- پروژه‌ی دفترچه تلفن را دوباره باز کنید. یک دکمه‌ی فرمان به نام CmdReport و با عنوان Show Report اضافه کنید.

۲- از منوی Project روی Add File کلیک کنید. denPhone را انتخاب کرده و روی OK کلیک کنید. به همین ترتیب rptPhone را اضافه کنید. پنجره‌ی Project را مشاهده کنید. این فایل‌ها زیر Designer لیست خواهند شد.

۳- کد زیر را به کار ببرید:

```
Private Sub CmdReport_Click()  
    rptPhone.show  
End Sub
```

۴- از متد Show برای نمایش گزارش داده‌ها استفاده کنید.

۵- برنامه کاربردی را ذخیره و اجرا کنید. روی دکمه‌ی Show Report کلیک کنید. گزارش

شبهه شکل ۷-۱۵ خواهد بود.



شکل ۷-۱۵

اکنون یک نسخه‌ی چاپی از دفترچه تلفن دارید. روی آیکن Printer کلیک کنید.

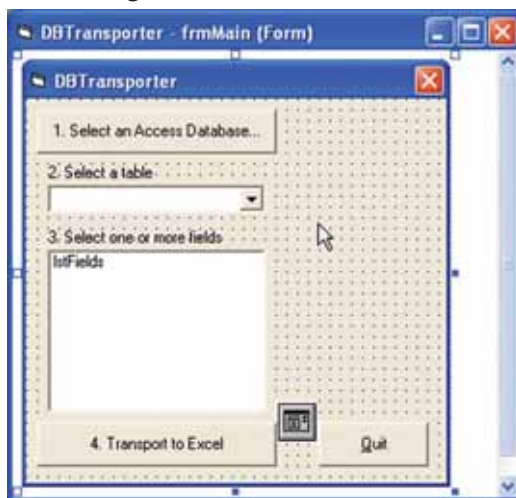
مثال ۴-۷- در این مثال یک برنامه‌ی سودمند به نام DBTransporter ایجاد خواهیم کرد که یک فایل بانک اطلاعاتی Access را باز کرده، جدول‌ها و فیلدهای مربوطه را از داخل فایل خوانده و سپس رکوردهای فیلدهای انتخاب شده را به صفحه گسترده Excel منتقل می‌کند. داده‌های بانک اطلاعاتی Access را با استفاده از DAO جستجو می‌کنیم. بعد از به دست آوردن داده‌ها از بانک اطلاعاتی از شیء‌های Excel Application، Workbook، Worksheet، و Range برای درج داده‌های بازبازی شده در یک صفحه گسترده‌ی Excel استفاده می‌کنیم. مراحل ایجاد این پروژه به صورت زیر است:

۱- پروژه‌ی جدیدی را در ویژوال بیسیک ایجاد کنید.

۲- روی فرم اولیه، سه دکمه‌ی فرمان، یک کادر ترکیبی، یک کادر لیست، یک کنترل CommonDialog و دو برچسب اضافه کنید.

۳- نام فرم را frmMain، کادر ترکیبی را cboTables و کادر لیست را lstFields قرار دهید. دکمه‌های فرمان را به ترتیب cmdOpenDB، cmdSave و cmdQuit نامگذاری کنید. نام کنترل CommonDialog را cdlgMain قرار دهید.

۴- مشخصه‌های Caption و Text کنترل‌ها را مطابق شکل ۱۶-۷ تنظیم کنید. مشخصه‌ی MultiSelect کنترل کادر لیست را با Extended-2 مقداردهی کنید.



شکل ۱۶-۷- می‌توان از کنترل CommonDialog برای جستجوی بانک اطلاعاتی برای دسترسی به داده‌ها و ارسال به Excel استفاده کرد.

۵- کد مربوط به دکمه‌ی فرمان Quit را بنویسید.

۶- کد زیر را به بخش اعلان فرم، اضافه کنید :

```
Private Const ACCESS_SYSTEM_PREFIX = "MSYS"
```

```
Private Const MAXIMUM_RECS = 20
```

```
Private Const TRANS_MSG = "DBTransporter is transporting..."
```

```
Private Const DEFAULT_XLS.FILENAME = "DBTrans"
```

```
Private gf_strDbPath As String
```

۷- کد زیر را به روال رویداد cmdopenDB_Click() اضافه کنید :

```
01 Dim ws As Workspace
```

```
02 Dim db As Database
```

```
03 Dim strTblName As String
```

```
04 Dim strFilter As String
```

```
05 Dim strMsg As String
```

```
06 Dim iReturn As Integer
```

```
07 Dim i%
```

```
08
```

```
09 strFilter = " Access DB (*.mdb)|*.mdb"
```

```
10 cdlgMain.Filter = strFilter
```

```
11
```

```
12 cdlgMain.ShowOpen
```

```
13
```

```
14 While gf_strDbPath = ""
```

```
15 gf_strDbPath = cdlgMain.filename
```

```
16 If gf_strDbPath = ""Then
```

```
17 strMsg = "You have not selected a file."
```

```
18 strMsg = strMsg & vbCrLf & vbCrLf
```

```
19 strMsg = strMsg & "Do you want to select one now?"
```

```
20 iReturn = MsgBox(strMsg, vbCritical + vbYesNo, "Selection Error")
```

```
21 If iReturn = vbNo Then
```

```
22 Exit Sub
```

```
23 Else
```

```
24 cdlgMain.ShowOpen
```

```
25 End If
```

```
26 End If
```

```
27 Wend
```

```

28
29 Set ws = DBEngine.Workspaces(0)
30 Set db = ws.OpenDatabase (gf_strDbPath)
31
32 cboTables.Clear
33
34 For i% = 0 to db.TableDefs.Count - 1
35 strTblName = db.TableDefs(i%).Name
36 If Left(UCase(strTblName),4) <> ACCESS_SYSTEM_PREFIX Then
37 cboTables.AddItem strTblName
38 End If
39 Next i%
40
41 db.Close
42 ws.Close

```

۸- کد زیر را به روال رویداد cboTables_Click() اضافه کنید :

```

01 Dim ws As Workspace
02 Dim db As Database
03 Dim tdef As TableDef
04 Dim i%
05 Dim strFieldName As String
06
07 lstFields.Clear
08 Set ws = DBEngine.Workspaces(0)
09 Set db = ws.OpenDatabase(gf_strDbPath)
10 Set tdef = db.TableDefs(cboTables.List(cboTables.ListIndex))
11 For i% = 0 To tdef.Fields.Count - 1
12 lstFields.AddItem tdef.Fields(i%).Name
13 Next i%
14
15 db.Close
16 ws.Close

```

۹- کد زیر را به روال رویداد cmdSave_Click() اضافه کنید :

```

01 Dim xla As New Excel.Application
02 Dim xlb As New Excel.Workbook

```

```

03 Dim xls As New Excel.Worksheet
04 Dim xlr As Excel.Range
05 Dim ws As Workspace
06 Dim db As Database
07 Dim rs As Recordset
08 Dim strSQL As String
09 Dim i%, j%, k%
10 Dim rc%
11 Dim r%, c%
12 Dim strFields() As String
13 Dim strTable As String
14 Dim strMsg As String
15 Dim strBuffer As String
16 Dim strFilter As String
17
18 For i% = 0 To lstFields.ListCount - 1
19     If lstFields.Selected(i%) = True Then
20         j% = j% + 1
21     End If
22 Next i%
23
24 If j% - 1 = -1 Then
25     strMsg = "You must select at least one field."
26     strMsg = strMsg & vbCrLf & vbCrLf
27     strMsg = strMsg & _
    "Please select a field in the Field Listbox"
28     MsgBox strMsg, vbCritical, "Field Selection Error"
29     lstFields.SetFocus
30 Exit Sub
31 End If
32
33 ReDim strFields( j%-1)
34
35 cmdQuit.Enabled = False
36 cmdOpenDB.Enabled = False
37 cboTables.Enabled = False

```

```

38 IstFields.Enabled = False
39
40 frmMain.MousePointer = vbHourGlass
41 strBuffer = frmMain.Caption
42 frmMain.Caption = TRANS_MSG
43
44 j% = 0
45
46 For i% = 0 To IstFields.ListCount-1
47 If IstFields.Selected(i%) = True Then
48 strFields(j%) = IstFields.List(i%)
49 j% = j% + 1
50 End If
51 Next i%
52
53 strTable = cboTables.Text
54
55 Set ws = DBEngine.Workspaces(0)
56 Set db = ws.OpenDatabase(gf_strDbPath)
57
58 Set xlb = xla.Workbooks.Add
59 Set xls = xlb.Worksheets.Add
60
61 xls.Activate
62
63 For i% = 0 To UBound(strFields)
64 strSQL = "SELECT" & strTable & "[" & strFields(i%) & "]"
65 ^
66 v
67 strSQL = strSQL & " FROM" & strTable
68 Set rs = db.OpenRecordset (strSQL)
69
70
71 rs.MoveLast
72 rs.MoveFirst
73
74
75 c% = i% + 1
76 j% = 0

```

```

73 'Set the column name
74 xls.Cells(1, c%) = strFields(i%)
75 Set xlr = xls.Cells(1, c%)
76 'Make the top row bold
77 xlr.Select
78 xlr.Font.Bold=True
79
80 If rs.RecordCount > MAXIMUM_RECS Then
81 rc% = MAXIMUM_RECS
82 Else
83 rc% = rs.RecordCount
84 End If
85
86 For r% = 2 To rc% + 1
87 xls.Cells(r%, c%) = rs(strFields(i%))
88 rs.MoveNext
89 Next r%
90 Next i%
91
92 frmMain.MousePointer = vbDefault
93 frmMain.Caption = strBuffer
94
95 strMsg = "Access file: "& vbCrLf & vbCrLf
96 strMsg = strMsg & gf_strDbPath & vbCrLf & vbCrLf
97 strMsg = strMsg & "has been successfully transported."
98 MsgBox strMsg, vbExclamation, "Transport successful"
99
100 cmdQuit.Enabled = True
102 cmdSave.Enabled = True
103 cmdOpenDB.Enabled = True
104 cboTables.Enabled = True
105 lstFields.Enabled = True
106
107 strMsg = "DBTransporter has transported"_
& "the Access data"
108 strMsg = strMsg & "to an Excel spreadsheet."

```

```

109 strMsg = strMsg & vbCrLf & vbCrLf
110 strMsg = strMsg & "Do you want to save"_
& "the Excel spreadsheet?"
111
112 If MsgBox(strMsg, vbQuestion & vbYesNo) = vbYes Then
113 strFilter = "Excel Spreadsheet (*.xls)|*.xls"
114 cdlgMain.Filter = strFilter
115 cdlgMain.filename = DEFAULT_XLS_FILENAME
116 cdlgMain.ShowSave
117
118 If cdlgMain.filename<>"" Then
119 xls.SaveAs (cdlgMain.filename)
120 strMsg = "Access data has been saved to file:"
121 strMsg = strMsg & vbCrLf & vbCrLf
122 strMsg = strMsg & cdlgMain.filename
123 MsgBox strMsg, vbExclamation, "Spreadsheet saved"
124 End If
125 End If
126
127 xls.Saved = True
128
129 xla.Quit

```

۱۰- پروژه را ذخیره کرده و اجرا کنید (شکل ۱۷-۷).



Author	Year Best
1 Jacobs, Russell	
2 Metzger, Philip W.	
3 Boode, John	
4 Sybow, Dan Parks	
6 Lloyd, John	
8 Thiel, James R.	
10 Ingham, Kenneth	
12 Wellin, Paul	
13 Kamen, Sari	
14 Gaylord, Richard	
15 Cuddy, Dave	
17 Gardner, Juanda Mercado	
19 Kouth, Donald E.	
21 Hakim, Jack	
22 Winchell, Jeff	
24 Clark, Claudia	
26 Scott, Jack	
27 Coolbaugh, James	
29 Ladd, Scott Robert	
31 Gabriel, Richard P.	

شکل ۱۷-۷. برنامه سودمند DBTransporter اولین سطر را سیاه (bold) می‌کند تا نام فیلدها از داده‌ها متمایز شود.

کد پروژه‌ی DBTransporter به سه بخش تقسیم می‌شود. اولین بخش، کدی است که به کاربران امکان انتخاب بانک اطلاعاتی و سپس مقداردهی خودکار کادر ترکیبی cboTables با جدول‌های بانک اطلاعاتی را می‌دهد. دومین بخش کد، کادر لیست IstFields را با فیلدهای جدول انتخاب شده پر می‌کند. آخرین بخش کد انتقال واقعی داده‌ها بین Access و Excel را انجام می‌دهد. توضیحات کد برنامه را به عنوان تمرین به عهده‌ی هنرجو واگذار می‌کنیم.

خودآزمایی و تحقیق

- ۱- پروژه‌ای ایجاد کنید که مشخصات دانش‌آموزان همکلاسی خودتان را در یک بانک اطلاعاتی ذخیره نماید. این پروژه امکان ویرایش، جستجو و گزارش چاپی لیست دانش‌آموزان را نیز ارایه کند.
- ۲- برنامه کاربردی طراحی و ایجاد کنید که بانک اطلاعاتی صورتحساب خرید خانه را مدیریت کند. گزینه‌ای برای به دست آوردن گزارش چاپی این بانک اطلاعاتی، اضافه کنید.

چاپ

در این فصل مشاهده خواهید کرد که چگونه می‌توان با چاپگر کار کرد. تاکنون مهارت‌های اصلی برنامه‌نویسی را کسب کرده‌اید و می‌توانید برنامه‌های عملی متنوعی بنویسید. چاپ نیز یکی از ویژگی‌های بارز برنامه‌های حرفه‌ای است.

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

- ۱- شیء چاپگر را در برنامه‌های خود به کار ببرد.
- ۲- نحوه‌ی تعیین تنظیمات چاپگر را توضیح داده و انجام دهد.
- ۳- دستور IsTypeOf را بیان کند.
- ۴- متدهای چاپگر را به کار ببرد.
- ۵- تکنیک‌های صحیح چاپ را بداند و به کار گیرد.
- ۶- ارتباط با کاربر در هنگام چاپ را بتواند انجام دهد.
- ۷- روش چاپ فرم را بیان کرده و انجام دهد.

۸-۱ شیء Printer

فرض کنید می‌خواهید کدی که در پنجره‌ی کد پروژه‌ای نوشته‌اید را چاپ کنید. برای این کار باید گزینه‌ی `Print` → `File` را انتخاب کنید. ممکن است کامپیوتر شما به چندین چاپگر دسترسی

داشته باشد. بنابراین، برنامه‌ی شما به تمام آن‌ها دسترسی خواهد داشت. به طور معمول، نمایر نیز به صورت یک چاپگر عمل می‌کند. بدین ترتیب حتی می‌توانید از برنامه‌های خود فاکس هم بفرستید! برنامه‌های Visual Basic می‌توانند چاپگر پیش فرض خاص خود را (که با چاپگر پیش فرض ویندوز متفاوت است) داشته باشند.

قبل از آن که به چاپ در برنامه‌های Visual Basic بپردازیم، درباره‌ی یک شیء خاص بحث می‌کنیم: مجموعه‌ی Printers. مجموعه‌ی Printers از تمام چاپگرهای موجود در سیستم اجراکننده‌ی برنامه، از جمله مودم‌هایی که می‌توانند فاکس بفرستند، تشکیل شده است. این مجموعه در هر سیستمی متفاوت است. حتی در فاصله‌ی دو اجرای برنامه هم این مجموعه می‌تواند دستخوش تغییر شود.

مجموعه‌ی Printers معادل فهرست چاپگرهای موجود در کادر محاوره‌ای Print است.

۱-۱-۸- دسترسی به مجموعه‌ی Printers

برای کار با مجموعه‌ی Printers هم مانند سایر مجموعه‌های Visual Basic باید از اندیس‌ها استفاده کنید. اندیس‌های مجموعه‌ی Printers نیز از ۰ شروع می‌شود. برای شمارش چاپگرها می‌توانید از حلقه For (از ۰ تا Printers.Count-1) یا حلقه‌ی For Each استفاده کنید. برای تعیین چاپگر پیش فرض باید از دستور Set استفاده کنید:

```
Set Printer = Printers(1) Change the default Printer
```

بعد از اجرای این دستور چاپگر دوم مجموعه‌ی چاپگرها، به عنوان چاپگر پیش فرض Visual Basic خواهد شد. البته تعیین چاپگر مناسب همیشه کار ساده‌ای نیست؛ اما می‌توان از تکنیک‌های خاصی برای این کار استفاده کرد.

۲-۱-۸- مشخصه‌های چاپگر

در جدول ۱-۸ تعدادی از مشخصه‌های مهم چاپگرها را مشاهده می‌کنید. برخی از این مشخصه‌ها دارای مقادیر نام‌دار هستند که می‌توانید در برنامه از آن‌ها استفاده کنید (استفاده از این مقادیر خوانایی برنامه را افزایش می‌دهد).

جدول ۱-۸- مشخصه‌های چاپگر

مشخصه	مفهوم
ColorMode	چاپگر سیاه و سفید است یا رنگی؟
Copies	تعداد کپی‌های چاپ. این مشخصه را فقط هنگام اجرای برنامه می‌توان تنظیم کرد.
CurrentX,CurrentY	مختصات چاپ نویسه بعدی.
DeviceName	نام چاپگر.
DriverName	نام راه‌انداز چاپگر. چندین چاپگر می‌توانند از یک راه‌انداز استفاده کنند.
Duplex	چاپ دو طرفه یا یک طرفه.
Height	ارتفاع کاغذ انتخاب شده (برحسب واحد ScaleMode).
Orientation	جهت چاپ
Page	شماره‌ی صفحه‌ی جاری.
PaperBin	مخزن کاغذ.
PaperSize	اندازه‌ی کاغذ.
Port	نام درگاه چاپگر.
PrinterQuality	کیفیت چاپ.
TrackDefault	مشخصه‌های چاپگر مطابق مشخصه‌های چاپگر پیش فرض خواهد بود یا خیر؟
Width	عرض کاغذ انتخاب شده (برحسب واحد ScaleMode).
Zoom	مقیاس بزرگ‌نمایی چاپ. (همه‌ی چاپگرها از این مشخصه پشتیبانی نمی‌کنند.)

هنگام شروع برنامه، مشخصات چاپگر پیش فرض مجموعه‌ی Printers مطابق مشخصات چاپگر پیش فرض ویندوز است، اما این وضعیت قابل تغییر است. به کد زیر توجه کنید. در این قطعه کد نام تمام چاپگرهای موجود در سیستم روی فرم چاپ خواهد شد.

- 1: Dim prnPrnr As Printer
- 2: For Each prnPrnr In Printers 'Steps through each
- 3: frmMyForm.Print prnPrnr.DeviceName
- 4: Next

به خط اول کد بالا توجه کنید: در این دستور یک متغیر از نوع Printer تعریف شده است. قبلاً هم دیدید که تقریباً از تمام انواع داده‌ی Visual Basic می‌توان متغیر تعریف کرد. به جای

حلقه‌ی For Each می‌توانید از یک حلقه‌ی For ساده هم استفاده کنید :

```
For prnPrntr = 1 To (Printers.Count-1)
```

البته شاید کد بالا هرگز در یک برنامه، کاربرد عملی نیابد ولی روش کار با چاپگرها را نشان می‌دهد. به یک مثال دیگر دقت کنید. در ادامه کدی را مشاهده می‌کنید که در میان تمام چاپگرهای سیستم، یک چاپگر رنگی را جستجو می‌کند.

```
1: Dim prnPrntr As Printer
2: For Each prnPrntr In Printers
3:   If prnPrntr.ColorMode = vbPRCMColor Then
4:     'Set color printer as system default.
5:     Set Printer = prnPrntr
6:     Exit For 'Don't look further
7:   End If
8: Next 'Step through all of them if needed
```

۸-۳- کنترل خروجی چاپگر

با استفاده از شیء printer می‌توانید بدون نگرانی از مشخصات فنی چاپگر با آن کار کنید. برای فرستادن خروجی به چاپگر باید از متدهای شیء Printer استفاده کنید. اما برنامه‌نویسی چاپگرها می‌تواند کاری وقت‌گیر و خسته‌کننده باشد.

بعد از انتخاب چاپگر پیش فرض، می‌توانید متن و گرافیک را روی چاپگر بفرستید. هنگام کار با شیء printer باید ابتدا تمام خروجی موردنظر را بسازید و سپس آن را یکباره به چاپگر ارسال کنید. برای ارسال سند به چاپگر دو متد وجود دارد: NewPage و EndDoc. (اگر هیچ‌یک از این متدها را به کار نبرده باشید و برنامه را پایان دهید باز هم سند به چاپگر ارسال خواهد شد.)

۸-۳-۱ چاپ کردن روی شیء Printer

ساده‌ترین راه ارسال خروجی به چاپگر، استفاده از متد Print است. به دستورات زیر توجه

کنید :

```
Printer.Print "This report shows sales figures for"
Printer.Print dteStart; "through" ;dteFinish;
Printer.Print "If you need more information, call ext. 319."
```

جزئیات کار به وسیله‌ی ویندوز کنترل خواهد شد. به‌عنوان مثال اگر چاپگر کاغذ نداشته باشد، این ویندوز است که پیام لازم را به کاربر می‌دهد تا وی اقدام به رفع مشکل (یا انصراف از چاپ) نماید.

```
SngTaxRate = 34
StrTitle = "Sands in thd Bench"
Printer.Print strTitle & "is the name of the book"
Printer.Print "Your tax rate is" & sngTaxRate
```

هر کاری که با متد Print فرم می توانستیم انجام دهیم، با شیء Printer هم قابل انجام است. به مثال زیر توجه کنید :

```
Printer.Print "Print a blank line"
```

مشاهده می کنید که متد Print متدی کلی است و می تواند با شیءهای مختلف کار کند. برای چاپ مطالب ارسال شده به چاپگر در ابتدای صفحه ی کاغذ جدید، باید از متد NewPage استفاده کنید :

```
Printer.NewPage "Go to top of next page"
```

۲-۳-۸- اندازه گذاری خروجی چاپگر

اندازه گذاری خروجی چاپگر اقدامی مهم در ایجاد خروجی مناسب است. برای تنظیم سیستم مختصات چاپگر نیز از مشخصه ی ScaleMode استفاده می شود. در جدول ۲-۸ مشخصه های سیستم مختصات چاپگر را ملاحظه می کنید.

جدول ۲-۸- مشخصه های سیستم مختصات چاپگر

مفهوم	مشخصه
حاشیه ی چپ چاپ.	ScaleLeft
مقیاس سیستم مختصات چاپگر.	ScaleMode
مختصه ی عمودی چاپ.	ScaleHeight
حاشیه ی بالای چاپ.	ScaleTop
مختصه ی افقی چاپ.	ScaleWidth

به عنوان مثال، برای تنظیم حاشیه ی بالا (به اندازه ی ۵ نویسه) و حاشیه ی چپ (به اندازه ی ۸ نویسه) می توانید از قطعه کد زیر استفاده کنید :

```
Printer.ScaleMode = vbCharacters 'Set scale to chars
Printer.ScaleTop = 5
Printer.ScaleLeft = 8
```

۳-۳-۸- مشخصه‌های CurrentX و CurrentY

هنگام شروع برنامه مشخصه‌های CurrentX و CurrentY هر دو ° هستند و به منتهی الیه چاپ - بالای کاغذ اشاره می‌کنند. این دو مشخصه همیشه از واحد تنظیم شده در ScaleMode استفاده می‌کنند و مختصات مکان چاپ نویسه بعدی را مشخص می‌کنند. در مثال زیر، یک پیام در سطر ۱۵ و ستون ۲۵ چاپ خواهد شد :

```
Printer.ScaleMode = vbCharacters
Printer.CurrentY = 14
'Remember the starting value = 0
Printer.CurrentX = 24
Printer.Print "Warning. Warning, there's danger ahead."
```

شیء printer دارای متدهای متعددی برای کنترل فرایند چاپ است. با متد NewPage قبلاً آشنا شدید. اگر در هر لحظه از چاپ، از ادامه‌ی کار منصرف شدید، می‌توانید با متد زیر همه‌ی کارهای انجام شده را لغو کنید :

```
Printer.KillDoc 'Don't send the output to the next page
```

متد KillDoc هر چه را که در حال ارسال به چاپگر باشد، از بین می‌برد. این متد نمی‌تواند کارهایی را که قبلاً به چاپگر فرستاده شده‌اند، حذف کند و قادر به متوقف کردن متد PrintForm نیز نیست.

مایکروسافت توصیه می‌کند که برای کارهای چاپی خود یک روال چند منظوره نوشته آن را در یک مدول قرار دهید. از این روال برای چاپ گرافیک هم می‌توانید استفاده کنید. در کد زیر، می‌توانید بدنه‌ی چنین روالی را مشاهده کنید.

```
1: Sub PrintAnywhere (Src As Object, Dest As Object)
2:   Dest.PaintPicture Src.Picture, Dest.Width/2, Dest.Height/2
3:   If TypeOf Dest Is Printer Then
4:     Printer.EndDoc
5:   End If
6: End Sub
```

فرض کنید فرمی با یک کادر تصویر دارید و می‌خواهید تصویر این کنترل را چاپ کنید یا فرمی دارید که می‌خواهید بعد از تکمیل شدن به وسیله‌ی کاربر آن را به چاپگر بفرستید. روال فوق، دو آرگومان می‌گیرد : مبدأ و مقصد چاپ. آرگومان مبدأ همواره فرم موردنظر است و آرگومان مقصد می‌تواند چاپگر باشد. چگونگی استفاده از این روال چنین است :

Call PrintAnywhere (frmUserForm, Printer) 'Print form

روال PrintAnywhere از متد PaintPicture برای چاپ خروجی استفاده می‌کند. متد PaintPicture شیء ارسال شده به آن را ترسیم می‌کند. این متد سه آرگومان می‌گیرد: شیء مورد نظر، عرض و ارتفاع رسم. در روال فوق، شیء ارسال شده با مقیاس 1/2 ترسیم خواهد شد. اگر مقصد روال، چاپگر باشد، دستور If (خط ۳) آن را به چاپگر خواهد فرستاد. دستور ... Is TypeOf روشی برای تشخیص کنترل مورد نظر است.

۴-۳-۸ استفاده از قلم^۱

قلم یکی از جنبه‌های مهم چاپ متن است که باید به آن توجه شود. جدول ۳-۸ مشخصه‌های قلم شیء Printer را نشان می‌دهد.

جدول ۳-۸ - مشخصه‌های قلم شیء چاپگر

مشخصه	مفهوم
Font	تنظیمات قلم را برمی‌گرداند.
FontBold	سیاه بودن قلم را تعیین می‌کند (True/False).
FontCount	تعداد قلم‌های چاپگر را برمی‌گرداند.
FontItalic	کج بودن قلم را تعیین می‌کند (True/False).
FontName	نام قلم جاری چاپگر.
Fonts	مجموعه قلم‌های چاپگر. از اندیس 0 شروع می‌شود و تا 1-FontCount پیش می‌رود.
FontSize	اندازه‌ی قلم.
FontStrikeThru	خط‌دار بودن قلم را تعیین می‌کند (True/False).
FontTransparent	شفاف بودن قلم را تعیین می‌کند (True/False).
FontUnderline	زیرخط‌دار بودن قلم را تعیین می‌کند (True/False).

از مشخصه‌های قلم می‌توان برای زیباتر کردن خروجی چاپ استفاده کرد. در قطعه کد زیر عبارت Visual Basic با حروف درشت به چاپگر فرستاده می‌شود:

```
Printer.FontItalic = True  
Printer.FontBold = True  
Printer.FonSize = 72 '1-inch letters  
Printer.Print "Visual Basic"
```

^۱ _Font

متدهای چاپگر فقط به چاپگر اختصاص ندارند و از آن‌ها برای فرم‌ها هم می‌توان استفاده کرد. به کد زیر توجه کنید و سعی کنید حدس بزنید که چه چیزی روی فرم چاپ خواهد شد. (اگر به جای فرم از چاپگر استفاده کنید، مشاهده خواهید کرد که دو خروجی چندان شبیه یکدیگر نیستند.)

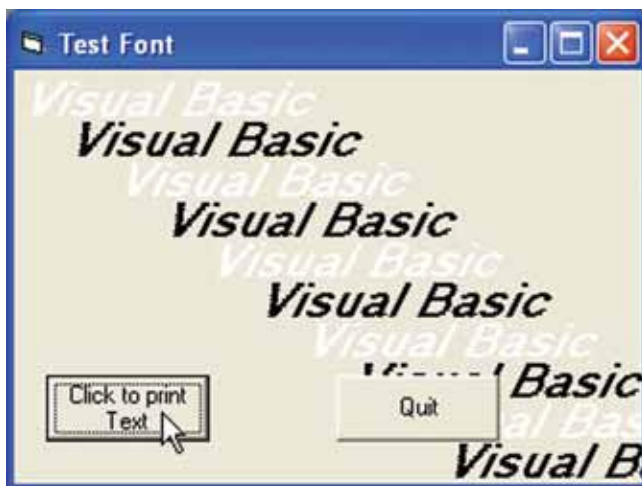
```
1: Private Sub cmdPrint_Click()
2:   'Produces an interesting output on
3:   'the form using the Print method
4:   Dim intCtr As Integer
5:   Dim intCurX As Integer
6:   Dim intCurY As Integer
7:
8:   'First set up the font
9:   frmPrint.FontItalic = True
10:  frmPrint.FontBold = True
11:  frmPrint.FontSize = 18
12:
13:  'Set measurements to TWIPs
14:  frmPrint.ScaleMode = vbTwips
15:
16:  'Save the current X and Y TWIP locations
17:  'each time the loop iterates
18:  For intCtr = 1 To 10
19:    intCurX = frmPrint.CurrentX
20:    intCurY = frmPrint.CurrentY
21:    'Print either black or white
22:    If (intCtr Mod 2) = 1 Then 'Even loop counter
23:      frmPrint.ForeColor = vbWhite
24:    Else
25:      frmPrint.ForeColor = vbBlack
26:    End If
27:    'Output the text in the large font
28:    frmPrint.Print "Visual Basic"
29:    '
30:    'Reposition the print location
31:    frmPrint.CurrentX = intCurX + 350
32:    frmPrint.CurrentY = intCurY + 300
```

```

33: Next intCtr
34: End Sub
35:
36: Private Sub cmdExit_Click()
37:     End
38: End Sub

```

در خط ۹ تا ۱۱ یک عبارت بزرگ و کج (italic) تهیه می‌شود تا به دفعات چاپ شود. در خط ۱۴ مقیاس سیستم با twips تنظیم می‌شود. از خط ۱۸ حلقه‌ای شروع می‌شود که عبارت Visual Basic را ده بار روی فرم می‌نویسد. عبارت‌های چاپ شده یک در میان سیاه/سفید هستند (اگر روی چاپگر رنگی کار می‌کنید، رنگ سفید را تغییر دهید). در هر بار تکرار حلقه، مقدار CurrentX و CurrentY افزایش می‌یابد (خطوط ۳۱ و ۳۲) تا عبارت‌ها روی هم چاپ نشوند. یکی از قسمت‌های جالب این روال دستور If خط ۲۲ است: با استفاده از عملگر Mod عبارت زوج و فرد از هم تفکیک می‌شوند تا تغییر رنگ یک در میان صورت گیرد. شکل ۸-۱ خروجی برنامه را نشان می‌دهد.



شکل ۸-۱- استفاده از متد Print

۸-۴ چاپ فرم

یکی از ساده‌ترین راه‌های ایجاد خروجی، چاپ مستقیم فرم است. Visual Basic دارای متدی به نام PrintForm است که کل یک فرم را به چاپگر می‌فرستد.

متد PrintForm روی شیء Form کار می کند. با دادن دستور PrintForm، ویژوال بیسیک بلافاصله شروع به چاپ فرم مشخص شده خواهد کرد. بنابراین قبل از اجرای این دستور، همه‌ی مقدمات کار را آماده کنید. در این قسمت خواهید دید که چگونه می توان از متد PrintForm به نحو مطلوب استفاده کرد و خروجی های مناسب به وجود آورد. (البته این متد بهترین راه ایجاد خروجی روی چاپگر نیست، اما همان طور که گفتیم ساده ترین روش است.) شکل کلی متد PrintForm چنین است :

```
[frmFormName.]PrintForm
```

دقت کنید که نام فرم الزامی نیست و اگر قید نشود، ویژوال بیسیک فرم جاری را چاپ خواهد کرد. به عنوان مثال، برای چاپ فرمی به نام frmAccPayable باید از دستور زیر استفاده کنید :

```
frmAccPayable.PrintForm 'Print the Accounts Payable form
```

و اگر فرم frmAccPayable فرم جاری برنامه باشد، دستور ساده زیر همان کار را انجام خواهد داد :

```
PrintForm 'Print the Accounts Payable form
```

یکی از مهم ترین معایب متد PrintForm این است که تمام کنترل های روی فرم را هم چاپ خواهد کرد. یکی دیگر از معایب این متد این است که علیرغم وضوح بالای چاپگر، چاپ فقط با وضوح 96 dpi انجام می شود و این سبب می شود تا کیفیت چاپ مانند آنچه در صفحه نمایش دیده می شود، نباشد. در هنگام چاپ فرم ها با متد PrintForm باید مشخصه AutoRedraw فرم با Ture تنظیم شود تا عناصر گرافیکی فرم در پشت عناصر متنی آن چاپ شوند. در کد زیر، مثالی از چاپ با متد PrintForm را مشاهده می کنید.

- 1: Dim blnAutoRedraw As Boolean 'Holds value of AutoRedraw
- 2:
- 3: frmBlank.Print "This is a Division Listing"
- 4: frmBlank.Print 'Blank line
- 5: frmBlank.Print "Division"; Tab(20); "Location"
- 6: frmBlank.Print "-----"; Tab(20);
- 7: frmBlank.Print "North"; Tab(20) ; "Widgets"
- 8: frmBlank.Print "South"; Tab(20) ; "Presses"
- 9: frmBlank.Print "East"; Tab(20); "Dye Tools"
- 10: frmBlank.Print "West"; Tab(20); "Grinders"
- 11: '

```

12: 'Save the form's AutoRedraw Property
13: '
14: blnAutoRedraw = frmBlank.AutoRedraw
15: '
16: 'Now Print the form
17: '
18: frmBlank.AutoRedraw = True
19: frmBlank.PrintForm
20:
21: 'Restore the AutoRedraw Property
22: '
23: frmBlank.AutoRedraw = blnAutoRedraw

```

به روش ذخیره کردن مشخصه AutoRedraw فرم قبل از اجرای متد PrintForm دقت کنید. با این کار می توان همه چیز را به حالت قبل از چاپ برگرداند. یکی از مهم ترین مواردی که باید هنگام چاپ در نظر داشته باشید، مقابله با خطاهای چاپ است. در کد زیر، روش کار را مشاهده می کنید.

```

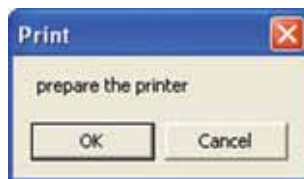
1: Private Sub cmdPrintForm_Click()
2:     Dim intBtnClicked As Integer
3:     On Error Goto ErrHandler 'Set up error handler.
4:     frmAccPayable.PrintForm 'Print form.
5:     Exit Sub
6: ErrHandler:
7:     intBtnClicked = MsgBox("A printer problem exists", vbExclamation,-
    "Print Error")
8: End Sub

```

متد PrintForm بهترین روش برای چاپ فرم های کاربردی (مانند فرم مشخصات و از این قبیل) است. اما دقت کنید که قبل از چاپ فرم تمام کنترل های غیر ضروری (مانند نوار عنوان فرم، منوها، دکمه ها و ...) را نامرئی کنید تا در فرم چاپی ظاهر نشوند. اجازه ندهید که برنامه قبل از کسب اجازه از کاربر چاپ را شروع کند. معمولاً کاربر باید خود را برای چاپ آماده کند. به عنوان مثال، چاپگر را روشن کند یا کاغذ در آن قرار دهد. برای اطلاع دادن به کاربر از کادر پیام استفاده کنید. در کد زیر، تابعی را برای این منظور مشاهده می کنید که می توانید با اصلاحاتی از آن استفاده کنید. این روال کادر پیام شکل ۲-۸ را نمایش خواهد داد.

این کادر پیام با این که چندان چشمگیر نیست اما به کاربر فرصت لازم برای آماده کردن مقدمات را می دهد.

```
1: Public Function PrReady() As Boolean
2:     'Make sure the user is ready to print
3:     Dim intIsReady As Integer
4:
5:     'The user will respond to the foolowing
6:     'message box when ready for the printing
7:     intIsReady = MsgBox("prepare the printer",vbOKCancel, "Print")
8:
9: If (intIsReady = vbCancel) Then
10:         PrReady = False
11: Else
12:         PrReady = true
13: End If
14: End Function
```



شکل ۲-۸- کاربر بعد از کسب آمادگی کار را ادامه می دهد.

این تابع یک مقدار Boolean برمی گرداند که آمادگی کاربر برای ادامه کار (یا انصراف) را تأیید می کند و می توانید در سایر قسمت های برنامه از مقدار برگشتی آن استفاده کرده و تصمیم مناسب را اتخاذ کنید. در کد زیر، برنامه ای را مشاهده می کنید که از مقدار برگشتی تابع PrReady () استفاده کرده است.

```
1: Private Sub cmdPrint_Click()
2:     'Print when ready, but do nothing
3:     'if the clicks is not ready
4:     If PrReady() Then
5:         'Call ReportPrint
6:     End If
```

7: End Sub

دستور If در خط ۴ با بررسی مقدار برگشتی تابع PrReady() چاپ را انجام خواهد داد.

خودآزمایی و تمرین

- ۱- اگر برنامه‌ی من هم‌زمان با سایر برنامه‌ها چیزی را به چاپگر بفرستد، چه اتفاقی خواهد افتاد؟
- ۲- آیا بهتر نیست ابتدا همه‌چیز را روی فرم قرار دهیم و در پایان فرم را به چاپگر بفرستیم؟
- ۳- یک برنامه چگونه می‌تواند تعداد چاپگرهای نصب شده در یک سیستم را تعیین کند؟
- ۴- کدام مشخصه مقیاس اندازه‌گیری چاپگر را تعیین می‌کند؟
- ۵- چگونه می‌توان چاپگر را وادار کرد یک صفحه‌ی جدید چاپ کند؟
- ۶- با کدام If می‌توانید نوع داده‌ی یک شیء را مشخص کنید؟
- ۷- کدام کلمه‌ی کلیدی به فرم جاری اشاره می‌کند؟
- ۸- متد PrintForm معمولاً با چه وضوحی چاپ می‌کند؟
- ۹- قبل از چاپ فرم، مشخصه AutoRedraw آن چه مقداری باید داشته باشد؟
- ۱۰- کدی بنویسید که نامتان را از ستون ۳۲ چاپگر چاپ کند.
- ۱۱- برنامه‌ای بنویسید که با پیدا کردن چاپگر رنگی یک متغیر Boolean را با True مقداردهی کند. اگر چاپگر رنگی وجود نداشته باشد، چاپگر پیش فرض نباید تغییر کند. چون این روال باید مقدار Boolean خود را به روال‌های دیگر بفرستد، آن را به صورت تابع بنویسید.