

## توانایی استفاده از انواع حلقه‌ها و ساختار Case Select و کنترل دکمه انتخاب

### هدف‌های رفتاری

- پس از مطالعه این واحد کار از فراگیر انتظار می‌رود که:
- ۱- بتواند از دستور Select Case استفاده کند.
  - ۲- بتواند از کنترل دکمه انتخاب استفاده کرده، رویدادها و ویژگی‌های آن را به کاربرد.
  - ۳- نحوه کار با حلقه‌های Do While...Loop.While...Wend.For...Next و غیره را توضیح دهد.
  - ۴- بتواند از دستورات خروج از حلقه Exit For و Exit Do استفاده کند.

## کلیات

یکی از مسایلی که در پروژه‌های واقعی با آن برخورد خواهید کرد تکراری بودن بعضی از دستورات و عملیات است. این دستورات تکراری ممکن است به تعداد دفعات معین تا رسیدن به شرایط خاصی انجام شوند. ساختارهای تکرار یا به عبارت دیگر حلقه‌ها، در این زمینه به شما کمک می‌کنند تا این‌گونه عملیات را راحت‌تر و مناسب‌تر انجام دهید. به علاوه معمولاً هنگام طراحی برنامه‌ها لازم است تا روند اجرای دستورات را تعیین کنید. ساختارهای تصمیم، امکانات اجرایی دستورات را با توجه به شرایط مورد نظر فراهم می‌کنند. از این گروه نحوه استفاده از دستور If را فراگرفته‌اید. نوع دیگری از ساختارهای تصمیم نیز به نام Select Case وجود دارند که در این واحدکار به توضیح آن می‌پردازیم.

### ۱-۶ دستور Select Case

این دستور مانند دستور If به برنامه‌نویس اجازه می‌دهد تا در شرایط مختلف دستورات مورد نظر خود را اجرا کند و عملیات مناسب را با داده‌ها و اطلاعات موجود انجام دهد. استفاده از دستور If زمانی که نیاز به بررسی شرط‌ها و انجام مقایسه‌های متعدد باشد سبب شلوغ شدن برنامه می‌شود. دستور Select Case زمانی استفاده می‌شود که یک متغیر یا حاصل یک عبارت، دارای مقادیر مختلفی باشد و بخواهیم براساس مقدار خاصی عملیات ویژه‌ای انجام دهیم. به عبارت دیگر Select Case جایگزین If‌هایی خواهد بود که می‌توانند مقادیر مختلف یک متغیر یا عبارت را بررسی کنند. در صورت به کارگیری دستور Select Case نوشتن کدها آسان‌تر و برنامه از خوانایی و دقت بیشتری برخوردار می‌شود. شکل کلی این دستور به این صورت است:

عبارت مورد مقایسه    Select Case

Case مقدار اول    :  
دستور(ات)


Case مقدار دوم    :  
دستور(ات)

Case مقدار سوم    :  
دستور(ات)


Case Else    :  
دستور(ات)

End Select    :  
دستور(ات)

در این دستور ابتدا مقدار عبارت مورد مقایسه با مقادیری که در مقابل هر Case قرار داده شده است، مقایسه می‌شود. اگر مقدار ذخیره شده در عبارت مورد نظر با مقدار ذکر شده در مقابل اولین Case برابر باشد، دستورات موجود در این Case اجرا می‌شوند و از بررسی سایر Case‌ها صرف نظر خواهد شد؛ اما اگر اولین مقایسه، نتیجه نادرست در پی داشته باشد، Case دوم بررسی می‌شود و در صورت درست بودن نتیجه دستورات، این Case اجرا می‌شود و در غیر این صورت Case بعدی بررسی می‌شود و به همین شکل تمام Case‌ها به ترتیب بررسی می‌شوند و اگر نتیجه بررسی تمام آن‌ها نادرست باشد، دستورات موجود در بخش Case Else اجرا خواهند شد و سپس برنامه ادامه می‌یابد.

**نکته** استفاده از بخش Case Else اختیاری است. 

برای ایجاد یک بازه از مقادیر عددی در Case‌ها، می‌توانید از کلمه کلیدی To استفاده کنید. به عنوان مثال برای مقادیر عددی بین ۱۲ تا ۱۴ می‌توانید از Case 12 To 14 استفاده کنید. برای استفاده از چند مقدار مختلف در یک Case می‌توانید از کاراکتر کاما (،)، کلمه کلیدی To، Is یا ترکیبی از آن‌ها استفاده کنید.

**مثال ۱:**  پروژه‌ای مطابق شکل ۶-۱ و جداول ۶-۱ و ۶-۲ طراحی کنید که سه نمره یک دانش‌آموز را دریافت کند و رتبه وی را مشخص کند. اگر معدل وی بین ۱۸ تا ۲۰ باشد، رتبه A و اگر بین ۱۶ الی ۱۸ باشد رتبه B و بین ۱۴ الی ۱۶ رتبه C و کوچک‌تر از ۱۴ رتبه D برای او نمایش داده شود. برای این کار عملیات زیر را به ترتیب انجام دهید:

۱ - یک پروژه جدید به همراه یک فرم و کنترل‌های آن مطابق شکل ۶-۱ ایجاد کنید.

جدول ۶-۱ خصوصیات فرم

خصوصیت	مقدار
Name	frmave
Caption	Average



شکل ۶-۱

جدول ۲-۶ خصوصیات کنترل‌ها

کنترل خصوصیت	Label	Label	Label	TextBox	TextBox	TextBox	Command Button
Name	lblit	lblwin	lblvb	txtit	txtwin	txtvb	cmdave
Caption	IT :	WINDOWS :	VB6:	—	—	—	&Average

۲ - سپس رویداد Click را به این صورت تنظیم کنید:

```
Private Sub cmdave_Click()
```

```
    Dim sngit As Single
```

```
    Dim sngwin As Single
```

```
    Dim sngvb As Single
```

```
    Dim sngav As Single ,strrank As String*1
```

```
    sngit = Val(txtit.Text)
```

```
    sngwin = Val(txtwin.Text)
```

```
    sngvb = Val(txtvb.Text)
```

```
    sngav = (sngit+sngwin + sngvb)/3
```

```
    Select Case sngav
```

```
        Case 18 To 20
```

```
            strrank = "A"
```

```
        Case 16 To 18
```

```
            strrank = "B"
```

```
        Case 14 To 16
```

```
            strrank = "C"
```

```
        Case Is < 14
```

```
            strrank = "D"
```

```
    End Select
```

```
    MsgBox «RANK IS : «+ strrank,»RANKING»
```

```
End Sub
```

در این رویداد پس از تعریف متغیرها ابتدا مقادیر تایپ شده در کادرهای متن با استفاده از تابع Val به عدد تبدیل شده و در متغیرهای مربوطه ذخیره می‌شوند سپس متوسط سه نمره محاسبه شده و در ادامه با استفاده از یک دستور Select Case مقدار معدل (sngav) مورد ارزیابی قرار می‌گیرد.

اگر مقدار آن بین ۱۸ تا ۲۰ باشد دستور موجود در شرط Case 18 To 20 را اجرا می‌شود و رتبه A را برای وی در متغیر رشته‌ای strrank ذخیره می‌کند، سپس بدون آن که سایر شرط‌ها بررسی شوند دستور Select Case خاتمه می‌یابد؛ اما اگر مقدار متغیر sngav بین ۱۶ تا ۱۸ باشد شرط موجود در اولین Case نادرست خواهد بود و شرط Case دوم یعنی Case 16 To 18 بررسی می‌شود و در نتیجه دستور مربوط به این Case اجرا می‌شود و بدون بررسی سایر Case‌ها دستور Select Case خاتمه می‌یابد و به همین شکل در صورت نادرست بودن شرط در Case دوم، سایر Case‌ها یکی یکی بررسی می‌شوند و در Case آخر اگر مقدار متغیر sngav کوچک‌تر از ۱۴ باشد شرط Case Is < 14 درست خواهد بود و رتبه مناسب برای وی در نظر گرفته می‌شود و پس از خاتمه دستور Select Case رتبه دانش‌آموز به وسیله یک کادر پیام نمایش داده می‌شود.

۳ - پروژه و فرم را با نام Average ذخیره کنید سپس آن را اجرا نمایید.

۴ - اعداد ۱۷، ۱۹ و ۲۰ را در کادرهای متن موجود تایپ کنید، سپس روی دکمه Average کلیک کرده و نتیجه را بررسی کنید.

۵ - مجدداً اعداد ۱۴، ۵ و ۸ را در کادرهای متن موجود تایپ کرده و روی دکمه Average کلیک و نتیجه را بررسی کنید.

۶ - اجرای برنامه را متوقف کرده و به پنجره ویژوال بیسیک بازگردید.



**مثال ۲:** پروژه‌ای طراحی کنید که هزینه حمل کالا به وسیله وسایل نقلیه مختلف را محاسبه کند. به این منظور کاربر وزن کالا و مسافت حمل را به همراه روش حمل آن معین می‌کند و سپس هزینه حمل کالا محاسبه شده، در اختیار وی قرار می‌گیرد. روش‌های حمل کالا و هزینه هر یک به ازای هر کیلومتر جابه‌جایی در جدول ۳-۶ ارائه شده است و برای محاسبه هزینه حمل کالا از فرمول زیر استفاده می‌شود.

هزینه به ازای یک کیلومتر جابه‌جایی مسافت جابه‌جایی وزن کالا = هزینه حمل کالا


### جدول ۳-۶

روش حمل کالا	هزینه حمل به ریال
اتوبوس	۱۰۰۰۰
قطار	۱۵۰۰۰
کشتی	۱۴۰۰۰
هواپیما	۲۰۰۰۰



شکل ۲-۶

اکنون این عملیات را به ترتیب انجام دهید:  
 ۱ - یک پروژه از نوع Standard EXE به همراه یک فرم، دو کادر متن و برچسب و یک دکمه فرمان مطابق شکل ۲-۶ ایجاد کنید.  
 کادر متن اول برای ورود وزن کالا و کادر متن دوم برای ورود مسافت حمل کالا استفاده می‌شود.

۲ - برای آن‌که کاربر توانایی انتخاب نوع روش حمل کالا را داشته باشد، از کنترل دکمه انتخاب (OptionButton) استفاده کنید. این کنترل اجازه می‌دهد تا امکان انتخاب یک گزینه از بین چند انتخاب برای کاربران فراهم شود. این کنترل معمولاً به صورت گروه‌هایی دوتایی یا بیشتر استفاده می‌شود و هر لحظه می‌توان یکی از کنترل‌ها را از مجموعه آن‌ها انتخاب کرد. برای استفاده از این کنترل در جعبه ابزار روی کنترل دکمه انتخاب  کلیک کنید و چهار کنترل از آن را مطابق شکل ۲-۶ روی فرم قرار دهید و خصوصیات آن‌ها را تنظیم کنید.

۳ - در پنجره خصوصیات، optbus را انتخاب کنید و خصوصیت Value آن را روی مقدار True تنظیم کنید. کنترل‌های دکمه انتخاب دارای خصوصیتی به نام Value هستند که یک خصوصیت منطقی است و می‌تواند True یا False باشد. اگر مقدار این خصوصیت روی True تنظیم شود نشان دهنده این است که دکمه انتخاب مربوطه انتخاب شده است و در غیر این صورت نشان‌دهنده عدم انتخاب کنترل است. همواره در یک گروه از دکمه‌های انتخاب، خصوصیت Value یکی از کنترل‌های دکمه انتخاب، True و برای سایر کنترل‌ها False می‌باشد.

۴ - اکنون باید دستورات مورد نظر را در رویداد Click دکمه فرمان Compute بنویسید.

بنابراین رویداد مزبور را به این صورت تنظیم کنید:

```
Private Sub cmdcom_Click()
    Dim intrans As Integer, sngprice As Currency
    Dim sngw As Single, sngl As Single
    If optbus.Value = True Then intrans=0
    If opttrain.Value = True Then intrans =1
    If optship.Value = True Then intrans =2
    If optplane.Value = True Then intrans =3
    sngw = Val(txtw.Text)
    sngl = Val(txtl.Text)
    Select Case intrans
        Case 0
            sngprice =sngw * sngl *10000
        Case 1
            sngprice=sngw*sngl*15000
        Case 2
            sngprice = sngw * sngl *14000
        Case 3
            sngprice = sngw * sngl *20000
    End Select
    MsgBox "Total Price =" + Str(sngprice)+ "Rials", , " Payment"
End Sub
```

در این رویداد با استفاده از دستور If مقدار خصوصیت Value هر یک از کنترل‌های دکمه انتخاب بررسی می‌شود و در صورت انتخاب هر یک از آن‌ها متغیر intrans متناسب با روش حمل انتخاب شده مقداردهی می‌شود؛ بنابراین در صورت انتخاب روش حمل با اتوبوس مقدار صفر، حمل با قطار مقدار ۱، حمل با کشتی مقدار ۲ و حمل با هواپیما مقدار ۳ در آن ذخیره می‌شود. سپس با استفاده از دستور Select Case مقدار متغیر intrans بررسی می‌شود تا با توجه به روش حمل انتخاب شده، هزینه حمل کالا محاسبه و در متغیر sngprice ذخیره شود، در نهایت این مقدار به وسیله یک کادر پیام نمایش داده خواهد شد.

۵- برای آن که ترتیب دریافت فوکوس در کنترل به صورت مناسب انجام شود، خصوصیت TabIndex دکمه فرمان Compute را روی عدد ۶ تنظیم کنید. در این صورت

دکمه فرمان Compute آخرین کنترلی است که در زمان فشردن کلید Tab، فوکوس را به دست می آورد.

۶ - پروژه و فرم را با نام transport ذخیره کنید.


۷ - پروژه را اجرا کنید و وزن کالا را ۲۰۰ کیلوگرم و مسافت جابه‌جایی را ۱۰۰ کیلومتر تایپ کنید، سپس روی دکمه انتخاب Plane کلیک کنید تا روش حمل کالا نیز انتخاب شود.

۸ - روی دکمه Compute کلیک کنید و نتیجه محاسبه را بررسی کنید.

۹ - مجدداً مقادیر ۳۵۰ و ۲۰۰ را برای وزن کالا و مسافت جابه‌جایی تایپ کنید و

این بار روش حمل با قطار را انتخاب کرده، روی دکمه Compute کلیک کنید و نتیجه را بررسی نمایید.

۱۰ - اجرای برنامه را خاتمه داده و به پنجره ویژوال بیسیک بازگردید.

**نکته:**  اگر بخواهید از کنترل‌های دکمه انتخاب به گونه‌ای استفاده کنید که به صورت گروه‌های جداگانه عمل کنند باید از کنترل دیگری به نام کنترل قاب یا Frame استفاده کنید، در غیر این صورت تمام کنترل‌های دکمه انتخاب که در یک فرم قرار دارند به صورت یک گروه واحد در نظر گرفته خواهند شد. برای ایجاد گروه‌های مستقل از کنترل‌های دکمه انتخاب، باید ابتدا کنترل قاب را روی فرم قرار دهید و سپس با روش درگ، کنترل‌های دکمه انتخاب را روی کنترل قاب بگذارید. کنترل قاب، رویداد و خصوصیت ویژه‌ای ندارد و مهم‌ترین خصوصیت آن خصوصیت Caption است که عنوان کنترل قاب را تعیین می‌کند.

## ۲-۶ ساختارهای تکرار در ویژوال بیسیک

بعضی اوقات لازم است تا عملیاتی را به صورت تکراری انجام دهید. در ویژوال بیسیک دستورات متعددی برای انجام عملیات تکراری قرار داده شده‌اند که بعضی از آنها توانایی اجرای عملیات تکراری را با تعداد دفعات معین دارند و بعضی دیگر عملیات موردنظر را تا رسیدن به شرایط خاصی فراهم می‌کنند. در این واحد کار به توضیح بعضی از آنها می‌پردازیم.




## ۱-۲-۶ حلقه For ... Next

از این حلقه زمانی استفاده می‌شود که لازم باشد دستورات را به تعداد دفعات معین و محدودی اجرا کنید. نحوه استفاده از این حلقه به این صورت است:

مقدار خاتمه To مقدار شروع = شمارنده حلقه For  
 .....  
 دستورات  
 Next شمارنده حلقه

برای استفاده از این گونه حلقه‌ها از یک متغیر به عنوان شمارنده حلقه استفاده می‌شود که به وسیله آن تعداد دفعات تکرار حلقه کنترل می‌شود. مقادیر شروع و خاتمه با توجه به نیاز برنامه‌نویس تنظیم می‌شود و معمولاً از مقادیر عددی صحیح استفاده می‌شود. اما می‌توان از هر نوع متغیر عددی یا عبارات ریاضی نیز استفاده کرد. این حلقه با کلمه کلیدی For آغاز شده و با کلمه کلیدی Next که به همراه نام متغیر شمارنده ذکر می‌شود، خاتمه می‌یابد و دستورات مورد نظر برای اجرا در بین آن‌ها قرار می‌گیرند. زمانی که اجرای برنامه به اولین خط از حلقه یعنی For می‌رسد، مقدار شروع در متغیر شمارنده ذخیره می‌شود و سپس این مقدار با مقدار خاتمه مقایسه می‌شود و در صورتی که کوچک‌تر یا مساوی با مقدار خاتمه باشد، دستورات بین For و Next اجرا می‌شود. با رسیدن به کلمه کلیدی Next، اجرای برنامه مجدداً به بخش For منتقل می‌شود، سپس یک واحد به مقدار قبلی متغیر شمارنده اضافه شده و در صورتی که این مقدار کوچک‌تر یا مساوی با مقدار خاتمه باشد، دستورات موجود در حلقه اجرا می‌شود و این عملیات تا زمانی که مقدار متغیر شمارنده از مقدار خاتمه بزرگ‌تر شود، ادامه می‌یابد. پس از خاتمه اجرای حلقه، اجرای دستورات بعد از کلمه کلیدی Next انجام می‌شود.

 **مثال ۳:** پروژه‌ای طراحی کنید که مجموع اعداد ۱ تا ۱۰ را محاسبه کرده و روی فرم نمایش دهد. به این منظور عملیات زیر را به ترتیب انجام دهید:



شکل ۳-۶

۱ - برنامه ویژوال بیسیک را اجرا کنید  
 سپس یک پروژه از نوع Standard EXE به همراه یک فرم و یک کنترل دکمه فرمان و یک کنترل برچسب مطابق شکل ۳-۶ ایجاد کنید.

۲ - رویداد Click دکمه فرمان cmdshow را به صورت زیر تنظیم کنید:

```
Private Sub cmdshow_Click()
```

```
Dim inti As Integer
```

```
Dim intsum As Integer
```

```
For inti = 1 To 10
```

```
intsum = intsum + inti
```

```
Next inti
```

```
lblsum. Caption = lblsum. Caption + Str (intsum)
```

```
End Sub
```

در این رویداد ابتدا متغیرهای مورد نیاز تعریف می‌شوند. به منظور محاسبه مجموع اعداد ۱ تا ۱۰ از یک حلقه For استفاده شده است. این حلقه از عدد ۱ شروع و تا رسیدن حلقه به مقدار ۱۰ تکرار می‌شود و در هر مرحله از اجرای حلقه نیز با استفاده از متغیر intsum مجموع اعداد محاسبه می‌شود و با پایان یافتن حلقه مقدار مجموع intsum پس از تبدیل به نوع داده رشته‌ای در کنترل برچسب نمایش داده می‌شود.

- ۳- پروژه و فرم را با نام sumnumbers ذخیره کنید سپس آن را اجرا کرده و آزمایش نمایید.  
 ۴- به اجرای پروژه خاتمه داده و به پنجره ویژوال بیسیک بازگردید.



**مثال ۴:** پروژه‌ای طراحی کنید که یک عدد طبیعی را از ورودی دریافت کند و مجموع اعداد زوج کوچک‌تر یا مساوی آن را نمایش دهد.

- ۱- برنامه ویژوال بیسیک را اجرا کرده، یک پروژه به همراه یک فرم، سه کنترل برچسب و یک کنترل کادر متن به همراه یک کنترل دکمه فرمان مطابق شکل ۴-۶ و جدول ۴-۶ ایجاد کنید.

جدول ۴-۶ خصوصیات فرم

خصوصیت	مقدار
Name	frmsn
Caption	Show Number



شکل ۴-۶

۲ - سپس رویداد Click دکمه Show Number را به این صورت تنظیم کنید:

```
Private Sub cmdshow_Click()
    Dim inti As Integer
    Dim intno As Integer
    Dim lngsum As Long
    intno = Val(txtno.Text)
    For inti = 1 To intno
        If intno mod inti=0
            lngsum=lngsum+inti
        End If
    Next inti
    lblsum.Caption=lblsum.Caption+Str(lngsum)
End Sub
```

۳ - پروژه و فرم را با نام shownumber ذخیره کنید سپس آن را اجرا کرده و آزمایش کنید.

۴ - اجرای پروژه را متوقف کرده و به پنجره ویژوال بیسیک باز گردید. اکنون شکل کلی حلقه For به شما معرفی می‌شود. همان‌طور که گفته شد متغیر شمارنده حلقه در هر بار اجرای حلقه به میزان یک واحد افزایش می‌یابد، اما گاهی اوقات لازم است تا میزان افزایش شمارنده به وسیله برنامه‌نویس تعیین شود یا از حلقه‌های کاهشی استفاده شود که در آن‌ها مقدار شروع از مقدار خاتمه بزرگ‌تر است و در هر بار اجرای حلقه مقدار شمارنده کم می‌شود. شکل کلی دستور For به این صورت است:

```
For میزان افزایش یا کاهش شمارنده Step مقدار خاتمه To مقدار شروع = شمارنده حلقه
    ↓
    دستورات
Next شمارنده حلقه
```

در واقع بخش Step امکان تنظیم میزان افزایش یا کاهش مقدار شمارنده حلقه را فراهم می‌کند. مقدار پیش فرض برای این بخش مقدار یک است. می‌توانید برای حلقه‌های افزایشی از مقادیر مثبت و برای حلقه‌های کاهشی از مقادیر منفی استفاده کنید. در این صورت حلقه زمانی متوقف می‌شود که مقدار شروع از مقدار خاتمه کوچک‌تر شود.



**نکته** در صورتی که مقدار شروع از مقدار خاتمه بزرگ تر باشد و مقدار Step مثبت باشد، حلقه اجرا نخواهد شد.

در صورت عدم استفاده از بخش Step مقدار افزایش شمارنده حلقه یک واحد خواهد بود

## دستور Print

دستور Print می تواند انواع مقادیر ثابت، متغیر و مقدار خصوصیت اشیا را روی فرم نمایش دهد. در صورتی که از این دستور به تنهایی استفاده شود فقط یک خط خالی نمایش داده می شود و اگر از کاراکتر سمی کالن (;) استفاده شود، خروجی ها پشت سرهم و بدون فاصله قرار می گیرند و اگر از کاراکتر کاما (,) استفاده شود، هر خط نمایشی در روی فرم، توانایی نمایش ۱۰ کاراکتر را خواهد داشت و اگر خروجی بزرگ تر از ۱۰ کاراکتر باشد، خروجی بعدی در یک ناحیه جلوتر نمایش داده خواهد شد. به عنوان مثال به این دستورات توجه کنید:

```
Print "Ali" ; "Reza"
```

```
Print "Visual", "Basic"
```

```
Print "Student" ;
```

```
Print "Information"
```

در صورت اجرای دستورات فوق خروجی حاصل به این شکل خواهد بود:

```
AliReza
```

```
Visual Basic
```

```
StudentInformation
```

عبارات موجود در دستور اول به دلیل استفاده از کاراکتر سمی کالن پشت سرهم و عبارات موجود در دستور دوم به دلیل استفاده از کاراکتر کاما با فاصله از هم قرار گرفته اند و در دو دستور آخر نیز چون Print سوم به کاراکتر سمی کالن ختم می شود، خروجی Print چهارم پشت سر آن نمایش داده می شود.

به عنوان مثال به این دستورات توجه کنید و نتیجه اجرای آن ها را در شکل ۵-۶ مشاهده کنید.

```
Print "1234567890" , "1234567890"
```

```
Print "12345678901234567890"
```

```
Print "12345678901" , "1234567890"
```



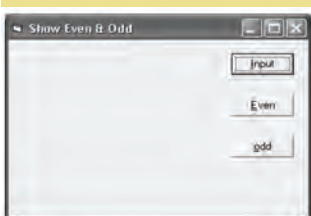
شکل ۵-۶

در دستور Print اول هر عبارت نمایشی از ده کاراکتر تشکیل شده است؛ بنابراین دو عبارت نمایشی به ترتیب در نواحی اول و دوم در کنار هم قرار می‌گیرند. اما در دستور Print آخر چون عبارت نمایشی اول از ۱۱ کاراکتر تشکیل شده است، عبارت نمایشی دوم در ناحیه ۳ نمایش داده می‌شود. دستور Print دوم برای تشخیص بهتر اندازه نواحی استفاده شده است.

### تمرین:



پروژه‌ای را طراحی کنید تا مطابق شکل ۶-۶ توانایی نمایش اعداد فرد و زوج کوچک‌تر از هر عدد طبیعی دلخواه موردنظر کاربر را داشته باشد. برای دریافت داده‌ها از کادر ورود داده استفاده کنید.



شکل ۶-۶

حلقه‌های For را می‌توان در داخل یکدیگر قرار داد و در این حالت با هر بار اجرای حلقه For اول، حلقه For داخل آن یک دور کامل اجرا می‌شود. شکل کلی نحوه استفاده از حلقه‌های متداخل به صورت زیر است:

میزان افزایش یا کاهش شمارنده Step مقدار خاتمه To شمارنده حلقه = مقدار شروع For

میزان افزایش یا کاهش شمارنده Step مقدار خاتمه To شمارنده حلقه = مقدار شروع For

.....  
دستور(ات)

Next شمارنده حلقه

Next شمارنده حلقه



شکل ۶-۷

**مثال ۵:** پروژه‌ای طراحی کنید که با استفاده



از حلقه For کاراکترهای ستاره را مطابق شکل ۶-۷ نمایش دهد. برای این کار عملیات بعد را انجام دهید.

۱ - یک پروژه از نوع Standard EXE به همراه یک فرم و یک کنترل دکمه فرمان مطابق شکل ۶-۷ ایجاد کنید.

۲ - رویداد Click دکمه فرمان را به این صورت تنظیم کنید:

```
Private Sub cmdsshow_Click()
```

```
Dim inti As Integer
```

```
Dim intj As Integer
```

```
For inti = 1 To 4
```

```
For intj= 1 To 5
```

```
Print "*";
```

```
Next intj
```

```
Print
```

```
Next inti
```

```
End Sub
```

۳ - پروژه و فرم را با نام showstart ذخیره کنید سپس برنامه را اجرا کرده و روی دکمه show کلیک کنید.

۴ - به اجرای برنامه خاتمه داده و به پنجره ویژوال بیسیک بازگردید.

**تمرین:**



پروژه‌ای طراحی کنید که شکل زیر را رسم نماید.

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

## ۲-۲-۶ حلقه While ... Wend و While ... Loop

این حلقه‌ها، دستورات را تا زمانی که شرط یا شرط‌های تعیین شده درست باشند، اجرا می‌کنند. از این نوع حلقه‌ها زمانی استفاده می‌شود که دفعات تکرار دستورات معین نباشد. شکل کلی این دستورات به صورت زیر است:

**While** (شرط یا شرط‌ها)  
⋮  
دستورات

**Do While** (شرط یا شرط‌ها)  
⋮  
دستورات

**Wend**

**Loop**

زمانی که اجرای برنامه به حلقه While برسد، ابتدا شرط یا شرط‌های موجود در جلوی کلمه کلیدی While ارزیابی می‌شوند و اگر نتیجه این ارزیابی درست باشد، دستورات بین While و Wend اجرا می‌شوند. با رسیدن به انتهای حلقه یعنی کلمه کلیدی Wend اجرای برنامه مجدداً به بخش While منتقل می‌شود و بررسی شرط یا شرط‌ها انجام می‌گیرد و به همین شکل اجرای حلقه تا زمانی که نتیجه ارزیابی شرط موجود در حلقه نادرست شود، ادامه می‌یابد. نحوه اجرای حلقه Do While مانند حلقه While است.



**مثال ۶:** پروژه‌ای طراحی کنید که هر بار مسافت بین دو شهر را براساس کیلومتر دریافت کند سپس مسافت دو شهر را به متر تبدیل کرده و نمایش دهد و این کار را تا دریافت مقدار صفر انجام دهد.

۱ - یک پروژه از نوع Standard EXE به همراه یک فرم مطابق با شکل ۸-۶ ایجاد کنید.



شکل ۸-۶

۲ - یک کنترل برچسب و دو کنترل دکمه فرمان روی فرم قرار داده و خصوصیت‌های آنها را تنظیم کنید.

### ۳ - رویداد دکمه Compute را به این صورت تنظیم کنید:

```
Private Sub cmdcompute_Click ()
```

```
Dim intno As Integer
```

```
Dim strdata As String
```

```
intno = 1
```

```
While (intno > ۰)
```

```
strdata = InputBox ("Enter Your Number:" , "Enter Data" , ۰)
```

```
intno = Val (strdata)
```

```
lblresult. Caption = intno*1000
```

```
Wend
```

```
End Sub
```

در این رویداد پس از تعریف متغیرها ابتدا مقدار متغیر `intno` روی مقدار ۱ تنظیم می‌شود تا حلقه `While` بتواند اجرا شود. زیرا شرطی که برای حلقه در نظر گرفته شده است `intno < ۰` می‌باشد اگر `intno` مقداره‌ی نشود هیچ وقت امکان ورود داده و انجام محاسبات وجود نخواهد داشت. در مرحله بعد حلقه `While` اجرا می‌شود و با استفاده از یک کادر ورود مسافت دو شهر دلخواه دریافت شده و در متغیر `strdata` ذخیره می‌شود سپس با استفاده از تابع `Val` محتویات متغیر `strdata` به عدد تبدیل شده و در متغیر `intno` قرار می‌گیرد سپس با تبدیل این مقدار به واحد متر، مسافت در کنترل برچسب نمایش داده می‌شود. در این مرحله با رسیدن به دستور `Wend` اجرای برنامه به ابتدای حلقه یعنی دستور `While` منتقل می‌شود در نتیجه شرط `intno < ۰` بررسی شده و اگر نتیجه بررسی شرط درست باشد عملیات تکرار خواهد شد اما اگر نتیجه بررسی شرط نادرست باشد (در صورت ورود عدد صفر) اجرای حلقه خاتمه می‌یابد.

۴ - پروژه و فرم را با نام `Distance` ذخیره کنید.

۵ - پروژه را اجرا کرده و روی دکمه `Compute` کلیک کنید و پس از ورود چند مقدار مختلف عدد صفر را وارد کنید تا مجدداً به پنجره برنامه بازگردید.

۶ - روی دکمه `Exit` کلیک کنید و از برنامه خارج شده و به پنجره ویژوال بیسیک

بازگردید.





تمرین:

پروژه‌ای طراحی کنید تا مضرب‌های کوچک‌تر از ۱۰۰ عدد ۹ را روی فرم نمایش دهد. به این منظور عملیات زیر را به ترتیب انجام دهید:

### ۳-۲-۶ حلقه Do ... Loop While

این حلقه مشابه حلقه Do While ... Loop است؛ با این تفاوت که دستورات داخل حلقه یک بار اجرا شده، سپس شرط حلقه بررسی می‌شود. بنابراین در صورت نادرست بودن شرط، دستورات حداقل یک بار اجرا می‌شوند. شکل کلی این دستور به صورت زیر است:

**Do**



دستورات)

**Loop While** (شرط یا شرطها)



**مثال ۷:** پروژه‌ای طراحی کنید که یک عدد طبیعی را دریافت کند و مجموع ارقام آن را نمایش دهد. برای این کار عملیات زیر را به ترتیب انجام دهید:

- ۱- یک پروژه از نوع Standard EXE به همراه یک فرم با مشخصات جدول ۱۹-۷ ایجاد کنید.
- ۲- در این پروژه به منظور اجرای دستورات از رویداد Activate فرم استفاده می‌شود. این رویداد بعد از رویداد Load فرم و زمانی که فرم نمایش داده شده و فوکوس را کسب می‌کند اجرا می‌شود. بنابراین رویداد Activate فرم را به این صورت تنظیم کنید:

```
Private Sub Form_Activate()
```

```
Dim lngi As Long
```

```
Dim lngno As Long
```

```
Dim strinput As String
```

```
strinput = InputBox("Enter Your Number :","input DATA",0)
```

```
lngno = Val(strinput)
```

```
Do
```

```
lngi = lngno Mod 10
```

```
Sum=Sum+lngi
```

```
Ingno = Ingno \ 10  
Loop While (Ingno > 0)  
End Sub
```

۳ - اجرای پروژه را متوقف کرده و به پنجره ویژوال بیسیک بازگردید.

تمرین:



پروژه طراحی کنید که مضارب کوچکتر از ۱۰۰ عدد ۷ را نمایش دهد.

## ۴-۲-۶ دستورات خروج از حلقه Exit For و Exit Do

دستورات Exit For و Exit Do به برنامه نویس امکان می دهند که قبل از آن که حلقه به طور عادی خاتمه یابد به اجرای حلقه خاتمه دهد و از آن خارج شود. از دستور Exit For برای خروج از حلقه For...Next و از دستور Exit Do برای خروج از حلقه های Do...Loop While، Do While...Loop، Do...Loop Until و Do Until...Loop استفاده می شود.

**مثال ۸:** پروژه digit را به گونه ای تغییر دهید تا با استفاده از دستور Exit Do اجرای حلقه خاتمه یابد. به این منظور عملیات زیر را به ترتیب انجام دهید:  
۱ - پروژه digit را باز کنید و رویداد Activate فرم را به این صورت تغییر دهید:



```
Private Sub Form_Activate()  
Dim lngi As Long  
Dim lngno As Long  
Dim strinput As String  
strinput = InputBox(«Enter Your Number :», «Input Data», 0)  
lngno = Val(strinput)  
Do  
    lngi = lngno Mod 10  
    sum = sum + lngi  
    lngno = lngno \ 10  
    If lngno < 0 Then Exit Do  
Loop While (True)  
End Sub
```

در این رویداد نیز مانند راه حل قبلی حلقه Do...Loop While به کار گرفته شده است. با این تفاوت که به جای شرط  $Ingno > 0$  از مقدار True استفاده شده است بنابراین حلقه می‌تواند بدون محدودیت بارها اجرا شود اما با توجه به حالت پروژه باید اجرای حلقه با توجه به تعداد ارقام عدد وارد شده تنظیم شود. به این منظور از یک دستور If دیگر در انتهای حلقه استفاده شده است که شرط  $Ingno < 0$  را بررسی می‌کند در صورتی که نتیجه این بررسی درست باشد (یعنی دیگر رقمی در عدد باقی نمانده باشد) با دستور Exit Do از اجرای مجدد حلقه جلوگیری به عمل می‌آورد.

۲ - تغییرات را ذخیره کرده و پروژه را اجرا و آزمایش کنید.

۳ - به اجرای پروژه خاتمه داده و به پنجره ویژوال بیسیک بازگردید.



شکل ۹-۶

**مثال ۹:** پروژه‌ای طراحی کنید که بزرگ‌ترین

عدد را بین n عدد دلخواه جستجو کرده و نمایش

دهد. برای این کار عملیات زیر را انجام دهید:

۱- یک پروژه جدید مطابق شکل ۹-۶ ایجاد کنید.

۲ - مطابق شکل ۹-۶ ابتدا کاربر تعداد اعداد خود را در کادر متن وارد می‌کند؛ سپس با کلیک روی دکمه Input و با استفاده از کادرهای ورود داده اعداد خود را وارد می‌کند و با پایان ورود اعداد، مقدار ماکزیمم روی فرم نمایش داده می‌شود. به علاوه در صورتی که کاربر کاراکتر Q را در کادر ورود داده تایپ کند، برنامه از دریافت اعداد بعدی خودداری می‌نماید. بنابراین رویداد Click دکمه Input را به این صورت تنظیم کنید:

```
Private Sub cmdinput_Click()
```

```
Dim inti As Integer, strno As String
```

```
Dim sngmax As Single
```

```
For inti = 1 To Val(txtno.Text)
```

```
strno = InputBox("Enter Your Number:", "InputBox", 0)
```

```
If strno = "Q" or strno = "q" Then Exit For
```

```
If inti = 1 Then
```

```
sngmax = Val(strno)
```

```
Else
```

```
If sngmax < Val (strno)Then sngmax =Val(strno)
```

```
End If
```

```
Next inti
```

```
MsgBox "Maximum Is :"+Str(sngmax)
```

```
End Sub
```

در این رویداد از یک حلقه For...Next برای دریافت اعداد استفاده خواهد شد. مقدار خاتمه حلقه با توجه به مقداری که کاربر در کادر متن مربوطه وارد می‌کند، تنظیم می‌شود. بخش عمده‌ای از دستورات موجود در حلقه برای محاسبه بزرگ‌ترین عدد به کار می‌روند، اما پس از آن که این رویداد اجرا شد، ابتدا با استفاده از یک کادر ورود داده مقدار اولین عدد را از کاربر دریافت می‌کند و بلافاصله مقدار تایپ شده در کادر ورود داده با یک دستور If بررسی می‌شود و اگر کاراکتر Q یا q تایپ شده باشد نتیجه شرط موجود درست بوده و در نتیجه دستور Exit For اجرا می‌شود که سبب خروج از حلقه و خاتمه دریافت اعداد خواهد شد؛ اما در غیر این صورت دستور If دوم بررسی می‌شود که در این صورت اگر اولین عدد داده شده باشد (یعنی شمارنده حلقه برابر با یک باشد) اولین عدد به عنوان بزرگ‌ترین عدد در نظر گرفته می‌شود و در متغیر sngmax ذخیره می‌شود، اما برای اعداد بعدی ابتدا مقدار تایپ شده در کادر ورود داده با مقدار ذخیره شده در متغیر sngmax مقایسه می‌شود و در صورتی که از آن بزرگ‌تر باشد، مقدار جدید به عنوان بزرگ‌ترین مقدار، در متغیر sngmax ذخیره خواهد شد. به این ترتیب بزرگ‌ترین مقدار، محاسبه و در یک کادر پیام نمایش داده می‌شود.

۳ - پروژه و فرم را با نام maximum ذخیره کنید، سپس آن را اجرا نمایید.

۴ - مقدار ۱۰ را در کادر متن تایپ کرده و روی دکمه Input کلیک کنید.

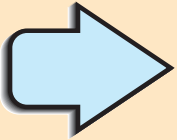
۵ - اعداد ۷، ۲ و ۱۸ را به ترتیب وارد کنید، سپس کاراکتر Q را در کادر ورود داده چهارم تایپ کنید و روی دکمه OK کلیک کنید. همان‌طور که مشاهده می‌کنید از ورود داده‌های بعدی جلوگیری به عمل می‌آید و بزرگ‌ترین عدد بین سه عدد وارد شده (۱۸) نمایش داده می‌شود.

۶ - پروژه را برای مقادیر دیگری آزمایش کنید، سپس اجرای پروژه را پایان داده و به پنجره ویژوال بیسیک بازگردید.

تمرین:



پروژه قبل را به گونه‌ای تنظیم کنید که توانایی محاسبه کوچک‌ترین عدد را نیز داشته باشد.



## Learn in English

### For...Next Statement

.Repeats a group of statements a specified number of times

#### Syntax

**For** counter = start **To** end [*step*]

[*statements*]

**Next** [*counter*]

The **For...Next** statement syntax has these parts:

Part	Description
counter	Required. Numeric variable used as a loop counter. The variable can't be a Boolean or an array element.
start	Required. Initial value of counter.
end	Required. Final value of counter.
step	Optional. Amount counter is changed each time through the loop. If not specified, step defaults to one.
statements	Optional. One or more statements between <b>For</b> and <b>Next</b> that are executed the specified number of times.

## واژه‌نامه

Compute	محاسبه کردن
Counter	شمارنده
Even	زوج
Exit	خروج
Final	نهایی، آخری
Information	اطلاعات
Initial	ابتدایی، اولین، اصلی
Odd	فرد
Optional	اختیاری
Payment	مبلغ، بهای پرداختی
Price	بها، قیمت
Rank	رتبه
Required	لازم داشتن
Specify	تعیین کردن
Statement	دستور
Step	مرحله
Transport	حمل کردن

## خلاصه مطالب

- از دستور Select Case برای ارزیابی چند مقایسه به صورت یکجا استفاده می‌شود.
- از کنترل دکمه انتخاب (Option Button) برای ایجاد گزینه‌های متعدد، برای انتخاب کاربر استفاده می‌شود.
- برای اجرای دستورات با دفعات تکرار معین از دستور For ... Next استفاده می‌شود.
- حلقه‌های Do...Loop While و Do While ... Loop، While ... Wend، حلقه‌های Do...Loop Until و Do Until...Loop را تا زمانی که نتیجه شرط یا شرط‌های ذکر شده در آن‌ها درست باشند، اجرا می‌کنند.
- حلقه‌های Do...Loop Until و Do Until...Loop را تا زمانی که نتیجه شرط یا شرط‌های ذکر شده در آن‌ها نادرست باشند، اجرا می‌کنند.

- دستور Print می‌تواند انواع پیام‌ها، عبارات رشته‌ای، مقدار متغیرها و خصوصیات فرم‌ها و کنترل‌ها را روی فرم نمایش دهد.
- از دستور Exit Do برای خروج از حلقه Do While...Loop، Do...Loop While، Do Until...Loop و Do...Loop Until استفاده می‌شود.
- از دستور Exit For برای خروج از حلقه For ... Next استفاده می‌شود.
- از کنترل قاب (Frame) برای دسته‌بندی کنترل‌ها استفاده می‌شود.
- برای اضافه کردن یک پروژه جدید از نوع Standard EXE می‌توان از گزینه New Project در منوی File ویژوال بیسیک یا از دکمه Add Standard EXE Project در نوار ابزار استاندارد استفاده کرد.
- به منظور دسترسی به خاصیت‌ها و متدهای کنترل یک فرم در فرم دیگر می‌توان به صورت زیر عمل نمود:

نام متد یا خاصیت. نام کنترل. نام فرم

## آزمون نظری

۱ - کدام حلقه شبیه به حلقه Do While ... Loop است؟

الف - For...Next      ب - While ...Wend      ج - Select Case      د - While...Loop

۲ - پس از اجرای برنامه زیر مقدار متغیر Sum چقدر است؟

Sum= ۰

I=5

Do While (I>3)

For k=1 to 2

Sum = Sum+k\*I

Next k

I=I-1

Loop

د- ۳۰

ج- ۲۹

ب- ۲۸

الف- ۲۷

۳ - حلقه زیر چند بار تکرار می شود؟

For i=7 To -2 Step -3

.....

Next i

د- ۵

ج- ۴

ب- ۳

الف- ۲

۴ - حلقه زیر چند بار تکرار می شود؟

While (True)

.....

Wend

الف- ۳۰ بار      ب- ۴۰ بار      ج- بی نهایت      د- تکرار نمی شود.

۵ - خروجی برنامه پس از اجرای حلقه زیر چیست؟

i=6

While (i>4)

j=6



i=i - 1

Do While (j>3)

Print i,j

j=j - 1

Loop

Wend

الف-	۵	۶	ب-	۵	۵	ج-	۴	۵	د-	۴	۶
	۵	۵		۵	۶		۴	۶		۴	۵
	۵	۴		۵	۴		۴	۴		۴	۴
	۴	۶		۴	۵		۵	۵		۵	۶
	۴	۵		۴	۶		۵	۶		۵	۵
	۴	۴		۴	۴		۵	۴		۵	۴

۶- خروجی برنامه پس از اجرای حلقه زیر چیست؟

For i=10 To 1 Step 2

Print "i" ;i

Next i

- الف- اعداد زوج بین ۱ و ۱۰      ب- اعداد فرد بین ۱ و ۱۰
- ج- اعداد ۱۰ تا ۱      د- هیچ‌گونه خروجی ندارد.
- ۷- در صورتی که مقدار متغیر intx برابر ۲ باشد، خروجی دستورات زیر چیست؟

Select Case intx

Case Is > 10 : Print intx\*10

Case Is <= 4 : Print intx\*4

Case Is = 7 : Print intx\*2

End Select

- الف- ۱۰      ب- ۸      ج- ۴      د- ۲
- دستور Exit Do سبب خروج از کدام حلقه می‌شود؟
- الف- For ...Next      ب- While ... Wend
- ج- Do While ... Loop      د- گزینه‌های ب و ج صحیح هستند.

۹ - کدام حلقه مشابه حلقه زیر است؟

```
For i=4 to 1 Step -1
    ⋮
Next i
```

الف -

```
i=4
While (i>=1)
    ⋮
i=i+ 1
Wend
i=4
```

ب (

```
Do While (i>=1)
    ⋮
i = i- 1
Loop
```

ج -

```
i =4
While (i<=1)
    ⋮
i=i - 1
Wend
```

د -

```
i = 4
Do While (i >=1)
    ⋮
i = i + 1
Loop
```

۱۰ - کدام گزینه در رابطه با دستورات زیر صحیح است؟

```
For i=1 To 15
    If i Mod 3=0 Then Print i
Next i
```

الف- اعداد ۱ تا ۱۵ را نمایش می‌دهد.

ب- مضارب عدد ۳ از ۱ تا ۱۵ را نمایش می‌دهد.

ج- اعداد ۱ تا ۱۵ را که بر ۳ بخش پذیر هستند نمایش می‌دهد.

د- باقیمانده تقسیم اعداد ۱ تا ۱۵ را بر ۳ نمایش می‌دهد.

۱۱- برای دسته‌بندی کنترل‌ها از کنترل ..... استفاده می‌شود.

الف- CheckBox      ب- Frame      ج- GroupBox      د- Label

۱۲- با استفاده از کدام خصوصیت می‌توان کنترل OptionButton را در حالت انتخاب قرار داد؟

الف- Check      ب- Index      ج- Value      د- Tab Index

۱۳- حلقه زیر چند بار تکرار می‌شود؟

k=2

Do

⋮

k=k\*2

Loop Until (k>10)

د- ۲

ج- ۳

ب- ۴

الف- ۵

۱۴- In the For Statement, the step argument can be.....number.

a- positive

b- negative

c- integer

d- positive or negative

۱۵- تفاوت بین حلقه‌های While با حلقه‌های Until چیست؟

۱۶- کاربرد دستور Exit Do و Exit For را بیان کنید.

۱۷- نحوه عملکرد دستور For را همراه با بخش Step توضیح دهید.

۱۸- تفاوت دستور Select Case را با IF بیان کنید.

۱۹- کنترل دکمه انتخاب (Option Button) و قاب (Frame) را توضیح داده و کاربرد

هر یک از آن‌ها را بیان کنید.

۲۰- نحوه اضافه کردن یک پروژ به جدید به پروژه موجود را توضیح دهید.

## آزمون عملی

- ۱ - پروژه‌ای طراحی کنید که مجموع اعداد سه رقمی را محاسبه کرده و نمایش دهد.
- ۲ - پروژه‌ای طراحی کنید که عدد طبیعی و دلخواهی را از کاربر دریافت کند و مقسوم علیه‌های زوج آن را محاسبه کرده و نمایش دهد.
- ۳ - پروژه‌ای طراحی کنید که یک جدول ضرب  $10 \times 10$  را چاپ کند.
- ۴ - پروژه‌ای طراحی کنید که دو عدد طبیعی را دریافت کند و کوچک‌ترین مضرب مشترک آن‌ها را نمایش دهد.
- ۵ - خروجی حاصل از اجرای حلقه‌های زیر چه خواهد بود؟

الف -

ب -

```
For I= 1 to 4
    For j=1 to 3
        Print "*";
    Next j
    Print
Next I

J=10
Do While (True)
    If j>5 Then Exit Do
    For i=1 to j
        Print i;
    Next i
    Print
    j=j+ 1
Loop
```

- ۶ - پروژه‌ای طراحی کنید که دو عدد را دریافت کرده و بزرگ‌ترین مقسوم‌علیه آن‌ها را نمایش دهد.