

### توانایی ایجاد و استفاده از انواع رویه‌ها در ویژوال بیسیک

#### هدف‌های رفتاری

- پس از مطالعه این واحد کار از فراگیر انتظار می‌رود که:
- ۱- انواع رویه‌ها را در زبان برنامه‌نویسی ویژوال بیسیک توضیح دهد.
  - ۲- رویه‌های رویداد را تعریف کند.
  - ۳- رویه‌های فرعی و رویه‌های تابعی را تعریف کند و تفاوت بین آن‌ها را توضیح دهد.
  - ۴- بتواند با استفاده از دستورات Exit Sub و Exit Function از رویه‌های فرعی و تابعی خارج شود.
  - ۵- انواع روش‌های ارسال متغیرها به رویه‌ها و تفاوت بین آن‌ها را بیان کند.
  - ۶- تفاوت رویه‌های عمومی و محلی را بیان کند و بتواند انواع رویه‌ها را به صورت عمومی و محلی تعریف کند.

## کلیات

معمولاً در طراحی پروژه‌های واقعی نیاز به دستورات عمل‌های فراوانی خواهید داشت و گاهی پیش می‌آید که مجموعه‌ای از دستورات عمل‌ها در بخش‌های مختلفی از پروژه یا حتی سایر پروژه‌ها به صورت تکراری استفاده می‌شوند. در این صورت می‌توان به جای تکرار، این مجموعه دستورات را یکبار در بخشی از پروژه نوشته و بارها مورد استفاده قرار داد. به این منظور در پروژه‌های بزرگ، برنامه به بخش‌های کوچک‌تر تقسیم می‌شود و در نتیجه زمان لازم برای خطایابی و خطازدایی و نوشتن دستورات عمل‌ها به شکل قابل توجهی کاهش می‌یابد. به علاوه ایجاد تغییرات در برنامه برای ارتقا و توسعه آن در آینده راحت‌تر و با خطای کمتر توأم خواهد بود. به این روش، برنامه‌نویسی ساخت یافته (Structural) می‌گویند. همان‌طور که می‌دانید این ویژگی در زبان ویژوال بیسیک نیز وجود دارد و با استفاده از انواع رویه‌ها (Procedure) می‌توانید به روش ساخت یافته نیز برنامه‌نویسی کنید. در ویژوال بیسیک، چهار نوع رویه وجود دارد که عبارتند از: رویه‌های فرعی (Sub Procedure)، رویه‌های تابعی (Function Procedure)، رویه‌های رویداد (Event Procedure) و رویه‌های آماده ویژوال بیسیک (Visual Basic Procedure). در این جا به توضیح سه گروه اول خواهیم پرداخت و در واحدکار بعد نحوه استفاده از رویه‌های آماده و ویژوال بیسیک را فراخواهید گرفت.

### ۷-۱ رویه‌های فرعی (Sub Procedure)

گاهی اوقات لازم است تا برنامه‌نویس دستورات مورد نظر خود را به صورت یک بخش مستقل از سایر دستورات در پروژه قرار دهد و در هر زمان که لازم باشد در هر جایی از پروژه (مانند سایر رویه‌ها) از آن استفاده کند. در این صورت استفاده از رویه‌های فرعی می‌تواند یکی از انتخاب‌های مناسب برای وی باشد.

به عنوان مثال رویه‌های رویداد از نوع رویه‌های فرعی هستند که بارها از آن استفاده کرده‌اید. رویه‌های فرعی دیگر را نیز می‌توانید در ماژول فرم، تعریف و استفاده کنید. اما همان‌طور که می‌دانید رویه‌های رویداد به‌طور خودکار در زمان لازم اجرا می‌شوند، اما برای اجرای رویه‌های فرعی که برنامه‌نویس تعریف می‌کند باید از دستورات خاصی استفاده شود تا بتواند رویه را در زمان مناسب اجرا کند.

در این جا لازم است تا نحوه تعریف یک رویه را به طور کلی بیاموزید، برای تعریف یک رویه فرعی به این شکل عمل کنید:

(... , نوع آرگومان As نام آرگومان اول) نام رویه فرعی Sub Private

⋮

### دستور(ات)

End Sub

تعریف یک رویه فرعی با کلمه کلیدی Private آغاز می‌شود. استفاده از این عبارت در تعریف یک رویه فرعی سبب می‌شود تا رویه مربوطه به عنوان یک رویه محلی تعریف شود؛ بنابراین فقط در سایر رویه‌های همان ماژول فرمی که در آن تعریف شده، قابل شناسایی و فراخوانی است. در ادامه تعریف رویه، کلمه کلیدی Sub و به دنبال آن نام رویه فرعی قرار می‌گیرد. برای نام‌گذاری رویه‌ها می‌توانید مانند متغیرها عمل کنید.

پس از آن‌که نام رویه را تعیین کردید باید نام و نوع آرگومان‌های رویه را مشخص کنید. در واقع آرگومان‌ها متغیرهایی هستند که به برنامه‌نویس اجازه می‌دهند تا از این طریق مقادیر موردنظر را به رویه فرعی ارسال کنند. قوانین تعریف و نام‌گذاری آرگومان‌ها دقیقاً مشابه متغیرهاست.

البته استفاده از آرگومان‌ها اجباری نبوده و یک رویه فرعی می‌تواند یک، دو یا چند آرگومان داشته باشد یا این‌که اصلاً هیچ آرگومانی نداشته باشد. اما در صورت استفاده از آن‌ها باید در هنگام فراخوانی رویه، دقت لازم را برای انتقال مقادیر مورد نظر به آرگومان‌ها به عمل آورید.

پس از خط اول تعریف رویه، دستورات مورد نظر که در رویه اجرا می‌شوند، قرار داده شده و خاتمه رویه نیز با عبارت End Sub تعیین می‌شود. بدین صورت رویه شما در ماژول فرم تعریف می‌شود.



مثال ۱: پروژه‌ای طراحی کنید که طول و عرض یک مستطیل را از کاربر دریافت کند و با استفاده از یک رویه فرعی، محیط و مساحت آن را محاسبه کرده و نمایش دهد. برای انجام این کار مراحل بعد را انجام دهید:

۱ - برنامه ویژوال بیسیک را اجرا کنید و یک پروژه از نوع Standard EXE به همراه یک فرم مطابق شکل ۱-۷ ایجاد کنید.



شکل ۱-۷

می بینید که دو کادر متن برای دریافت طول و عرض مستطیل در نظر گرفته شده است و پس از ورود داده‌ها با کلیک روی دکمه Calculate محیط و مساحت مستطیل محاسبه خواهد شد. ۲- برای محاسبه محیط و مساحت مستطیل با استفاده از یک رویه فرعی، ابتدا باید آن را تعریف کنید بنابراین به پنجره ماژول فرم رفته و تعریف رویه را به این صورت تایپ کنید، سپس کلید Enter را بفشارید.

Private Sub rectangle()

عبارت End Sub در پایین دستوری که تایپ کرده‌اید به طور خودکار ظاهر می‌شود. در واقع این دو دستور محدوده رویه فرعی را تعیین می‌کنند و دستورات مربوط به رویه در بین این دو خط قرار می‌گیرند. رویه‌های فرعی نیز مانند رویه‌های رویداد دارای یک نام هستند. نام‌گذاری رویه‌های فرعی مانند نام‌گذاری متغیرهاست و از همان قوانین پیروی می‌کنند. ۳- در این مرحله باید دستورات مربوط به محاسبه محیط و مساحت مستطیل را در رویه فرعی rectangle تایپ کنید؛ بنابراین رویه را به صورت بعد تنظیم کنید:

Private Sub rectangle()

Dim sngl As Single, sngw As Single

Dim sngarea As Single, sngperi As Single

sngl = Val(txtl.Text)

sngw = Val(txtw.Text)

sngarea = sngl \* sngw

sngperi = 2 \* (sngl + sngw)

lblarea.Caption = "Area Is " + Str(sngarea)

lblperi.Caption = "Perimeter Is " + Str(sngperi)

End Sub

در این رویه ابتدا داده‌های ورودی در کادرهای متن به عدد تبدیل می‌شوند و در متغیرهای sngl و sngw ذخیره می‌شوند، سپس عملیات محاسبه مساحت و محیط انجام و در نهایت با استفاده از کنترل‌های برچسب نمایش داده می‌شوند.  
پس از تعریف یک رویه فرعی لازم است نحوه استفاده از دستورات موجود در رویه یا به عبارت بهتر اجرای رویه فرعی را بدانید به این عمل فراخوانی رویه نیز می‌گویند. برای انجام این کار می‌توانید با توجه به نیاز خود یکی از روش‌های زیر را به کار گیرید:

(... , مقدار مربوط به آرگومان اول) نام رویه Call

... , مقدار مربوط به آرگومان اول نام رویه

فراخوانی می‌تواند با دستور Call یا بدون آن انجام شود سپس نام رویه و در صورت استفاده از آرگومان‌ها، مقادیر یا متغیرهای متناسب با هر یک را در داخل پرانتز تایپ کرده و با کاراکتر کاما (,) از یکدیگر جدا کنید.

۴ - برای آن‌که بتوانید دستورات موجود در رویه rectangle را اجرا کنید باید در رویداد Click دکمه Calculate، این دستور را تایپ کنید: Call rectangle

دستور Call در هنگام اجرا، رویه rectangle را فراخوانی می‌کند، به عبارت دیگر اجرای برنامه به رویه rectangle منتقل می‌شود و پس از رسیدن به عبارت End Sub در این رویه اجرای برنامه مجدداً به محل فراخوانی باز می‌گردد و دستور بعد از Call اجرا می‌شود.

۵ - پروژه و فرم را با نام rectangle ذخیره نمایید.

۶ - پروژه را اجرا نمایید. اعداد ۴ و ۶ را به عنوان طول و عرض مستطیل وارد کنید و روی دکمه Calculate کلیک کنید و نتیجه محاسبه را بررسی نمایید.

۷ - پروژه را برای مقادیر دیگر نیز بررسی کنید، سپس به پنجره ویژوال بیسیک بازگردید.

تمرین:



پروژه‌ای طراحی کنید که با استفاده از یک رویه فرعی، محیط و مساحت هر دایره به شعاع دلخواه R را محاسبه کند.



**مثال ۲:** پروژه‌ای طراحی کنید که عدد دلخواهی را از کاربر دریافت کرده و زوج یا فرد بودن آن را با استفاده از یک رویه فرعی مشخص کند. برای انجام این کار مراحل بعد را به ترتیب انجام دهید:

- ۱ - یک پروژه از نوع Standard EXE و یک فرم مطابق شکل ۲-۷ ایجاد کنید.
- ۲ - روی منوی Tools در پنجره ویژوال بیسیک کلیک کنید و سپس گزینه Add Procedure را برگزینید تا کادر محاوره Add Procedure نمایش داده شود.
- ۳ - در کادر متن Name نام رویه را تایپ کنید. برای رویه این مثال، از عبارت oddoreven استفاده کنید (شکل ۳-۷).
- ۴ - در بخش Type گزینه Sub را برای ایجاد رویه فرعی انتخاب کنید.
- ۵ - در بخش سوم از تنظیمات کادر محاوره Add Procedure برای میدان دید و شناسایی رویه (Scope) دکمه انتخاب Private را برگزینید.



شکل ۲-۷



شکل ۳-۷

۶ - در پایان روی دکمه OK در کادر محاوره Add Procedure کلیک کنید تا رویه فرعی در ماژول فرم اضافه شود.

۷ - اکنون به پنجره ماژول فرم بروید و رویه oddoreven را به صورت زیر تنظیم کنید:

Private Sub oddoreven(sngno As Single)

If (sngno Mod 2) = 0 Then

    MsgBox "your number is even .» , , «Message»

Else

    MsgBox "your number is odd .» , , «Message»

End If

End Sub

رویه فرعی oddoreven دارای یک آرگومان با نام sngno می‌باشد که از نوع Single تعریف شده است، از این آرگومان برای انتقال عددی که می‌خواهید زوج یا فرد بودن آن را داخل رویه معین کنید، استفاده می‌شود. داخل رویه نیز با استفاده از عملگر Mod و یک If باقی‌مانده تقسیم عدد مورد نظر که در آرگومان sngno ذخیره شده است بر عدد دو بررسی می‌شود. اگر این مقدار برابر صفر باشد کادر پیام، عدد را به عنوان یک عدد زوج و در غیر این صورت عدد را به عنوان یک عدد فرد معرفی خواهد کرد.

۸- در این مرحله باید رویداد Click دکمه فرمان Compute به شکلی تنظیم شود تا در صورت کلیک روی دکمه Compute رویه فرعی oddoreven فراخوانی شده و زوج یا فرد بودن عدد مربوطه را معین کند، بنابراین رویه رویداد Click دکمه Compute را به این صورت تنظیم کنید:

```
Private Sub cmdcom_Click()
```


```
Dim snginput As Single
```

```
snginput = Val(txtno.Text)
```

```
Call oddoreven(snginput)
```

```
End Sub
```

در این رویداد پس از تعریف متغیر snginput، با استفاده از تابع Val عبارت تایپ شده در کادر متن به مقدار عددی تبدیل و در متغیر snginput ذخیره می‌شود. در مرحله بعد باید رویه فرعی oddoreven را فراخوانی کرد. برای این کار با استفاده از دستور Call رویه فراخوانی می‌شود و برای آن که عدد مورد نظر کاربر که در متغیر snginput ذخیره شده است در اختیار رویه قرار گیرد، نام متغیر snginput پس از نام رویه و در داخل پرانتز ذکر می‌شود، در این صورت در هنگام اجرای دستور Call بین متغیر snginput و متغیر sngno ارتباط برقرار می‌شود و در واقع هنگام اجرای رویه، مراجعه به متغیر sngno به منزله مراجعه، خواندن و ذخیره‌سازی اطلاعات در متغیر snginput است.

 **نکته** در صورتی که یک رویه دارای آرگومان باشد، اگر بعد از تعریف رویه، نام رویه را در ماژول فرم تایپ کنید؛ ویژوال بیسیک یک راهنمایی به صورت ToolTip را که شامل نام رویه و نام و نوع آرگومان‌های آن می‌باشد، نمایش می‌دهد (شکل ۴-۷).



شکل ۴-۷

- ۹ - پروژه و فرم را با نام oddeven ذخیره کرده، سپس آن را اجرا کنید.
- ۱۰ - در کادر متن فرم برنامه، عدد ۴ را تایپ کنید و روی دکمه Compute کلیک کنید. در این صورت رویه فراخوانی می‌شود و عدد ۴ که در متغیر snginput ذخیره شده است، در آرگومان sngno کپی می‌شود و سپس نوع عدد تعیین شده و در کادر پیام، پیغام «عدد زوج است» نمایش داده می‌شود.
- ۱۱ - پروژه را برای چند عدد دیگر بررسی کنید و به محیط برنامه‌نویسی ویژوال بیسیک باز گردید.

## ۲-۷ رویه‌های تابعی (Function Procedure)

نوع دیگری از رویه‌ها، رویه‌های تابعی هستند. رویه‌های تابعی شبیه به رویه‌های فرعی هستند با این تفاوت که از آن‌ها در مواردی که لازم باشد، استفاده می‌شود و پس از انجام مجموعه‌ای از محاسبات، مقداری به محل فراخوانی رویه، بازگشت داده می‌شود. قبلاً از بعضی رویه‌های تابعی آماده در ویژوال بیسیک استفاده کرده‌اید، برای مثال می‌توان به رویه تابعی Val، Str و نظایر آن‌ها اشاره کرد. تابع Val با دریافت یک رشته از کاراکترهای رقمی آن را به مقدار عددی تبدیل می‌کند و تابع Str با دریافت یک مقدار عددی آن را به رشته تبدیل می‌نماید. در ویژوال بیسیک امکان تعریف رویه‌های تابعی مورد نیاز توسط برنامه‌نویس نیز وجود دارد.



تعریف و فراخوانی رویه‌های تابعی نیز شبیه به رویه‌های فرعی است. برای تعریف رویه‌های تابعی می‌توانید رویه را در مازول فرم تایپ کنید یا از کادر محاوره Add Proc - dure استفاده کنید. شکل کلی تعریف رویه‌های تابعی به این صورت است:

**نوع مقدار بازگشتی As ( . . . , نوع آرگومان As نام آرگومان) نام تابع Private Function**

.....  
دستور(ات)  
مقدار بازگشتی = نام تابع  
.....

**End Function**

تعریف یک رویه تابعی مانند رویه فرعی بوده و با کلمه کلیدی Private آغاز شده و باعث می‌شود رویه تابعی به صورت محلی تعریف شود. در ادامه تعریف رویه تابعی کلمه کلیدی Function و به دنبال آن نام رویه تابعی قرار می‌گیرد. قوانین نام‌گذاری رویه‌های تابعی مانند رویه‌های فرعی و متغیرهاست.

پس از تعیین نام رویه تابعی باید نام و نوع آرگومان‌های رویه را مشخص کنید تا با استفاده از آن‌ها مقادیر مورد نیاز به رویه ارسال شوند. نحوه تعریف و استفاده از آرگومان‌ها در رویه‌های تابعی مانند رویه‌های فرعی می‌باشد و استفاده از آن‌ها و تعداد و نوع آرگومان‌ها اختیاری است. پس از تعریف رویه تابعی باید دستورات مورد نظر خود را در رویه قرار داده و خاتمه رویه را با عبارت End Function تعیین کنید.

اما نکته مهمی که در رابطه با رویه‌های تابعی وجود دارد و در رویه‌های فرعی موجود نیست، مربوط به نحوه بازگشت یک مقدار به محل فراخوانی می‌شود. همان‌طور که در شکل کلی تعریف رویه تابعی می‌بینید، باید مقداری را که می‌خواهید بازگشت دهید به نام تابع انتساب دهید. با رسیدن به این دستور و اجرای آن مقدار تعیین شده به محل فراخوانی بازگشت داده می‌شود. علاوه بر این باید نوع مقدار بازگشتی در انتهای خط اول تعریف رویه و پس از معرفی آرگومان‌ها معین شود. برای فراخوانی یک رویه تابعی می‌توانید به این شکل عمل کنید:

... , مقدار مربوط به آرگومان اول) نام رویه تابعی = نام یک متغیر متناسب با نوع داده بازگشتی

هم‌چنین می‌توانید رویه تابعی را بدون آن‌که مقداری را به عنوان نتیجه محاسبات بازگشت دهد، تعریف کنید. در این صورت فراخوانی رویه تابعی می‌تواند به صورت فراخوانی یک رویه فرعی باشد.

( ... , مقدار مربوط به آرگومان اول) نام رویه Call

... , مقدار مربوط به آرگومان اول نام رویه

**نکته:** مقدار بازگشتی یک تابع می تواند به عنوان مقدار ارسالی برای تابع دیگری استفاده شود. نوع مقدار بازگشتی در تعریف تابع با توجه به نوع داده ای که به محل فراخوانی بازگشت می دهد تعیین می شود. اگر نوع مقدار بازگشتی در تعریف تابع تعیین نشود به طور پیش فرض از نوع داده Integer استفاده می شود.

**مثال ۳:** پروژه ای طراحی کنید که با استفاده از یک تابع، باقیمانده تقسیم دو عدد دلخواه را محاسبه کرده و نمایش دهد. برای این کار مراحل بعد را انجام دهید:



شکل ۵-۷

۱ - برنامه ویژوال بیسیک را اجرا کنید و یک پروژه از نوع Standard EXE به همراه یک فرم مطابق شکل ۵-۷ ایجاد کنید. دو کادر متن روی فرم قرار دارند که اعداد موردنظر در آنها وارد خواهند شد و کاربر با کلیک روی دکمه Remainder باقی مانده تقسیم دو عدد را مشاهده خواهد کرد.

۲ - به پنجره ماژول فرم بروید و تابع remainder را به این صورت برای محاسبه باقیمانده تقسیم دو عدد تعریف کنید:

```
Private Function remainder (intno1 As Integer, intno2 As _ Integer)As Integer
```

```
Dim intresult As Integer
```

```
intresult = intno1 Mod intno2
```

```
remainder = intresult
```

```
End Function
```

تابع remainder دارای دو آرگومان  $intno1$  و  $intno2$  است که برای دریافت مقدار مقسوم و مقسوم علیه به کار می‌روند؛ سپس در بخش دستورات این رویه ابتدا یک متغیر محلی دیگر برای ذخیره‌سازی باقی‌مانده تقسیم، تعریف می‌شود و به دنبال آن با استفاده از عملگر Mod باقی‌مانده تقسیم دو عدد در این متغیر (intresult) ذخیره می‌شود. اما برای آن‌که نتیجه محاسبه به محل فراخوانی بازگشت داده شود از دستور  $remainder = intresult$  استفاده می‌شود و نتیجه محاسبه که در متغیر intresult ذخیره شده است به نام تابع نسبت داده می‌شود. به علاوه چون متغیر intresult از نوع عدد صحیح (Integer) معرفی شده است، بنابراین در انتهای خط اول تعریف تابع نوع مقدار بازگشتی نیز از نوع Integer معرفی شده است.

۳- در این مرحله رویداد Click دکمه فرمان Remainder را به صورت زیر تنظیم کنید:

```
Private Sub cmdrem_Click()
```

```
Dim intm As Integer, intn As Integer
```

```
Dim intmod As Integer
```

```
intm = Val(txtno1.Text)
```

```
intn = Val(txtno2.Text)
```

```
intmod = remainder(intm, intn)
```

```
lblres.Caption = "Answer Is " + Str(intmod)
```

```
End Sub
```

با اجرای این رویه ابتدا مقادیری که در کادرهای متن تایپ شده‌اند، بعد از تبدیل در متغیرهای intm و intn ذخیره می‌شوند؛ سپس تابع remainder با متغیرهای intm و intn فراخوانی می‌شود و در نتیجه اجرای برنامه به بخش تعریف رویه تابعی remainder منتقل شده و بین متغیرهای ارسالی intm و intn با آرگومان‌های  $intno1$  و  $intno2$  ارتباط برقرار می‌کند، سپس اجرای دستورات موجود در رویه تابعی آغاز می‌شود و با انجام محاسبات و رسیدن به دستور  $remainder = intresult$  و انجام آن، اجرای برنامه به محل فراخوانی یعنی رویداد Click دکمه remainder (intmod = remainder(intm, intn)) بازمی‌گردد و مقدار موجود در متغیر intresult در متغیر intmod ذخیره می‌شود و در پایان این مقدار در کنترل برچسب نمایش داده شده و رویداد خاتمه می‌یابد. نوع متغیر intmod با نوع مقدار بازگشتی در تعریف تابع و همین‌طور نوع متغیر intresult هماهنگی لازم را دارد.

۴ - پروژه و فرم را با نام remainder ذخیره کنید، سپس آن را اجرا نمایید.

۵ - اعداد ۱۲ و ۵ را به ترتیب در کادرهای متن وارد کنید، سپس روی دکمه Remainder کلیک کرده و نتیجه محاسبه را مشاهده نمایید.

۶ - پروژه را برای مقادیر دیگری نیز آزمایش کنید، سپس به پنجره ویژوال بیسیک بازگردید.



تمرین:

پروژه فوق را به گونه‌ای تنظیم کنید که اگر عدد اول از عدد دوم کوچک‌تر باشد با استفاده از یک رویه فرعی، خطای کاربر به وی گزارش شود و محاسبات نیز انجام نشود. پروژه‌ای طراحی کنید که یک عدد دلخواه را از کاربر دریافت کند و با استفاده از یک رویه تابعی قدر مطلق آن را محاسبه کرده و نمایش دهد. پروژه‌ای طراحی کنید که با استفاده از یک تابع، قرینه هر عدد دلخواه را نمایش دهد. رویه‌ای تابعی بنویسید که هر عدد دلخواه  $x$  را دریافت کند و مقدار معکوس آن را  $\frac{1}{x}$  محاسبه کرده و بازگشت دهد.

## ۳-۷ روش‌های ارسال مقادیر به رویه‌های فرعی و تابعی

تاکنون نحوه استفاده از رویه‌های فرعی و تابعی و چگونگی انتقال مقادیر مورد نیاز به رویه‌ها را فرا گرفتید. در واقع دو روش مختلف برای ارسال مقادیر مورد نیاز به رویه‌ها وجود دارد که هر یک ویژگی‌هایی را به همراه دارند. اولین روش فراخوانی متغیرها با مرجع (Call By Reference) و دومین روش فراخوانی متغیرها با مقدار (Call By Value) است.

### ۱-۳-۷ روش فراخوانی با مرجع

در این روش، ارتباط مستقیمی بین متغیر ارسالی و آرگومان متناظر آن در رویه مربوطه برقرار می‌شود به گونه‌ای که متغیر و آرگومان متناظر با آن به محل مشترکی در حافظه اشاره می‌کنند. بنابراین هرگونه تغییر در محتویات آرگومان‌ها در رویه، سبب تغییر در متغیرهای ارسالی خواهد شد. در واقع شما تاکنون با این روش، مقادیر خود را به رویه‌ها

ارسال کرده‌اید.

در صورتی که بخواهید متغیرها را با روش ارسال با مرجع به یک رویه ارسال نمایید، کلمه کلیدی ByRef را در ابتدای نام آرگومان‌ها در تعریف رویه ذکر کنید. البته در صورت عدم استفاده از کلمه کلیدی ByRef نیز نحوه ارسال متغیرها به طور پیش فرض از نوع فراخوانی با مرجع می‌باشد.

فرض کنید می‌خواهید با استفاده از یک رویه فرعی محتویات دو متغیر را با یکدیگر جابه‌جا کنید، برای این منظور رویه زیر را در نظر بگیرید:

```
Private Sub moving(ByRef sngx As Single,ByRef sngy As Single)
```

```
Dim sngtemp As Single
```

```
sngtemp = sngy
```

```
sngy = sngx
```

```
sngx = sngtemp
```

```
End Sub
```

رویه فرعی moving دو آرگومان دارد که برای دریافت مقادیر دو متغیر به کار می‌روند و در ابتدای نام هر یک از آن‌ها از کلمه کلیدی ByRef استفاده شده است، این عبارت تعیین می‌کند که نحوه ارسال متغیرها به رویه به صورت فراخوانی با مرجع باشد.

هم‌چنین با تعریف یک متغیر واسطه با نام sngtemp در داخل رویه فرعی، ابتدا مقادیر یکی از متغیرها را در متغیر واسطه (sngtemp) ذخیره کرده، سپس مقدار متغیر sngx را در متغیر sngy و در آخر مقدار متغیر واسطه را در متغیر sngx کپی می‌کند. به این صورت مقادیری که در آرگومان‌های sngx و sngy قرار دارند با یکدیگر تعویض می‌شوند؛ اما باید ببینیم که چگونه از این رویه می‌توان برای تعویض مقدار دو متغیر در خارج از رویه استفاده کرد. فرض کنید که برای این کار رویه فرعی moving به صورت زیر فراخوانی می‌شود:

```
Dim x As Single
```

```
Dim y As Single
```

```
x = 28
```

```
y = -10
```

```
Call moving(x, y)
```

```
Print x, y
```

پس از تعریف و مقداردهی دو متغیر  $x$  و  $y$ ، رویه فرعی فراخوانی شده است. در این جا اجرای برنامه به محل رویه منتقل می شود و چون از کلمه کلیدی ByRef در ابتدای نام آرگومانها در تعریف رویه استفاده شده است، ارتباط مستقیمی بین متغیرهای ارسالی و آرگومانهای رویه moving به وجود می آید؛ در نتیجه تغییر در محتویات آرگومانهای  $sngx$  و  $sngy$  سبب تغییر در محتویات متغیرهای ارسالی  $x$  و  $y$  و تعویض محتویات دو متغیر می شوند. بنابراین پس از خاتمه اجرای رویه، دستور Print مقادیر جابه جا شده را برای متغیرهای  $x$  و  $y$  در روی فرم نمایش خواهد داد.

**نکته** اگر رویه فرعی moving به صورت  $Call\ moving(x+1, y+1)$  فراخوانی شود مقدار متغیرها پس از فراخوانی رویه همان مقادیر قبل از فراخوانی خواهد بود.  
اگر مقادیر ثابت به رویه فرعی moving ارسال شود کلمه های کلیدی ByRef هیچ تأثیری روی نحوه ارسال آنها نمی گذارد.  
در روش فراخوانی با مرجع، هم نام یا غیرهم نام بودن متغیرهای ارسالی با آرگومانهای رویه اهمیتی ندارد.



### تمرین:

پروژه ای طراحی کنید تا بتوانید با استفاده از یک فرم، یک کنترل دکمه فرمان و با استفاده از رویه فرعی moving تعویض محتویات دو متغیر را با استفاده از روش فراخوانی با مرجع بررسی کنید.

## ۲-۳-۷ روش فراخوانی با مقدار

گاهی اوقات لازم است تا بین متغیرهای ارسالی و آرگومانهای متناظر آنها ارتباطی به وجود نیاید و در صورت تغییر در مقدار آرگومانها، در محتویات متغیرهای ارسالی تغییر حاصل نشود. در این صورت امکان استفاده از روش فراخوانی با مرجع مفید نیست. روش دیگری که برای انجام این گونه عملیات مناسب است، روش فراخوانی با مقدار می باشد. در این روش با ذکر کلمه کلیدی ByVal به جای عبارت ByRef از ایجاد ارتباط مستقیم بین متغیرهای ارسالی و آرگومانها جلوگیری به عمل می آید و فقط مقادیر موجود در متغیرهای ارسالی در داخل آرگومانهای متناظر آنها کپی خواهد شد. به عنوان

مثال اگر تعریف رویه فرعی moving را به این صورت تغییر دهید:

```
Private Sub moving(ByVal sngx As Single, ByVal sngy As Single)
```

در صورت فراخوانی رویه فرعی moving با حالت فوق، در واقع عدد ۲۸ در آرگومان sngx رویه و عدد ۱۰ - در آرگومان sngy رویه کپی خواهند شد و پس از تعویض محتویات این دو آرگومان در داخل رویه و خروج از رویه، مقادیر مربوط به این آرگومان‌ها از بین خواهد رفت و با بازگشت به محل فراخوانی و اجرای دستور Print همان مقادیر قبل از فراخوانی برای متغیرهای x و y نمایش داده می‌شوند.



تمرین:

رویه moving را در پروژه‌ای که در تمرین قبل ایجاد کرده‌اید به شکل فراخوانی با مقدار تغییر دهید سپس آن را اجرا کرده و نتیجه را با حالت قبل بررسی کنید.



**مثال ۴:** یک رویه فرعی بنویسید تا دو عدد را دریافت کرده، میانگین آن‌ها را محاسبه کند و نمایش دهد. برای این کار یک رویه average را به صورت زیر بنویسید:

```
Private Sub average(ByVal sngno1 As Single, ByVal sngno2 _
```

```
As Single, ByVal sngave As Single)
```

```
sngave = (sngno1 + sngno2) / 2
```

```
End Sub
```

در این رویه از سه آرگومان استفاده شده است. آرگومان‌های sngno<sub>1</sub> و sngno<sub>2</sub> برای ارسال اعداد به داخل رویه و به صورت فراخوانی با مقدار تعریف شده‌اند، زیرا لازم نیست این دو آرگومان در متغیرهایی که در زمان فراخوانی به رویه ارسال می‌شوند، تغییری ایجاد کنند. اما آرگومان سوم یعنی sngave با روش ارسال با مرجع تعریف شده است؛ زیرا پس از محاسبه معدل دو عدد در محل فراخوانی مورد استفاده قرار می‌گیرد. برای چنین رویه‌ای، فراخوانی به صورت زیر خواهد بود:

```
Call average (x , y , average)
```

در فراخوانی فوق ابتدا مقادیر متغیرهای ارسالی x و y در آرگومان‌های sngno<sub>1</sub> و sngno<sub>2</sub>

کپی می‌شوند و بین متغیر average و آرگومان sngave در رویه به گونه‌ای ارتباط

برقرار می‌شود که تغییر در مقدار آرگومان sngave، محتویات متغیر ارسالی average را نیز تحت تأثیر قرار می‌دهد و سبب اعمال همان تغییرات روی آن خواهد شد. می‌بینید که با این روش نیز می‌توان با استفاده از رویه‌های فرعی نتیجه محاسبات را به محل فراخوانی بازگشت داد. به این صورت پس از محاسبه متوسط دو عدد و ذخیره این مقدار در متغیر sngave رویه خاتمه یافته و به محل فراخوانی باز می‌گردد. اکنون می‌توانید در محل فراخوانی رویه با استفاده از متغیر average به معدل دو عدد دسترسی پیدا کنید.



تمرین:

یک پروژه از نوع Standard EXE به همراه یک فرم به گونه‌ای طراحی کنید تا با استفاده از رویه average بتوان معدل هر دو عدد دلخواه را محاسبه و روی فرم مشاهده کرد.

## ۴-۷ نحوه استفاده از نام آرگومان‌ها در رویه‌ها

تاکنون با استفاده از رعایت ترتیب در ارسال آرگومان‌های رویه‌ها توانستیم داده‌ها را به آرگومان‌های متناظرشان نسبت دهیم، شما می‌توانید با استفاده از نام آرگومان‌ها در زمان فراخوانی، ترتیب ذکر داده‌ها را بدون در نظر گرفتن ترتیب قرارگیری آن‌ها در تعریف رویه، به رویه ارسال کنید.

این روش باعث می‌شود در زمان زیاد بودن تعداد آرگومان‌ها کار فراخوانی آسان‌تر و احتمال اشتباه در ارسال داده‌ها به آرگومان‌ها کمتر شود. برای استفاده از نام آرگومان‌ها در زمان فراخوانی می‌توانید به صورت زیر عمل کنید: مقدار ارسالی = نام آرگومان  
فرض کنید می‌خواهید یک رویه تابعی بنویسید که به وسیله آن بتوان مجموع اعداد زوج بین دو عدد طبیعی دلخواه را محاسبه کرد. برای این کار به این دستورات توجه کنید.

$$(m < n)$$

Function Myfunc(m As Integer, n As Integer) As Single

Dim i As Integer, sum As Single

If m Mod 2 = 0 Then m = m+2 Else m = m+ 1

sum= 0



```
For i = m To n-1 Step 2
```

```
sum=sum+i
```

```
Next i
```

```
Myfunc = sum
```

```
End Function
```

در تعریف این تابع دیده می‌شود دو آرگومان  $m$  و  $n$  (با فرض  $m < n$ ) به تابع ارسال می‌شوند. در تابع ابتدا زوج یا فرد بودن  $m$  به وسیله یک `if` بررسی می‌شود تا بتوان عدد زوج بعد از  $m$  را تعیین کرد اگر  $m$  زوج باشد حاصل  $M \bmod 2 = 0$  درست خواهد بود و با اضافه شدن دو واحد به  $m$  عدد زوج بعدی محاسبه می‌شود اما اگر  $m$  فرد باشد نتیجه  $M \bmod 2 = 0$  نادرست خواهد بود و با اضافه شدن یک واحد به  $m$  باز عدد زوج بعدی محاسبه می‌شود، سپس مجموع اعداد زوج بین  $m$  و  $n$  با یک حلقه `For` و توسط دستور  $Sum = Sum + i$  به دست می‌آید و در پایان مقدار `Sum` به عنوان مجموع اعداد زوج بین  $m$  و  $n$  بازگشت داده می‌شود. حلقه `For` نیز تا رسیدن به مقدار  $n - 1$  اجرا شده تا در صورت زوج بودن  $n$  خود  $n$  در مجموع وارد نشود.

بعد از درک نحوه کارکرد تابع به فراخوانی آن می‌پردازیم. تاکنون نحوه فراخوانی چنین توابعی را فرا گرفته‌اید، مثلاً برای نمایش مجموع اعداد زوج بین ۲ و ۱۲ از فرمان بعد استفاده می‌کنیم:

```
Print Myfunc (2,12)
```

اگر در فراخوانی جای اعداد ۲ و ۱۲ عوض شود نتیجه کار نامناسب خواهد بود. اما با فراخوانی با استفاده از نام آرگومان‌ها امکان اشتباه از بین می‌رود. برای نمونه در مثال فوق فراخوانی با استفاده از نام آرگومان‌ها به صورت زیر خواهد بود:

```
Print Myfunc (n := 12 , m := 2)
```

می‌بینید که برای فراخوانی با استفاده از نام آرگومان‌ها باید از نام آرگومان به همراه «:=» در قبل از مقدار ارسالی استفاده کنید. البته فراخوانی قبل را می‌توان به صورت زیر نیز نوشت:

```
Print Myfunc (m := 2 , n := 12)
```

تمرین:



رویه تابعی `Myfunc` را به گونه‌ای تنظیم کنید که بدون فرض  $M < N$  نیز بتواند

محاسبات را به‌طور صحیح انجام دهد.

## ۵-۷ خروج از یک رویه با استفاده از دستورات Exit Sub و Exit Function

گاهی اوقات لازم است تا قبل از این که تمام دستورات یک رویه اجرا شده و به انتهای رویه برسید، از رویه خارج شده و به محل فراخوانی بازگردید. به این منظور برای خروج از یک رویه فرعی از دستور Exit Sub و برای خروج از یک رویه تابعی از دستور Exit Function استفاده کنید.



**مثال ۵:** یک رویه تابعی تعریف کنید تا تعداد ارقام هر عدد طبیعی دلخواه را محاسبه کرده و نمایش دهد. برای این کار رویه تابعی digits را به صورت زیر تعریف کنید:

```
Private Function digits(ByVal sngno As Integer) As Integer
```

```
    Dim intdigits As Integer
```

```
    If sngno < 0 Then
```

```
        digits = -1
```

```
        Exit Function
```

```
    End If
```

```
    While (sngno > 0)
```

```
        0 ngno = sngno \ 1
```

```
        intdigits = intdigits + 1
```

```
    Wend
```

```
    digits = intdigits
```

```
End Function
```

این تابع دارای یک آرگومان است که برای انتقال عدد مورد نظر به داخل تابع استفاده شده است. علاوه بر این روش ارسال عدد مورد نظر نیز ارسال با مقدار خواهد بود تا تغییر روی مقدار آرگومان sngno در رویه تابعی، تأثیری روی متغیر ارسالی نگذارد. در ضمن نوع مقدار داده بازگشتی یعنی تعداد ارقام نیز از نوع صحیح است؛ بنابراین نوع داده بازگشتی در انتهای خط اول رویه و از نوع Integer تعریف می‌شود.

به هر حال در صورت فراخوانی این تابع و اجرای آن ابتدا مقدار متغیر ارسالی در

آرگومان sngno کپی می‌شود، سپس با اجرای یک دستور If مقدار عدد ارسال شده بررسی می‌شود و در صورتی که مقدار آن کوچک‌تر از صفر باشد، مقدار ۱- را بازگشت می‌دهد. از این مقدار می‌توان در محل فراخوانی برای اطمینان از نحوه ارسال صحیح مقدار به تابع استفاده کرد سپس با استفاده از دستور Exit Function اجرای رویه خاتمه یافته و به محل فراخوانی بازمی‌گردید. این عمل سبب خواهد شد تا کاربر از ارسال مقادیر عددی اشتباه مطلع شود، اما اگر مقدار ارسالی بزرگ‌تر از صفر باشد با استفاده از یک حلقه While...Wend تعداد ارقام عدد محاسبه شده و در متغیر intdigits ذخیره می‌شود و پس از خاتمه حلقه این مقدار با استفاده از دستور digits= intdigits بازگشت داده خواهد شد.



تمرین:

پروژه‌ای طراحی کنید تا با استفاده از یک رویه فرعی بتواند تعداد ارقام هر عدد طبیعی دلخواه را محاسبه کرده و در یک کادر پیام نمایش دهد، به علاوه در صورت ارسال مقادیر منفی به رویه، پیام خطایی نمایش داده شود و اجرای رویه نیز خاتمه یابد.

## ۶-۷ رویه‌های محلی و عمومی

رویه‌هایی که تاکنون تعریف شده‌اند از نوع محلی یا به عبارت دیگر Private می‌باشند و فقط در همان ماژول که تعریف می‌شوند قابل شناسایی و فراخوانی هستند، اما گاهی اوقات لازم است تا یک رویه تابعی یا فرعی را که در یک ماژول فرم تعریف شده است در ماژول فرم دیگری فراخوانی کنید.



شکل ۶-۷

در این صورت در زمان اجرای برنامه و با فراخوانی رویه مزبور پیام خطایی مطابق شکل ۶-۷ نمایش داده می‌شود.

### ۱-۶-۷ نحوه ایجاد رویه‌های عمومی در ماژول فرم


برای آن که عملیات فراخوانی یک رویه از داخل ماژول فرم دیگری امکان‌پذیر باشد به جای کلمه کلیدی Private، از کلمه کلیدی Public استفاده کنید. به این شکل رویه به


عنوان یک رویه عمومی معرفی خواهد شد و علاوه بر فراخوانی آن در ماژولی که تعریف شده است در سایر ماژول‌ها امکان فراخوانی خواهد داشت، به چنین رویه‌هایی رویه‌های عمومی می‌گویند.

برای فراخوانی رویه‌های عمومی باید قبل از ذکر نام رویه، نام ماژول فرمی را که رویه در آن تعریف شده است، تایپ کنید و سپس کاراکتر نقطه را تایپ کرده و نام رویه را ذکر نمایید. این مطلب را می‌توان به این صورت بیان کرد:

**Call** (مقادیرارسالی) نام رویه فرعی. نام فرمی که رویه فرعی در آن تعریف شده است

(مقادیرارسالی) نام رویه تابعی. نام فرمی که رویه تابعی در آن تعریف شده است.

 **نکته** در صورتی که حوزه شناسایی (Scope) یک رویه در زمان تعریف، تعیین نشود رویه به طور پیش فرض به صورت عمومی تعریف خواهد شد.

 **مثال ۶:** پروژه‌ای طراحی کنید که میزان مصرف برق هر ساختمان دلخواهی را محاسبه کرده، مقادیر فعلی و قبلی را که از کنتور برق قرائت شده‌اند، وارد کند. با محاسبه میزان مصرف، هزینه برق مصرفی در ساختمان مورد نظر به دست می‌آید. به این منظور مراحل بعد را به ترتیب انجام دهید:

۱ - یک پروژه جدید از نوع Standard EXE ایجاد کنید، سپس یک فرم مطابق شکل ۷-۷ برای دریافت مقادیر ورودی ایجاد کنید.



شکل ۷-۷

در این فرم از چهار کادر متن برای دریافت تاریخ قرائت کنتور و میزان مصرف برق در مراجعه قبلی و مراجعه فعلی استفاده می‌شود و از دکمه Compute برای انجام محاسبات و نمایش نتیجه آن روی فرم دیگری استفاده می‌شود.

۲- برای انجام محاسبات از یک رویه تابعی و یک رویه فرعی استفاده کنید، این رویه‌ها را در ماژول فرم و به این صورت تنظیم کنید:

```
Public Function expense (ByVal sngpv As Single, ByVal sngcv As Single) As Currency
```

```
    Dim sngresult As Single , intlevel As Integer
```

```
    Dim curpay As Currency
```

```
    sngresult = sngcv - sngpv
```

```
    Select Case sngresult
```

```
        Case Is < 500 :
```

```
            intlevel = 1
```

```
        Case Is < 1000 :
```

```
            intlevel = 2
```

```
        Case Is < 1500 :
```

```
            intlevel = 3
```

```
        Case Else
```

```
            intlevel = 4
```

```
    End Select
```

```
    Call payment (intlevel, sngresult, curpay)
```

```
    expense = curpay
```

```
End Function
```

```
Private Sub payment(intlevel As Integer, ByVal sngresult _
```

```
    As Single, ByRef curpay As Currency)
```

```
    Select Case intlevel
```

```
        Case 1 : curpay = sngresult * 100
```

```
        Case 2 : curpay = sngresult * 250
```

```
        Case 3 : curpay = sngresult * 500
```

```
        Case 4 : curpay = sngresult * 1000
```

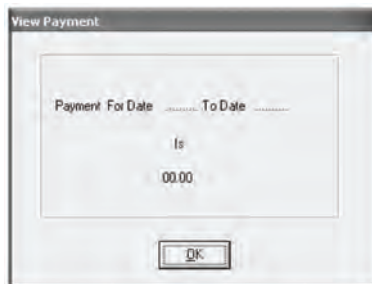
```
    End Select
```

```
End Sub
```

رویه تابعی expense به صورت عمومی تعریف شده و دارای دو آرگومان از نوع Single برای دریافت مقادیر مصرف قبلی و فعلی است. این تابع با توجه به مقدار مصرف برق بر اساس کیلووات، سطح مصرف و هزینه آن را محاسبه کرده و به صورت یک مقدار از نوع Currency بازمی گرداند، پس از فراخوانی این تابع ابتدا مقادیر قرائت شده به ترتیب در آرگومان های sngpv و sngcv کپی می شوند، سپس تفاضل این دو مقدار از یکدیگر کسر شده و به عنوان میزان مصرف در متغیر sngresult ذخیره می شوند و بر همین اساس با استفاده از یک دستور Select Case سطح مصرف (intlevel) نیز تعیین می شود تا با توجه به این سطح، هزینه مصرف برق محاسبه شود. به این منظور چهار سطح مصرف از کم تا خیلی زیاد در نظر گرفته شده است، پس از محاسبه سطح مصرف برق، نوبت محاسبه هزینه آن می شود. این کار با استفاده از یک رویه فرعی انجام می شود. رویه payment با دریافت سطح مصرف (intlevel) و مقدار مصرف برق (sngresult)، هزینه آن را با استفاده از یک دستور Select Case بر اساس سطح مصرف محاسبه کرده و در متغیر curpay ذخیره می کند. به علاوه متغیر curpay به صورت فراخوانی با مرجع به رویه ارسال می شود تا میزان هزینه در محل فراخوانی قابل استفاده باشد، این رویه از نوع محلی تعریف شده است. پس از آن که رویه payment خاتمه یافت، اجرای برنامه به محل فراخوانی آن یعنی رویه تابعی expense بازگشته و با استفاده از دستور  $expense = curpay$  مقدار هزینه مصرف برق را به محل فراخوانی بازمی گرداند.

۳- نام کادرهای متن مربوط به تاریخ قرائت و میزان مصرف قبلی را به ترتیب روی tx - pd و txtpv و تاریخ قرائت و میزان مصرف فعلی را به ترتیب روی txtcd و txtcv تنظیم کنید.

۴ - برای نمایش میزان هزینه مصرف برق در یک فرم جداگانه، احتیاج به یک فرم جدید خواهید داشت، بنابراین یک فرم جدید مطابق شکل ۸-۷ ایجاد کنید، سپس فرم و پروژه را با نام viewpayment ذخیره کنید.



شکل ۸-۷

سپس رویداد Activate این فرم و رویداد Click دکمه OK را به صورت زیر تنظیم کنید:

```
Private Sub Form_Activate()
```

```
Dim sngpay As Single
```

```
Dim sngpv As Single
```

```
Dim sngcv As Single
```

```
sngpv = Val(frmpower.txtpv.Text)
```

```
sngcv = Val (frmpower.txtcv.Text)
```

```
sngpay = frmpower.expense (sngpv, sngcv)
```

```
lblview.Caption = "Payment For Date " + Trim (frmpower.txtpd.Text) + _
```

```
" To Date "+Trim (frmpower.txtcd.Text)
```

```
lblpay.Caption = sngpay
```

```
End Sub
```

```
Private Sub cmdok_Click()
```

```
Unload Me
```

```
frmpower.SetFocus
```

```
frmpower.txtpd.SetFocus
```

```
End Sub
```

در این فرم از رویداد Activate فرم View Payment استفاده شده است تا پس از نمایش فرم، گزارش محاسبات در روی فرم به نمایش درآید. در این رویه با فراخوانی تابع expense محاسبات انجام می‌شود. همان‌طور که ملاحظه می‌کنید، برای فراخوانی این تابع ابتدا نام فرمی که تابع در مازول آن تعریف شده، یعنی frmpower ذکر شده است. البته این کار زمانی مؤثر است که رویه به صورت عمومی (Public) تعریف شده باشد در این صورت در زمان تایپ نام فرم (frmpower) و کاراکتر نقطه یک لیست حاوی نام رویه expense که در مازول frmpower تعریف شده است، نمایش داده خواهد شد (شکل ۹-۷). نکته دیگری که در فراخوانی رویه expense مهم است، نحوه ارسال مقادیر تایپ شده در کادرهای متن است.

برای دسترسی خصوصیات کنترل‌های یک فرم در فرم دیگر می‌توانید ابتدا نام فرم، سپس نام کنترل و بعد نام خصوصیت را ذکر کنید؛ به این صورت مقادیر مصرف برق با استفاده از تابع Val به نوع عددی تبدیل شده و به تابع ارسال می‌شوند. مقدار هزینه مصرف برق با استفاده از تابع expense محاسبه شده و برگردانده می‌شود و در نهایت این مقدار در متغیر sngpay ذخیره می‌شود. در ادامه اجرای رویه Activate تاریخ‌های مربوط به قرائت کنتور به وسیله کنترل برچسب lblview و هزینه مصرف برق به وسیله کنترل برچسب lblpay نمایش داده می‌شوند.



شکل ۹-۷

رویداد Click دکمه OK نیز به‌گونه‌ای تنظیم شده است تا در صورتی که کاربر روی این دکمه کلیک کند، فرم View Payment بسته شده و فوکوس با متد SetFocus به فرم frmpower و سپس کنترل کادر متن txtpd داده شود.

۵ - به ماژول فرم frmpower بروید و دستور frmview.show را در رویداد Click دکمه Compute و دستور Unload Me را در رویداد Click دکمه Exit تایپ کنید.

۶ - فرم و پروژه را با نام power ذخیره کنید. پروژه را اجرا کنید و داده‌ها را مطابق شکل ۱۰-۷ در کادرهای متن تایپ کرده و روی دکمه Compute کلیک کنید، نتیجه محاسبات مطابق شکل ۱۱-۷ در فرم دوم قابل مشاهده خواهد بود.

۷ - با کلیک روی دکمه Exit به اجرای پروژه خاتمه دهید و به پنجره ویژوال بیسیک بازگردید.





شکل ۷-۱۰



شکل ۷-۱۱



**تمرین:** پروژه power را به گونه‌ای تنظیم کنید که رویه فرعی payment در ماژول فرم View Payment تعریف شود.

## ۷-۶-۲ نحوه ایجاد رویه‌های محلی و عمومی با استفاده از ماژول کد

روش دیگری نیز در ویژوال بیسیک وجود دارد که در آن به جای تعریف رویه‌های موردنظر در ماژول فرم از ماژول کد (Code Module) استفاده می‌شود. ماژول کد یکی دیگر از اجزای تشکیل دهنده در پروژه‌هاست. ماژول‌های کد فقط از دستورالعمل‌ها تشکیل می‌شوند و شکل ظاهری مانند فرم‌ها ندارند. یکی دیگر از کاربردهای ماژول کد، تعریف متغیرهای عمومی است. نحوه تعریف و فراخوانی رویه‌ها در یک ماژول کد مانند ماژول فرم است.



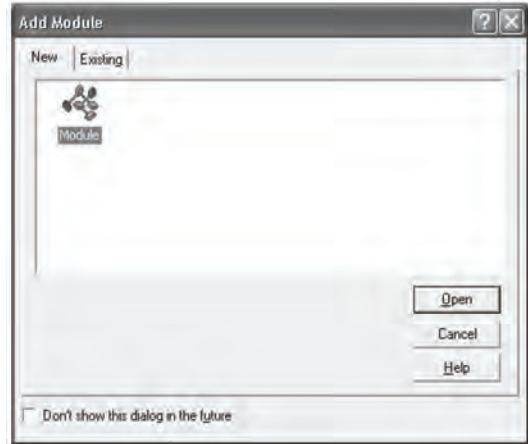
**مثال ۷:** پروژه power را به گونه‌ای تغییر دهید تا رویه‌های مربوطه را با استفاده از ماژول کد تعریف و فراخوانی کند. به این منظور عملیات زیر را به ترتیب انجام دهید:

- ۱- ابتدا یک ماژول کد به پروژه اضافه کنید به این منظور در پنجره ویژوال بیسیک و در پنجره پروژه کلیک راست کنید سپس به ترتیب روی گزینه Add و بعد روی گزینه Module کلیک کنید (شکل ۷-۱۲). کادر محاوره Add Module مطابق شکل ۷-۱۳ نمایش داده می‌شود.

- ۲- در کادر محاوره Add Module، روی دکمه Open کلیک کنید تا یک ماژول کد به پروژه power اضافه شود. اکنون به پنجره خصوصیات بروید و خصوصیت Name ماژول کد را روی مقدار mdlpower تنظیم کنید.



شکل ۱۲-۷




شکل ۱۳-۷

۳ - در پنجره پروژه روی آیکن ماژول کد کلیک راست کنید و گزینه Save ModuleAs...

را انتخاب کنید، سپس ماژول کد را با نام power ذخیره کنید. همان طور که در کادر  
محواره ذخیره سازی ماژول مشاهده می کنید، پسوند فایل های ماژول کد BAS است.

۴ - در این مرحله به ماژول فرم frmpower بروید و رویه های expense و payment را  
به ماژول کد mdlpower انتقال دهید.

۵ - به ماژول فرم frmview بروید و عبارت frmpower را از ابتدای نام رویه expense  
حذف کنید. توجه داشته باشید اگر رویه ای در ماژول کد به صورت عمومی تعریف شود،  
تمام ماژول ها می توانند آن را فراخوانی کنند و نیازی به استفاده از نام ماژول کد در زمان  
فراخوانی رویه نیست.

 **نکته** در صورتی که یک متغیر در یک ماژول کد به صورت عمومی تعریف شود، در تمام رویه  
قابل شناسایی و استفاده است

۶ - تغییرات را ذخیره کرده و پروژه را اجرا کنید و با همان مقادیر مثال ۵ آزمایش  
کنید.

۷ - در پایان روی دکمه Exit کلیک کنید تا اجرای پروژه خاتمه یابد، سپس به پنجره  
ویژوال بیسیک بازگردید.

## ۷-۷ رویه‌های رویداد (Event Procedures)

تاکنون از این‌گونه رویه‌ها بارها استفاده کرده‌اید و همان‌طور که می‌دانید یکی دیگر از ویژگی‌های زبان برنامه‌نویسی ویژوال بیسیک پشتیبانی از رویدادهاست. این نوع از رویه‌ها در زمان ایجاد یک شیء مانند کنترل یا فرم همراه با آن‌ها به طور هم‌زمان به وجود می‌آیند و زمانی که رویداد مربوط به یک شیء رخ می‌دهد، رویه رویداد و دستورات موجود در آن به طور خودکار اجرا می‌شوند. شکل کلی یک رویه رویداد به این صورت است:

(آرگومان‌های رویداد) نام رویداد - نام شیء Private Sub

⋮

دستورات

End Sub

شیء می‌تواند یک کنترل یا یک فرم باشد. در رویه‌های رویداد کنترل‌ها، نام شیء، مقدار خصوصیت Name کنترل و در رویه‌های رویداد فرم‌ها، از کلمه کلیدی Form به عنوان نام شیء استفاده می‌شود.


در بعضی از رویه‌های رویداد بخش دیگری به نام آرگومان‌های رویداد نیز وجود دارند. آرگومان‌ها در واقع، متغیرهایی هستند که با توجه به نوع رویداد، اطلاعاتی را به داخل رویه منتقل می‌کنند تا برنامه‌نویس با استفاده از آن‌ها مقاصد خود را در رویداد پیگیری کند. رویه‌های رویداد مربوط به ماوس و صفحه کلید از این دسته رویدادها هستند که از آرگومان‌ها استفاده می‌کنند.

البته تاکنون از رویه‌های رویدادی مانند Click, DblClick, Load, Unload و نظایر آن‌ها در رابطه با فرم‌ها و کنترل‌ها استفاده کرده‌اید. در مباحث آینده نیز نحوه استفاده از رویه‌های رویداد دیگری را فرامی‌گیرید. در این‌جا لازم است به ارایه توضیحاتی در رابطه با مهم‌ترین رویدادهای فرم‌ها و کنترل‌ها بپردازیم.

### ۷-۷-۱ رویه رویداد Activate

این رویه رویداد مربوط به فرم‌هاست و زمانی اجرا می‌شود که فرم، فوکوس می‌گیرد. وقتی پروژه‌ای از چند فرم تشکیل شده باشد و فوکوس از یک پنجره به پنجره دیگر

برنامه منتقل شود، رویداد Activate مربوط به پنجره‌ای که فوکوس را در اختیار گرفته است، اجرا خواهد شد. لازم به ذکر است که این رویه رویداد فاقد هر گونه آرگومان می‌باشد. این رویداد پس از رویداد Load اجرا می‌شود.

 **نکته** رویداد Activate در زمان انتقال فوکوس از پنجره یک برنامه به پنجره برنامه دیگر اجرا نخواهد شد.

### ۲-۷-۲ رویه رویداد Click

این رویه نیز فاقد آرگومان است و زمانی اجرا می‌شود که کاربر روی فرمی که فعال است یا کنترل موجود در آن کلیک کند.

### ۳-۷-۲ رویه رویداد Deactivate

این رویداد نیز فاقد آرگومان و مخصوص فرم‌هاست و عملکردی شبیه به رویداد Activate دارد، با این تفاوت که زمان اجرای آن هنگامی است که فرم، فوکوس خود را بین پنجره‌های یک برنامه از دست می‌دهد و اطلاعات فرم در حافظه باقی می‌ماند.

### ۴-۷-۲ رویه رویداد DbClick

این رویه رویداد در فرم‌ها و بسیاری از کنترل‌ها وجود دارد و فاقد هرگونه آرگومان است. این رویداد زمانی رخ می‌دهد که کاربر روی فرمی که فعال است یا کنترل موجود در آن، عمل دابل کلیک انجام دهد.

### ۵-۷-۲ رویه رویداد Load

این رویه نیز مربوط به فرم‌هاست و زمانی اجرا می‌شود که اطلاعات مربوط به فرم وارد حافظه اصلی شده و فرم، روی دسک‌تاپ نمایش داده شود. در صورتی که بخواهید دستوراتی را در زمان اجرای برنامه انجام دهید، از این رویداد استفاده کنید.


### ۶-۷-۲ رویه رویداد Unload

این رویداد نیز مخصوص فرم‌هاست و فاقد آرگومان می‌باشد. زمانی که فرم را می‌بندید

یا به عبارت دیگر اطلاعات فرم از حافظه کامپیوتر خارج شود، این رویداد اجرا می‌شود. در صورتی که بخواهید دستوراتی را در زمان خاتمه برنامه یا بسته شدن فرم اجرا کنید، از این رویه استفاده کنید.

### ۷-۷-۷ رویه رویداد GotFocus

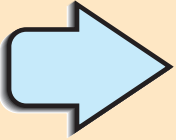
این رویداد نیز بین فرم‌ها و کنترل‌ها مشترک است و زمانی که فرم یا کنترل فوکوس را در اختیار می‌گیرند، اجرا می‌شود. در هنگامی که یک فرم، فوکوس را به دست می‌آورد ابتدا رویداد Activate و سپس رویداد GotFocus اجرا می‌شود.

 **نکته** در صورتی که در یک فرم رویداد Activate و GotFocus به‌طور هم‌زمان مورد استفاده قرار گیرند فقط رویداد Activate اجرا خواهد شد.

در صورتی که در یک فرم کنترلی وجود داشته باشد که بتواند فوکوس بگیرد (مانند کنترل Text Box) رویداد GotFocus اجرا نمی‌شود.

### ۷-۷-۸ رویه رویداد LostFocus

این رویداد نیز بین فرم‌ها و کنترل‌ها مشترک است و زمانی که فرم و کنترل، فوکوس خود را از دست می‌دهند، اجرا می‌شود. هنگامی که یک فرم، فوکوس را از دست می‌دهد ابتدا رویداد LostFocus و سپس رویداد Deactivate اجرا می‌شود.



## Learn in English

### **Introduction to Procedures**

You can simplify programming tasks by breaking programs into smaller logical components. These components called procedures can then become building blocks that let you enhance and extend Visual Basic. Procedures are useful for condensing repeated or shared tasks, such as frequently used calculations, text and control manipulation, and database operations. There are several types of procedures used in Visual Basic:

- Sub procedures do not return a value.
- Function Procedures return a value.
- Event Procedure.
- Visual Basic Procedure

## واژه‌نامه

Component	جزء
Condense	خلاصه کردن
Enhance	افزودن
Extend	توسعه دادن
Frequently	تکراری
Introduction	معرفی، آشنایی
Manipulation	دستکاری کردن
Private	خصوصی
Procedure	رویه
Public	عمومی
Scope	میدان دید

## خلاصه مطالب

- در برنامه‌نویسی به روش ساخت یافته، برنامه به بخش‌های کوچک‌تر به نام رویه تقسیم می‌شود.
- رویه‌ها در ویژوال بیسیک به چهار گروه رویه‌های فرعی و تابعی، رویه‌های رویداد و رویه‌های آماده و ویژوال بیسیک تقسیم می‌شوند.
- رویه‌های رویداد در زمان ایجاد یک شیء ایجاد می‌شوند و در زمانی که رویداد مربوطه رخ می‌دهد به‌طور خودکار اجرا می‌شوند.
- رویه‌های فرعی و تابعی را می‌توانید با تایپ در ماژول فرم یا با استفاده از کادر محاوره Add Procedure تعریف کنید.
- برای فراخوانی رویه‌های فرعی از دستور Call استفاده می‌شود.
- برای تعریف یک رویه محلی کلمه کلیدی Private را در ابتدای تعریف رویه قرار دهید.
- یک رویه محلی فقط در همان ماژولی که تعریف می‌شود قابل شناسایی و فراخوانی است.
- برای تعریف یک رویه عمومی، کلمه کلیدی Public را در ابتدای تعریف رویه قرار دهید.
- یک رویه عمومی در تمام ماژول‌های موجود در پروژه، قابل شناسایی و فراخوانی است.
- برای فراخوانی رویه‌ها از دوروش فراخوانی با مقدار یا فراخوانی با مرجع استفاده می‌شود.
- در فراخوانی رویه‌ها با مقدار یک کپی از داده‌ها در آرگومان قرار داده می‌شود.
- در فراخوانی رویه‌ها با مرجع تغییر در مقدار آرگومان‌های رویه سبب تغییر مقدار متغیرهای ارسالی می‌شود.
- در هنگام فراخوانی رویه‌ها می‌توان از نام آرگومان‌ها استفاده کرد.
- برای خروج از یک رویه فرعی و تابعی به ترتیب از دستورات Exit Sub و Exit Function استفاده می‌شود.

## آزمون نظری

۱ - کدام رویداد هنگامی که یک فرم، فوکوس خود را از دست می‌دهد، اجرا می‌شود؟

الف - Activate      ب - Deactivate      ج - Load      د - Unload

۲ - کدام گزینه در رابطه با تفاوت رویه‌های فرعی و رویه‌های تابعی درست است؟

الف - ارسال متغیرها به رویه      ب - بازگشت یک مقدار به محل فراخوانی  
ج - تعریف آرگومان‌ها      د - محل تعریف رویه

۳ - پس از فراخوانی رویه تابعی myname به صورت «Basic»، «Visual»، «Call myname»

مقدار بازگشتی چیست؟

Public Function myname (ByRef strfname As String, ByVal strlname As String) As String

Dim strname As String

strname = strfname + strlname

myname = strname

End Function

الف - «Visual»      ب - «Visual Basic»      ج - «Basic»      د - «VisualBasic»

۴ - برای تعریف یک رویه عمومی از کلمه کلیدی ..... استفاده می‌شود.

الف - Private      ب - Public      ج - Dim      د - Sub

۵ - استفاده از کدام گزینه برای تعریف یک رویه فرعی مناسب است؟

الف - Sub ... End Sub      ب - Function ... End Function

ج - Private      د - Public

۶ - پسوند فایل‌های مازول کد در ویژوال بیسیک چیست؟

الف - VBC      ب - BAS      ج - CLM      د - FBC

۷ - در صورتی که در یک رویه تابعی، نوع داده بازگشتی تعیین نشود از چه نوعی به طور

پیش فرض استفاده می‌شود؟

الف - Boolean      ب - Integer      ج - Byte      د - String

۸ - به طور پیش فرض نحوه ارسال متغیرها به یک رویه از چه نوعی است؟

الف - ارسال با مقدار      ب - ارسال با مرجع      ج - محلی      د - استاتیک

۹ - در صورت تعریف یک رویه با کلمه کلیدی Private در مازول کد.....

الف - رویه مربوطه در تمام مازول‌های کد قابل فراخوانی است.



- ب- رویه مربوطه در تمام ماژول‌ها قابل فراخوانی است.  
 ج- رویه مربوطه در تمام ماژول‌های فرم قابل فراخوانی است.  
 د- رویه مربوطه فقط در همان ماژول کدی که تعریف شده است قابل فراخوانی است.  
 ۱۰- پس از فراخوانی تابع test به صورت test (x As Long) مقدار بازگشتی چند است؟  
 (با فرض این که  $x = 3$  باشد).

Public Function test (ByVal y As Long) As Long

test = y \* y \* y

End Function

الف- ۳                      ب- ۹                      ج- ۲۷                      د- ۷۱

۱۱- نوع مقدار بازگشتی در رویه تابعی زیر چیست؟

Private Function Compute (into As Integer) As Boolean

الف- منطقی                      ب- عدد صحیح                      ج- رشته                      د- عدد اعشاری

۱۲- عملکرد رویه فرعی Subchange چیست؟

Public Sub Subchange (ByRef m As String, ByRef n As String)

Dim strchange As String

strchange = m

m = n

n = strchange

End sub

- الف- محتویات m و n را در رویه فرعی عوض می‌کند.  
 ب- محتویات m را در n قرار می‌دهد.  
 ج- محتویات m و n را در محل فراخوانی عوض می‌کند.  
 د- محتویات m و n را در رویه فرعی و هم در محل فراخوانی عوض می‌کند.  
 13 - Which of the following procedures can return a value?

a- Sub Procedure

b- VB Procedure

c- Functions

d- Event Procedure

- ۱۴- انواع رویه‌ها در ویژوال بیسیک را نام برده و هر یک را به طور خلاصه شرح دهید.  
 ۱۵- تفاوت بین رویه‌های فرعی و تابعی را بیان کنید.  
 ۱۶- انواع روش‌های فراخوانی رویه‌ها را توضیح دهید.  
 ۱۷- فرق بین رویه‌های عمومی و محلی را بیان کنید.  
 ۱۸- نحوه ایجاد و استفاده از یک ماژول کد را توضیح دهید.

## آزمون عملی

۱ - پروژه‌ای طراحی کنید که دمای یک اتاق را براساس فارنهایت دریافت کرده و با استفاده از یک رویه تابعی مقدار آنرا به سلسیوس محاسبه کرده و نمایش دهد.

$$c = \frac{5}{9}(F - 32)$$

۲ - پروژه‌ای طراحی کنید که یک عدد را دریافت کرده و با استفاده از یک رویه تابعی بخش‌پذیری آنرا بر اعداد ۳ و ۷ مشخص نماید.

۳ - پروژه‌ای طراحی کنید که یک عدد به عنوان مقدار زمان بر اساس ثانیه دریافت کرده و با استفاده از یک رویه فرعی مقدار ساعت آنرا محاسبه نموده و نمایش دهد.

۴ - پروژه‌ای طراحی کنید که با استفاده از یک رویه تابعی زوج یا فرد بودن هر عدد دلخواهی را مشخص کند.

۵ - پروژه‌ای طراحی کنید که موجودی یک مشتری بانک را براساس ریال دریافت کند و با استفاده از یک رویه فرعی آنرا به واحد یورو تبدیل کرده و نمایش دهد.  
( ۱۷۰۰ ریال = ۱ یورو )