



فصل

مفاهیم شیء گرای و ماژول کلاس

اهداف رفتار ی: پس از آموزش این فصل هنرجو می تواند:

- مفاهیم اصلی شیء گرای را توضیح دهد.
- Component Object Model را تعریف کند.
- تفاوت بین ماژول کلاس و کد را شرح دهد.
- یک مؤلفه COM در ویژوال بیسیک ایجاد کند.
- نمونه ای از یک کلاس را ایجاد کند.
- مشخصه ها و متدهای شیء را به کار گیرد.
- روال رویدادی برای مدیریت رویدادهای شیء بنویسد.

۱-۳- برنامه نویسی شیء گرا

برنامه نویسی شیء گرا دارای اصول و ویژگی هایی است که با پیدایش ویندوز و مفاهیمی مانند «چند وظیفه ای» که در سیستم عامل وجود دارد، مطرح شد. در این روش، برخلاف روش قبلی، به جای استفاده از تابع اصلی که وظیفه کنترل تمام برنامه را بر عهده دارد از مفاهیمی هم چون کلاس، مشخصه و شیء استفاده می شود. بنابراین برای یادگیری بهتر این روش، ابتدا باید با مفاهیم گفته شده آشنا شویم.

در محیط زندگی با مفهوم شیء آشنا هستیم. همان طور که می دانید یک شیء چیزی است مادی که دارای مشخصه‌هایی است، از جمله این که می تواند در مقابل برخی رویدادها که ممکن است برایش رخ دهد، از خود واکنش‌هایی نشان دهد. پس با این حال می توان گفت که هر شیء به همراه سه جنبه زیر شناخته می شود:

- مشخصه‌ها
- رفتار یا متد
- روابط

مشخصه‌ها، ویژگی‌هایی هستند که مشخص کننده حالت فعلی شیء است. به عنوان مثال، می توان گفت رنگ یک میز قهوه‌ای است یا قد یک شخص ۱۷۰ سانتی متر است. در این مثال، میز و انسان شیء هستند و قهوه‌ای بودن و ۱۷۰ سانتی متر بودن قد، مشخصه آن‌هاست. رفتار یک شیء، نحوه پاسخ آن شیء در مقابل رویدادهایی است که ممکن است برایش رخ دهد. به عنوان مثال، شیء میز در مقابل رویداد وارد آمدن نیروی بیش از حد، می شکند (رفتار).

تمرین: حال خودتان مثال‌هایی را برای شیء انسان و رویدادهایی که می تواند



برای وی اتفاق افتد و رفتارهای متقابل او بیاید.

توجه داشته باشید که ممکن است شیء خاصی در مقابل بعضی رویدادها هیچ رفتاری از خود نشان ندهد. این در صورتی است که برای رویداد مورد نظر هیچ رفتاری تعریف نشده باشد. روابط هر شیء نیز نشان دهنده ارتباط آن شیء با شیء‌های دیگر است. به عنوان مثال، یک شخص می تواند مالک یک شیء مانند میز باشد که در این صورت رابطه مالکیت بین شیء و شخص برقرار است.

شیء‌هایی که در دنیای واقعی وجود دارند، از انواع متفاوت هستند. حتی شیء‌های هم نوع ممکن است مشخصه‌ها، رفتارها و روابط متفاوتی داشته باشند. با توجه به این نکته، برای شیء‌ها تقسیم بندی خاصی را در نظر می گیریم و اصطلاح کلاس را تعریف می کنیم. کلاس مجموعه تمام شیء‌های هم نوع است. هر چند این شیء‌ها، مشخصه‌ها، رفتار و روابط متفاوتی داشته باشند.

به عنوان مثال، انسان یک کلاس است و هر شخص به خصوصی از این مجموعه، شیئی از کلاس مذکور محسوب می شود. با توجه به تعاریف و مفاهیمی که ذکر شد، می توان روش برنامه نویسی

شیء‌گرا را به صورت زیر بیان کرد :

هر برنامه شیء‌گرا شامل تعدادی شیء با مشخصه‌ها و متدهای متفاوت است به نحوی که روابط خاصی بین آن‌ها برقرار می‌باشد.

متدها، مجموعه‌ای از دستورالعمل‌های برنامه‌نویسی هستند که باید در هنگام بروز رویدادهایی آشکار شوند. مجموعه این دستورالعمل‌ها، رفتار آن شیء را در برابر رویداد به خصوصی نشان می‌دهند.


شیء‌گرایی از بهترین مفاهیم برنامه‌نویسی ساخت یافته به وجود آمده و با چندین مفهوم قوی ترکیب شده تا امکان سازماندهی برنامه‌ها به طور کارآمد را فراهم کند. به طور کلی، هنگامی که با روش شیء‌گرا برنامه می‌نویسید، مسأله را به بخش‌های تشکیل دهنده آن تجزیه می‌کنید. هر مؤلفه‌ای شامل دستورالعمل‌ها و داده‌های مرتبط با خودش است. از طریق این عملیات، پیچیدگی کاهش یافته و می‌توان برنامه‌های بزرگ‌تر را مدیریت کرد. همه زبان‌های برنامه‌نویسی شیء‌گرا در سه چیز مشترک هستند : کپسوله‌سازی، چندریختی و وراثت.

۱-۱-۳- کپسوله‌سازی (Encapsulation): همان طوری که می‌دانید تمام برنامه‌ها از دو

عنصر اصلی تشکیل می‌شوند : عبارات برنامه (کد) و داده‌ها. کد بخشی از برنامه است که عملیات را اجرا می‌کند و داده‌ها اطلاعاتی است که به وسیله این عملیات تحت تأثیر قرار می‌گیرند. کپسوله‌سازی، مکانیزم برنامه‌نویسی است که کد و داده‌ها را با هم در یک جا قرار داده و هر دو را از استفاده نادرست و تداخل خارجی ایمن نگه می‌دارد.


در یک زبان شیء‌گرا، کد و داده‌ها ممکن است با هم در چنین روشی محدود شوند که یک جعبه را ایجاد می‌کنند. درون جعبه تمام داده‌های موردنیاز و کد قرار دارد. هنگامی که در این روش، کد و داده‌ها با هم پیوند برقرار می‌کنند، یک شیء به وجود می‌آید. به عبارت دیگر، یک شیء ابزاری است که از کپسوله‌سازی پشتیبانی می‌کند.

اتومبیل دارای چهار چرخ، فرمان، ترمز، بدنه، در، موتور، شمع‌ها، ... و متدهای حرکت چرخ‌ها، روشن و خاموش کردن، عقب و جلو بردن، ... است و تمام اجزای اتومبیل با هم در ارتباطند. راننده از برخی از متدها مثل عقب و جلو بردن، روشن و خاموش کردن استفاده می‌کند و نیاز ندارد که از چگونگی عملکرد موتور یا شمع‌ها (کد متدها) برای حرکت اتومبیل آگاه باشد. لذا اگر عملکرد موتور اتومبیل تغییر کند در رفتار راننده برای استفاده از اتومبیل تأثیری ندارد.

 **نکته:** کپسوله‌سازی سبب می‌شود که اشیاء بدون آنکه از چگونگی عملکرد یکدیگر اطلاع داشته باشند با هم ارتباط برقرار کنند.

۲-۱-۳- چندریختی (Polymorphism): کمیتی است که به یک رابط امکان

می‌دهد تا برای یک کلاس عمومی (مفهوم کلاس را در همین فصل توضیح خواهیم داد)، از عملیات یکسانی استفاده کند. عمل خاص، توسط ذات حقیقی شیء تعیین می‌شود. یک مثال ساده از چندریختی در فرمان اتومبیل است. فرمان اتومبیل برای تمام اتومبیل‌ها بدون توجه به روشی که مورد استفاده قرار می‌دهند، یکسان است. فرمان برای اتومبیلی که به‌طور دستی کار می‌کند یا با نیروی برق یا هر چیز دیگری، عمل یکسانی را انجام می‌دهد. بنابراین، بعد از این که شما چگونگی عمل کردن را یاد گرفتید، می‌توانید فرمان هر نوع اتومبیلی را کنترل کنید. همین هدف می‌تواند در برنامه‌نویسی نیز اعمال شود. در برنامه‌نویسی شیء گرا می‌توان کلاس‌های عمومی را تعریف کرد مانند کلاس انسان که شامل تمام انسان‌ها (مرد و زن) می‌شود. انسان‌ها دارای مشخصه‌هایی مثل دست، پا، سر، ...، پوشش، احساسات، ... هستند. پوشش و احساسات و بسیاری از ویژگی‌های انسان‌ها در مرد و زن با هم تفاوت دارد در حالی که هر دو دارای این ویژگی‌ها هستند لذا کلاس مرد و کلاس زن زیر کلاس‌های کلاس انسان هستند. در مثال دیگر کلاس میوه‌ها را در نظر بگیرید که شامل همه میوه‌ها می‌شود و طعم از مشخصه‌های آن است. کلاس میوه‌ها شامل زیر کلاس‌های سیب، گلابی، موز، ... است که هر کدام طعم مخصوص به خود دارند.

 **نکته:** با استفاده از چند ریختی می‌توان برای متد تعریف شده در کلاس عمومی، پیاده‌سازی متفاوت در زیر کلاس‌ها داشت.

۳-۱-۳- وراثت: عملی است که یک شیء می‌تواند مشخصه‌های شیء دیگری را

به‌دست آورد. به‌همین دلیل، از مفهوم دسته‌بندی سلسله‌مراتبی پشتیبانی می‌کند. اگر درباره وراثت بیشتر فکر کنید، اطلاعات بیشتری راجع به دسته‌بندی سلسله‌مراتبی (از بالا به پایین) به‌دست خواهید آورد. به‌عنوان مثال، سیب قرمز بخشی از دسته‌بندی سیب است که آن هم بخشی از کلاس میوه‌هاست. میوه‌ها هم در کلاس بزرگ‌تری به‌نام غذا قرار دارند. کلاس غذا دارای مشخصات اصلی (خوراکی، پروتئین

و غیره) است که به طور منطقی به زیر کلاس‌های غذا اعمال می‌شود. علاوه بر این مشخصات، کلاس میوه دارای مشخصه‌های (آبدار، شیرین و غیره) است که آن را از سایر غذاها متمایز می‌کند. کلاس سیب نیز مشخصه‌های خاصی را برای یک سیب تعریف می‌کند که عبارتند از: رسیدن روی درخت و غیره. یک سیب قرمز، تمام خصوصیات کلاس‌های بالاتر را به ارث می‌برد و فقط مشخصه‌هایی که منحصر به فرد هستند را تعریف خواهد کرد.

بدون استفاده از وراثت، هر شیء به‌طور مجزا بایستی تمام مشخصه‌های خودش را تعریف کند. با استفاده از وراثت، شیء فقط نیاز به تعریف مشخصه‌هایی دارد که در داخل آن کلاس منحصر به فرد هستند. این سبب می‌شود که صفات عمومی را از پدرشان به ارث ببرند. بنابراین، مکانیزم وراثت به یک شیء امکان می‌دهد تا نمونه‌ خاصی از یک حالت عمومی تر باشد.

۲-۳- مازول‌های کلاس

برنامه‌نویسی شیء‌گرا برای مدل‌سازی مفاهیم و مسأله‌ها در دنیای واقعی به کار می‌رود. از اجزای اصلی برنامه‌نویسی شیء‌گرا کلاس است. تاکنون با کلاس‌های زیادی در ویژوال بیسیک آشنا شده‌اید. کلاس‌هایی مانند Form، Command Button، Text Box، List Box که هر کدام دارای مشخصه‌ها، متدها و رویدادهای خاص خود هستند و می‌توان به هر تعدادی که لازم است نمونه‌هایی از این کلاس‌ها را در برنامه استفاده کرد. به عنوان مثال می‌توان روی فرم، دو کادر متن و یک دکمه فرمان قرار داد که نمونه‌هایی از کلاس‌های کادر متن و دکمه فرمان هستند. نمونه‌های یک کلاس را شیء می‌گویند همان‌طور که به نمونه‌های انواع داده‌ها مثل Integer، String متغیر می‌گویند. به عبارت دیگر کلاس مجموعه تمام شیء‌های هم نوع است ولی شیء نمونه‌ای از یک کلاس است مانند سیب که نمونه‌ای از کلاس میوه‌ها است.

در فصل ۲ با نوع داده تعریف شده به وسیله کاربر (UDT) آشنا شدید. کلاس، شبیه یک نوع داده تعریف شده به وسیله کاربر است با این تفاوت که علاوه بر داشتن داده‌ها، کلاس دارای زیرروال‌هایی نیز می‌باشد.

کلاس‌ها شامل متغیرها و زیرروال‌ها هستند. متغیرها را اعضای داده‌ای (مشخصه) و زیرروال‌های کلاس را که می‌توانند از نوع زیر برنامه (SUB) یا تابع (Function) باشند، متد می‌گویند. بنابراین، کلاس، مجموعه‌ای از مشخصه‌ها و متدهاست.

به عنوان مثال به نوع داده Music که به صورت زیر تعریف شده است توجه کنید.

Type Music

Composer As String

Piece As String

End type

در برنامه روالی به نام Report نوشته شده که مقادیر عناصر نوع داده Music را در کادر پیام نمایش می دهد.

Dim my Music As Music

Private Sub Report ()

Dim Msg As String

Msg = "Composer: " & my Music.Composer & vbCrLf

Msg = Msg & "Piece: " & my Music.Piece

Msg Box Msg نمایش مقادیر در کادر پیام

End sub

اگر بخواهید روال Report به عنوان عنصری از نوع داده Music باشد باید آن را به صورت کلاس Music تعریف کنید.

۱-۲-۳- ایجاد ماژول کلاس: در ویژوال بیسیک، کلاس ها به صورت یک ماژول پیاده سازی می شوند و تمام اجزای کلاس در داخل ماژول تعریف می شوند و ماژول کلاس به پروژه اضافه می شود.

ماژول های کلاس (پرونده های .cls) اصول و اساس برنامه نویسی شیء گرا در ویژوال بیسیک است. ماژول های کلاس را می توان با یک نقشه ساخت خانه مقایسه کرد. دقیقاً مانند یک خانه که از طریق نقشه ساخته می شود، شیء های جدید نیز از طریق ماژول های کلاس ایجاد می شوند. مشخصه ها و متدهای این شیء های جدید را می توانید تغییر دهید. یک ماژول کلاس شبیه یک ماژول کد استاندارد (پرونده .bas) است، به همین دلیل هر دو دارای عملیاتی هستند که می توانند به وسیله ماژول های دیگری در داخل برنامه کاربردی مورد استفاده قرار گیرند.

● میدان دید داده های یک ماژول استاندارد تمام برنامه است. این بدین معنی است که این داده ها تا پایان دوره حیات برنامه وجود دارند (معتبر هستند).

● برای هر شیئی که از کلاس ایجاد می‌شود، داده‌های ماژول کلاس به وجود می‌آیند. هر داده‌ی شیء فقط در دوره‌ی حیات شیء وجود دارد (با ایجاد شیء، به وجود می‌آید و با از بین رفتن شیء، از بین می‌رود).

● متغیرهایی که به صورت Public در ماژول استاندارد تعریف می‌شوند، در هر جایی از پروژه قابل رؤیت هستند. ولی متغیرهایی که به صورت public در ماژول کلاس تعریف می‌شوند فقط با مراجعه به نمونه‌ی خاصی از کلاس یا شیء قابل دسترس هستند.



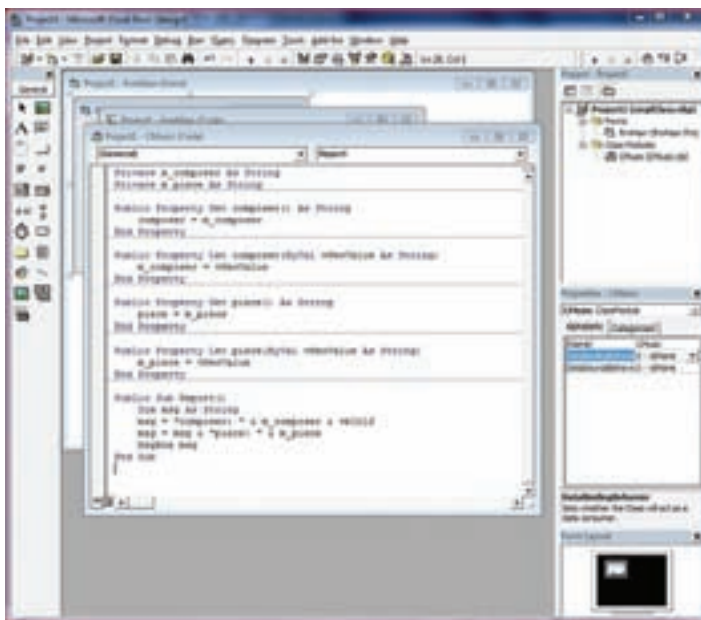
۱- پروژه جدیدی را ایجاد کرده، نام آن را Smpl Class. vbp قرار دهید. فرم پیش فرض را به FrmMain تغییر دهید.

۲- از منوی Project گزینه Add Class Module را انتخاب کنید.

۳- Class Module را انتخاب کرده، دکمه Open را کلیک کنید.

۴- در پنجره Properties مشخصه Name ماژول کلاس را CMusic قرار دهید و سپس

کلاس را در فایل به نام CMusic.cls ذخیره کنید (شکل ۳-۱).



شکل ۳-۱- ایجاد مدول کلاس

۲-۲-۳- اضافه کردن مشخصه‌ها به کلاس: کلاس CMusic دارای دو مشخصه

Composer و Piece از نوع String می‌باشد. برای این مشخصه‌ها در کلاس متغیرهایی در نظر بگیرید.

```
Private m_Composer AsString
```

```
Private m_Piece AsString
```

اجزای کلاس (مشخصه‌ها و متدها) می‌توانند به صورت Private یا Public تعریف شوند متغیرهایی که به صورت Private تعریف می‌شوند فقط به وسیله متدهای کلاس قابل استفاده هستند ولی متغیرهایی که به صورت Public تعریف می‌شوند علاوه بر متدهای کلاس، در سایر زیر برنامه‌های موجود در برنامه قابل دسترسی هستند. توصیه می‌شود که متغیرهای کلاس به صورت Private تعریف شوند تا فقط از طریق متدهای کلاس قابل دسترسی باشند.

در کلاس CMusic مقادیر واقعی مشخصه‌های Composer و Piece در متغیرهای m_Composer و m_Piece (متغیرهای عضو کلاس) قرار دارند و از آنجا که این متغیرها به صورت Private تعریف شده‌اند برای دسترسی به آنها (خواندن و تغییر مقدار آنها) روال‌های Get و Let به کلاس اضافه می‌شود.

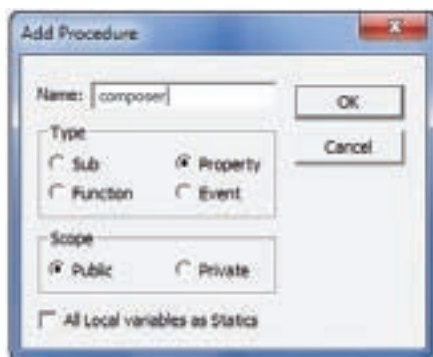
برای اضافه کردن این روال‌ها مراحل زیر را انجام دهید.

۱- پنجره Code ماژول کلاس را باز کنید.

۲- از منوی Tools گزینه Add Procedure را انتخاب کنید.

۳- گزینه Property را در کادر محاوره‌ای Add Procedure انتخاب کنید و مشخصه

Composer را در کادر متن Name وارد کنید. روی Ok کلیک کنید (شکل ۲-۳) متدهای Get Composer و Let Composer به پنجره کد اضافه می‌شود.



شکل ۲-۳- اضافه کردن مشخصه به کلاس


```
Public Property Get Composer ( ) As Variant
```

```
End Property
```

```
Public Property let Composer (Byval vNew Value As Variant)
```

```
End Property
```

روال Get برای خواندن مقدار مشخصه Composer است لذا در بدنه روال دستور زیر را

بنویسید.

```
Composer = m_Composer
```

این دستور مقدار واقعی مشخصه Composer را که در متغیر m_Composer قرار دارد در مشخصه Composer قرار می دهد.

روال Let برای مقداردهی کردن مشخصه Composer است لذا متغیر m_Composer را با آرگومان ورودی این روال مقداردهی کنید.

```
m_Composer = vnewValue
```

توجه کنید که ویژگیال بیسیک تمام مشخصه ها را از نوع Variant در نظر می گیرد. با توجه به این که مشخصه Composer از نوع String است کد را اصلاح کنید.
کد برای مشخصه Composer به صورت زیر تکمیل می شود:

```
Public Property Get Composer ( ) As String
```

```
Composer = m_Composer
```

```
End Property
```

```
Public Property Let Compose (By Val Vnew Vlaue As String)
```

```
m_Composer = Vnew Value
```

```
End Property
```

۴- مشخصه Piece را مانند مشخصه Composer به کلاس اضافه کنید.

```
Public Property Get Piece ( ) As String
```

```
Piece = m_piece
```

```
End Property
```

Public Property Let Piece (By Val vNewValue As String)

m_piece = piece

End property

۳-۲-۳- اضافه کردن متدها به کلاس : کلاس مجموعه‌ای از مشخصه‌ها و متدها می‌باشد.

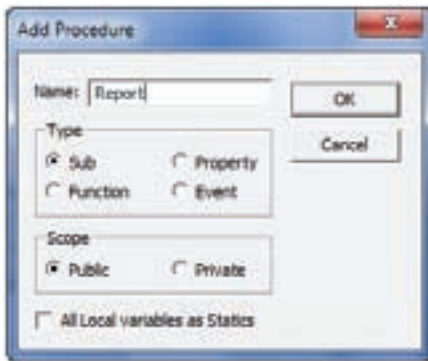
کلاس CMusic دارای متد Report برای نمایش مشخصه‌های کلاس در کادر پیام است. برای

اضافه کردن این متد به کلاس CMusic مراحل زیر را انجام دهید.

۱- پنجره کد ماژول کلاس را باز کنید.

۲- از منوی Tools گزینه Add Procedure را انتخاب کنید.

۳- در کادر محاوره‌ای Add Procedure



نوع Sub را انتخاب کرده و Report را در کادر

متن Name وارد کرده، سپس Ok را کلیک کنید.

(شکل ۳-۳)

شکل ۳-۳- اضافه کردن متد به کلاس

کد زیر به پنجره اضافه می‌شود.

Public Sub Report ()

End Sub

۴- متد Report را تکمیل کنید.

Public Sub Report ()

Dim msg As String

msg = "Composer: " & m_composer & vbCrLf

msg = msg & "Piece: " & m_piece

Msg Box msg

End Sub

۴-۲-۳- ایجاد شیء از یک کلاس: پس از ایجاد کلاس، می‌توان نمونه‌ای از آن را به‌عنوان

شیء در برنامه استفاده کرد. از دو روش برای ایجاد شیء می‌توان استفاده کرد:

۱- استفاده از AsNew:

```
Public object_name As New Class_name
```

● As New: کلمه کلیدی برای ایجاد شیء

● Object_name: نام شیء که مشابه نام متغیر در اعلان متغیر می‌باشد.

● Class_name: نام کلاسی که شیء از نوع آن کلاس ایجاد می‌شود (مانند نوع داده در اعلان

متغیر می‌باشد)

```
Public MyMusic As New CMusic
```

شیء MyMusic را از نوع کلاس CMusic ایجاد می‌کند.

۲- استفاده از دستور انتساب Set:

```
Dim object_name As Class_name
```

```
Set object_name = New Class_name
```

در خط اول همانند اعلان متغیر، شیء به نام object_name از نوع کلاس Class_name

را تعریف می‌کند.

خط دوم مانند دستور مقداردهی به متغیر می‌باشد.

● Set: کلمه کلیدی برای مقداردهی به اشیاء

● New: کلمه کلیدی برای ایجاد یک نمونه از کلاس

● object_name: نام شیء

● Class_name: نام کلاس

```
Dim MyMusic As CMusic
```

```
Set MyMusic = New CMusic
```



روی فرم دکمه فرمان گذاشته و Caption آن را Report قرار دهید. رویداد Click دکمه سبب

نمایش مشخصات قطعه موزیک در کادر محاوره‌ای می‌شود.

در پنجره کد فرم برنامه زیر را بنویسید.

```
Public MyMusic As New CMusic
```

```
Private Sub Form_Load ( )  
    MyMusic.Composer = "Banan"  
    MyMusic.Piece = "Elahenaz"  
End Sub
```

```
Private Sub CmdReport_Click ( )  
    MyMusic.Report ( )  
End Sub
```

کد زیر مثال بالا را با استفاده از دستور انتساب Set نشان می‌دهد.

```
Public Mymusic As CMusic  
Private Sub Form_Load ( )  
    Set MyMusic = New CMusic  
    MyMusic.Composer = "Banan"  
    MyMusic.Piece = "Elahenaz"  
End Sub
```

```
Private Sub CmdReport_Click ( )  
    MyMusic.Report ( )  
End Sub
```

تمرین: مثال فوق را به گونه‌ای تغییر دهید که مشخصات قطعه موزیک را با استفاده از دو کادر متن به وسیلهٔ کاربر دریافت کند.



روال Let که برای تغییر مشخصه‌های کلاس نوشته می‌شود در هنگام استفاده از کلاس به کار نمی‌رود و از خود مشخصه استفاده می‌شود.

```
MyMusic.Composer = "Banan"
```

شیء MyMusic برای مقداردهی مشخصهٔ Composer در دستور بالا از روال Let

استفاده می‌کند.

تمرین: با قرار دادن breakpoint روی روال Let در مثال فوق بررسی کنید



این روال در چه زمانی فراخوانی می‌شود؟

هنگام نوشتن کدی که از شیء‌ها استفاده می‌کند، بهتر است حافظه‌ای مورد استفاده به وسیله شیء‌ها را پس از پایان کار با آن‌ها آزاد سازید. بعد از این که کارتان با یک شیء به پایان رسید از عبارت Set برای مقداردهی Nothing به متغیر شیء استفاده کنید. مثال زیر، حافظه مورد استفاده به وسیله شیء MyMusic را آزاد می‌کند:

Set MyMusic = Nothing

۵-۲-۳- ایجاد رویدادها: یک شیء اعلان می‌کند که بعضی از عملیات از طریق به کارگیری

رویدادها ارائه می‌شوند. مشخصه‌ها و متدها، رابط‌های درونی هستند زیرا آن‌ها به خارج شیء ارتباطی ندارند. به طور واضح، رویدادها رابط‌های بیرونی هستند، زیرا آن‌ها درون شیء مقداردهی اولیه و خارج از شیء مدیریت می‌شوند. به عنوان مثال، هنگامی که مشخصه Caption یک فرم را تغییر می‌دهید، شیء فرم شامل کدی است که این تغییر را اعمال می‌کند و شما این کد را مشاهده نمی‌کنید. ولی روال‌های رویداد را خودتان کدنویسی می‌کنید.

ماژول‌های کلاس شامل دو رویداد درونی هستند: Initialize و Terminate.

● رویداد Initialize هنگامی که نمونه‌ای از کلاس ایجاد و قبل از این که مشخصه‌ای مقداردهی شود، رخ می‌دهد. هنگام نوشتن یک ماژول کلاس، رویداد Initialize را برای مقداردهی اولیه هر داده‌ای که به وسیله کلاس مورد استفاده قرار می‌گیرد، به کار گیرید. هم چنین می‌توان از این رویداد برای بارگذاری فرم‌های مورد استفاده به وسیله کلاس نیز استفاده کرد.

● هنگامی که متغیر شیء خارج از میدان دید باشد یا با مقدار Nothing مقداردهی شود، رویداد Terminate رخ می‌دهد. هنگامی که ماژول کلاس را می‌نویسید، از این رویداد برای ذخیره اطلاعات، unload کردن فرم‌ها یا اجرای وظایفی که هنگام خاتمه کلاس رخ خواهند داد، استفاده کنید.



کلاس CSpecialDate برای نمایش تاریخ تعریف شده است این کلاس دارای مشخصه Date

Format برای تعیین قالب نمایش تاریخ می‌باشد. کد ماژول کلاس به صورت زیر است.

Option Explicit

Private m_Format As String

اعلان مشخصه کلاس

متد Get برای خواندن مشخصه DateFormat

Public Property Get DateFormat () As String

DateFormat = m_Format

End Property

متد Let برای نوشتن (مقداردهی کردن) مشخصه DateFormat

Public Property let DateFormat (Byval vNew Value As String)

m_Format = vNewValue

End Property

متد GetSpecialDate برای نمایش تاریخ با فرمت m-Format

Public Function GetSpecialDate () As String

GetSpecialDate = Format (Now, m_Format)

End Function

روال رویداد Initialize

Private Sub Class-Initialize ()

m_Format = "yy-mm-dd"

End Sub

با توجه به کد ماژول کلاس CSpecialDate، این کلاس علاوه بر مشخصه DateFormat،

دارای متد GetSpecialDate برای نمایش تاریخ با فرمت تعیین شده در مشخصه DateFormat

است. این متد به صورت Function نوشته شده است.

روال رویداد Initialize برای این کلاس نوشته شده است و در آن مشخصه DateFormat با

مقدار اولیه «yy-mm-dd» برای فرمت نمایش تاریخ مقداردهی شده است.

توجه کنید که مقدار واقعی مشخصه DateFormat در متغیر m_Format قرار دارد. این

متغیر به صورت Private تعریف شده است، لذا تنها روال‌های موجود در ماژول کلاس به آن دسترسی

دارند و نمونه‌هایی که از کلاس ایجاد می‌شوند نمی‌توانند مستقیماً از این متغیر استفاده کنند و به جای آن از خود مشخصه DateFormat استفاده می‌کنند که سبب می‌شود کلاس از متدهای Let و Get این مشخصه برای تغییر دادن و خواندن مشخصه استفاده کند.

برای استفاده از کلاس CSpecialDate روی فرم برچسب Lable1 و دکمه فرمان CmdShowDate را قرار داده و کد زیر نوشته شده است؛

Privat Sub CmdShowDate_Click ()

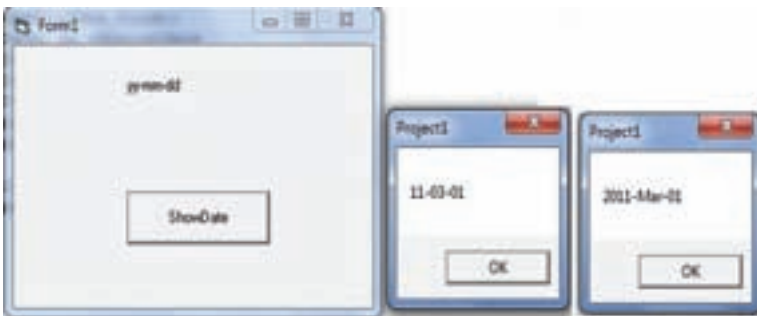
```

Dim oSpecDate1 As cSpecialDate      'اعلان متغیری از نوع کلاس'
Dim oSpecDate2 As cSpecialDate      'اعلان متغیری از نوع کلاس'
Set oSpecDate1 = New cSpecialDate   'ایجاد شیء اول (ospecDate1)'
Set oSpecDate2 = New cSpecialDate   'ایجاد شیء دوم (ospecDate2)'
Label1.Caption = oSpecDate1.DateFormat /DateFormat 'نمایش مقدار مشخصه'
MsgBox oSpecDate1.GetSpecialDate     'نمایش تاریخ در کادر پیام'
oSpecDate2.DateFormat = "yyyy-mm-dd" 'تغییر مقدار مشخصه DateFormat'
MsgBox oSpecDate2.GetSpecialDate     'نمایش تاریخ در کادر پیام'

```

End Sub

در رویداد Click دکمه فرمان دو شیء به نام‌های oSpecDate1 و oSpecDate2 از نوع کلاس CSpecialDate ایجاد شده است. برای شیء oSpecDate1 مقدار مشخصه DateFormat را تغییر نداده و تاریخ را با همان قالب تعیین شده در روال رویداد Initialize کلاس نمایش می‌دهد، ولی برای شیء oSpecDate2 مقدار مشخصه DateFormat را تغییر داده سپس تاریخ را با قالب جدید نمایش می‌دهد (شکل ۳-۴).



شکل ۳-۴- فرم و کادرهای پیام مثال ۳-۳

خودآزمایی

- ۱- مفاهیم زیر را توضیح دهید :
کپسوله‌سازی - چندریختی - وراثت
- ۲- هدف از ماژول کلاس در ویژوال بیسیک، چیست؟
- ۳- اگر یک مدل شیء مبتنی بر دوچرخه ایجاد کرده باشید، چگونه آن را پیاده‌سازی خواهید کرد؟
- ۴- ماژولی برای کلاس مستطیل بنویسید که دارای متدهایی برای محاسبه محیط و مساحت مستطیل باشد؟
- ۵- کلاس CTime را ایجاد کنید که دارای متغیرهایی برای ذخیره ساعت، دقیقه و ثانیه است و متد ShowTime که زمان را با توجه به مقدار متغیرهایش نمایش می‌دهد.