

VisualBasic对象使用 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/137/2021\\_2022\\_VisualBasi\\_c97\\_137576.htm](https://www.100test.com/kao_ti2020/137/2021_2022_VisualBasi_c97_137576.htm)

使用对象 在 Visual Basic中创建应用程序之际，也就是和对象打交道之时了。可以使用 Visual Basic 提供的对象，比如控件、窗体和数据访问对象。也可以在一个 Visual Basic 应用程序内部控制另一个应用程序对象。甚至还可建立自己的对象，定义它们的属性和方法。对象是什么 对象是代码和数据的组合，可以作为一个单位来处理。对象可以是应用程序的一部分，比如可以是控件或窗体。整个应用程序也是一个对象。下表列出了在 Visual Basic 中可能用到的几种类型的对象：

对象从何而来 Visual Basic 中的每个对象都是用类定义的。用饼干模子和饼干之间的关系作比喻，就会明白对象和它的类之间的关系。饼干模子是类。它确定了每块饼干的特征，比如大小和形状。用类创建对象。对象就是饼干。

下面再用两个例子进一步说明 Visual Basic 中类与对象之间的关系。

- 1.在 Visual Basic 的“工具箱”上，控件代表类。直到在窗体上画出这些被称作控件的对象为止，它们实际上并不存在。在创建控件之时也就是在复制控件类，或建立控件类的实例。这个类实例就是应用程序中引用的对象。
- 2.在设计时操作的窗体是类。在运行时，Visual Basic 建立窗体的类实例。“属性”窗口显示 Visual Basic 应用程序中的对象的类和 Name 属性，如图 5.8 所示。把对象看作是类的原原本本的复制品，从而建立所有对象。一旦它们以单个对象的形式存在，属性就可改变。例如，如果在窗体上画了三个命令按钮，则每个命令按钮对象都是命令按钮类的实例。每个对象都

具有一组由类定义的公共的特征和功能（属性、方法和事件）。但是，每个对象都有自己的名字，都能分别设置成有效或无效，都能放在窗体的不同位置，等等。为简单起见，本章之外的大部分内容将不过多引用对象的类。例如，只要记住术语“ListBox 控件”意味着“列表框类的一个实例”。用对象能做什么 对象可提供现成代码，省却书写麻烦。例如，可以自己创建打开文件和保存文件的对话框，但实际上大可不必。取而代之的是利用 Visual Basic 提供的 CommonDialog 控件（一个对象）。虽然用户也能撰写日程管理和资源管理的程序，但也大可不必。在此亦可使用 Microsoft Project 提供的 Calendar、Resources 和 Task 对象。Visual Basic 可以组合来自其它来源的对象 Visual Basic 提供了把来自不同资源的对象组合起来的工具。现在可把 Visual Basic 的各种强有力的特性以及支持自动化（先前以 OLE 自动化闻名）的应用程序结合起来，建立定制的解决方法。自动化是部件对象模式(COM)的一个特性，它是应用程序使用的工业标准，用来陈列对象以开发工具和其它应用程序。可把 Visual Basic 内部的控件结合在一起，也可使用其它应用程序提供的对象。考虑把下列对象放入 Visual Basic 窗体：1.Microsoft Excel Chart 对象 2.Microsoft Excel Worksheet 对象 3.Microsoft Word Document 对象 可用这些对象建立如图 5.9 所示的支票簿应用程序。由于不必书写代码，重建已由 Microsoft Excel 和 Word 等对象提供的功能，因而节省很多时间。使用对象初步 Visual Basic 对象支持属性、方法和事件。在 Visual Basic 中，称对象的数据（设置和属性）为属性，称各种可在对象上操作的过程为方法。事件是可被对象识别的动作，例如单击鼠标和按下键盘键

，还可编写代码来响应事件。改变对象的属性就可改变对象的特性。用收音机打比方，收音机的一个属性是音量。用 Visual Basic 的行话来说，就是收音机有个“Volume”属性，改变其值就可调节音量大小。假定收音机的音量值可设置在 0 到 10 之间。如果能够通过 Visual Basic 控制收音机，则可在一个过程中写代码，把“Volume”属性值从 3 提高到 5，使声音更响一些：`Radio.Volume = 5` 除了属性以外，对象还有方法。方法和属性都是对象的一部分。一般说来，方法就是要执行的动作，而属性就是要设置或检索的特性。以拨号打电话为例。可以说电话有一个“拨号”方法，拨一个 7 位电话号码的语法就是：`Phone.Dial 5551111` 对象还有事件。当对象的某方面有变动时就触发了事件。例如，收音机可能有“VolumeChange”事件。电话可能有“Ring”事件，等等。

用属性控制对象 个别属性随着可以设置和取得它们的值的时间不同而不同。有的属性可在设计时设置。可在“属性”窗口设置这些属性的值而无需编写任何代码。而有的属性在设计时是不可用的；因此，这些属性只有通过代码在运行时设置。在运行时可以设置并可获得值的属性叫做读写属性。在运行时只能读取的属性叫做只读属性。设置属性值 在想改变对象的外观或特性时设置属性的值。例如，通过改变 TextBox 控件的 Text 属性，就可以改变文本框的内容。用下列语法设置属性值：`object.property = expression` 下面是设置属性的语句：`Text1.Top = 200` 设置 Top 属性为 200 缇（twips）。`Text1.Visible = True` 显示文本框。`Text1.Text = "hello"` 在文本框中显示"hello"。读取属性值 想要在代码执行附加动作（例如给另一个对象赋值）之前得知对象的状态，这时就要读取属

性值。例如，在运行代码之前能够返回 TextBox 控件的 Text 属性值，以确定文本框的内容，这里，代码可能改变值。在大多数情况下可以用以下语法获得属性值：`variable = object.property` 属性值可以作为较复杂的表达式的一部分，而不必将属性赋予变量。下面的代码计算控件数组中的一个新成员的 Top 属性，它等于前一个成员的 Top 属性加上 400。

```
Private Sub cmdAdd_Click () [语句] optButton (n).Top =  
optButton (n-1).Top 400 [语句] End Sub
```

提示 如果不只一次使用一个属性值，而且将这个值存储到一个变量中，则代码执行起来会更快。用方法来执行动作 方法能够影响属性值。例如在用收音机打比方的例子中，SetVolume 方法改变了 Volume 属性。与此类似，在 Visual Basic 中列表框具有 List 属性，而 Clear 和 AddItem 方法可以改变这一属性。在代码中使用方法 在代码中使用方法时如何书写语句，这取决于该方法要求多少参数，以及是否返回一个值。如果方法不要求参数，则用以下语法编写代码：`object.method` 下例中，用 Refresh 方法重画图片框：`Picture1.Refresh` 强迫重画控件。有些方法，如上面的 Refresh，既无参数又不返回值。如果方法要用多个参数，就用逗号将它们分开。例如，Circle 方法就要用代表窗体上圆的位置、半径和颜色的参数：画一个半径为 1200 缇的蓝色圆。`Form1.Circle (1600, 1800), 1200, vbBlue` 如果要保存方法的返回值，就必须把参数用括号括起来。例如，GetData 方法从剪贴板返回一张图片：`Picture = Clipboard.GetData (vbCFBitmap)` 如果没有返回值，则参数不会出现在括号中。例如，AddItem 方法没有返回值。`List1.AddItem "yourname"` 在列表框中添加 "yourname"。详细

信息 关于 Visual Basic 对所有方法提供的语法和参数，请参阅《VisualBasic 6.0语言参考手册》。对象之间的关系 在同一窗体上放置两个命令按钮时，它们是两个具有不同 Name 属性设置值（Command1 和 Command2）的分离的对象，但都分享同一个类命令按钮。它们也共享这样一个特征，即它们在同一窗体上。在本章前面已看到，窗体上的控件也包含在窗体中。这就把控件放在一个层次上。要引用控件就必须先引用窗体，就如同打电话拨号一样，首先要拨国家/地区号码或区域号码，然后再拨具体的电话号码。这两个命令按钮也共享这样一个特征，即它们都是控件。所有控件都具有公共特性，使它们不同于窗体以及 Visual Basic 环境中的其它对象。以下章节中将说明 Visual Basic 如何用集合使相关的对象分组。对象的层次 对象层次提供了一种组织结构，它决定了对象间的相互关系以及访问它们的方法。在大多数情况下不必考虑 Visual Basic 对象的层次。但是：1.在操作其它应用程序的对象时，应当熟悉那个应用程序的对象层次。关于定位对象层次的信息，请参阅第十章“用部件编程”。2.在使用数据访问对象时，应当熟悉数据访问对象层次。在 Visual Basic 中，一个对象包含其它对象的一般情况时有发生。下面我们要详细讨论。使用对象集合 集合对象有自己的属性和方法。对象集合中的对象作为集合的成员被引用。集合中的每个成员从 0 开始顺次编号；这就是成员的索引号。例如，控件集合包含已给定窗体上的所有控件，如图 5.10 所示。如果对集合中的所有对象都必须执行相同操作，则可用集合简化代码。例如，下列代码遍历控件集合，并在列表框中列出每个成员的名字。 Dim MyControl as Control For Each MyControl In

Form1.Controls 把每个控件的名字送到列表框。 List1.AddItem MyControl.Name Next MyControl 应用集合成员的属性与方法有两种通用方法可用来对集合对象成员寻址： 1.指定成员的名称。以下两个表达式等价： Controls ("List1") Controls!List1 2.利用成员的索引号： Controls (3) 一旦能够从整体上对全体成员寻址，而且还能够个别地对单个成员寻址，就可用下面随便一种方式应用属性和方法： 将 ListBox 控件的 Top 属性设置为 200。 Controls!List1.Top = 200 - 或者 - Dim MyControl as Control For Each MyControl In Form1.Controls () 将每一个成员的 Top 属性设置为 200。 MyControl.Top = 200 Next MyControl

包含其它对象的对象在 Visual Basic 中，一些对象包含其它对象。例如，窗体通常包含一个或数个控件。把一个对象当作其它对象的容器，这样做的好处在于，在代码中引用容器即可明了要使用哪个对象。例如，图 5.11 说明在一个应用程序中可有两个不同的窗体，一个用来输入应付帐交易，另一个用来输入应收帐交易。两个窗口可以有名为 IstAcctNo 的列表框。通过引用包含列表框的窗体就可准确指定要使用的列表框： frmReceivable.IstAcctNo.AddItem 1201 - 或者 - frmPayable.IstAcctNo.AddItem 1201

Visual Basic 中的公共集合在 Visual Basic 中一个对象包含其它对象的一般情况时有发生。下表简要叙述 Visual Basic 中最常用的集合。在 Visual Basic 中也可实现对象控制。详细信息 关于对象容器的详细信息，请参阅第八章“再论编程”中的“使用集合替代数组”。关于打印机集合的信息，参阅第十二章“使用文本和图形”。关于窗体集合和控件集合的详细信息，请参阅《Visual Basic 6.0 语言参考手册》。容器属性在窗体内，可用 Container 属性改

变对象的容器。以下控件可包含其它控件：1.Frame 控件  
2.PictureBox 控件 3.ToolBar 控件（仅对专业版和企业版而言）  
如何在窗体上将命令按钮从一个容器移动到另一个容器，该例对此作了演示。打开新工程，并在窗体上画出 Frame 控件、PictureBox 控件和CommandButton 控件。下列代码在窗体单击事件中，这些代码使计数器变量增加，并通过 SelectCase 循环使命令按钮在容器之间移动。

```
Private Sub Form_Click ()  
Static intX as Integer  
Select Case intX  
Case 0  
Set Command1.Container = Picture1  
Command1.Top= 0  
Command1.Left= 0  
Case 1  
Set Command1.Container = Frame1  
Command1.Top= 0  
Command1.Left= 0  
Case 2  
Set Command1.Container = Form1  
Command1.Top= 0  
Command1.Left= 0  
End Select  
intX = intX + 1  
End Sub
```

详细信息 请参阅《Visual Basic 6.0语言参考手册》中的“Container 属性”。对象之间的通讯除了在 Visual Basic 中使用和创建对象外，还可与其它应用程序通讯并在自己的应用程序中操作其它应用程序的对象。在应用程序之间共享数据的能力是 Microsoft Windows 操作系统的关键性能之一。有了 Visual Basic，与其它应用程序进行通讯变得极为灵活。详细信息 关于使用其它应用程序的对象及其通讯的详细信息，请参阅第十章“用部件编程”。100Test 下载频道开通，各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)