

第 2 章 对象及其操作

Visual Basic 是面向对象的编程语言，它改变了传统意义上的编程机制，采用了“事件驱动”机制。在 VB 中不仅提供了大量的控件对象，而且还提供了创建自定义对象的方法和工具，为开发应用程序带来了方便。因此，对于一个学习 VB 编程的人来说，了解对象及其特点的相关概念是必要的。这一章将讨论 VB 中最基本的两种对象，即窗体和控件。

2.1 对象

对象是 VB 程序设计的核心。对象具有属性、方法和事件三个要素。在这一节里，将主要介绍有关对象的基本概念、对象的三要素以及设置等知识。

2.1.1 对象的概念

对象可以是一组代码和数据的集合，用来描述自然的或抽象的实体，也可以是一切在编程中可操纵的实体。从广义上来讲，包括窗体中的按钮、标签、文本框等控件，甚至包括屏幕、打印机、扫描仪等计算机的外接设备。

VB 中所说的对象，通常是指狭义上的对象，即窗体和控件。窗体即为程序设计阶段程序员操作的界面平台；而在程序运行阶段，窗体则转化为用户与应用程序交互的平台。控件是 VB 或用户预先定义好的、以图标方式存放在工具箱中、用于完成程序设计中特定功能的工具。

在 VB 中，对象分为两种。一种是系统设计好的，称为预定义对象。用户可以直接调用或对其进行操作。另一种是由用户自定义的，称为自定义对象。

对象具有属性、方法和事件三个要素。建立了一个对象，其操作需要通过与该对象相关联的属性、方法和事件来进行。其中，属性是对象的参数，由一段预先编写的子程序代码组成。事件是由系统设置好的、能够被对象识别的动作。方法是由系统提供的用来完成特定操作的子过程和函数。我们可以把属性看成一个对象的性质，把方法看成对象要执行的操作，把事件看作对象的响应。

VB 中的“对象”与“面向对象程序设计”中的“对象”是一样的。只是在使用 C++ 这类面向对象的语言时，必须自己创建和设计“对象”，而在 VB 中，这些对象可以直接被使用。因此，设计 VB 应用程序就显得相当轻松。

1. 对象属性

在日常生活中，如订购的对象为家具城的棕色木茶几，则只需要在家具城寻找颜色属性为棕色、材料属性为木质的茶几即可。同理，可以订购黑色的玻璃茶几等。可见，属性是对象的特性，不同的对象有不同的属性。在 VB 中，对象常见的属性有 Name（名称）、Color（颜色）、Font（字体）、Enabled（有效）等。

对象的属性可以通过属性窗口来修改，也可以通过代码来修改，典型的对象属性设置语法结构如下：

对象名.属性名称=属性值

具体的设置方法见 2.1.3 节。

2. 对象事件

事件 (Event), 是一种加在对象上的“作用”, 该作用是由 VB 预先设置好的、能够被识别的动作。例如, 在对象上 Click (单击) 时, 该单击操作就是一种加在对象上的“事件”。此外, 在 VB 中常见的事件还有: DblClick (双击)、KeyDown (按下按钮)、KeyUp (松开按钮)、MouseMove (移动鼠标) 等。

为了响应对象上的这些事件以完成某种特定的操作, 程序设计人员必须预先编写好一段独立的程序代码, 这样的程序代码称为事件过程。每个对象可以识别一个或者多个事件, 因此可以使用一个或多个事件过程对用户或系统的事件做出响应。在程序设计中, 根据应用程序的需要, 编写相应的事件过程是应用程序设计的主要任务之一。

3. 对象方法

当设置对象的某些属性时, 其实就是将某些数据加入对象, 而事件过程是从外面加到对象身上的一段程序。对象原来内含的函数或过程叫做“方法”。

对象的“属性”或“事件过程”都可以重新设置或修改, 但是“方法”的内容却是固定、不能修改的, 而且是看不见的。事实上, 用户也只能“调用”它, 即运行对象内含的这些固定功能的程序。

对象方法的调用格式如下:

对象名称.方法名称[参数]

使用“方法”来控制 VB 中的某个对象, 实质上是运行该对象的某个内部函数或过程, 例如:

`Button1.Enabled`

以上代码将激活 Button1 控件, 使 Button1 控件有效。至于让 Button1 控件从无效状态变到有效状态的细节, 程序设计人员无须知道。VB 提供了大量的“方法”, 有些“方法”可以适用于多种甚至所有类型的对象, 而有些“方法”只能适用于少数几种对象。

2.1.2 对象的建立和编辑

1. 对象的建立

在窗体上建立对象的步骤如下:

- (1) 将鼠标定位在工具箱内要制作控件对象对应的图标上, 单击选择。
- (2) 将鼠标移动到窗体所需的位置处, 按住左键拖拽到所需的大小后释放鼠标。

建立对象更方便的方法是直接在工具箱双击所需的控件图标, 则立即在窗体出现一个大小为默认尺寸的对象框。

2. 对象的选定

要对某对象进行操作, 只要单击欲操作的对象就可选中, 这时选中的对象出现八个方向的控制柄。

若要同时对多个对象进行操作, 则要同时选中多个对象, 有两种方法:

- (1) 拖动鼠标指针, 将欲选定的对象包围在一个虚框内即可。
- (2) 先选定一个对象, 按下 Ctrl 键, 再单击其他要选定的控件。

3. 复制或删除对象

(1) 复制对象。选中要复制的对象, 单击工具栏的“复制”按钮, 再单击“粘贴”按钮, 这时会显示是否要创建控件数组的对话框, 选择“否”则复制了标题相同而名称不同的对象。

（2）删除对象。选中要删除的对象，然后按 Del 键。

4. 对象的命名

每个对象都有自己的名字，有了它才能在程序代码中引用该对象。建立的控件都有默认的名字，诸如 Form1、Form2 之类的窗体默认名等。用户也可在属性窗口通过设置 Name（名称）属性来给对象重新命名，名字必须以字母或汉字开头，由字母、汉字和数字组成，长度 ≤255 个字符，其中可以出现下划线。

2.1.3 对象属性设置

对象的属性设置在程序设计阶段或程序运行阶段都可以进行。

在程序设计阶段，可以通过 VB 开发环境中的属性窗口来设置对象的属性，不需要编写任何代码。对象的一些外观属性，在属性窗口设置了相应的值后，在窗体设计窗口中立即可以预览到设置效果。属性窗口主要用来设置对象属性的初始值和一些在整个程序运行过程中不进行修改的属性。

在程序运行阶段，可以通过程序中的代码来设置对象的属性值，从而实现在程序运行时改变对象属性。

在属性窗口中设置对象的属性，首先必须选择要设置的对象，然后激活属性设置窗口。

以下是几种属性窗口激活的方法：

- 执行“视图”菜单中的“属性窗口”命令。
- 单击工具栏上的“属性窗口”按钮。
- 按 F4 键。

属性的设置方法一共有 3 种，下面分别来介绍这 3 种方法的具体使用。

1. 直接键入属性值

对象的有些属性，如 Caption（标题）、Text（文本内容）都必须由用户输入。在建立该对象时，VB 可能为其设置好了默认值。为了提高程序的可读性，最好为其赋一个具有特定意义的名称，可以在属性窗口中键入新属性值。

例如，用户在窗体上设置了两个 Label 控件，如图 2-1 所示。其中，窗体的标题显示为 Form1 字样，两个 Label 控件显示为 Label1 和 Label2 字样，这是因为窗体的 Caption（标题）属性默认值为 Form1，而两个 Label 控件的 Caption 属性默认值分别为 Label1 和 Label2。

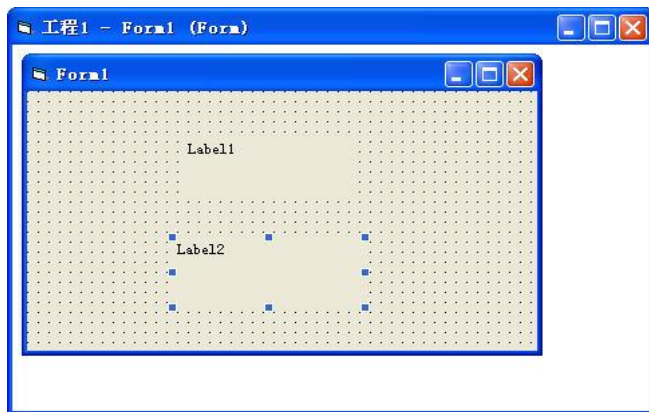


图 2-1 在窗体上放置两个 Label 控件

如果要修改图 2-1 中窗体的 Caption 属性，用户可以按照如下步骤操作：

(1) 单击工具栏上的“属性窗口”按钮，出现如图 2-2 所示的属性窗口。

(2) 单击属性窗口“对象框”右侧的下拉箭头，弹出的下拉列表中将显示所有的对象名称和类别。

(3) 搜索“Form1 Form”项（这里 Form1 为对象名称，Form 为对象类别）。

(4) 确定了 Form1 对象后，在属性窗口的左侧搜索 Caption 属性，在属性窗口右侧的“属性设置”框中输入“改变 Caption 属性示例”。

用同样的方法，将 Label1 对象和 Label2 对象的 Caption 属性修改为“控件 1”和“控件 2”，最后结果如图 2-3 所示。



图 2-2 属性设置窗口



图 2-3 修改对象的 Caption 属性

2. 在下拉列表中选择属性值

对象的有些属性取值是有限的，对于这样的属性，可以在下拉列表中选择所需要的属性值，例如，为了设置窗体对象的 BorderStyle（边界类型）属性，可以按如下步骤操作（如图 2-4 所示）：

(1) 启动 VB，激活属性窗口。

(2) 在属性窗口中找到 BorderStyle，并单击该属性条。其右侧一列显示 BorderStyle 的当前属性值，同时在右端出现一个向下的箭头。

(3) 单击右端的箭头，将显示该属性可能取值的下拉列表。

(4) 单击列表中的某一项，即可把该项设置为 BorderStyle 属性的值。

3. 利用对话框设置属性值

对于和图形（Picture）、图标（Icon）或者字体（Font）有关的属性，设置框的右侧会显示省略号，单击该省略号，屏幕上将显示一个对话框，可以利用这个对话框设置所需要的属性。例如，在属性列表中找到 Font 属性，如图 2-5 所示，单击 Font 属性右侧的省略号，将显示如图 2-6 所示的“字体”对话框，在这个对话框中设置对象的 Font 属性，包括字体、字体样式、大小及效果等。



图 2-4 从下拉列表中选择属性值



图 2-5 用对话框设置属性



图 2-6 “字体”对话框

2.1.4 对象属性的读取

对象属性值的读取，主要是为了在执行某个操作之前得到对象的状态，必要时可以对其进行修改。如对一个 Label 控件，可以设置两个变量来读取它的宽和高，再判断一下是否符合要求。如果不符合要求，可以将其重新设置。用代码读取对象属性的典型语法结构为：

变量=对象名.属性名

例如，下列语句就是将当前 Label1 控件的高度值赋给变量 a：

```
a=Label1.Height
```

属性值也可以作为表达式的一部分。以下代码是将 Label 控件的高度值在原来的基础上调低一点：

```
Label1.Height=Label1.Height-10
```

表达式中，10 的单位为 twip。

2.1.5 对象的事件、事件过程和事件驱动

1. 事件

对于对象而言，事件就是发生在该对象上的事情。在 VB 中，系统为每个对象预先定义好了一系列的事件。例如，单击 (Click)、双击 (DoubleClick)、改变 (Change)、获取焦点 (GotFocus)、键盘按下 (KeyPress) 等。

2. 事件过程

当在对象上发生了事件后，应用程序就要处理这个事件，而处理的步骤就是事件过程。它是针对某一对象的过程，并与该对象的一个事件相联系。VB 应用程序设计的主要工作就是为对象编写事件过程中的程序代码。事件过程的形式如下：

```
Sub 对象名_事件 ([参数列表])
```

```
... '事件过程代码
```

```
End Sub
```

例如单击“cmdOk”命令按钮，使命令按钮的字体大小改为 20 磅，则对应的事件过程如下：

```
Sub cmdOk_Click()
```

```
cmdOk.FontSize = 20 '设置命令按钮的字体大小为 20 磅
```

```
End Sub
```

编写对象事件过程必须先打开对象所在窗体的代码窗口，然后选择对象的事件，最后为事件添加代码。

打开对象所在窗体的代码窗口的方法有：

- 单击工程资源管理器窗口的“查看代码”按钮，系统会弹出代码窗口。
- 执行“视图”菜单中的“代码窗口”命令，系统也会弹出程序代码窗口。
- 双击 VB 开发环境中的任何一个对象，也可打开该对象对应的代码窗口。这种方法是 VB 编程中最常用的一种方法。

打开代码窗口后，窗口的代码编辑区中将出现与用户双击对象相对应的某一事件过程。如图 2-7 所示，为单击 Label1 控件后，系统弹出该对象事件过程。

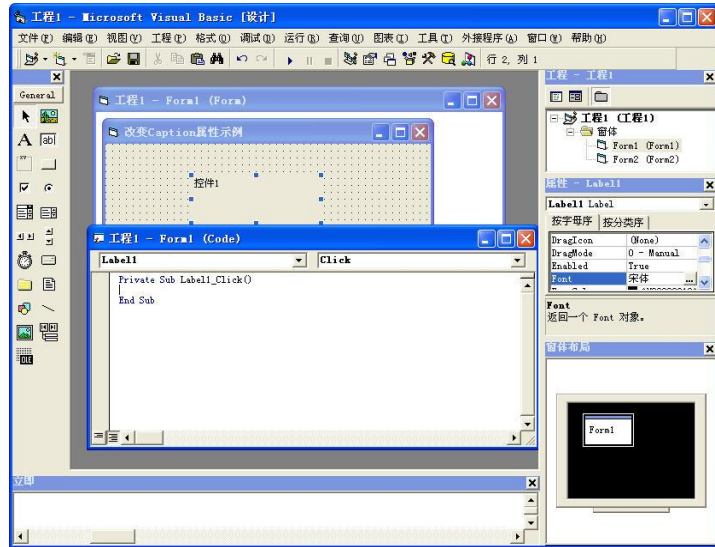


图 2-7 单击 Label1 控件后出现的程序代码编辑窗口

如果代码窗口中所出现的事件过程不是用户想要编写的事件过程，用户还可以在代码窗口中自行选择想要编写的事件过程。操作步骤如下：

- (1) 单击代码窗口左上方的对象列表框，在对象列表中选择要编写事件过程的对象。
- (2) 单击代码窗口右上方的事件列表框，在事件列表中选择要响应的事件名称。

如图 2-8 所示是对象框和事件框的操作图。对象和事件选择完毕后，在代码窗口的编辑区中将出现一个事件过程。用户就可以往该事件过程中添加事件代码了。



图 2-8 对象的选择和事件的选择

3. 事件驱动程序设计

在传统的面向过程的应用程序中，应用程序自身控制了执行哪一部分代码和按何种顺序执行代码，即代码的执行是从第一行开始，随着程序流执行代码的不同部分。程序执行的先后顺序由设计人员编写的代码决定，用户无法改变程序的执行流程。

在 VB 中,程序的执行发生了根本的变化。程序的执行先等待某个事件的发生,然后再去执行处理此事件的事件过程,即事件驱动程序的设计方式。这些事件的顺序决定了代码执行的顺序,因此应用程序每次运行时所经过的代码的路径可能都是不同的。

VB 程序的执行步骤如下:

- (1) 启动应用程序,装载和显示窗体。
- (2) 窗体(或窗体上的控件)等待事件的发生。
- (3) 事件发生时,执行对应的事件过程。
- (4) 重复执行步骤(2)和(3)。

如此循环执行,直到遇到“END”结束语句结束程序的运行或单击“结束”按钮强行停止程序的运行。

2.2 窗体

窗体是 Visual Basic 中的主要对象,是图形界面的基本组成部分。窗体具有自己的属性、事件和方法。

2.2.1 窗体的结构与属性

窗体结构与 Windows 应用程序的窗口十分类似。窗体结构包含控制菜单框、标题栏、最大化/最小化按钮、关闭按钮以及边框。

各类控件必须建立在窗体上。运行程序时,每个窗体对应一个窗口。存储结构上,一个窗体对应一个窗体模块。如图 2-9 所示,为一个窗体的示意图。



图 2-9 窗体的结构

1. 窗体的创建

启动 VB 6.0 后,在“新建工程”对话框中选择“标准 EXE”选项,单击“打开”按钮,便创建了第一个窗体。一个工程可以包含多个窗体,创建新窗体的步骤如下:

- (1) 执行“工程”下拉菜单中的“添加窗体”选项,打开“添加窗体”对话框。
- (2) 在“添加窗体”对话框中,选择“新建”选项卡,对话框的列表框中列出了各种窗体的类型,选中“窗体”选项用于建立一个空白新窗体,选择其他选项则建立一个预定义了某些功能的窗体。
- (3) 单击“打开”按钮,一个新窗体被添加到当前工程中。

添加新窗体后,就可以对窗体的属性进行修改。创建一个窗体如图 2-10 所示,创建多个

窗体如图 2-11 所示。

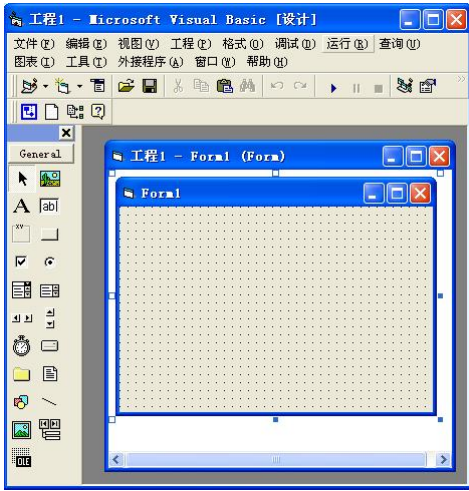


图 2-10 创建一个窗体

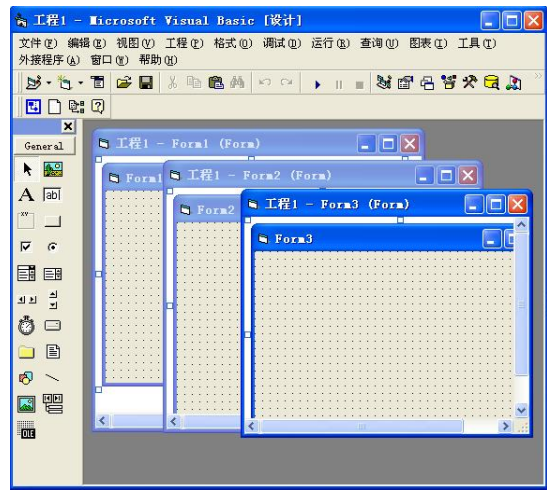


图 2-11 创建多个窗体

2. 窗体的属性

窗体属性除了可以通过属性窗口进行设置之外，还可以在窗体事件过程中通过程序代码设置。这两种设置方式适用于大部分的属性，也有一些属性只能用程序代码或者属性窗口设置中的一种来设置。通常把只能通过属性窗口设置的属性称为“只读属性”，如 Name 属性。窗体的属性多达 50 多种，表 2-1 按字母顺序列出了部分常用的窗体属性，供读者学习参考，如果需要更详细的窗口属性说明，请参看 MSDN 帮助系统。

表 2-1 窗体的属性

属性名	说明
Appearance	设置窗体中的控件是否以三维立体方式显示。取值 1-3D 表示默认设置，以三维立体方式显示窗体上的控件；0-Flat 表示以二维方式显示窗体上的控件。该属性不能通过代码在运行时设置，必须通过属性窗口在设计时设置
AutoRedraw	设置窗体中显示的信息是否重画。当返回被其他窗体遮住的窗体时，如果此属性设置为 True，则自动刷新或者重画窗体中的所有信息，否则不重画
BackColor	设置窗体的背景颜色
BorderStyle	设置窗体的边界样式。取值为 0-None 时表示窗体不仅没有边界，而且没有标题栏；取值为 1-Fixed Single 时表示边界是固定的单线，运行时不能改变窗体大小；2-Sizable 表示是默认设置，窗体的大小可以在运行时改变；3-Fixed Dialog 表示是固定的对话框，大小不能改变；4-Fixed ToolWindow 表示是固定的工具箱窗口，大小不能改变；5-Sizeable ToolWindow 表示是大小可变的工具窗口。该属性的设置会影响其他属性的设置值。如设置为 0、3、4 或 5 时，MinButton、MaxButton 和 ShowInTaskbar 属性将自动设置为 False，表示这类窗体没有“最小化”“最大化”按钮
Caption	设置窗体标题栏中显示的文本内容
ClipControls	当 Paint 事件发生时，确定窗体中的内容是全部重画还是只重画最新暴露的区域。取值为 True 时表示是默认设置，重画整个窗体内容；False 表示重画最新暴露的区域
ControlBox	设置窗体左上角是否显示控制菜单图标。取值为 True 时表示默认设置，窗体中含有窗体图标和控制按钮；False 则表示窗体中不含窗体图标和控制按钮

续表

属性名	说明
DrawMode	用于设置绘图模式，以便控制图形的显示效果。默认是用窗体的前景色来画线
DrawStyle	用于设置线条的线型，默认为实线
DrawWidth	用于设置线条的宽度，默认为 1
Enable	设置窗体是否对鼠标或者键盘事件做出响应。取值为 True 时表示默认设置，对事件做出响应；False 表示对事件不产生响应。这时只能在窗体中显示文本和图形
FillColor	用于设置图形的填充颜色，默认是黑色
FillStyle	用于设置图形的填充方式，默认是透明（Transparent）
Font	设置窗体中文本显示时使用的字体，包括字体的名称和大小，以及是否为粗体、斜体或者粗斜体，是否带有删除线和下划线
FontTransparent	当窗体上要显示文本的位置已有图形或文本存在时，可以用 FontTransparent 属性来确定如何处理原来的图形或文本。取值为 True 时表示是默认设置，将要显示的内容与原来的图形或者文本重叠显示；False 表示将要显示的内容覆盖原来的图形或者文本
ForeColor	设置窗体的前景颜色。设置方法类似于 BackColor 属性
Height、Width	设置窗体的初始高度和宽度
Icon	设置窗体的图标。Icon 属性的设置值通常为图标文件名（扩展名为*.ico）
KeyPreview	如果窗体中的某个控件在使用时发生了键盘事件，则可以用 KeyPreview 属性来确定窗体是否接收键盘事件。取值为 True 表示窗体先接收键盘事件，然后再由当前正使用的控件来处理；False 表示默认设置，当前正使用的控件接收键盘事件，窗体不处理该事件
Left、Top	设置窗体的左上角位置
MaxButton	设置窗体中是否含有最大化按钮。取值为 True 表示默认设置，有最大化按钮；False 表示窗体中不含有最大化按钮
MDIChild	设置当前窗体是否是 MDI 窗体的子窗体。取值为 True 表示当前窗体为 MDI 窗体的子窗体；False 表示默认设置，不是 MDI 窗体的子窗体
MinButton	设置窗体中是否含有最小化按钮。取值 True 表示默认设置，窗体中含有最小化按钮；False 表示窗体中不含有最小化按钮。如果 BorderStyle 属性设置为 0、3、4 和 5，则此属性无效
MouseIcon	指定一个自定义的图标或者光标作为鼠标指针的形状。当 MousePointer 属性设置为 99 时，就要使用 MouseIcon 属性来确定鼠标指针的形状
MousePointer	设置鼠标指针的形状。如果将其设为 99，则可以使用 MouseIcon 属性来指定自定义的鼠标指针形状
Moveable	设置窗体是否可移动。取值为 True 表示默认设置，窗体运行时能够移动；False 表示窗体运行时不可移动
Name	设置当前窗体的名称。代码中可以通过 Name 属性设置的名称来引用相应的窗体
Picture	设置是否在窗体内显示一个图形。默认设置为 None，表示窗体中不显示图形。如果在代码中设置，要使用 LoadPicture 函数
RightToLeft	设置文本显示方向和显示外观
ScaleHeight	设置窗体绘图区域的高度。改变属性的设置值后，将重新定义新的度量单位

续表

属性名	说明
ScaleWidth	设置窗体绘图区域的宽度。改变属性的设置值后，将重新定义新的度量单位
ScaleLeft&ScaleTop	设置窗体绘图区域的左上角位置。默认为 (0,0)，表示窗体左上角即为原点
ScaleMode	设置窗体的度量单位，默认为点素 (Twip)
ShowInTaskbar	设置窗体最小化时是否在任务栏上显示窗体图标
StartPosition	设置窗体首次显示时的位置
Visible	设置窗体是否可见。取值为 True 表示默认设置，窗体可见；False 表示窗体不可见
WindowState	设置程序运行时窗体的显示状态。取值为 0-Normal 表示默认设置，以正常方式显示；1-Minimized 表示以最小化方式显示；2-Maximized 表示以最大化方式（占据整个屏幕）显示

例 2.1 设计一个程序，程序进入运行状态后，自动将窗体的大小设置为屏幕大小的一半并使窗体居中显示，并且在窗体上装载一幅图片。单击窗体时窗体上的图片消失，窗体的背景色会随机发生变化。

进入 VB 编程环境后，根据题目要求在代码窗口中编写窗体的 Load 事件过程和窗体的 Click 事件过程，如下所示：

```
Private Sub Form_Load()
    Form1.Width = Screen.Width * 0.5           '设置窗体的宽度
    Form1.Height = Screen.Height * 0.5        '设置窗体的高度
    Form1.Left = (Screen.Width - Width) / 2    '在水平方向上居中显示
    Form1.Top = (Screen.Height - Height) / 2   '在垂直方向上居中显示
    Form1.Picture = LoadPicture("c:\windows\clouds.bmp") '装载图片
End Sub

Private Sub Form_Click()
    Form1.Picture = LoadPicture("")           '卸载图片
    Form1.BackColor = QBColor(Rnd * 15)      '随机改变窗体背景颜色
End Sub
```

Load 是窗体能够响应的一个事件，只要启动应用程序，窗体被装入内存，就会触发 Load 事件。Load 事件过程通常用来对对象的属性和变量进行初始化。

程序中 LoadPicture 是 VB 的一个函数，用于将指定的图片文件调入内存。LoadPicture 的一般使用格式是：

[对象].Picture = LoadPicture ("文件名")

对象是指窗体、图片框、图像框等，默认为窗体。括号中双引号里的内容是图形文件名（一般应写完整路径）。本例中是指定装载 C 盘 Windows 目录中的 clouds.bmp 图形文件。如果双引号中为空，则表示对象不装载图形。

窗体的 Picture 属性决定窗体的背景图形，该属性也可以在属性窗口中设置。VB 支持以下几种图像文件格式：位图文件 (*.bmp)、GIF 压缩位图文件 (*.gif)、JPEG 压缩位图文件 (*.jpg)、图元文件 (*.wmf) 和图标文件 (*.ico)。

要想改变窗体的尺寸，就需要对窗体的 Height 和 Width 这两个属性进行设置。Height 指窗体的高度，Width 指窗体的宽度，单位为 twip（1 英寸约等于 1440 twip）。本例中是把窗体的高度和宽度都设为屏幕窗口尺寸的一半。程序的运行效果如图 2-12 所示。



图 2-12 登录后的窗体

2.2.2 窗体事件

与窗体有关的事件较多,窗体最常用的事件有 Load (装入)、Activate (活动) 和 Deactivate (非活动)、Click (单击)、DbClick (双击)、Unload (卸载) 和 Paint (绘画) 事件。

1. Load 事件

Load 事件用来在启动程序时对属性和变量进行初始化。在装入窗体后,只需要运行程序,就会自动触发 Load 事件,并执行窗体的 Form_Load 事件过程。Form_Load 事件过程执行完之后,如果窗体模块中还存在其他事件过程,则 VB 将暂停程序的执行,并等待触发下一个事件过程。如果 Form_Load 事件过程内不存在任何指令,则系统将显示该窗体。

2. Activate 事件

在 Load 事件发生之后,系统会自动触发 Activate 事件。Load 事件发生时窗体是不活动的,而 Activate 事件发生时窗体已经是活动的。在不活动的窗体上不能使用 Print 方法显示信息,在活动的窗体上才能使用 Print 方法。

3. Click 事件

Click 事件是单击左键时发生的事件。程序运行后,当单击窗体本身(不是窗体上的控件)的某个位置时,将触发 Click 事件,执行窗体的 Form_Click 事件过程。如果单击的是窗体内的控件,则只能调用相应控件的 Click 事件过程。

4. DbClick 事件

DbClick 事件是双击左键时发生的事件。程序运行后,当双击窗体本身(不是窗体上的控件)的某个位置时,将触发 DbClick 事件,执行窗体的 Form_DbClick 事件过程。

5. UnLoad 事件

UnLoad 事件是从内存中清除一个窗体时触发的事件。如果重新装入该窗体,则窗体中所有的控件都要重新初始化。

6. Paint 事件

当窗体被移动或放大,或者窗体移动时覆盖了另一个窗体时,触发该事件。

例 2.2 “窗体变化”实例。该实例程序的用户界面只有一个窗体。运行程序初始,该窗体的宽度为 4800Twip,高度为 3200Twip,标题为“Visual Basic 欢迎您”。单击该窗体后,窗体的宽度变为 3600Twip,高度变为 2400Twip,标题变为“窗体变化实例”。要求程序设计的过程中使用代码修改窗体的属性。

程序设计步骤如下:

(1) 用户界面的设计。启动 VB,在“新建工程”对话框中选择“标准 EXE”选项,单击“打开”按钮。此时出现一个窗体,为需要设计的应用程序界面。新建窗体的名称属性和标

题属性的默认值均为 Form1。

(2) 代码的设计。双击窗体，在代码区出现如下语句：

```
Private Sub Form_Load()
End Sub
```

(3) 该代码为窗体载入时自动触发的事件过程，在两条语句中插入设置窗体的代码，在程序运行初期完成对窗体的设置，添加的代码如下所示：

```
Private Sub Form_Load()
    Form1.Width = 4800
    Form1.Height = 3200
    Form1.Caption = "Visual Basic 欢迎您"
End Sub
```

(4) 接下来实现单击该窗体后，窗体的标题、宽和高改变的过程。单击过程下拉列表框，选择 Click 事件，在代码区中将出现如下两条语句：

```
Private Sub Form_Click()
End Sub
```

(5) 在两条语句中插入窗体变化的代码，如下所示：

```
Private Sub Form_Click()
    Form1.Width = 3600
    Form1.Height = 2400
    Form1.Caption = "窗体变化实例"
End Sub
```

按 F5 键运行程序，窗体如图 2-13 所示。单击窗体上的任意位置，窗体形状和标题都将发生变化，如图 2-14 所示。



图 2-13 窗体初始效果图

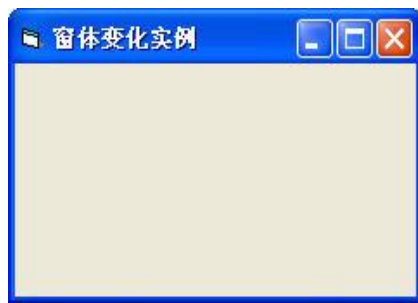


图 2-14 变化后的窗体

2.2.3 窗体方法

窗体上常用的方法有 Print、Cls 和 Move 等，其语法结构见 2.5 节“常用方法”。

1. Print 方法

Print 方法用来显示文本内容。其形式为：

[对象.]Print 表达式

2. Cls 方法

Cls 方法用来清除窗体或图片框在运行时由 Print 方法显示的文本或用绘图方法所产生的图形。其形式为：

[对象.] Cls

省略对象默认为窗体。

3. Move 方法

Move 方法用来移动窗体或控件对象的位置，也可改变对象的大小。其形式为：

[对象.] Move 左边距离[,上边距离[,宽度[,高度]]]

一般也可以由 Left 和 Top 属性更方便地实现移动的效果，但不能同时改变控件的大小。

2.3 控件

窗体和控件都是 VB 中的对象，共同构成了用户的界面。有了控件，VB 的功能变得十分强大，同时还便于操作。控件以图标的形式存放于“工具箱”中，每种控件都有特定的对应图标。启动 VB 后，工具箱位于窗体的左侧，如图 2-15 所示。

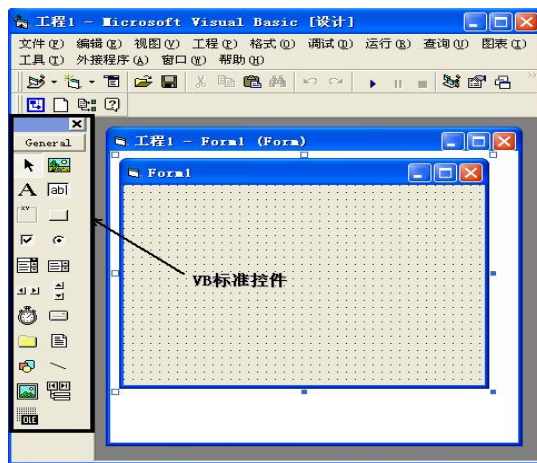


图 2-15 VB 标准控件

2.3.1 控件的分类

VB 6.0 的控件共分为 3 类：

- 标准控件（也称为内部控件），例如文本框、命令按钮以及图片框。这些控件由 VB 的.exe 文件提供。启动 VB 后，内部控件就出现在工具箱中，不能添加也不能删除。
- ActiveX 控件，也称为 OLE 控件或定制控件，是扩展名为.ocx 的独立文件，其中包括各种 VB 版本提供的控件和仅在专业版及企业版中提供的控件，另外还包括第三方提供的 ActiveX 控件。
- 可插入对象，这些对象可以添加到工具箱中，所以也可以将它们当作控件使用。其中一些对象支持 OLE，使用这类控件可以在 VB 应用程序中控制另外一个应用程序（例如 Microsoft Word）的对象。

启动 VB 后，工具箱中列出的是内部控件，如图 2-15 所示。工具箱实际上是一个窗口，称为工具箱窗口，可以通过单击其右上角的关闭按钮关闭工具箱。如果想打开工具箱，可执行“视图”菜单中的“工具箱”命令或单击标准工具栏中的“工具箱”按钮。

2.3.2 标准控件

表 2-2 列出了标准工具箱中内部控件的名称和作用。

表 2-2 标准工具箱中内部控件名称及其说明

图标	名称	功能
	PictureBox 图片框	用于显示图形、文本或者作为其他控件的容器
	Label 标签	用于显示不被修改的文本
	TextBox 文本框	用于输入、显示文本，文本框中内容可以被编辑
	Frame 框架	用于作为其他控件的容器，常为选项按钮或者复选框分组
	CommandButton 按钮	用于接收事件
	CheckBox 复选框	用于接收选择，一组复选框可以选择多个，也可一个不选
	OptionButton 选项按钮	用于接收选择，一组选择按钮只能选择一个
	ComboBox 组合框	用于显示供用户选择的列表项，也可以输入新的列表项
	ListBox 列表框	用于显示供用户选择的列表
	HScroll 水平滚动条	用于提供快速水平定位或提供输入数值
	VScroll 垂直滚动条	用于提供快速垂直定位或提供输入数值
	Timer 计时器	用于按设定的时间间隔产生计时器事件
	DriveListBox 驱动器列表框	用于显示当前可用的驱动器
	DirListBox 目录列表框	用于显示目录列表
	FileListBox 文件列表框	显示当前路径下的文件名列表
	Shape 形状控件	用于在窗体上画各种类型的形状
	Line 直线控件	用于在窗体上画各种类型的直线
	Image 图像控件	用于显示位图、图标等图形
	OLE 容器控件	用于将其他应用程序的对象添加到 VB 的应用程序中
	Data 数据控件	用于连接数据库，并在窗体的其他控件中显示数据库信息

以上简单介绍了工具箱中的标准控件图标以及功能。在以后的章节中，将陆续介绍这些控件在实际应用程序中的使用。

2.3.3 控件的命名和控件值

1. 控件的命名

每个控件都有一个名字。这个名字就是控件的 Name（名称）属性。一般情况下，控件的 Name 属性都有默认值，如 Command1、Text1、Picture1 等。为了增强程序的可读性，较为通常的做法是采用具有一定意义的名字作为对象的 Name 属性，以便于看出对象的类型或其他信息。因此，Microsoft 建议（不是规定）用 3 个小写字母作为对象的 Name 属性的前缀。表 2-3 列出了窗体和内部控件建议使用的前缀。

表 2-3 标准控件的命名示例

图标	前缀	实例
Form（窗体）	frm	frmTitle
PictureBox（图片框）	pic	picCat
Label（标签）	lbl	lblFileName
Frame（框架）	fra	fraComputer
CommandButton（命令按钮）	cmd 或 btn	cmdOK
CheckBox（复选框）	chk	chkBackup
OptionButton（单选按钮）	opt	optChinese
ComboBox（组合框）	cbo	cboLanguage
ListBox（列表框）	lst	lstCode
HScroll（水平滚动条）	hsb	hsbDegree
VScroll（垂直滚动条）	vsb	vsbHeight
Timer（计时器）	tmr	tmrTime
DriveListBox（驱动器列表框）	drv	drvFloppy
DirListBox（目录列表框）	dir	dirHard
FileListBox（文件列表框）	fil	filExample
Shape（形状控件）	shp	shpSquare
Line（直线控件）	lin	linConnect
Image（图像控件）	img	imgHouse
TextBox（文本框控件）	txt	txtText1
OLE（容器控件）	ole	oleObject
Data（数据控件）	dat	datBiblio

2. 控件的值

一般情况下，控件的属性值设置是按照如下语法格式来进行的：

控件.属性

例如，一个文本文件的文本属性可以这么定义：

```
TxtText1.Text="VB6.0"
```

其中，“TxtText1”是控件的名称，Text是控件的属性，该语句把“VB6.0”赋给了文本框的Text属性。

为了便于使用，VB为每个控件规定了一个默认属性，在设置该属性时，不需要给出属性名，通常将该属性称为控件的值。控件值是一个控件最重要或最常用的属性。例如，文本框的控件值为Text，在设置该控件的Text属性时，不需要写成“名称.Text”的形式，只要给出控件名称即可。上面例子中的语句可以改为：

```
TxtText1="VB6.0"
```

控件值虽然节省代码，但是影响了程序的可读性，因此用户最好不使用这种方法，而是按照“控件名称.属性”的格式来编写代码，确保程序更容易理解。

部分常用控件的值见表 2-4。

表 2-4 部分控件的控件值

控件	控件值
PictureBox (图片框)	Picture
Label (标签)	Caption
Frame (框架)	Caption
CommandButton (命令按钮)	Value
CheckBox (复选框)	Value
OptionButton (单选按钮)	Value
ComboBox (组合框)	Text
ListBox (列表框)	Text
HScroll (水平滚动条)	Value
VScroll (垂直滚动条)	Value
Timer (计时器)	Enabled
DriveListBox (驱动器列表框)	Drive
DirListBox (目录列表框)	Path
FileListBox (文件列表框)	FileName
Shape (形状控件)	Shape
Line (直线控件)	Visible
Image (图像控件)	Picture
TextBox (文本框控件)	Text
Data (数据控件)	Caption

2.4 控件的画法和基本操作

在设计用户界面时，要在窗体上画出所需的控件，本节将介绍控件的画法和基本操作。

2.4.1 控件的画法

如果要在窗体中插入一个控件，通常有两种办法，下面以 Label 控件为例来说明。

第一种方法的步骤如下所示：

- (1) 单击如图 2-16 所示工具箱中的 Label 控件按钮。

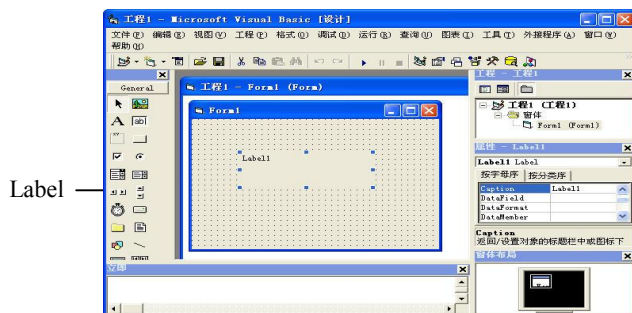


图 2-16 画出指定大小的控件

(2) 将鼠标指针移到窗体上时，鼠标指针变成了十字形状。

(3) 将十字形状指针移到要绘制标签控件的左上角位置，按下左键并拖曳，窗体上将出现一个方框。

(4) 将鼠标向右下方移动，直到一个合适的位置，松开左键，即可在窗体上画出一个 Label 控件。

(5) 如果想要在窗体编辑区拖动鼠标可画出多个对应的控件，按住 Ctrl 键，单击工具箱中的控件按钮，松开 Ctrl 键，并按照上述步骤进行绘制即可。

第二种建立控件的方法比较简单，即双击工具箱中的某个控件图标（本例为 Label 控件），窗体的中央就会出现一个默认大小的控件，如图 2-17 所示。与第一种方法不同的是，用第二种方法所画控件的大小和位置是固定的。

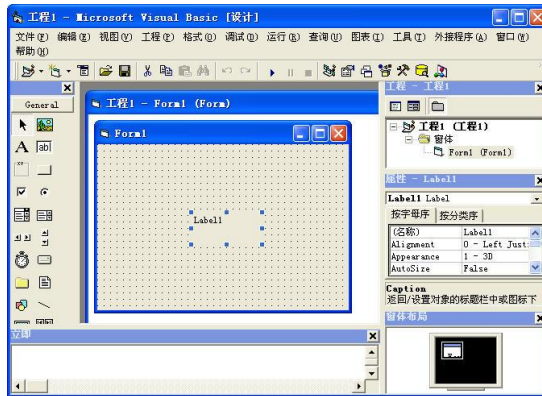


图 2-17 插入默认大小的控件

窗体的工作区中布满了“对齐”的小点，这些点是用来做参考的，以对齐控件。如果不想看到这些小点或想改变点与点之间的距离，可以执行“工具”菜单中的“选项”命令，在打开的对话框中单击“通用”选项卡，在“窗体网格设置”栏内设置相关的选项，设置窗口如图 2-18 所示。

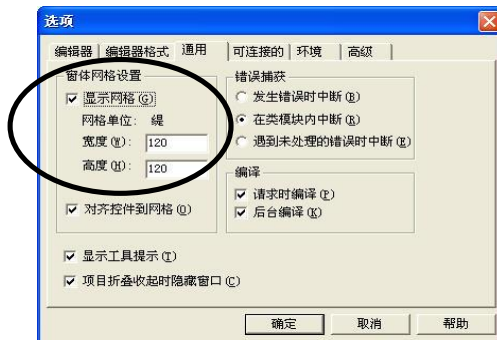


图 2-18 窗体网格设置对话框

2.4.2 控件的基本操作

在一个窗体上画好一个或多个控件后，它们的大小和位置可能不符合要求，但可以对控件进行缩放或移动位置等操作。

1. 控件的选择

当用户画完一个控件后，在该控件的边框上会出现 8 个黑色的小方块。这些小方块被称为句柄，用于调整控件的大小。

有时候，可能需要对多个控件进行操作，这就需要选择多个控件，常用的选择方法有：

- 按住 Ctrl 键，单击要选择的多个不连续控件。
- 按住 Shift 键，单击要选择的多个连续控件。
- 如果要选择的控件比较集中，则将鼠标指针移到要选择控件的左上角，按住左键向下方拖动，拖动过程中会出现一个矩形框。当所有要选择的控件均包含在该矩形框中时，松开左键。

在被选择的这些控件中，有一个控件的周围是 8 个实心句柄（其他控件是 8 个空心句柄），该控件被称为“基准控件”，如图 2-19 所示。当对被选择的控件进行对齐、调整大小等操作时，将以“基准控件”为准。

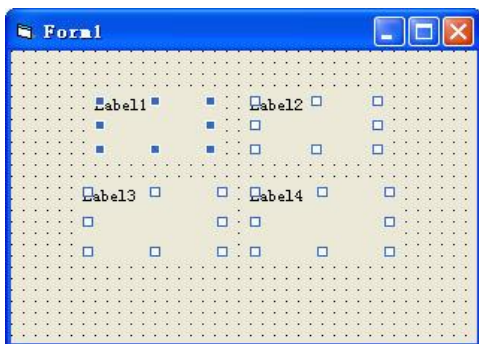


图 2-19 选择了多个 Label 控件

2. 控件的拖放和移动

控件画出后，其大小和位置不一定符合设计要求，此时可以对控件进行放大、缩小或移动其位置等操作。

在前面画控件的过程中已经看到，控件的周围有 8 个黑色的句柄。要缩放控件，只需将鼠标指针对准控件的句柄，出现双向箭头时，拖动鼠标即可改变控件的高度和宽度。按下 Shift+“方向箭头”键也可改变控件的大小。

如果要移动控件，选中控件后，单击该控件，并按下不动，拖动控件到所需的位置，然后松开左键。此外，按下 Ctrl+“方向箭头”键也可以移动控件的位置。或者在属性窗口中修改控件的 Left、Top、Width 和 Height 属性值也可以改变控件的位置。

3. 控件的复制和删除

VB 允许对画好的控件进行“拷贝”，操作步骤如下：

(1) 单击需要复制的控件（本例程序中为 Label1 控件）。

(2) 执行“编辑”菜单中的“复制”命令。执行该命令后，VB 将把目标控件拷贝到 Windows 的剪贴板中。

(3) 执行“编辑”菜单中的“粘贴”命令，屏幕上将显示一个对话框，如图 2-20 所示。询问是否要建立控件数组，单击“否”按钮后，就把活动控件复制到窗体的左上角，如图 2-21 所示。



图 2-20 “询问是否创建控件数组”对话框



图 2-21 复制 Label 控件

为了删除一个控件，必须先选中该控件，然后按 Del 键，即可把该控件清除。或者右键单击选中的控件，选择“删除”即可。清除后，其他某个控件周围自动产生 8 个黑色的句柄（如果存在其他控件的话），如图 2-22 所示。

4. 通过属性窗口改变对象的位置和大小

在属性窗口中，使用四种属性来表示控件或者窗体对象的大小和位置，即 Width、Height、Top 和 Left。对象的位置由 Top 和 Left 属性来决定，大小由 Width 和 Height 属性来决定。其中 (Top, Left) 是控件或窗体相对于左上角的坐标，Width 是水平方向的长度，Height 是垂直方向的长度。它们之间的数学关系，如图 2-23 所示。



图 2-22 删除 Label 控件

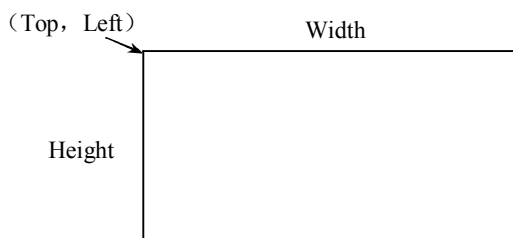


图 2-23 对象的位置和大小

需要注意的是，对于窗体来说，(Top, Left) 是相对于屏幕左上角的坐标；对于控件来说，(Top, Left) 是相对于窗体左上角的坐标。

2.5 常用方法

在面向对象编程过程中，仅通过对象的属性实现对象的控制是不够的，对象对事件的响应过程需要用相应的方法来进行描述。对于不同的对象，VB 提供了不同的方法，但使用方法的语法结构是相同的。其形式为：

[对象.]方法[参数列表]

本节介绍常用的方法，其他一些方法在介绍相关对象时再具体介绍。

1. Print 方法

Print 方法的作用是在对象上输出信息。其形式为：

[对象.]Print[{Spc(n)|Tab(n)}][表达式列表];|,|]

其中：“对象”可以是窗体、图形框或打印机。若省略了对象，则在窗体上输出。

Spc(n)函数：用于在输出时插入 n 个空格，允许重复使用。

Tab(n)函数：用于在输出表达式列表前向右移动 n 列，允许重复使用。

表达式列表：要输出的数值或字符串表达式，若省略，则输出一个空行，多个表达式之间用空格、逗号、分号分隔，也可出现 Spc 和 Tab 函数。

;(分号)：表示光标定位在上一个显示的字符后。

,(逗号)：表示光标定位在下一个打印区的开始位置处，打印区每隔 14 列开始。

注意：①Spc 函数与 Tab 函数的作用类似，可以互相替代。但 Tab 函数从对象的左端开始计数，而 Spc 函数表示两个输出项之间的间隔。②Print 方法在 Form_Load 事件过程中不起作用。

2. Cls 方法

Cls 方法用于清除运行时在窗体或图形框中显示的文本或图形。形式如下：

[对象.]Cls

其中：“对象”为窗体或图形框，省略为窗体。

例如：

```
Picttrue1.Cls    清除图形框内显示的图形或文本
Cls              清除窗体上显示的文本
```

注意：①Cls 方法只清除运行时在窗体或图形框中显示的文本或图形，不清除窗体在设计时的文本和图形。②Cls 方法使用后，CurrentX、CurrentY 属性被设置为 0。

3. Move 方法

Move 方法用于移动窗体或控件，并可改变其大小。形式如下：

[对象.]Move 左边距离[,上边距离][,宽度[,高度]]





其中：“对象”可以是窗体及除时钟、菜单外的所有控件，省略对象为窗体。

左边距离、上边距离、宽度、高度是以数值表达式表示，以 twip 为单位。如果对象是窗体，则“左边距离”和“上边距离”以屏幕左边界和上边界为准，否则以窗体的左边界和上边界为准，宽度和高度表示可改变其大小。

习题二

一、选择题

1. 为了消除窗体上的一个控件，下列正确的操作是（ ）。
 - A. 按回车键
 - B. 按 Esc 键
 - C. 选择（单击）要清除的控件，然后按 Del 键
 - D. 选择（单击）要清除的控件，然后按回车键
2. 在 VB 集成环境的设计模式下，双击窗体上的某个控件打开的窗口是（ ）。
 - A. 工程资源管理器窗口
 - B. 属性窗口
 - C. 工具箱窗口
 - D. 代码窗口

3. 设窗体上有一个命令按钮数组，能够区分数组中各个按钮属性的是（ ）。
A. Name B. Index
C. Caption D. Left
4. 在 VB 集成环境中要结束一个正在运行的工程，可单击工具栏上的一个按钮，这个按钮是（ ）。
- A.  B.  C.  D. 
5. 用来设置文字字体是否斜体的属性是（ ）。
- A. FontUnderline B. FontBold C. FontSlope D. FontLtalic
6. 以下能在窗体 Form1 的标题栏中显示"VisualBasic 窗体"的语句是（ ）。
- A. Form1.Name="VisualBasic 窗体"
B. Form1.Title="VisualBasic 窗体"
C. Form1.Caption="VisualBasic 窗体"
D. Form1.Text="VisualBasic 窗体”
7. 以下不能在“工程资源管理器”窗口中列出的文件类型是（ ）。
- A. bas B. res C. fmn D. ocx
8. 以下叙述中错误的是（ ）。
- A. 在工程资源管理器窗口中只能包含一个工程文件及属于该工程的其他文件
B. 以.BAS 为扩展名的文件是标准模块文件
C. 窗体文件包含该窗体及其控件的属性
D. 一个工程中可以含有多个标准模块文件
9. 以下叙述中错误的是（ ）。
- A. 双击鼠标可以触发 DbIclick 事件
B. 窗体或控件的事件的名称可以由编程人员确定
C. 移动鼠标时，会触发 MouseMove 事件
D. 控件的名称可以由编程人员设定
10. 以下不属于 VB 系统的文件类型是（ ）。
- A. .frm B. .bat C. .vbg D. .vbp

二、填空题

1. 在窗体上画一个文本框和一个图片框，然后编写如下两个事件过程：

```
Private Sub Form_Click()  
Text1.Text="VB 程序设计"  
End Sub  
Private Sub Text1_Change()  
Picture1.Print "VB Programming"  
End Sub
```

程序运行后，单击窗体，则在文本框中显示的内容是_____，而在图片框中显示的内容是_____。

2. 在窗体上画一个文本框和一个图片框，然后编写如下两个事件过程：

```
Private Sub Form_Load()  
Text1.Text="计算机"  
End Sub
```

```
Private Sub Text1_Change()
Picture1.Print"等级考试"
End Sub
```

程序运行后，在文本框中显示的内容是_____，而在图片框中显示的内容是_____。

3. 在面向对象方法中，类的实例称为_____。

4. 在窗体上画一个名称为 **Command1** 的命令按钮和一个名称为 **Text1** 的文本框。程序运行后，**Command1** 为禁用（灰色）。当向文本框中输入任意字符时，命令按钮 **Command1** 变为可用。请在_____处填入适当的内容，将程序补充完整。

```
Private Sub Form_Load()
Command1.Enabled=False
End Sub
Private Sub Text1_____()
Command1.Enabled=True
End Sub
```

5. 在面向对象方法中，_____描述的是具有相似属性与操作的一组对象。