

# 第 3 章

## Visual Basic 编程基础

### 本章要点

本章介绍了 Visual Basic 程序设计的基本步骤、程序书写规范，并用通俗易懂的语言详细讲解了 Visual Basic 面向对象程序设计中的几个重要而抽象的概念：对象与类，属性、方法和事件，事件驱动机制，窗体模块、标准模块和类模块等。

### 学习目标

1. 了解 Visual Basic 程序设计的基本步骤、类模块的概念。
2. 理解对象与类的概念、事件驱动机制。
3. 掌握 Visual Basic 程序书写规范，属性、事件和方法，以及窗体模块和标准模块的用法。



### 3.1 Visual Basic 程序设计的基本步骤

利用 Visual Basic 进行程序设计大致包括以下基本步骤：

设计应用程序的界面。如对窗体、命令按钮、菜单进行设计并设置它们的属性。

编写事件过程代码。即编写事件发生后所要执行的程序代码。这些代码用于对发生的各种事件做出响应。如鼠标单击“新建”菜单的事件发生后，可以执行该菜单的单击事件过程以建立一个新的文件。

调试并保存应用程序中的各个组成文件，并编译生成可执行文件。编译生成的可执行文件可以脱离 Visual Basic 开发环境直接在 Windows 中运行。



### 3.2 对象与类

#### 1. 对象 (Object)

Visual Basic 具有“面向对象”的特性，Visual Basic 应用程序的基本单元是对象，用 Visual Basic 编程就是用“对象”组装程序。这种“面向对象”的编程方法与传统的全部用代码编制程序的方法有很大区别，就像用集成电路芯片组装电视机和用三极管、二极管组装电视机的区别一样。显然，“面向对象”的编程方法比传统的编程方法更简单、更方便，并且编写出的程序也更加稳定。因此，“对象”可以被看作 Visual Basic 程序设计的核心。

在 Visual Basic 程序设计中，对象中还可以包含其他对象，包含其他对象的对象称为容器。举个例子来说，如果把“人”当作要研究的一个对象，人又包含头、手、腿、脚等部位，其中的每个部位又可以单独作为被研究的对象。在 Visual Basic 程序设计中，整个应用程序就是一个对



象，应用程序中又包含着窗体（Form）、命令按钮（Command Button）、菜单（Menu）等对象。

## 2. 类（Class）

在 Visual Basic 中，对象是由类创建的，因此可以说对象是类的具体实例，这就好比蛋糕和做蛋糕的模具之间的关系。

各种不同的对象分属于各种不同的种类。同一类对象可能具有一些不同的特性（或者说同一类对象不一定具有完全相同的特性）；具有某些相同特性的对象，不一定是同一类对象。这就好比人和猴子虽然都有身高、性别等特性，但二者之间还存在着智商、语言等特性差异，这两个对象之间的特性相差很多，根本就不能算作同一类对象。而男人和女人，虽然有着性别、生理等方面的差异，但他们绝大部分特性相同，可以算作同一种类。由此，可以归结为一句话：同一类对象的绝大部分特性相同。



## 3.3 属性、方法和事件

在 Visual Basic 中，可以通过属性、方法和事件来说明和衡量一个对象的特性。

### 1. 属性（Property）

属性是指用于描述对象的名称、位置、颜色、字体等特性的一些指标。通过改变对象的属性值可以改变对象的特性。

有些属性可以在设计时通过属性窗口来设置，不用编写任何代码；而有些属性则必须通过编写代码，在运行程序时进行设置。可以在运行时读取和设置取值的属性称为读写属性，只能读取的属性称为只读属性。

同样以“人”这类对象为例。人具有的各种特性，都可以称为属性，如身高、性别、年龄、学历等。这些属性数据以属性值的方式记录在属性栏中。同一类对象的绝大部分属性栏相同，但其中记录的属性值不一定相同。例如，程序中的命令按钮（Command Button）都具有高度（Height）、宽度（Width）、字体（Font）等属性，但每个不同的命令按钮的高度、宽度、字体等属性的具体取值不一定相同。

总之，属性指明了对象“是什么样的”，常用于定义对象的外观。

### 2. 方法（Method）

方法是用来控制对象的功能及操作的内部程序。例如，人具有说话、行走、学习、睡觉等功能，在 Visual Basic 中，对象所能提供的这些功能和操作，就称作“方法”。以窗体为例，它具有显示（Show）或隐藏（Hide）的方法。

总之，方法指明了对象“能做什么”，常用于定义对象的功能和操作。

### 3. 事件（Event）

事件是指发生在某一对象上的事情。事件又可分为鼠标事件及键盘事件。例如，在命令按钮（Command Button）这一对象上可能发生鼠标单击（Click）、鼠标移动（Mouse Move）、鼠标按下（Mouse Down）等鼠标事件，也可能发生键盘按下（KeyDown）等键盘事件。

人可以对外界事件做出反应，例如，某人听见其他人叫自己的名字时会立即答应，或扭头寻找。同样，Visual Basic 中的对象也具有响应一些外部事件的能力，例如，命令按钮或菜单被鼠标单击（Click）时会执行相应的程序代码来完成指定的操作。

总之，事件指明了对象“什么情况下做”，常用于定义对象发生某种反应的时机和条件。



### 3.4 事件驱动机制

#### 1. 事件驱动机制与传统编程方式的异同

在传统的程序设计过程中，程序是按照预先编写的代码逐条依次执行的，即按照预定的流程执行。

而 Visual Basic 是按照事件驱动机制运行程序的。Visual Basic 的每一个窗体和控件都有一个预定义的事件集，如鼠标单击 (Click) 事件、鼠标双击 (Dblclick) 事件等。如果其中有一个事件发生，而且在关联的事件处理过程中存在代码，则 Visual Basic 执行对应的代码。在事件驱动机制中，系统先执行哪一段代码并不取决于预定的顺序，而是由用户操作来决定的。例如，单击某个按钮的操作产生该按钮的单击 (Click) 事件，此时被执行的代码就是该按钮的单击 (Click) 事件处理程序，随后又发生了某个菜单的单击 (Click) 事件，则接下来被执行的代码就是该菜单的单击 (Click) 事件处理程序。这就是 Visual Basic 的事件驱动机制。在一个事件处理过程内部，Visual Basic 语言与其他传统语言类似，程序也是按照预定的流程执行的。

#### 2. 事件产生的方式

在 Visual Basic 中，产生一个事件大致有以下几种情况：

程序操作者触发。操作者可以通过键盘操作或鼠标操作产生一个事件，如鼠标单击 (Click) 事件、鼠标双击 (Dblclick) 事件、键盘按下 (KeyDown) 事件等。

由系统触发。系统自身也可以触发事件，如在一个定时提醒程序中，可以利用“定时器控件”在时间满足提醒条件时由系统自动触发一个事件提醒用户需要处理的事情。

代码间接触发。有些事件可以由程序代码间接触发，如当代码装载窗体时会产生该窗体的 Load (装载) 事件等。

#### 3. 事件驱动程序的执行过程

事件驱动程序的执行过程分为以下 4 步：

启动应用程序，装载和显示窗体；

应用程序（主要是窗体或控件）接收和响应发生的事件；

如果在相应的事件处理过程中存在代码，就执行这些代码；

应用程序等待下一个事件的发生。如此周而复始地运行，直到程序被关闭。

#### 4. 按照事件驱动机制编写程序的基本方法

按照事件驱动机制编程就是根据需要完成事件处理程序的编写工作。程序设计人员应仔细分析程序运行中可能产生的每一个事件，对于某些事件，在事件发生时希望程序执行若干代码以便实现某种功能，则应将这些代码编写成该事件的事件处理程序；有一些事件，事件发生后不需执行任何代码，不必为这些事件编写代码，系统将忽略这些事件，且对其不会有任何反应。



### 3.5 组织 Visual Basic 程序代码

Visual Basic 是通过程序模块进行代码组织的，其程序模块分为三种类型：窗体模块、标准模块和类模块。

Visual Basic 程序的操作界面是窗体，在窗体中可以包含控件（如命令按钮、菜单等），



每个窗体和控件都包含各自的事件处理过程。一个简单的应用程序可以只有一个窗体，所有程序代码都保存在该窗体及其控件的事件处理过程中，这就是窗体模块。

对于一个庞大复杂的应用程序，则需要创建多个窗体，这样就有可能出现在几个窗体中都要执行的公共代码。为了避免重复，这些公共代码被保存在一个独立于窗体的模块中，这个模块被称为标准模块。

Visual Basic “面向对象”编程实际上是用具体的对象来组装程序。对于经常使用的一些程序代码，可以用类模块的形式将其固定下来，使之成为一类对象所固有的功能特性。这样，就可以在程序设计过程中，用类来产生一些类的实例（程序中引用的具体的对象），进而用具体的对象来完成组装程序的工作，而那些经常使用的程序代码也会随着对象的应用一起发挥功用。

### 1. 窗体模块

窗体模块的文件扩展名为“.frm”。

窗体模块中主要包括通用过程和本窗体及其控件的事件处理过程。在窗体模块中可以定义变量、常数和外部过程的窗体级声明。在用 Visual Basic 编写应用程序时，要注意写入窗体模块的代码是该窗体及其控件专用的，对于多个窗体所共用的程序代码可以用“标准模块”的形式来完成，而不是放在某个窗体模块中。

### 2. 标准模块

标准模块的文件扩展名为“.bas”。

标准模块是应用程序内其他模块访问公用过程和声明的容器。它们可以包含变量、常数、外部过程和全局过程的全局声明或模块级声明。写入标准模块的代码不必针对特定的应用程序，如果能够注意不用具体名称引用窗体和控件，则在许多不同的应用程序中可以重复使用标准模块。

### 3. 类模块

类模块的文件扩展名为“.cls”。

在 Visual Basic 中，类模块是面向对象编程的基础。可以在类模块中编写代码以建立新的对象。这些新对象可以包含自定义的属性和方法。实际上，Visual Basic 提供的窗体、控件都是类模块的一种，通过类模块可以根据自己的需要建立自己的对象。



## 3.6 Visual Basic 程序代码书写规范

### 1. 命名规范

在编写 Visual Basic 程序时，要声明和命名许多元素。声明过程、变量和常数的名字时，必须遵循以下四条规则：

必须以字母开头；

不能包含嵌入的句号或者类型声明字符；

不能超过 255 个字符，控件、窗体、类和模块的名字不能超过 40 个字符；

不能和受到限制的关键字同名。所谓受到限制的关键字是指 Visual Basic 中预先定义的词，又称保留字，它们是 Visual Basic 语言的组成部分。

## 2. 注释规范

在程序代码中使用注释是一个优秀程序设计人员的良好习惯。这样既可以方便开发者，也可以方便以后可能检查或修改源代码的其他程序设计人员。在 Visual Basic 语言中，注释符为“'”，当 Visual Basic 遇到这个符号时将忽略其后面的所有内容。程序设计人员可以随意地在“'”后面添加中文或英文注释内容。

### 【例 3.1】

```
Private Sub Form_Load()
    '这是一个 Visual Basic 的示例程序
    '它主要完成打印学生名册的功能
    Command1.Caption = "打印"
    Command2.Caption = "取消"
End Sub
```

#### 说明<<<

“'”后面的内容为程序注释内容，程序运行时，将不会被执行。

## 3. 分行书写规范

为了便于程序的阅读和书写，经常要将 Visual Basic 语句分成多行，分行的方法是在代码中用续行符（一个空格后面跟一个下画线）将长语句分成多行。

### 【例 3.2】

```
For Each xm In .xm
    If xm <> "" Then Debug.Print "    " & _
        xm.Name & " = " & xm
Next xm
```

#### 说明<<<

- ① 本例中第 2 行和第 3 行实际上是一个完整的语句。
- ② 在同一行内，续行符后面不能加注释。

## 4. 合并行规范

一行语句有时很短时，希望将多条语句合并写到同一行上。这时只需要将各语句用“:”分开即可。例如：

```
Command1.Caption="打印":Commadn2.Caption= "取消"
```

这等同于下面的语句：

```
Command1.Caption="打印"
Commadn2.Caption= "取消"
```



### 习题 3

#### 1. 填空题

(1) 对象是 Visual Basic 应用程序的\_\_\_\_\_。在 Visual Basic 中可以用\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_来说明和衡量一个对象的特性。

(2) 属性分为\_\_\_\_\_属性和\_\_\_\_\_属性两种。方法是\_\_\_\_\_的内部程序。

(3) 在 Visual Basic 中, 事件产生的方式主要有\_\_\_\_\_、\_\_\_\_\_和\_\_\_\_\_。

(4) Visual Basic 的程序模块有三种: \_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_。

(5) 窗体模块的文件扩展名为\_\_\_\_\_、标准模块的文件扩展名为\_\_\_\_\_、类模块的文件扩展名为\_\_\_\_\_。

(6) 在 Visual Basic 语言中, 注释符为\_\_\_\_\_, 分行符为\_\_\_\_\_, 并行符为\_\_\_\_\_。

(7) 通过\_\_\_\_\_模块可以根据需要建立自己的对象。

#### 2. 简答题

(1) 什么是对象的属性?

(2) 简述事件驱动程序的执行过程。

(3) 简述事件驱动机制与传统编程方式的异同。

电子工业出版社版权所有  
盗版必究