

# Visual Basic 6.0

## 高级实用教程

曾伟民 邓勇刚 等编著

抖斗书屋 审校



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

URL: <http://www.phei.com.cn>

# Visual Basic 6.0 高级实用教程

曾伟民 邓勇刚 等编著

抖斗书屋 审校

JS82/09

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

## 内 容 提 要

Visual Basic 6.0 是微软公司主流软件系列——Visual Basic 系列的最新产品,它是在 5.0 版本的基础上做进一步完善的,并增加了网络应用方面的内容。微软所推出的 6.0 版本力图成为全球可视化编程的标准。

全书详细介绍了 Visual Basic 6.0 的特点、功能及使用方法,使读者能够在短期内熟悉运用 Visual Basic 6.0 的各种高级功能,进行程序设计。本书对 Visual Basic 6.0 的新特性、界面高级编程、数据库访问、Internet 开发、多媒体设计等功能进行了全面阐述。在讲解这些内容时,我们列举了丰富的实例。书中图文并茂,内容繁简得当,由浅入深,非常适合计算机编程人员及计算机爱好者学习和参考使用,尤其适合相关专业的培训班及各大院校作为教材使用。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,翻印必究。

### 图书在版编目(CIP)数据

Visual Basic 6.0 高级实用教程/曾伟民、邓勇刚等编著. —北京:电子工业出版社,1999.10

ISBN 7-5053-5695-X

I. V… II. ①曾… ②邓… III. BASIC 语言-程序设计-教材 IV. TP312

中国版本图书馆 CIP 数据核字(1999)第 34221 号

书 名: **Visual Basic 6.0 高级实用教程**

编 著 者: 曾伟民 邓勇刚 等

审 校 者: 抖斗书屋

责任编辑: 郭 晶

特约编辑: 刘文智

印 刷 者: 北京大中印刷厂

出版发行: 电子工业出版社 URL: <http://www.phei.com.cn>

北京市海淀区万寿路 173 信箱 邮编 100036

经 销: 各地新华书店

开 本: 787×1092 1/16 印张: 33.5 字数: 857.6 千字

版 次: 1999 年 10 月第 1 版 2000 年 4 月第 2 次印刷

书 号: ISBN 7-5053-5695-X  
TP·2932

印 数: 4000 册 定价: 45.00 元

凡购买电子工业出版社的图书,如有缺页、倒页、脱页、所附磁盘或光盘有问题者,请向购买书店调换。若书店售缺,请与本社发行部联系调换。电话 68279077

# 序

当今世界,使用何种计算机语言的人最多?肯定是 Basic。自从微软推出 Visual Basic 以来, Visual Basic 就成了广大程序开发人员的首选语言。据统计,仅在数据库信息系统开发领域, Visual Basic 就占了将近 90%的份额。所见即所得,面向对象,组件开发等特性大大缩短了应用软件的开发周期。98 年 10 月,微软又推出了 Visual Basic 的最新版本 Visual Basic 6.0。这是一个激动人心的产品,利用它,用户会发现编程其实并不枯燥。

有人说, Visual Basic 面向的是初级用户,只能开发一些简单的应用程序,不像 C/C++ 那样能开发一些高级程序。其实不然,几乎所有用 C/C++ 开发的应用程序,都能用 Visual Basic 开发。

那么,如何用 Visual Basic 开发一些高级程序, Visual Basic 6.0 又有哪些新功能呢?本书将以 Visual Basic 6.0 为参考,结合作者多年使用 Visual Basic 开发的经验,循序渐进地为您做出回答。

## 1. 谁使用这本书

虽然本书更多地谈论一些高级话题,适用于中高级读者,但深入浅出,初学者亦能看懂(当然,您必须掌握 Visual Basic 的一些基本语法及 Visual Basic 集成环境的使用方法)。

本书适用于如下读者:

- (1) 您刚了解了 Visual Basic 的基本用法,想更深入地学习 Visual Basic。
- (2) 您刚使用了 Visual Basic 5.0,想了解 Visual Basic 6.0 的新内容。
- (3) 您已经系统学习了 Visual Basic,想了解 Visual Basic 的一些编程技巧。
- (4) 您想编写一个数据库应用系统,但不知道使用何种数据库访问方法。
- (5) 您想用 Visual Basic 快速开发 Internet 应用程序。

## 2. 本书的组织结构

本书主要论述的是 Visual Basic 的一些高级话题和 Visual Basic 6.0 的新特性。因此,本书将重点放在这两方面。全书共分 14 章。

### 第 1 章 Visual Basic 6.0 快速入门

这一章适用于第一次接触 Visual Basic 6.0 的读者,讨论如何用 Visual Basic 6.0 建立您的第一个 Visual Basic 6.0 应用程序,并简要介绍了 Visual Basic 语言本身的一些特性。

### 第 2 章 Visual Basic 6.0 的新内容

Visual Basic 6.0 相对 Visual Basic 5.0 又增加了一些激动人心的新内容,如新概念、新函数、新控件等,本书将一一为您介绍。

### 第 3 章 活用 Visual Basic 语言

即使对 Visual Basic 很熟悉的人,也不一定能将此语言用得得心应手。本章介绍了使用 Basic 语言的一些技巧。

### 第 4 章 用户界面高级编程

用户界面在很多方面都是应用程序中最重要的一部分,它是用户最终看到的東西,美观、易用是准则。本章介绍了一些基本的界面元素及一些最新的界面元素,并讨论了使用

它们的某些技巧。

## 第 5 章 文件操作

本章主要讨论对二进制文件的读写操作，并解决一些读写文件时碰到的常见问题，如对 String 类型的处理等。

## 第 6 章 ActiveX

COM 和 ActiveX 是现在的热门话题。本章讨论了 COM 模型，如何制作 ActiveX 控件，如何使用 ActiveX 自动化，ActiveX 控件，ActiveX 文档以及如何编写 ActiveX EXE，ActiveX DLL 应用程序等。

## 第 7 章 Internet 和网络编程

Internet 和网络编程是当今的一大热点。本章讨论了如何使用 Winsock、FTP、WebBrowser 等控件，讨论了如何收发 E-mail，以及网络编程中的一些小技巧。

## 第 8 章 数据库访问技术

本章告诉您如何选择数据库访问方法。系统介绍了 DAO、RDO、ODBC、ADO 等各种数据库访问技术，重点讨论了 ADO 的使用方法。

## 第 9 章 数据显示和报表

本章简要介绍了如何显示从数据库中得到的数据。

## 第 10 章 多媒体编程

本章讨论了多媒体编程中常用到的两个控件：MCI 和 ActiveMovie 控件。

## 第 11 章 帮助文件

本章详细讨论如何制作 Visual Basic 帮助文件并在应用程序中使用帮助。另外还讨论了一些高级功能。

## 第 12 章 注册表

本章简要讨论了 Visual Basic 对注册表的操作。

## 第 13 章 错误处理

错误总是难免的。为了保证应用程序的健壮性，必须对异常错误进行处理。本章讨论如何处理异常错误。

## 第 14 章 实用 Visual Basic 编程技术

本章介绍了六个实用程序并讨论了其中用到的编程技巧。

子曰：“知之者不如好之者，好之者不如乐之者”，希望读者能发现用 Visual Basic 6.0 编写应用程序的乐趣。

本书由中科辅龙计算机技术有限公司抖斗书屋策划，主要部分由曾伟民、邓勇刚编写。另外参加编写的人员还有：李寅、赖春容、崔佳、简煊明、郭亮、周子毅、陈诚、饶恒毅、孟名、李军、朱勇、杨胜荣、罗杰军、陈永刚、刘磊、漆文荣、余立春、甘江武、陈卫国、陈雨田等。全书由史惠康、石利文、郭美山统稿，徐平、杨桂莲校排。

由于时间仓促，作者水平有限，本书错漏之处在所难免，欢迎广大读者批评指正。

史惠康

# 第 1 章 Visual Basic 6.0 快速入门

Visual Basic 是基于窗体的可视化程序开发环境。其编写程序的典型方法是根据预先设计的图形用户界面，用鼠标将工具箱中的各种控件拖放到窗体的指定位置，并将其调整成所需的大小，设置必要的属性，再加入处理用户事件的简单代码。Visual Basic 的突出特点是将大量的 Windows 编程细节隐藏起来，通过数据驱动的方式调用不同的程序代码，使得程序开发人员可以将主要精力放在程序的关键处理步骤上，而且其处理方法直观、方便。Visual Basic 虽然与传统的 Basic 语言有本质的区别，但是它继承了 Basic 语言简单、易用的优点。

这一章适用于第一次接触 Visual Basic 6.0 的读者，讲述如何用 Visual Basic 6.0 建立您的第一个 Visual Basic 6.0 应用程序，并简要介绍了 Visual Basic 语言本身的一些特性。

## 1.1 建立您的第一个 Visual Basic 6.0 应用程序

Visual Basic 程序设计环境包含需要快速有效的构造 Windows 程序的所有工具。在用 Visual Basic 之前，有些概念需要知道：

- Windows 是如何运转的：窗口、事件和消息

微软 Windows 操作系统管理着许多的窗口，每个窗口都有一个唯一的标识符（窗口句柄或 hWnd），系统连续地监视着每个窗口的活动或事件，事件可以是用户的鼠标点击或键盘输入，甚至可以是程序控制、其他窗口的行为等而导致的。

每当一个事件产生，就会导致一条消息发送到操作系统。系统处理这个消息，并把它广播到其他的窗口中。每个窗口基于自己的指令采取合适的行动来处理这个特殊的消息，例如，当被其他窗口覆盖时重画自己。

可以想象，处理不同窗口的不同消息是非常麻烦的。幸运的是，Visual Basic 屏蔽了这些底层的消息处理，许多的消息都由 Visual Basic 自动地处理，其他的可以根据程序员方便与否作为事件过程来响应。这样，您不需要知道太多的细节便可以快速创建出功能很强的应用程序。

- 理解事件驱动（Event-Driven）模式

在事件驱动模式下，代码并不按照预定的顺序执行，而是通过执行不同的代码部分来响应不同的事件。事件可以由用户、系统消息、甚至应用程序本身触发，这些事件的顺序决定着代码的执行顺序。

- 交互式的开发

不像传统的编程语言（如 C/C++ 等）的开发过程包括三个不同的步骤：写代码、编译、测试代码，Visual Basic 的开发过程是交互式的，Visual Basic 解释执行代码，所以你可以一边开发，一边执行，随时测试代码的效果。

有了这些概念，我们就可以通过一个实际工程的开发过程来感受一下 Visual Basic 的方

便与强大。

### 1.1.1 创建一个标准 EXE 项目工程

先了解一下集成开发环境。

单击 Windows “开始”按钮，指向“程序”，再指向 Microsoft Visual Studio 6.0 文件夹，单击 Microsoft Visual Basic 6.0 图标，就可以启动 Visual Basic 6.0，这时会出现“新建工程”对话框，提示新建工程的类型，如图 1-1 所示。

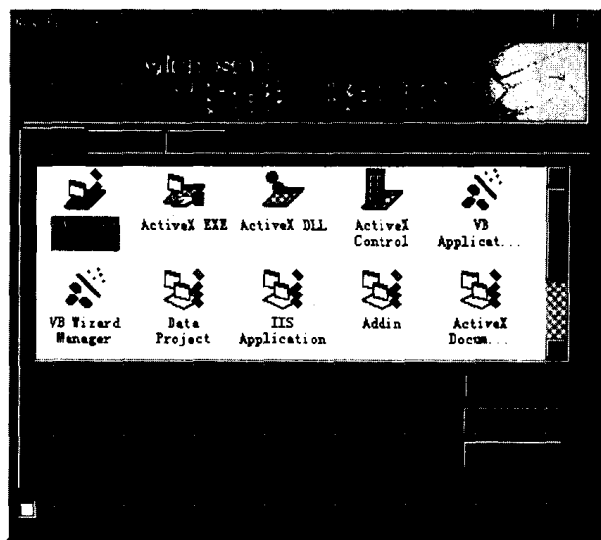


图 1-1 新工程对话框

选择“打开”，接受默认设置，即新建一个标准的 32 位 Visual Basic 应用程序。这时，一个新的工程连同一些窗口和工具被自动打开，如图 1-2 所示。

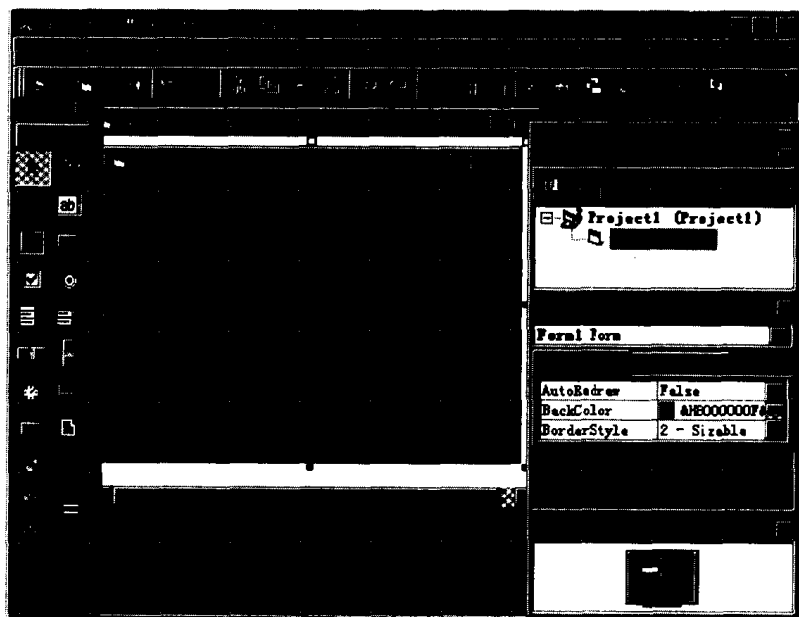


图 1-2 集成开发环境

可以看出,中间是用户界面窗体。在 Visual Basic 中,窗体就是一个自定义的窗口,在这个窗口上可以创建程序的用户界面。左边是工具箱,其中有各种工具或控件,可以在窗体中添加程序的用户界面部件。右上角是工程窗口,其中列出了所有在开发过程中用到的所有文件,可以通过工程菜单命令来添加、删除或保存不同的文件(模块)。工程窗口中可以一次装载结果工程文件,通过单击工程文件名,在他们之间来回切换。

工程窗口下面是属性窗口,可以改变用户界面部件的特性或属性集。属性集就是用户界面的某一对象的性质描述。属性窗口包括一个对象下拉列表框,其中列出了窗体上的所有用户界面部件(对象)。当建立用户界面时,可以通过属性窗口改变属性集,如按钮上的显示字符(Caption),或在程序运行时,通过代码改变它。

右下角是窗体布局窗口,用来调整设计窗口在屏幕上的位置。

上面这些窗口可以按照个人的喜好和方便调整位置和大小。

### 1.1.2 添加控件到窗体中

我们来编写一个幸运“6”游戏,就像老虎机,当随机产生的3位数字之一是“6”时,显示一副漂亮的图画。我们需要两个按钮,一个是游戏开始,另外一个退出。还有三个 Label,表示产生的三个随机数,还有一个时钟 Timer,定时(例如每20毫秒)的随机改变三个数字大小,点击“开始”按钮时,时钟启动,直到再次点击按钮。它还有一幅图片,如图1-3所示。

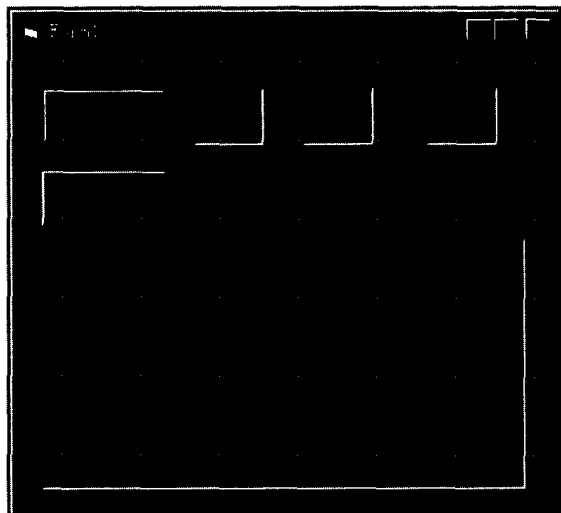


图 1-3 幸运的“6”界面

#### 一、编排控件

利用鼠标或键盘改变用户界面元素的大小和位置,也可以按住 Ctrl 键,使用鼠标同时点击选中几个对象(例如上图中的三个 Label),通过菜单“Format\Align”来调整选中对象,使它们上对齐(也可以左、右或下对齐),通过菜单“Format\Make Same Size”使得几个对象的宽度、高度或同时一样大。所有的安排调整都是为了界面美观和方便。

#### 二、设置控件属性

到目前为止,控件的几个关键属性如下面的表 1-1 所示。



表 1-1 控件的属性

| 对 象                  | 属 性         | 设 置 值                 |
|----------------------|-------------|-----------------------|
| Command1             | Caption     | 开始                    |
| Command2             | Caption     | 退出                    |
| Label1,Label2,Label3 | BorderStyle | 1-Fixed Single        |
|                      | Alignment   | 2-Center              |
|                      | Font        | MS Sans Serif 8-point |
| Label4               | Caption     | 幸运的"6"                |
|                      | Font        | 隶书-24                 |
| Picture1             | Picture     | "C:\Pic\Sky.bmp"      |
|                      | Stretch     | True                  |
|                      | Visible     | True                  |
| Timer1               | Interval    | 20                    |

### 1.1.3 编写事件响应代码

我们还要为命令按钮编写响应代码。当用鼠标点击 Command1 按钮（即“开始”）时，修改其 Caption 为“停止”，启动时钟，让 Label1、Label2、Label3 分别定时显示为三个随机的一位整数（通过修改其 Caption 属性实现），直到再次单击按钮，如果其中有一个是“6”，那么让图片显示出来（通过修改其 Visible 属性实现），再发出一声鸣叫声（Beep）。

用鼠标双击 Command1 按钮，显示 Form1 的代码编写窗口，在 Private Sub 和 End Sub 之间输入一些程序行，最后结果如下：

```
Private Sub Command1_Click()
    If Command1.Caption = "开始" Then
        Picture1.Visible = False ' 不让图片显示出来
        Command1.Caption = "停止"
        Timer1.Enabled = True
    Else
        Command1.Caption = "开始"
        Timer1.Enabled = False
        If (Label1.Caption = 6) Or (Label2.Caption = 6) Or (Label3.Caption = 6) Then
            Picture1.Visible = True
            Beep
        End If
    End If
End Sub
```

注意到程序行后面以 ' 开头的部分是注释，方便调试和阅读。

我们希望当鼠标点击 Command2（即“退出”）按钮时程序结束，这时只需要执行 End 语句即可，如下：

```
Private Sub Command2_Click()
```

```
End
```

```
End Sub
```

当程序刚开始时，三个 Label 不显示任何数字，同时图片隐藏起来。这部分代码是在窗体装载时执行，如下：

```
Private Sub Form_Load()
```

```
Label1.Caption = ""
```

```
Label2.Caption = ""
```

```
Label3.Caption = ""
```

```
Picture1.Visible = False
```

```
Timer1.Enable = False
```

```
End Sub
```

在时钟触发过程中，随机产生三个 0 到 9 之间的数字，显示在三个标签上，如下：

```
Private Sub Timer1_Timer()
```

```
Label1.Caption = Int(Rnd * 10)
```

```
Label2.Caption = Int(Rnd * 10)
```

```
Label3.Caption = Int(Rnd * 10)
```

```
End Sub
```

至此，全部代码编写结束，可以按热键 F5 调试运行，看看效果，碰碰运气。

### 1.1.4 保存工程

在菜单“File”中选择“Save Project As”命令，弹出一个对话框提示输入文件名和位置。输入 MyProject 后按回车。同样用菜单“File\Save Form1.frm As”可以保存窗体文件。下次启动 Visual Basic 时，可以将 MyProjec 工程文件调出来调试、运行。

### 1.1.5 生成可执行文件

利用菜单命令“File\Make MyProject.exe”可以生成可执行文件，该可执行文件可以做为 Windows 应用程序单独执行，而不依赖于 Visual Basic 集成开发环境。

## 1.2 Visual Basic 语言特性

Visual Basic 是“可视化 Basic”。它有两方面的含义：其一，“可视化”意思是当创建图形用户界面元素时，不用写大量的代码行来描述和控制界面元素的位置、大小等属性，只是简单的在屏幕上增加已经建好的对象，就像用“画笔”按照自己的构思一点点的添加元素，能立刻感受到变化；其二，Basic (Beginner All-Purpose Symbolic Instruction Code) 意思是对一个初学者来说可以实现任何目标的符号化指令代码，相对于 C/C++、Pascal 等来说是很基本和简单的。Visual Basic 从最早的 Basic 语言演变到现在也包括了许多的语句、函数和关键字，它们大多数是和 Windows 的图形用户界面 (GUI) 有关。

了解一门编程语言，首先必须了解用该语言编写应用程序的结构和框架。应用程序是

一些完成某（或一些）任务的指令集，其结构是这些指令的组织方式，包括如何存储和执行顺序。因为 Visual Basic 应用程序是基于对象的，所以代码结构与屏幕上的表示相对应。对象包括属性、方法和事件。屏幕上看到的窗体就是定义其大小和内部行为的特性的表示。对于每个窗体，用一个相关的模块（后缀名是.frm）来存储它的代码。

每个窗体模块包括事件过程——响应某特定事件时执行的指令。窗体包括控件，每个控件都有相应的一些事件过程。窗体模块中还可以包括窗体中任何事件响应过程可以调用的通用过程。

有时，不同的窗体之间有一些共同的代码要执行，这时，可以创建一个不同于.frm 的模块，即标准模块（后缀名是.BAS），它不和任何一个特定的窗体或控件关联，其中的过程或函数可以被任何一个窗体模块调用。

在 Visual Basic 中，还有另外一种模块：类模块（后缀名是.CLS），它是用来创建可被其他过程调用的对象的，和标准模块不同在于，它可以包括属性、方法和事件。

以上这三种模块中都可以包括两部分：

- 声明 在这部分可以定义变量、常量、常数、类型，窗体模块中要用到的动态连接库（DLL）过程在这里声明，还有类和标准模块等的声明。
- 过程 包括窗体中可以作为一个整体单元运行的子过程（Sub）、函数（Function）和属性过程（Property procedure）等。

**注意：**

职业版和企业版的 Visual Basic 还包括 ActiveX 文档、ActiveX 设计器和用户控件，这些模块用不同的后缀名来表示，从写代码的角度来说，这些模块可以看作和窗体模块一样。

现在来看看事件驱动（Event-Driven）的应用程序是怎样工作的。

Windows 操作系统是事件驱动的。顾名思义，事件驱动就是指令的执行由事件激发。Visual Basic 中的每个窗体和控件都有一组预先定义好的事件集，如果其中一个事件发生而且和它相关联的事件过程有代码，那么 Visual Basic 激活那段代码的执行。对象可以自动识别预定义的事件，但相应的过程得由程序员来编写。

典型的事件驱动的应用程序执行过程如下：

1. 应用程序启动，主窗体被装载并显示。
2. 窗体（或其中的控件）接收到一个事件，这可能是用户（例如使用键盘或鼠标等）、系统（例如时钟事件）或代码间接（例如 Load 事件）产生的。
3. 如果和该事件关联的过程有代码，则执行该过程。
4. 应用程序等待下一个事件。

**注意：**

有许多事件是同时发生的，例如当 Db1Click 事件发生时，MouseDown、MouseUp 和 Click 事件也会发生。

为方便使用，下面的表 1-2 列出了 Visual Basic 中常用的事件。

表 1-2 Visual Basic 控件的常用事件

| 事 件       | 何时发生               |
|-----------|--------------------|
| Change    | 用户修改了文本输入框或组合框文本内容 |
| Click     | 用户基本的鼠标按钮点击了对象     |
| DbClick   | 用户基本的鼠标按钮双击了对象     |
| DragDrop  | 用户拖动一个对象到其他位置      |
| DragOver  | 用户在一个对象上面拖动控件      |
| GotFocus  | 对象获得焦点             |
| KeyDown   | 当某个对象有焦点时用户按下键盘    |
| KeyPress  | 当某个对象有焦点时用户按下键盘并释放 |
| KeyUp     | 当某个对象有焦点时用户释放键盘    |
| LostFocus | 对象失去焦点             |
| MouseDown | 鼠标指针在对象上面时用户按下鼠标   |
| MouseMove | 鼠标指针在对象上面时用户移动鼠标   |
| MouseUp   | 鼠标指针在对象上面时用户释放鼠标   |

### 1.2.1 变量和常量

变量在程序中存储临时值。在使用变量之前，应该明确地声明，指定其数据类型，在内存中分配足够的空间。声明变量使用 Dim 语句：Dim varname [As type]，其中变量名 varname 是必需的，可以不指定变量类型，Visual Basic 自动设定该变量为变体数据类型（见下一节）。

变量也可以不用 Dim 语句声明，而直接通过变量赋值来创建一个变量。例如下面一个函数：

```
Function SafeSqr(num)
    TempVal = Abs(num)      ' 取绝对值
    SafeSqr = Sqr(TempVal)  ' 求平方
End Function
```

变量 TempVal 未经声明，Visual Basic 自动使用这个名字创建一个变量，就像被显式声明过一样。这非常方便，但可能因为拼写错误导致很微妙的错误，例如，在上面求平方的例子中，如果求平方的代码写成：SafeSqr = Sqr(TmpVal)，看上去没有什么不同，但是 TempVal 拼写时少了一个字母 e，这使得该函数总是返回 0，因为 Visual Basic 碰到 TmpVal 时，不知道是拼写错误还是一个新的变量，它总是为该名字创建变量，并赋初始值为 0。

为避免这种拼写错误，可以约定每个使用的变量都必须显式声明，这样，Visual Basic 在碰到一个未经声明的变量时，就警告提示。为了显式声明变量，只需要在一个标准模块、窗体模块或类模块的声明部分前加上下面的语句：Option Explicit。或者使用菜单命令“Tools\Option”，在“Editor”选项卡中选中“Require Variable Declaration”复选框，这样会自动的在任何新创建的模块中添加 Option Explicit 语句，但在已经创建好的模块中，还得手工添加。

一个变量的作用域是指该变量起作用的范围，即能够识别并访问该变量的那部分代码。

在一个模块中声明的变量只在该模块中所有的过程、函数中起作用，使用 **Public** 关键字修饰的变量在整个应用程序中有效。使用 **Static** 关键字修饰的变量当过程结束时保留原值。

当变量的作用范围不同时，变量的名字可以相同。在不同的过程中可以使用相同名字的变量，这些变量类型可以相同，也可以不同。在不同过程中定义的局部变量，实际是毫不相干的。变量的作用域还可以交叉，这时使用就近原则，就是说，使用最近的变量声明。例如下面一段程序：

```
Public Dim temp As Integer
Sub Form_Load
    Dim temp As Integer
    Temp = 12          ' 局部变量
    Form1.temp = 100   ' 显式指定用模块级变量
End Sub
```

局部变量 **temp** 和模块级变量 **temp** 冲突，以局部声明的变量优先。

有时程序中有些常数会经常出现，或者代码用到一些本身毫无意义的很难记住的数字，这时为了增加程序的易读性，也便于维护方便，可以使用常量。常量是一种不能修改和赋新值的变量。常量有两种类型，一种是内部或系统定义的常量，由应用程序和控件提供。用户也可以建立自定义的常量。

声明常量的语法是：

[Public/Private] Const 常量名 [As 类型] = 表达式

常量的类型和变量一样，可以是数值、字符或时间、日期等（具体见下一节）。例如：

```
Const PI = 3.1415926
```

```
Const Birthday = #01/17/75#
```

```
Private Const ErrorMessage = '内存分配错误'
```

常量也有作用范围，和变量是一样的。

## 1.2.2 数据类型

所有的变量和常量都有名字和数据类型，数据类型决定了变量和常量将如何存储。如果一个变量声明时没有指定数据类型，则使用缺省的类型：**Variant**，它就像变色龙，在不同的环境下可以表示许多不同的数据类型。当为一个类型为 **Variant** 的变量赋值时，不需要进行类型转换，**Visual Basic** 自动执行所有必须的转换。

在 **Visual Basic** 中，有一些定义好的数据类型，简单介绍如下：

### 一、用于数值计算的数字类型

计算机经常要作一些数字处理，所以一般的高级语言都预先定义好不同的数字类型。**Visual Basic** 支持以下几种数字型的数据类型：**Integer**（整型）、**Long**（长整型）、**Single**（单精度浮点型）、**Double**（双精度浮点型）和 **Currency**（货币型）等。

整数的运算速度较快，而且比其他类型占用内存要小。所以如果用户事先知道变量要存放整数，而不是带小数点的数字，就应当将它声明为 **Integer** 或 **Long** 型。如果变量包含小数，可以将其声明为 **Single**、**Double** 或 **Currency** 类型。**Currency** 支持小数点右边 4 位和小数点左边 15 位，它是一个精确的定点数据类型，适用于货币计算。

## 二、Byte（字节型）

如果变量包含二进制数据，可以声明其为 Byte（字节）数组类型。在进行 ANSI 和 UNICODE 格式转换时，用 Byte 变量存储二进制数据可以保留数据的原貌。如果使用 String 类型存储变量，在进行转换时，变量中的任何二进制数据都会遭到破坏。

## 三、适用于字符串操作的 String 类型

字符串变量类型的字符串长度是 0~255。默认情况下，String 变量是一个可变长度的字符串，随着对字符串赋予新值，它的长度可增可减。例如：

```
Public PromptString As String '长度不确定
```

然后，对其赋值：

```
PromptString = "请使用合法软件！"
```

如果要声明一个长度固定的可以使用下面的语法：

```
String * 字符串长度
```

例如，声明一个长度为 128 的字符串，可以下面语句：

```
Dim filename As String * 128
```

如果赋值一个长度小于 128 的字符串给 filename 变量，那么，Visual Basic 会在 filename 后面补上足够的空格以达到长度为 128，如果赋值一个大于定义长度的字符串给 filename，Visual Basic 会简单的截掉多余的字符。因为多余空格经常补在定长字符串结尾，处理这些变量时，函数 Trim 和 Rtrim 很有用处（这是去掉多余空格的函数，详情见 API 手册）。

在标准模块中，固定长度的字符串可以定义为 Public 或 Private，在窗体模块和类模块中，固定长度的字符串必须定义为 Private。

如果字符串表示的是一个数值，那么可以将该字符串赋值给数值变量，同样 Visual Basic 允许一个数值变量赋给字符串变量。例如在一个窗体中放置一个命令按钮、文本框和列表框，为命令按钮 Click 事件添加下面的一段代码：

```
Private Sub Command1_Click()  
    Dim intX As Integer  
    Dim strY As String  
    strY = "100.23"  
    intX = strY ' 把字符串值赋给数值型变量  
    List1.AddItem Cos(strY) ' 增加字符串中数值的余弦到列表框  
    strY = Cos(strY) ' 把余弦值赋给字符串变量  
    Text1.Text = strY ' 字符串变量显示在文本框中  
End Sub
```

Visual Basic 会自动的强制变量转换为合适的类型。字符串类型和数值类型相互转换时，最好使用警告提示，因为如果字符串中是非数值时会导致运行时错误。

## 四、Boolean 逻辑类型

如果一个变量包括简单的 yes/no,true/false 或 on/off 信息，可以声明它为 Boolean 类型。缺省时，Boolean 变量值是 False。下面的例子中，blnRunning 是 Boolean 类型变量，存储 yes/no 设置：

```
Dim blnRunning As Boolean ' 看看录音设备是否正在运转
```

```
If Recorder.Direction = 1 Then  
    blnRunning = True  
Endif
```

### 五、保存日期的 Date 类型

日期和时间可以用特定的 Date 时间类型或 Variant 类型来存储。Date 类型的变量存储为 64 位浮点数值形式，可以表示的日期范围从公元 100 年 1 月 1 日到公元 1999 年 12 月 31 日，而时间可以从 00:00:00 到 23:59:59。可以使用下面的方式赋值给 Date 类型的变量，注意日期文字必须使用符号 # 括起来，否则 Visual Basic 将不能正确的识别。

```
Dim today As Date  
today = #12/21/98#  
today = #1998-12-21 12:30:00 PM#  
today = #98,12,21#  
today = #December,21,1998#
```

当其他数值类型的变量转换为 Date 类型的变量时，Visual Basic 将使用小数点左边的值表示日期信息，而右边的值表示时间。Visual Basic 默认午夜为 0，中午为 0.5，并且使用负整数表示 1899 年 12 月 30 日之前的日期。例如下面的程序可以在用户单击窗体后显示一个由数字转化过来的日期：

```
Private Sub Form_Click()  
    Dim thisDate As Date  
    Dim thisValue As Double  
    thisValue = 36012.58  
    thisDate = thisValue  
    Print thisDate  
End Sub
```

程序运行结果如图 1-4 所示。

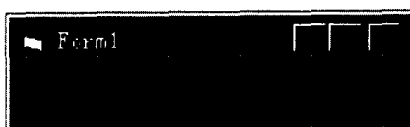


图 1-4 显示由数值转化过来的日期

### 六、对象 Object 类型

对象变量存储的是所指对象的 32-bit (4 字节) 地址。被声明为 Object 类型的变量可以通过赋值语句 (Set) 指向程序能识别的任何实际对象，例如：

```
Dim objDb As Object  
Set objDb = OpenDatabase ("C:\VB6\Biblio.mdb")
```

当声明对象变量时，尽量使用特定的类而不是通用的对象，例如使用 TextBox 而不是 Control，或者在前面的例子中，使用 Database 而不是 Object，这样，Visual Basic 可以在程序运行时确定对象的属性和方法。有关 Visual Basic 提供的类列在对象浏览器 (Object Browser) 中。

当使用其他应用程序的对象时, 尽量声明对象为对象浏览器中特定的类, 而不是 Variant 或通用的对象, 这保证了 Visual Basic 识别所指的特定类型的对象, 而且在运行时可以确定相关的引用。

### 七、万能的 Variant 类型

Variant 变量可以存储所有系统定义的数据类型的数据, Visual Basic 自动执行任何必须的转换。例如:

```
Dim SomeValue      ' 缺省为 Variant 类型
SomeValue = "17"    ' SomeValue 包含"17" (长度为 2 的字符串)
SomeValue = SomeValue - 15      ' SomeValue 现在是数值 2
SomeValue = "U" & SomeValue     ' SomeValue 变成"U2" (长度为 2 的字符串)
```

尽管处理 Variant 类型的变量时不需要太关心实际的数据类型, 有些陷阱应该避免: 如果要执行算术运算或函数, Variant 变量必须包含数值; 在合并字符串时, 使用 & 操作符而不是 +。

除了能表示其他标准的数据类型以外, Variant 变量还可以是三种特殊的值: Empty、Null 和 Error:

#### ● Empty

有时需要知道是否有个值已经赋给某个创建好的变量, Variant 类型变量在被赋值前有 Empty 值, 它和 0 值、长度为 0 的字符串 (“”) 或空值 Null 不同, 可以通过 IsEmpty 函数来测试, 例如:

```
If IsEmpty(Z) Then Z = 0
```

当 Variant 变量包含 Empty 值时, 可以在表达式中被当作 0 或长度为 0 的字符串来使用, 依赖于具体的环境。

任何值 (包括 0、0 长度字符串或 Null) 赋给 Variant 变量时, Empty 值立刻消失。也可以重新赋值为 Empty。

#### ● Null

Null 常用于数据库应用程序中表示未知或缺少数据, 它有某些唯一的属性:

- (1) 对包含 Null 的表达式, 计算结果总是 Null, 于是说 Null 通过表达式 “传播”。
- (2) 将 Null 值、含 Null 的 Variant 变量或计算结果为 Null 的表达式作为参数传递给大多数函数时, 将会返回 Null。

可以用 Null 关键字指定 Null 值, 例如:

```
Z = Null
```

如果要测试 Variant 变量是否包含 Null 值, 可以使用 IsNull 函数, 例如:

```
If IsNull(X) And IsNull(Y) Then
```

```
    Z = Null
```

```
Else
```

```
    Z = 0
```

```
Endif
```

如果将 Null 值赋给 Variant 以外的其他任何类型变量, 则将出现可以捕获的错误。

除非明确将 Null 赋给变量, 否则变量不会设置成 Null 值, 所以如果不在应用程序中使



用 Null, 就不必书写测试和处理 Null 的程序。

#### ● Error

Error 是用来指示在过程中出现错误时的特殊值。然而, 不像其他种类的错误那样, 程序并不产生普通的应用程序级的错误处理。这可以让程序员根据错误值采取另外的行动。可以用 CVer 函数将实数转换成错误值来产生 Error 的值。

### 1.2.3 对象

#### 一、什么是对象?

对象是可以被当作一个单元的代码和数据的组合, 它可以是应用程序中的窗体或控件, 也可以是整个应用程序。在 Visual Basic 中可以使用的对象的例子有: 命令按钮、窗体、数据库和图表等。

对象可以看作是现实生活中具体或抽象的东西, 它们有两个共同的特点: 第一, 它们都有自己的状态, 例如一辆车有大小、颜色、型号等特征; 第二, 它们都有自己的行为, 例如, 车可以加速前进或后退或停止等。现代的程序设计方法已经从以前的“算法+过程”进化到面向对象的方法, 即对现实生活中具体的对象模型化, 将其看作代码和数据的组合, 数据就是对象的属性, 而代码就是对象的方法, 对象的行为及状态可以被封装起来, 便于理解 (因为是现实模型的抽象)、管理 (因为不需要知道它具体是怎么实现的) 和重用 (因为对象之间可以有继承)。在 Visual Basic 中最常用的两种对象是窗体和控件。窗体是创建应用程序界面的基础, 像 Visual Basic 开发环境中所看到的 Form1 等; 控件是“工具箱”里一个个的小按钮, 可以按照需要添加按钮、列表框、时钟等控件。

创建对象需要模板, 类是对象的模板, 它是比对象更抽象的概念。类是用来创建对象的样板, 它包含所创建对象的状态描述和方法的定义、实现等, 而对象只不过是类的一个具体实例。

可以使用“车”的例子来说明类和对象之间的关系。在现实世界里, 我们可以看到丰田、林肯、桑塔那、解放、东风等车, 它们虽然看上去各有特色, 但是可以把它们抽象为一个类“车”。有自己的“属性”, 例如颜色、重量、型号等, 还有自己的“方法”, 例如加速、倒退、刹车等。虽然特征各不一样, 但都可以通过这些共同的抽象的属性及共同的行为方式来描述。我们甚至可以再进一步抽象出“小轿车”和“大卡车”等概念。

所有的对象创建时都同样是类的副本, 一旦作为一个个体对象存在, 它们的属性就可以改变。例如, 在窗体中添加三个命令按钮, 每个命令按钮都是类 CommandClass 的一个实例, 每个对象共享着类所定义的特征集和兼容性 (属性、方法和事件), 然而, 每个对象有自己的名字、可以被单独的 Enable 或 Disable、可以放在窗体的不同位置等等。

在 Visual Basic 集成开发环境中, “工具箱”中每个控件按钮都可以看作是一个类, 单击按钮后在窗体上画下的诸如按钮、列表框等元素, 都是由相应的类创建的对象, 这可以从属性窗口中看出来, 如图 1-5 所示。