

计算机技术入门提高精通系列丛书

Visual Basic 4.0 使用速成

Visual Basic 4.0

使用速成

张晓涪 董树怀 黄普 编著

12

F/1



人民邮电出版社

7/23/12
ZXF/1

计算机技术入门提高教程系列丛书

Visual Basic 4.0 使用速成

张晓涪 董树怀 黄 普 编著

人民邮电出版社

5.66 / 15

计算机技术入门提高精通系列丛书

Visual Basic 4.0 使用速成

◆ 编 著 张晓涪 董树怀 黄 普
责任编辑 俞 彬


◆ 人民邮电出版社出版发行 北京崇文区夕照寺街14号
北京顺义向阳胶印厂印刷
新华书店总店北京发行所经销

◆ 开本:787×1092 1/16
印张:18.75

字数:459千字 1997年9月第1版
印数:6 001-11 000册 1998年5月北京第2次印刷

ISBN 7-115-06679-5/TP·514

定价:28.00元



Visual Basic 4.0 是 Windows 3. x、Windows 95 和 Windows NT 应用程序的可视化开发环境,本书介绍用 Visual Basic 4.0 在 Windows 95 中文版下进行程序设计的技术。该书从初学者的角度,以开发几个具体的应用程序为线索,循序渐进地对利用 Visual Basic 4.0 开发 Windows 95 中文版应用程序的编程技术进行讲解,从设计菜单、表单、控件等用户界面到应用程序的具体功能,均有详细介绍。本书叙述简洁明快,适合于有一定编程基础的 Visual Basic 初学者学习。

Visual Basic 是美国微软公司推出的新一代可视化编程语言,本书是 Visual Basic 4.0 的入门教材。在写作本书时我们努力让本书具有下面几个特点:

● 这本书虽然涉及的方面不是很多,但特别注重 Visual Basic 基础概念的阐述和深入讲解,希望能帮助读者打下坚实的 Visual Basic 的编程基础。

● 本书在很多地方对于 Visual Basic 3.0 和 4.0 进行了比较。有些区别是很细微的,如果不明确指出也许读者并不能体会到。希望这些叙述能使 Visual Basic 3.0 的程序员在使用 4.0 版时少走一此弯路。

● 由于本书的入门性质,许多内容不便讲述,但是本书中许多地方都给读者指出了进一步的学习方向,如果读者把这些都掌握了就可以成为一名真正的 Visual Basic 程序员。

● 本书在讲解时力求跟上发展的步伐,对于一些不合潮流的内容尽量不讲或者只花费很少的笔墨。最明显的例子就是本书中除了在错误陷阱一节中讲到 On Error GoTo 语句涉及 GoTo 语句之外,任何地方都没有提到。GoTo 语句在我看来没有任何可利用的价值,它的害处总比益处多,因此也希望读者不要使用 GoTo 语句。

● 本书是结合例子讲述的,重要的例子包括“我的记事本”和“我的计算器”。书中一直在利用每一章新学到的知识完善“我的记事本”程序。在讲解例子时我们尽量花费较多的笔墨讲解编程原理和如何克服编程中的困难,至于程序的具体实现则希望读者能自己读懂程序代码。在程序清单中有极为详细的注释,会对读者有很大帮助。

本书中某些内容(例如附录)是针对 Visual Basic 4.0 Enterprise 版讲解的,与 Visual Basic 4.0 的标准版和专业版稍有不同,但在本书中绝大部分内容并不涉及到这种不同。

由于 Visual Basic 4.0 有 32 位版和 16 位版两种,虽然用户在编写程序时很少会涉及到两种版本中源代码不同的地方,但是实际上还是有许多地方要注意加以区分。用户可以为不同的版本编写两套源码,但更一般的方式是使用 # IF...Then... # Else 宏语句来告诉 Visual Basic 32 位和 16 位时应该执行哪些语句。可采用下面的格式:

```
# IF Win16 Then
    ' 16 位的 Windows 程序代码
# ELSE IF Win32 Then
    ' 32 位的 Windows 程序代码
# ELSE
    ' 其他操作系统平台的程序代码。
# End IF
```

一、本书中的约定

本书中许多地方提到选择菜单时使用下面的格式:

菜单|菜单项

菜单指菜单条上的菜单,菜单项指该菜单的弹出子菜单中的一个菜单项的名字。例如 Tools|Options... 意味着用户应该选择 Tools 菜单,然后选择 Options... 菜单项。

本书中讲到一些语句函数过程的格式时使用了一些符号。中括号[]表明括号中的部分是可以被省略的。竖线表示在竖线两边的部分可以任意选择一个。例如:Instr([开始位置,]字符串,特定的字符串[,查找方法])表明 Instr 函数在使用时开始位置和查找方法参数是可以省略的。下面的子程序定义语句:

[Private|Public] Sub 子程序名(参数表)

则指明 Private 和 Public 是只需要使用一个的,并且二者都可以省略。省略后 Visual Basic 的处理则依据不同的语句而有各自不同的缺省值。

二、如何成为一名 Visual Basic 编程专家

要成为一名 Visual Basic 编程专家,读者需要付出艰苦的努力。在读完本书之后,成为专家之前读者还需要学以下的內容:

- 在 Visual Basic 之中如何调用 Windows 动态链接库(*.DLL 或 *.EXE)中的过程。
- Visual Basic 之中使用 DDE 进行程序间的通信。
- Visual Basic 之中如何使用 OLE 技术。
- 使用 OLE 自动化技术,类封装技术。
- 编写作为 OLE 自动化服务器的动态链接库。
- 全面了解有关数据库的知识,编程方法。尤其深入了解 Jet 2.5/3.0 以及 SQL 语言。

| | |
|------------------------------------|----|
| ●第一章 Visual Basic 入门 | 1 |
| 1.1 Visual Basic 编程概念 | 1 |
| 1.1.1 Windows 程序的共同特点 | 1 |
| 1.1.2 Visual Basic 中的软件构件 | 1 |
| 1.1.3 对象 | 2 |
| 1.1.4 对象的属性和方法 | 4 |
| 1.1.5 事件驱动编程 | 5 |
| 1.2 Visual Basic 4.0 的新特色 | 7 |
| 1.3 安装 Visual Basic 4.0 | 10 |
| 1.4 从实例中学习 Visual Basic | 12 |
| 1.4.1 程序设计步骤 | 12 |
| 1.4.2 设计举例 | 13 |
| ●第二章 熟悉 Visual Basic 用户环境 | 17 |
| 2.1 主窗口 | 17 |
| 2.1.1 标题栏 | 17 |
| 2.1.2 菜单条 | 18 |
| 2.1.3 工具栏 | 18 |
| 2.2 表单 | 19 |
| 2.3 工具箱 | 20 |
| 2.4 属性窗口 | 22 |
| 2.5 项目窗口 | 24 |
| 2.6 代码窗口 | 25 |
| 2.7 Visual Basic 4.0 的联机帮助系统 | 26 |
| 2.7.1 用 Help 菜单启动帮助系统 | 26 |
| 2.7.2 帮助系统窗口 | 28 |
| 2.8 设置 Visual Basic 环境 | 29 |
| ●第三章 表单及控件的重要属性、事件和方法 | 31 |
| 3.1 表单和控件重要的属性 | 31 |
| 3.1.1 Name 与 Caption 属性 | 31 |
| 3.1.2 对象大小和位置的属性 | 33 |
| 3.1.3 有关字体的属性 | 34 |
| 3.1.4 关于颜色的属性 | 37 |
| 3.1.5 Enabled 与 Visible 属性 | 43 |

| | | |
|-------|---------------|----|
| 3.1.6 | 其他重要属性 | 44 |
| 3.1.7 | 表单的 ASCII 码表示 | 48 |
| 3.2 | 表单和控件的重要事件 | 50 |
| 3.2.1 | 焦点事件 | 50 |
| 3.2.2 | 鼠标和键盘事件 | 53 |
| 3.3 | 表单和控件的重要方法 | 61 |
| 3.4 | 表单的特殊属性和事件 | 67 |
| 3.4.1 | 表单特殊的重要属性 | 68 |
| 3.4.2 | 表单的特殊事件 | 69 |

● **第四章 Visual Basic 应用程序的基本结构** 70

| | | |
|-------|--------------------------|----|
| 4.1 | 模块类型 | 70 |
| 4.1.1 | 过程 | 70 |
| 4.1.2 | 表单模块和标准模块的组成 | 73 |
| 4.1.3 | 如何在编程时插入一个过程 | 73 |
| 4.1.4 | 过程中的代码 | 74 |
| 4.1.5 | 一点约定 | 75 |
| 4.2 | 数据类型与变量、常量 | 75 |
| 4.2.1 | 数据类型 | 76 |
| 4.2.2 | 变量 | 77 |
| 4.2.3 | 变量的作用域 | 80 |
| 4.2.4 | 静态变量 | 81 |
| 4.2.5 | 字符串 | 81 |
| 4.2.6 | 可变数据类型 (Variant) | 82 |
| 4.2.7 | 常量 | 83 |
| 4.2.8 | 自定义数据类型 | 84 |
| 4.2.9 | 过程调用中的参数的传递 | 85 |
| 4.3 | 一些基本语句 | 86 |
| 4.3.1 | 赋值语句 | 86 |
| 4.3.2 | 对象变量的声明和赋值 | 86 |
| 4.3.3 | Load 和 Unload 语句 | 87 |
| 4.3.4 | End 和 Stop 语句 | 88 |
| 4.3.5 | Beep 语句 | 89 |
| 4.3.6 | Exit Sub 和 Exit Function | 89 |
| 4.4 | 数组 | 89 |
| 4.4.1 | 多维数组 | 90 |
| 4.4.2 | 动态数组 | 90 |
| 4.4.3 | 数组的管理 | 91 |
| 4.4.4 | 控件数组 | 92 |

| | | |
|----------------------------------|-----------------------|-----|
| 4.5 | 运算符 | 94 |
| 4.5.1 | 数学运算符 | 94 |
| 4.5.2 | 关系运算符 | 95 |
| 4.5.3 | 字符串运算符 | 97 |
| 4.5.4 | 逻辑运算符 | 98 |
| 4.5.5 | 运算符的优先级 | 98 |
| 4.6 | 判断 | 99 |
| 4.6.1 | IF... Then 结构 | 99 |
| 4.6.2 | If... Then... Else 结构 | 100 |
| 4.6.3 | Select case 结构 | 101 |
| 4.7 | 循环 | 102 |
| 4.7.1 | Do... Loop 循环结构 | 102 |
| 4.7.2 | For... Next 循环结构 | 104 |
| 4.7.3 | For Each... Next 结构 | 104 |
| 4.8 | 程序流程控制结构的嵌套和强制退出 | 105 |
| 4.8.1 | 控制结构的嵌套 | 105 |
| 4.8.2 | 强制退出控制结构 | 106 |
| 4.9 | 一个例子:我的计算器 | 106 |
| ●第五章 Visual Basic 内部过程和函数 | | 114 |
| 5.1 | 字符串函数 | 114 |
| 5.1.1 | 建立字符串 | 114 |
| 5.1.2 | 分析字符串 | 115 |
| 5.1.3 | 操作字符串 | 116 |
| 5.2 | 数据类型转换函数 | 117 |
| 5.3 | 数学函数 | 120 |
| 5.4 | 金融函数 | 121 |
| 5.5 | 日期/时间函数 | 121 |
| 5.5.1 | 获取当前的日期和时间 | 121 |
| 5.5.2 | 设置当前的日期和时间 | 122 |
| 5.5.3 | 时间的计算 | 122 |
| ●第六章 菜单 | | 124 |
| 6.1 | 认识菜单 | 124 |
| 6.2 | 菜单编辑器 | 125 |
| 6.3 | 从实例中学习设计菜单 | 127 |
| 6.3.1 | 用菜单编辑器设计菜单框架 | 127 |
| 6.3.2 | 实现菜单选中状态的切换 | 131 |
| 6.3.3 | 实现菜单项允许和禁止状态的转换 | 131 |

| | | |
|-------------|-----------------------------|------------|
| 6.3.4 | 实现弹出式菜单 | 132 |
| 6.4 | 一个例子：“我的记事本” | 132 |
| 6.4.1 | 文本框 | 132 |
| 6.4.2 | 剪贴板 | 135 |
| 6.4.3 | 例子：我的记事本 | 136 |
| ●第七章 | 利用控件完成用户界面 | 144 |
| 7.1 | 有多个表单的项目 | 144 |
| 7.1.1 | 处理多表单的方法和语句 | 144 |
| 7.1.2 | 访问多表单程序中的数据 | 145 |
| 7.1.3 | 表单的显示方式 | 145 |
| 7.1.4 | 表单显示时的位置 | 146 |
| 7.1.5 | 改变启动表单 | 147 |
| 7.2 | 对话框 | 147 |
| 7.2.1 | 消息对话框 | 147 |
| 7.2.2 | 输入对话框 | 150 |
| 7.3 | 构造用户界面的常用控件 | 151 |
| 7.3.1 | 命令按钮 | 151 |
| 7.3.2 | 标签 | 152 |
| 7.3.3 | 文本框 | 152 |
| 7.3.4 | 一个小例子：十进制十六进制转换器 | 153 |
| 7.3.5 | 选项按钮 | 154 |
| 7.3.6 | 复选框 | 156 |
| 7.3.7 | 滚动条 | 156 |
| 7.3.8 | 列表框和组合框 | 158 |
| 7.3.9 | 其他常用的控件 | 162 |
| 7.4 | Visual Basic 常用定制控件介绍 | 162 |
| 7.4.1 | Windows 95 控件 | 163 |
| 7.4.2 | 系统功能控件 | 163 |
| 7.4.3 | 其他控件 | 163 |
| 7.5 | 多文档界面(MDI) | 164 |
| 7.6 | 为“我的记事本”加入查找功能 | 165 |
| ●第八章 | 文件操作 | 178 |
| 8.1 | 文件操作语句及函数 | 178 |
| 8.1.1 | 文件操作语句 | 178 |
| 8.1.2 | 文件操作函数 | 179 |
| 8.2 | 文件系统控件 | 180 |
| 8.2.1 | 文件列表框 | 181 |

| | | |
|-------------|---------------------------|------------|
| 8.2.2 | 目录列表框 | 182 |
| 8.2.3 | 驱动器列表框 | 183 |
| 8.2.4 | 协同所有文件系统控件 | 183 |
| 8.3 | 文件操作的公用对话框 | 184 |
| 8.4 | 存取文件 | 187 |
| 8.4.1 | 文件存取格式 | 187 |
| 8.4.2 | 打开和关闭 | 188 |
| 8.5 | 顺序文件 | 190 |
| 8.5.1 | 分隔域的顺序文件 | 190 |
| 8.5.2 | 不分隔域的顺序文件 | 192 |
| 8.5.3 | 文件长与文件末尾 | 194 |
| 8.5.4 | 向顺序文件末尾添加 | 195 |
| 8.6 | 随机文件 | 196 |
| 8.6.1 | 定义随机文件的域结构 | 196 |
| 8.6.2 | 读写随机文件 | 196 |
| 8.7 | 二进制文件 | 198 |
| 8.7.1 | 文件位置 | 198 |
| 8.7.2 | 读写二进制文件 | 198 |
| 8.7.3 | 怎样决定使用何种文件 | 199 |
| 8.8 | 为“我的记事本”加入文件操作功能 | 200 |
| ●第九章 | 利用系统资源 | 222 |
| 9.1 | 应用程序对象 | 222 |
| 9.1.1 | 版本信息属性 | 223 |
| 9.1.2 | 产品信息属性 | 223 |
| 9.1.3 | 运行信息属性 | 224 |
| 9.2 | 屏幕对象 | 224 |
| 9.3 | 打印机对象 | 225 |
| 9.4 | 打印设置公用对话框 | 227 |
| 9.5 | 定时器控件 | 228 |
| 9.6 | 与其他的 Windows 应用程序通信 | 230 |
| 9.6.1 | Shell 函数 | 231 |
| 9.6.2 | AppActivate 语句 | 231 |
| 9.6.3 | SendKeys 语句 | 232 |
| 9.6.4 | 综合运用 | 234 |
| 9.7 | 使用 Windows API | 234 |
| 9.8 | DoEvents 函数 | 235 |
| 9.9 | 键盘事件 | 235 |
| 9.10 | 鼠标事件 | 237 |

| | | |
|--------|-------------------------|-----|
| ●第十章 | 图形控件与绘图操作 | 239 |
| 10.1 | 图形基础 | 239 |
| 10.2 | Visual Basic 坐标系统 | 240 |
| 10.2.1 | 什么是坐标系统 | 240 |
| 10.2.2 | 定制坐标系统 | 241 |
| 10.3 | 图片框控件 | 243 |
| 10.3.1 | 在设计阶段装入图像 | 244 |
| 10.3.2 | 在运行阶段装入图像 | 244 |
| 10.3.3 | PaintPicture 方法 | 244 |
| 10.4 | 向图像框控件及表单中装入图像 | 247 |
| 10.4.1 | 使用图像框 | 247 |
| 10.4.2 | 在表单中显示图像 | 247 |
| 10.5 | 在表单和图片框中创建图形和文字 | 247 |
| 10.5.1 | Line 方法 | 248 |
| 10.5.2 | Circle 方法 | 250 |
| 10.5.3 | Pset 方法 | 251 |
| 10.5.4 | 显示文字 | 252 |
| 10.5.5 | 使用相对坐标 | 253 |
| 10.5.6 | DrawStyle 与 DrawMode 属性 | 254 |
| 10.6 | 线条与形状控件 | 254 |
| 10.7 | 一个简单动画的例子 | 255 |
| ●第十一章 | 程序调试及错误处理 | 259 |
| 11.1 | 程序调试 | 259 |
| 11.1.1 | 程序错误的类型 | 259 |
| 11.1.2 | 设计阶段、执行阶段和中断模式 | 260 |
| 11.1.3 | 调试窗口 | 261 |
| 11.1.4 | 设置断点 | 263 |
| 11.1.5 | 快速观察 | 264 |
| 11.1.6 | 调用对话框 | 264 |
| 11.1.7 | 程序跟踪 | 265 |
| 11.2 | 错误处理 | 266 |
| 11.2.1 | 错误陷阱 | 266 |
| 11.2.2 | 错误对象 | 268 |
| 11.2.3 | 用 Resume 语句恢复运行 | 269 |
| ●第十二章 | 数据访问 | 270 |
| 12.1 | 数据库概述 | 270 |

| | | |
|-----------|---------------|------------|
| 12.2 | 数据库中数据的访问 | 272 |
| 12.3 | 使用数据管理器 | 273 |
| 12.4 | 使用数据控件和数据敏感控件 | 277 |
| 附录 | 菜单 | 283 |

第一章 Visual Basic 入门

在 Visual Basic 出现之前,各种程序语言的编程概念基本是一致的,所不同的只是语言关键字、语法以及一些细节问题。Visual Basic 带来了编程的新思想、新概念和新方法,它非常适于编写 Windows 应用程序,使原本艰深复杂的 Windows 编程变得轻松多了。

在这一章中将要介绍 Visual Basic 的新编程概念,这些概念的建立将有助于读者了解到 Visual Basic 的精髓所在。此外,还将介绍一些 Visual Basic 4.0 新增的功能,以及 Visual Basic 4.0 的安装方法。

1.1 Visual Basic 编程概念

1.1.1 Windows 程序的特点

在开始学习 Visual Basic 之前,我们希望每位读者都能对 Windows 有一定的了解。因为 Microsoft 公司推出 Visual Basic 的目的就是为了方便用户开发 Windows 应用程序,熟悉 Windows 的读者可以更快地学会 Visual Basic。如果有的读者还不太熟悉 Windows,也不必为此感到气馁。我们并不要求读者有 Windows 编程的经验,这些读者只需花上几个小时,使用一下 Windows 以及几种 Windows 应用程序就可以了。重要的是在使用过程中对 Visual Basic 最终要完成的开发产品——Windows 程序建立起感性上的认识,这一点对于一个 Visual Basic 初学者也就足够了。

Microsoft 公司推出的 Windows 系列产品因其界面美观,使用方便,赢得了计算机用户的广泛青睐。现在各公司开发的 Windows 程序用“多如牛毛”这个词来形容一点也不过份,这些程序虽然各不相同,但它们之间存在着很多的相似相同之处。

所有的 Windows 程序中都有若干窗口,窗口可以放大、缩小,还可以移动。所有的窗口又是由一些基本的构件(Component)组成,这些构件包括:下拉式菜单、复选框、选项按钮、文本框、命令按钮等等。所以可以说,Windows 程序是由许多相同的基本构件以不同方式组合而成。

再进一步,各构件能对用户的操作作出反应。例如:用鼠标单击菜单名可打开一个下拉式菜单,一个选项按钮可在“开/关”状态间切换,在文本框中可输入并显示文本。

综上所述,可以认为所有 Windows 程序都是由这些“有反应的构件”组成。传统的 Windows 编程中,程序员必须为实现每一构件编写一大段代码,这对时间精力是多么大的浪费!因此,Microsoft 开发出了 Visual Basic 这一工具,把编程者从令人乏味的重复工作中解脱出来。

1.1.2 Visual Basic 中的软件构件

用某些编程者的术语来说,Visual Basic 是一种“应用程序的构成框架”。Windows 应用

程序的所有基本组成部分在 Visual Basic 中都可得到,编程者可以拿来即用,而不必专门为显示一个对话框而编上大段程序,这样就可以避免大量的重复劳动,把宝贵的时间花在实现程序的特殊功能上。

这种编程思想与小孩搭积木非常相似,Visual Basic 提供的软件构件就相当于积木块,用 Visual Basic 构件编程就相当于用积木块搭出各种各样的积木玩具。

用软件构件编写程序的方法越来越受到人们的重视。软件构件并不是一种新东西,它在过去的很长一段时间里就已有,只不过可能称呼不同而已。如今,Windows 编程的复杂性使之成为必须,而以 Visual Basic 为代表的高水平编程工具的发展使之成为可能。

软件构件是一种完成特定任务的“自包含”软件模块。它要完成的任务可能非常简单,也可能非常复杂。自包含这一点对于软件构件是非常重要的,因为每个构件具有独立性,使用者可以从中获得很大的好处,在编程时可以将各构件直接插入到程序中而不会有任何麻烦。可以利用的软件构件越多,编程就越容易。

在 Visual Basic 中把这些软件构件称为对象(Object),对象的概念在后面马上就会介绍,在这里我们仍沿用软件构件这一名词。Visual Basic 中可以使用三个层次的软件构件,第一层次是在 Visual Basic 安装时自带的一些构件,这些构件都是 Windows 应用程序最常见的组成部分,编程者可以直接把它们用到自己的程序中去。

第二层次中,Visual Basic 允许创建编程者自己的软件构件。当编程者逐渐获得更多的编程经验之后,可能会发现自己在不同的程序中重复干着相同的事情,这正是传统的 Windows 编程使人陷入困境之处。Visual Basic 向编程者提供了创建自包含构件的工具,编程者可以把经常重复的部分定制成一个构件,在重复使用时可以像前面所述那样,在程序中直接使用该构件。

Visual Basic 中第三层次的软件构件是一些定制的控制(Custom Control),这些构件被称为 OCX 或 VBX 控件。它们多数是由其他软件公司或个人为 Visual Basic 开发而成,使用方法与其他构件一样,用户可以购买获得,有些也可以免费获得。

Visual Basic 编程时有这么多现成的软件构件可用,所以比用 C++ 等传统语言编程要轻松多了,这正是 Visual Basic 吸引人之处。

虽然我们反复强调了软件构件的好处,但仅有软件构件还是不够的。继续阅读本书,读者就可以看到,只需编写少量的代码就能完成一个看起来非常不错的 Windows 窗口界面程序:程序中可以有多个窗口,每个窗口中可以有选项按钮、命令按钮、文本框、列表框等构件。但如果在程序中不编写自己的代码,那么除了看起来还不错以外,程序什么事也做不了,Visual Basic 编程必须完成两件事。

- 从 Visual Basic 提供的构件中挑选合适的来组成自己的程序。
- 编写 Basic 代码把各构件联系起来,完成特定的任务。

软件构件与代码结合才能组成完整程序。

1.1.3 对象

Visual Basic 对象(Object)也就是前面所述的软件构件,本书在前面之所以采用软件构件这一名词,是为了让读者建立“对象是 Visual Basic 程序的基本组成部分”这一概念。今后我们将使用对象这一专业术语来代替软件构件的说法。

Visual Basic 对象与 OOP(面向对象编程, Object Oriented Programming)中的对象是一样的,所不同的是以 C++ 为代表的 OOP 编程时,必须由编程者亲自设计编写对象的代码,而 Visual Basic 的对象大多是现成的。

那么在计算机程序中,什么是对象呢?在现实事物中, Object 是指物体,物体的概念是众人皆知的,桌子、板凳、房子和汽车等等这些都是物体。物体可以大到宇宙,可以小到细菌病毒,自然界中每种物体常常是由更小的物体构成的。比如说:地球是太阳系的一个构件,太阳系是银河系的一个构件,银河系又是更大的物体——宇宙的一个构件,这样的例子举不胜举。

计算机程序设计中的对象概念的产生正是受到了自然界的启发:为什么不能把现实世界中的事物反映到程序设计中去呢?这种想法最终成为了面向对象编程的指导思想。

面向对象编程是程序设计的最新形式,它受到了编程人员的广泛欢迎,这是因为它能够达到程序设计必须尽力达到的四个目标:

- 可检验性(verifiable);
- 可修正性(revisable);
- 可重复使用性(reusable);
- 可移植性(portable);

程序必须可以检验,这样才能证实程序达到了设计期望的功能;程序必须可修正,这样设计者以及使用者才可以调试修改;程序可以重复使用,这样就允许编程者只需写一次程序,今后需要时可以重新使用该程序;程序可移植,这样才能以最小的代价在不同的操作平台之间转换和移植。

对象是数据与数据操作方式的综合, Visual Basic 中使用的对象不再是通常而言的一种抽象的东西,而是一种类似于工具框。菜单按钮之类的真实的,可触摸的东西。 Visual Basic 对象中包含了数据以及处理数据的方法。 Visual Basic 为设计用户界面提供了许多预定义的对象,比如说我们在前面提到过, Visual Basic 的对象直接加到 Windows 程序中,就可以设计出命令按钮,文本框等 Windows 常见构件。这些对象都是 Windows 用户界面这一现实事物中可见的,确实存在的对象。每个对象都可完成某种特定的任务,例如,在文本框中可向程序输入数据,命令按钮可使程序完成相应的操作。

Visual Basic 中最重要的对象有表单(Form)和控件(control)这两种。表单就是窗口,它是构成程序与用户之间界面的基础,表单是一个包含有创建用户界面所需的其他对象以及应用程序代码的窗口。某些表单可以充满整个屏幕,其中还可以包含另外的表单,某些表单可以是小一些的定制对话框。如图 1.1 中所示,除了 text1 文本框控件以及 Command1 命令按钮控件以外,其余的部分都是表单。

控件也是读者很熟悉的标准用户界面对象。图 1.1 中的 text1 文本框以及 Command1 命令按钮就是常见的两种控件,其他控件对象还有列表框、菜单、滚动条等等,在这里就不一一列举了。

除了表单与控件之外, Visual Basic 4.0 中还有一些应用程序环境的系统资源对象,如剪贴板对象、屏幕对象和打印机对象等等,这些将在第九章中加以介绍。

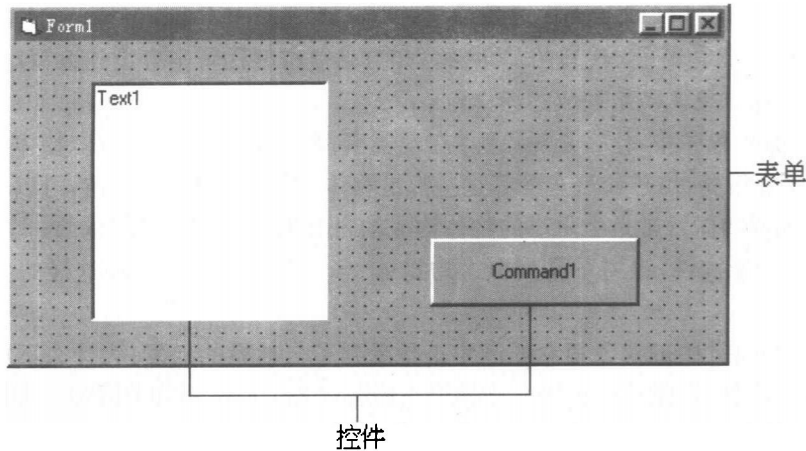


图 1.1 表单和控件

1.1.4 对象的属性和方法

一、属性

对象是一个抽象化的概念,只有在设置了对象的属性(Property)之后,才能生成真实的对象。对象的属性也可以认为是个别对象区别于其他对象的特征。例如,“人”是一个抽象的概念,人按性别可分为男人和女人,按年龄可以分为老年人、中年人、青年人和儿童,按肤色可分为白种人、黄种人和黑种人,性别、年龄和肤色就是对象“人”的属性。

所有的 Visual Basic 对象都各自有一个属性集合,对象的属性决定了它的外观和行为。例如,几乎所有的对象都有 Top 和 Left 这两个属性,它们决定了对象的左上角在屏幕中所处的位置,此外,Height 与 Width 这两个属性决定了对象的大小。有的属性普遍存在于大多数或全部对象之中,而有的属性只存在于个别对象之中。

对象的属性在 Visual Basic 程序与编程者的信息交流方面也起着重要作用。例如,如果一个编程者在文本框控件中输入文本之后,程序可以通过文本框控件的 Text 属性获得输入的文本,这里所说的程序是指编程者所写的 Basic 程序代码。

一个 Visual Basic 的使用者可能会发现,在编程时相当多的时间被花在了处理对象属性上,这也正说明了对象属性的重要性。用面向对象编程的观点来看,属性是对象的外部可见数据,即在对象外部可以存取的数据。编程者对于对象的了解和控制被限制在对象提供的属性之中,对象内部的结构究竟如何我们是无从知道的,也不需要知道。

还有一点需要提醒读者注意,由于对象内部的数据只能由对象自身来操作,所以我们不能直接改变对象的属性,只能利用给对象发送某些信息这种间接方式来改变对象的某个属性,这类似于以下情况:即某人想让别人做某事,他不能利用控制对方思想的方式来达到目的,但是可以采取给对方一些建议或请求的形式来实现。使用对象属性的方法是将属性名附在对象名之后,中间用小数点隔开,例如,如 Text1 是一个文本框对象,则可以用下列表达式访问 Text1 对象的 Text 属性:

```
Text1.Text
```

若表达式后面有一个等号和一个数据,则对象将内部的 Text 属性值改变成等号后指定