

计算机技术入门提高精通系列丛书

# Visual Basic 5.0

## 中文版 编程指南

康博创作室 编著  
孙江宏 审校



人民邮电出版社  
PEOPLE'S POSTAL  
TELECOMMUNICATIONS  
PUBLISHING HOUSE

312

c/1

7/P312  
KPC/1

计算机技术入门提高精通系列丛书

# Visual Basic 5.0 中文版 编程指南

康博创作室 编著

孙江宏 审校

人民邮电出版社

JS266/28  
04

计算机技术入门提高精通系列丛书  
**Visual Basic 5.0 中文版编程指南**

---

- ◆ 编 著 康博创作室  
审 校 孙江宏  
责任编辑 李 际
- ◆ 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: 14  
字数: 344 千字 1998 年 5 月第 1 版  
印数: 1—11 000 册 1998 年 5 月北京第 1 次印刷

ISBN 7-115-07061-X/TP • 657

---

定价: 22.00 元



## 内容提要

本书是一本介绍 Visual Basic 5.0 中文版的编程指南。全书共分 10 章。书中首先介绍了 Visual Basic 5.0 中文版的基本开发环境和编程基础;然后介绍了窗体和控件的使用方法、基本图形用户界面的设计,以及 Visual Basic 5.0 的绘图功能;最后以几个小程序为例,介绍了 Visual Basic 5.0 中文版的开发过程。

本书内容翔实,叙述生动,示例丰富,适用于那些想尽快掌握 Visual Basic 5.0 中文版新功能的用户,同时也可作为广大 Visual Basic 程序员的简明参考手册。

## 前 言

Visual Basic 是 Microsoft 公司推出的一种简单、直观的编程语言。Visual Basic 5.0 版是其目前的最新版本,它完备的功能以及众多的新特征使它一经面市便受到广大用户的喜爱。

本书是在 Visual Basic 5.0 中文版的基础上完成的。书中首先简述了 Visual Basic 的历史及 Visual Basic 5.0 的开发环境,介绍了 Visual Basic 的编程基础知识,讨论了窗体的属性以及如何在窗体上操纵控件、基本的图形用户界面的组件和设计方法,涉及到命令按钮、标签、文本框、复选框、单选框、框架、滚动条、列表框、组合框等;然后介绍了 Visual Basic 图形用户界面中的高级元素——菜单和对话框的设计过程;接下来,讲解了如何用 Visual Basic 绘制图形;最后介绍了图标时钟和地址簿应用程序的编制过程,编制这两个小程序用到了前面介绍的各种知识,并且逐步引入了程序设计中的各种方法和技巧。

本书是我们多年来运用 Visual Basic 进行编程工作的简要总结。Visual Basic 是一种功能强大的计算机编程语言,我们力图用较小的篇幅简明扼要地讲解许多关键的功能和特征,希望读者能从中获益。本书是集体合作的结晶,参加本书编写、资料收集以及整理工作的有冯志强、曹康、张强、刘芝泉、游国权、常征、周秀会、陈君、王睿、蔡汇锦、曹锋、刘亚军、张涛、钟杰、李东升、李志诚、李建锋、朱琳、李娟、刘秀英、陈则贵、徐学虎、刘兵、陈红等。

由于编者水平有限,书中难免存在错误和不足之处,恳请广大读者批评指正。

编者

1998 年 2 月

|                                           |    |
|-------------------------------------------|----|
| ●第一章 Visual Basic 简介 .....                | 1  |
| 1.1 Visual Basic 发展简史 .....               | 1  |
| 1.2 Visual Basic、VBA 和 VBScript 的比较 ..... | 2  |
| 1.2.1 VB 和 VBA 的比较 .....                  | 2  |
| 1.2.2 最新的 Visual Basic——VBScript .....    | 3  |
| 1.2.3 何时使用何种 Visual Basic .....           | 4  |
| 1.3 Visual Basic 5.0 的 8 种工程 .....        | 4  |
| 1.4 小结 .....                              | 5  |
| ●第二章 Visual Basic 5.0 的开发环境 .....         | 7  |
| 2.1 启动 Visual Basic .....                 | 7  |
| 2.2 Visual Basic 的系统环境 .....              | 8  |
| 2.2.1 主窗口 .....                           | 9  |
| 2.2.2 窗体 .....                            | 11 |
| 2.2.3 工程窗口 .....                          | 12 |
| 2.2.4 属性窗口 .....                          | 13 |
| 2.2.5 快捷菜单 .....                          | 14 |
| 2.3 菜单的使用 .....                           | 14 |
| 2.3.1 “文件”菜单 .....                        | 14 |
| 2.3.2 “编辑”菜单 .....                        | 17 |
| 2.3.3 “查看”菜单 .....                        | 20 |
| 2.3.4 “工程”菜单 .....                        | 22 |
| 2.3.5 “格式”菜单 .....                        | 24 |
| 2.3.6 “工具”菜单 .....                        | 25 |
| 2.4 小结 .....                              | 30 |
| ●第三章 Visual Basic 5.0 编程基础 .....          | 33 |
| 3.1 代码编辑器窗口及示例 .....                      | 33 |
| 3.2 Visual Basic 5.0 编程基础 .....           | 44 |
| 3.2.1 代码行的分段及合并 .....                     | 44 |
| 3.2.2 使用注释 .....                          | 44 |
| 3.2.3 Visual Basic 5.0 中的数制 .....         | 45 |
| 3.2.4 Visual Basic 5.0 的命名约定 .....        | 45 |
| 3.3 Visual Basic 5.0 中的数据结构和类型 .....      | 45 |

|       |                               |    |
|-------|-------------------------------|----|
| 3.3.1 | Visual Basic 5.0 中的数据类型 ..... | 45 |
| 3.3.2 | Visual Basic 5.0 中的变量 .....   | 49 |
| 3.3.3 | 常量 .....                      | 50 |
| 3.4   | Visual Basic 5.0 中的控制语句 ..... | 51 |
| 3.4.1 | 判断语句 .....                    | 51 |
| 3.4.2 | 循环语句 .....                    | 53 |
| 3.5   | Visual Basic 5.0 中的数组 .....   | 55 |
| 3.5.1 | 静态数组 .....                    | 55 |
| 3.5.2 | 动态数组 .....                    | 56 |
| 3.5.3 | 使用数组 .....                    | 56 |
| 3.6   | Visual Basic 5.0 中的过程 .....   | 57 |
| 3.6.1 | 过程的声明 .....                   | 57 |
| 3.6.2 | 过程的使用 .....                   | 58 |
| 3.6.3 | 过程的参数传递 .....                 | 58 |
| 3.7   | 小结 .....                      | 59 |
| ●第四章  | 创建窗体和控件 .....                 | 61 |
| 4.1   | 开发工程的基本步骤 .....               | 61 |
| 4.2   | 窗体 .....                      | 62 |
| 4.2.1 | 窗体的建立、添加和删除 .....             | 62 |
| 4.2.2 | 窗体的属性 .....                   | 62 |
| 4.2.3 | 窗体的事件 .....                   | 65 |
| 4.2.4 | 样本示例:设计一个显示不同大小字体的窗体 .....    | 67 |
| 4.3   | 使用控件 .....                    | 68 |
| 4.3.1 | 什么是控件 .....                   | 69 |
| 4.3.2 | 定制控件 .....                    | 70 |
| 4.3.3 | 向窗体添加控件 .....                 | 71 |
| 4.3.4 | 控件的操作 .....                   | 71 |
| 4.4   | 小结 .....                      | 73 |
| ●第五章  | 建立界面 .....                    | 75 |
| 5.1   | 命令按钮、标签和文本框的设计 .....          | 75 |
| 5.1.1 | 命令按钮 .....                    | 75 |
| 5.1.2 | 文本框 .....                     | 79 |
| 5.1.3 | 标签 .....                      | 82 |
| 5.1.4 | 加减法算术运算器 .....                | 84 |
| 5.2   | 复选框、单选框和框架 .....              | 86 |
| 5.2.1 | 复选框 .....                     | 86 |
| 5.2.2 | 单选框 .....                     | 87 |
| 5.2.3 | 框架 .....                      | 88 |
| 5.2.4 | 文本编辑器的设计 .....                | 89 |
| 5.3   | 滚动条、列表框和组合框 .....             | 92 |

|                                   |     |
|-----------------------------------|-----|
| 5.3.1 滚动条 .....                   | 92  |
| 5.3.2 列表框 .....                   | 94  |
| 5.3.3 组合框 .....                   | 96  |
| 5.4 小结 .....                      | 97  |
| ●第六章 添加菜单和对话框 .....               | 99  |
| 6.1 建立菜单 .....                    | 99  |
| 6.1.1 建立菜单标题 .....                | 101 |
| 6.1.2 控件与控件名 .....                | 102 |
| 6.1.3 建立 Exit 项 .....             | 102 |
| 6.1.4 将代码添加到菜单项 .....             | 103 |
| 6.1.5 添加 Print 菜单项 .....          | 103 |
| 6.2 完成菜单 .....                    | 104 |
| 6.2.1 将分隔线插入菜单 .....              | 104 |
| 6.2.2 添加控件数组 .....                | 104 |
| 6.2.3 更改颜色 .....                  | 106 |
| 6.2.4 复选菜单项 .....                 | 107 |
| 6.3 弹出式菜单 .....                   | 109 |
| 6.4 MsgBox 函数 .....               | 110 |
| 6.4.1 显示消息 .....                  | 110 |
| 6.4.2 函数返回值 .....                 | 112 |
| 6.5 InputBox 函数 .....             | 113 |
| 6.4 小结 .....                      | 115 |
| ●第七章 绘制图形 .....                   | 117 |
| 7.1 Visual Basic 的坐标系统和绘图模式 ..... | 117 |
| 7.1.1 为什么 Windows 坐标是倒置的 .....    | 118 |
| 7.1.2 什么是缇 .....                  | 118 |
| 7.2 在窗体中画线 .....                  | 121 |
| 7.2.1 Line 控件的属性 .....            | 121 |
| 7.2.2 Line 命令 .....               | 123 |
| 7.3 Step 关键字的说明 .....             | 126 |
| 7.4 CurrentX 和 CurrentY 属性 .....  | 127 |
| 7.5 设置刻度 .....                    | 129 |
| 7.6 其他绘制图形的方法 .....               | 130 |
| 7.6.1 Pset 方法 .....               | 130 |
| 7.6.2 Shape 控件 .....              | 131 |
| 7.6.3 Circle .....                | 132 |
| 7.7 小结 .....                      | 133 |
| ●第八章 建立时钟程序 .....                 | 135 |
| 8.1 设计图标时钟 .....                  | 135 |
| 8.2 处理图标 .....                    | 136 |

|                                     |     |
|-------------------------------------|-----|
| 8.2.1 设置标题 .....                    | 136 |
| 8.3 读时钟 .....                       | 136 |
| 8.4 使用计时器 .....                     | 137 |
| 8.4.1 建立计时器 .....                   | 137 |
| 8.4.2 设置计时器 .....                   | 138 |
| 8.5 显示时间 .....                      | 139 |
| 8.5.1 使用时间函数 .....                  | 140 |
| 8.5.2 从日期数字中获取信息 .....              | 141 |
| 8.6 绘制表盘 .....                      | 142 |
| 8.6.1 设置图标 .....                    | 142 |
| 8.6.2 绘制时钟指针 .....                  | 143 |
| 8.6.3 显示秒针 .....                    | 145 |
| 8.6.4 用颜色绘图 .....                   | 146 |
| 8.6.5 使用 Xor 删除线 .....              | 146 |
| 8.6.6 绘制时针和分针 .....                 | 148 |
| 8.7 相关函数 .....                      | 149 |
| 8.7.1 Format\$ 函数 .....             | 149 |
| 8.7.2 时间/日期函数 .....                 | 150 |
| 8.8 小结 .....                        | 150 |
| ●第九章 文件操作 .....                     | 151 |
| 9.1 处理磁盘文件 .....                    | 151 |
| 9.1.1 建立磁盘文件 .....                  | 152 |
| 9.1.2 写入文件 .....                    | 153 |
| 9.1.3 读文件 .....                     | 155 |
| 9.1.4 关闭文件 .....                    | 157 |
| 9.2 数据组包:用户自定义类型 .....              | 157 |
| 9.2.1 定义新的类型 .....                  | 158 |
| 9.2.2 设计 AddressInfo 类型 .....       | 159 |
| 9.2.3 定长字符串 .....                   | 160 |
| 9.2.4 读取和写入复合变量 .....               | 161 |
| 9.2.5 去掉字符串末尾的空格 .....              | 162 |
| 9.3 创建新的子例程 .....                   | 163 |
| 9.3.1 DBOpen 子例程 .....              | 163 |
| 9.3.2 建立 DBClose 子例程 .....          | 164 |
| 9.3.3 建立 DBRead 和 DBWrite 子例程 ..... | 164 |
| 9.4 小结 .....                        | 165 |
| ●第十章 对多个记录的处理 .....                 | 167 |
| 10.1 建立模块 .....                     | 167 |
| 10.1.1 在其他窗体内部引用控件 .....            | 168 |
| 10.1.2 使用子例程参数 .....                | 170 |

|         |                                   |     |
|---------|-----------------------------------|-----|
| 10.1.2  | 修改数据库模块 .....                     | 171 |
| 10.2    | 多记录方式的 Database.bas 的修改 .....     | 171 |
| 10.2.1  | 在 DBOpen 中添加支持多个记录的功能 .....       | 171 |
| 10.2.2  | 测试代码 .....                        | 172 |
| 10.2.3  | 建立一个 DBLen() 函数 .....             | 173 |
| 10.2.4  | 修改 DBRead 子例程 .....               | 174 |
| 10.2.5  | 修改 DBWrite 子例程 .....              | 175 |
| 10.2.6  | 在 DBClose 中增加注解 .....             | 175 |
| 10.3    | 向窗体中添加多个记录 .....                  | 176 |
| 10.3.1  | 添加窗体级变量 .....                     | 176 |
| 10.3.2  | 修改 GetRecord .....                | 176 |
| 10.3.3  | 建立空记录 .....                       | 178 |
| 10.3.4  | 修改 SaveRecord .....               | 179 |
| 10.3.5  | 注意记录的变化 .....                     | 180 |
| 10.3.6  | 建立菜单 .....                        | 180 |
| 10.3.7  | 添加新记录 .....                       | 182 |
| 10.3.8  | 遍历记录 .....                        | 182 |
| 10.3.9  | 使 Exit 菜单项有效 .....                | 183 |
| 10.3.10 | 支持 Cut、Copy、Paste 和 Undo 命令 ..... | 184 |
| 10.4    | 小结 .....                          | 185 |
| ●附录 A   | 键盘使用指南 .....                      | 187 |
| A.1     | 代码编辑键 .....                       | 187 |
| A.2     | 代码窗口常用键 .....                     | 187 |
| A.3     | 代码窗口快捷键 .....                     | 188 |
| A.4     | 代码窗口浏览键 .....                     | 189 |
| A.5     | 窗体窗口键 .....                       | 189 |
| A.6     | 全局键 .....                         | 190 |
| A.7     | Windows 键 .....                   | 190 |
| A.8     | 立即窗口快捷键 .....                     | 191 |
| A.9     | 菜单编辑键 .....                       | 191 |
| A.10    | 列表键 .....                         | 192 |
| A.11    | 代码窗口中起作用的快捷键 .....                | 192 |
| A.12    | 工程管理器组合键 .....                    | 193 |
| A.13    | 属性窗口快捷键 .....                     | 193 |
| A.14    | 工具箱键 .....                        | 194 |
| A.15    | 监视窗口中的快捷键 .....                   | 195 |
| ●附录 B   | Visual Basic 5.0 中的运算符 .....      | 197 |
| B.1     | 关系运算符 .....                       | 197 |
| B.2     | 算术运算符 .....                       | 197 |
| B.3     | 逻辑运算符 .....                       | 198 |

|                                 |     |
|---------------------------------|-----|
| B.4 文本运算符 .....                 | 199 |
| ●附录 C 变量类型 .....                | 201 |
| C.1 Visual Basic 中的变量类型 .....   | 201 |
| C.2 Windows API 的变量类型 .....     | 201 |
| ●附录 D Visual Basic 中的常用函数 ..... | 203 |

## Visual Basic 简介

今天,在计算机界中,图形用户界面操作系统如 Windows 等已经基本上占据了市场,传统的文本方式下的 DOS 在逐渐退出历史舞台,C:> 提示符再也无法引起人们的兴趣。一幅图片胜过千言万语,带有图标图形界面工作平台为广大计算机用户提供了发挥个人才能的广阔天地。人们再也不必为了输入一个自己并不熟悉的文本命令而浪费时间了。这就是当今的图形用户界面(GUI)的意义。

然而,人们是不会轻易满足的。当用户在使用方面的随意性得到了最大限度的满足以后,在个人开发方面的随意性要求便逐渐地强烈起来。同时,人们发现自己正在逐渐走入一个使用计算机的误区:偏离了主要的工作任务,却把大量的精力和时间花在了界面上。开发 Windows 应用程序要比开发 DOS 应用程序困难得多。因此,人们迫切需要一个高效的基于 GUI 的应用程序开发工具,使自己摆脱困境,以轻松、便利和愉快的方式完成自己的设计构想。

这就是 Visual Basic 出现的背景。

### 1.1 Visual Basic 发展简史

1991 年,Microsoft 公司在亚特兰大的 Windows World'91 中发布了第一个 Visual Basic 版本,即 1.0 版。它提供了比 Quick Basic 强大得多的功能。这是 Windows 程序开发者的一大福音,从那时起,编程界发生了巨大的变化。人们完全跳出了以前 C 语言程序员那种繁复和茫茫无期的开发工作,进入了一个全新的天地。Windows 程序员终于有了一种简单、直观而又功能强大的编程语言。

Visual Basic 是 RAD(Rapid Application Development,快速应用程序开发)系列的第一个产品。这些产品使普通用户能够开发复杂的 Windows 应用程序。Visual Basic 出现之后,得到了迅速的发展,1992 年,Microsoft 公司推出了 Visual Basic 2.0。1993 年,Visual Basic 3.0 问世。1995 年,Visual Basic 4.0 推出。目前,人们所见到的最高版本是 Visual Basic 5.0。我们相信将来人们还会看到 Visual Basic 更新的版本诞生。

Visual Basic 允许用户将菜单、文本框、命令按钮、选项按钮、复选框、文件和目录框增加到空白的窗体中,利用网格处理表格数据,与其他 Windows 应用程序进行通信,以及访问数据库。用户可以在屏幕上

显示多个窗口,这些窗口能访问剪贴板并能访问同时在运行的其他大多数 Windows 程序中的信息。

使用 Visual Basic,您可以在短时间内成为 Windows 程序员。它打破了传统的设计方式,程序员不需要再跟着程序的流程循序渐进地开发,而是可以根据不同的事件运行不同的过程。

在 Visual Basic 的运行过程中,直到某个函数或模块被调用时才将这个函数调入内存,当然也允许多个程序使用该函数,这样就大大节省了内存空间。这就是我们常说的动态链接方式。它有很多好处:

- ① 节省内存空间,共享程序资源;
- ② 提高程序执行速度;
- ③ 可以任意扩充程序;
- ④ 可集成的 DLL 资源广阔。

Visual Basic 利用了动态数据交换(Dynamic Data Exchange, DDE)和对象链接与嵌入(Object Linking and Embedding, OLE)技术,实现了不同程序之间的数据交换和对象共享的目的。

可以说,Visual Basic 的程序设计概念就是面向对象的概念。对象就是数据(data)和程序代码(code)互相结合的综合体。Windows 上面的每一个图标,包括窗口本身都是对象。如果没有任何事情发生,对象处于停顿状态。当存在外来事件时,程序段执行。它的执行顺序是由外来事件决定的,因此是“事件驱动”的。

编写 Visual Basic 程序非常简单。首先将各个对象放在空白窗体上,然后将程序代码分别添加给对象或图标,将它们组合起来,就可以随意运行了。

虽然 Visual Basic 使开发 Windows 程序很容易,但是经过几年的发展,它已经发展成为一种真正专业化的开发语言和环境。

## 1.2 Visual Basic、VBA 和 VBScript 的比较

目前,Microsoft 公司已经开发了 3 种面向 Windows 的 BASIC 版本:Visual Basic、Visual Basic for Application 和 VBScript。

Visual Basic 是 Microsoft 公司试图为 Windows 产品创建一个通用的宏语言。VBA 是基于高度成功的 Visual Basic 产品的。这个第二代版本同 VB 一样易用,而且两者共享大多数核心功能。

虽然 Visual Basic for Application(VBA)在 1993 年才首次发行,却因为与 VB 的联系而迅速成为一种成熟的语言。Microsoft 公司允许其他公司购买并将 VBA 包含在其内部的应用程序中。这意味着,当前的 VB 和 VBA 代码可以移植到其他应用程序中,而这些程序可以由 Microsoft 公司开发,也可以由其他公司开发。

VBScript 是 Visual Basic 的微缩版本,可以将它看作批处理语言。它使用 VB 的基本语法。VBScript 代码通常直接嵌入在 Web 页中,就像 JavaScript 一样,但它不一定必需如此。

### 1.2.1 VB 和 VBA 的比较

VB 和 VBA 在许多重要的方面很相似,尤其是具有一些共享功能,而这个特性使 VBA 成为一种强大的语言。它们共享如下功能:

- ① 同 OLE 紧密结合(包括增强的 OLE Automation 支持);
- ② 具有 Visual Basic 语言基础子集的 VBA 类型库;
- ③ 能使用为 VB4.0 或更高版本开发的 ActiveX 控件;
- ④ 使用同样的 VBE(Visual Basic Editor)界面,使不同的实现保持一致性。

有共同点,并不意味着两者完全相同,这样就没有什么意义了。表 1.1 列出了 Visual Basic 和 VBA 的特征和区别。

表 1.1 VB 和 VBA 的特征

| 语 言                          | 定 义                                                                                                                                    |
|------------------------------|----------------------------------------------------------------------------------------------------------------------------------------|
| Visual Basic                 | VB 最重要的特征是它可以创建独立的可执行文件或自动化服务器,可以将它们发行给不拥有 Microsoft 应用程序的用户                                                                           |
| Visual Basic for Application | 1993 年秋作为替代 Microsoft Excel 的 XLM 宏语言出现,VBA 97 现在包含在 Access,Excel,PowerPoint 和 Word 中,其他产品也会带有 VBA;VBA 和 VB 之间最重要的不同是 VBA 需要另外的产品来开发代码 |

有一个简单的方法来区分 VB 和 VBA,VB 是独立的编程产品,可以生成可独立执行的文件,而 VBA 是应用程序的宏控制语言。

从 Visual Basic 5.0 开始,VB 包含了 VBA 的所有功能(以共享类型库的形式)。完全用 VBA 编写的代码可以被所有使用 VBA97 类型库的产品共享。

### 1.2.2 最新的 Visual Basic——VBScript

VBScript 使用 VBA 类型库中功能的一个子集。它的代码是解释型的。

Microsoft 公司开发 VBScript 的主要目的是提供与 Microsoft Windows 开发工业标准(即 Visual Basic)兼容的 Web 语言。然而,新的语言要嵌入 Web 页中,因此它必须缩得足够小以致于能同 JavaScript 的解释过程一样快。为了达到这个目的,Visual Basic 中较少使用的功能都被删除了。

VBScript 是 VB 的一套小型的基本功能,用它们可以共同完成更复杂的任务。这意味着用 VBScript 编程比用 VB 困难一些,并且需要更多的创造性。

VBScript 对 VB/VBA 产品缩小的另一个原因是对安全方面的考虑。如果 Microsoft 公司让 VBScript 程序员调用 Windows API 或执行文件 I/O,那么一些“黑客”可以轻易用 VBScript 编写病毒,这将破坏所有执行 VBScript 页面的系统。因此,Microsoft 公司将 VB 中的所有存在被滥用潜在危险的功能都删除掉了。

VBScript 同其他版本的 VB 的最大不同在于 VBScript 只支持 Variant 数据类型。因为这样简了解释器的代码,所以做了这样的更改会使执行更快。然而,这个修改并不是对 VB 程序员的限制,因为 Variant 数据类型十分灵活,它可以支持 VB 使用的所有数据类型。

VBScript 的另一个主要删节是 VBA 类型库中的许多预定义常量。为了使 VB 代码尽可能易于移植,在 VBScript 中删除预定义的常量是必要的。虽然 VBScript 不支持定义常量,但是可以定义相同的名字的变量,从而使原来的 VB 代码实际上未做改动。

VB 最重要的功能都包含在 VBScript 中,因此仍然可以用 VBScript 编写功能强大的程序。唯一不同的是使用 VBScript 编程必须更具创造性,这要求用户以新的方式思考。用户也许会发现为克服这些限制而发明的技巧非常好,以至于想将它们移植到 VB 应用程序中。VBScript

是新的挑战,但它很有趣。

要在 Web 页面中使用 VBScript,必须加入 HTML 代码,它通知浏览器 Web 页支持 VBScript。此外,VBScript 不支持在 VB 或 VBA 中习惯使用的窗体。这意味着代码中必须包括有关使用哪一种控件和将它们放在什么地方的信息。

### 1.2.3 何时使用何种 Visual Basic

在什么场合使用什么样的 Visual Basic 是由用户的需要决定的,如表 1.2 所示。

表 1.2 各种语言的用途

| 语 言                          | 用 途                                                                                                               |
|------------------------------|-------------------------------------------------------------------------------------------------------------------|
| Visual Basic                 | 如果应用程序需要最大限度的灵活性,或者需要创建独立的可执行文件或共享组件,那么 Visual Basic 正合适                                                          |
| Visual Basic for Application | 如果应用程序依赖于使用 VBA 作为控制宏语言的应用程序,那么就使用 VBA。除非用户没有这些应用程序,否则 VBA 也许是应用程序的正确选择。VBA 包含对 OLE Automation 的支持。因此,可能性实际上是无终止的 |
| VBScript                     | 如果对 Web 页编程或在 Outlook 中编程,那么必须用 VBScript。虽然 VBScript 不如 VB 和 VBA 的功能丰富,它仍然可以用来编写复杂的应用程序                           |

在 VBA 和 VBScript 之间作出选择很容易,判断何时使用 VB 而不是 VBA 则稍微复杂一些。考虑 VB 的最好方法是将它看成一个对象工厂。VB 最适用于创建对象(例如,定制控件共享的 OLE 对象等等)。VB 也是高级的定制解决方案的最佳选择,后者需要快速和独立。但是 VBA 对于给功能强大的产品,如 Microsoft Excel 甚至于 Internet Explorer 中的 Web 页面添加功能最有用。

## 1.3 Visual Basic 5.0 的 8 种工程

当进入 Visual Basic 5.0 时,用户可以选择创建以下 8 种工程中的 1 种:

### 1. Standard EXE

创建 Windows 程序的项目类型。

### 2. ActiveX EXE

远程自动程序,它将任务作为多级应用程序的一部分执行。它过去叫 OLE 自动化服务器。

### 3. ActiveX DLL

创建为 DLL 的远程自动化程序。只有从另一个程序中调用时才可以运行这种程序。它不能独立运行。

### 4. ActiveX Control

可用于创建 Visual Basic 程序或任何 ActiveX 应用程序的控件。

### 5. VB Application Wizard

用于运行 Wizard 建立应用程序的框架。

### 6. ADD-IN

用于为 Visual Basic 自身提供另外的功能。有一个例子是 Visual Data Manager。

**7. ActiveX Document DLL**

创建运行于 Internet Explorer 中的应用程序。

**8. ActiveX Document EXE**

创建运行于 Internet Explorer 中的应用程序。

实际上,这就是第一次启动 Visual Basic 时的基本设计环境。

## 1.4 小 结

在本章中,我们介绍了 Visual Basic 5.0 的发展简史,以及当前通用的 VB、VBA 和 VBScript 之间的差别和共同点。还简要描述了 Visual Basic 5.0 的新增功能,分析了 Visual Basic 5.0 的 8 种工程。从下一章开始,我们将进入到 VB 的开发环境中,迈出使用 VB 创建程序的第一步。

