

全国计算机等级考试教材

(二级)

Visual Basic 语言程序设计

严洪华 严志勇 编著

内 容 简 介

本书是严格按照教育部考试中心制定的“全国计算机等级考试二级考试大纲”中“Visual Basic 语言程序设计”的要求而编写的。

本书共分 12 章，深入浅出地介绍了 Visual Basic 可视化的面向对象的编程技术和方法，详细地讲解了 Visual Basic 6.0 中文版的编程环境、基本语法、常用标准控件、数组、过程、菜单、对话框、窗体设计与多文档界面、鼠标与键盘事件、数据文件处理、ActiveX 控件及数据库编程等设计技术。全书内容紧扣考试大纲，既有 Visual Basic 6.0 的基础知识，又有结合实际应用插入的大量例题讲解，因此它是参加计算机等级考试人员考前必备的教材，也可作为 Visual Basic 初学人员的培训教材和有编程经验的专业人员开发 Visual Basic 应用程序的参考书。

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，侵权必究。

图书在版编目 (CIP) 数据

(二级) Visual Basic 语言程序设计/严洪华, 严志勇编著. —北京: 电子工业出版社, 2003. 5

全国计算机等级考试教材

ISBN 7-5053-8649-2

. Visual 严... 严... . BASIC 语言—程序设计—水平考试—教材 . TP312

中国版本图书馆 CIP 数据核字 (2003) 第 026821 号

责任编辑: 坚 如 特约编辑: 王保秀

印 刷:

出版发行: 电子工业出版社 <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: 21.5 字数: 550 千字

版 次: 2003 年 5 月第 1 版 2003 年 5 月第 1 次印刷

定 价: 27.00 元

凡购买电子工业出版社的图书, 如有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系。
联系电话: (010) 68279077

目 录

| | |
|---------------------------------|----|
| 第 1 章 概述 | 1 |
| 1.1 Visual Basic 的由来与发展 | 1 |
| 1.2 Visual Basic 的主要功能与特点 | 2 |
| 1.3 Visual Basic 的启动与退出 | 3 |
| 1.3.1 Visual Basic 的启动 | 4 |
| 1.3.2 应用程序编制前的类型选择 | 5 |
| 1.3.3 Visual Basic 的退出 | 7 |
| 1.4 Visual Basic 开发环境的主窗口 | 7 |
| 1.5 窗体设计器窗口 | 12 |
| 1.6 工程资源管理器窗口 | 12 |
| 1.7 属性窗口 | 13 |
| 习题 | 14 |
| 第 2 章 Visual Basic 的编程基础 | 16 |
| 2.1 对象 | 16 |
| 2.1.1 对象的概念 | 16 |
| 2.1.2 对象属性 | 17 |
| 2.1.3 对象事件 | 17 |
| 2.1.4 对象方法 | 17 |
| 2.1.5 对象属性的设置 | 18 |
| 2.1.6 对象属性的读取 | 20 |
| 2.1.7 编写对象的事件过程 | 20 |
| 2.2 窗体 | 21 |
| 2.2.1 窗体的结构与属性 | 21 |
| 2.2.2 窗体的事件 | 25 |
| 2.3 控件 | 25 |
| 2.3.1 控件的分类、命名与控件值 | 26 |
| 2.3.2 常用标准控件的通用属性与事件 | 27 |
| 2.3.3 窗体中的控件布置 | 29 |
| 2.3.4 窗体和控件的基本操作 | 30 |
| 2.4 工程及其管理方法 | 32 |
| 2.5 事件驱动 | 36 |
| 习题 | 37 |
| 第 3 章 Visual Basic 的基本语法 | 38 |
| 3.1 数据类型 | 38 |
| 3.1.1 基本数据类型 | 38 |
| 3.1.2 用户自定义数据类型 | 40 |

| | |
|-------------------------------|----|
| 3.1.3 枚举类型 | 41 |
| 3.2 常量 | 42 |
| 3.3 变量 | 43 |
| 3.3.1 变量的命名 | 43 |
| 3.3.2 变量的显式声明 | 44 |
| 3.3.3 变量的默认声明 | 45 |
| 3.3.4 变量的作用域 | 45 |
| 3.3.5 变体类型变量 | 47 |
| 3.4 运算符与表达式 | 48 |
| 3.4.1 算术运算符 | 48 |
| 3.4.2 关系运算符 | 49 |
| 3.4.3 逻辑运算符 | 50 |
| 3.4.4 表达式 | 50 |
| 3.4.5 表达式的执行顺序 | 50 |
| 3.5 常用语句 | 51 |
| 3.5.1 赋值语句 | 51 |
| 3.5.2 注释语句 | 52 |
| 3.5.3 暂停语句 | 52 |
| 3.5.4 结束语句 | 53 |
| 3.6 常用内部函数 | 53 |
| 3.6.1 数学函数 | 53 |
| 3.6.2 转换函数 | 54 |
| 3.6.3 字符串函数 | 55 |
| 3.6.4 日期和时间函数 | 56 |
| 3.6.5 随机数函数 | 56 |
| 习题 | 57 |
| 第4章 数据的输入和输出 | 59 |
| 4.1 Print 方法的数据输出 | 59 |
| 4.1.1 Print 方法 | 59 |
| 4.1.2 与Print 方法有关的函数 | 60 |
| 4.1.3 其他的方法和属性 | 62 |
| 4.2 Input Box 函数的数据输入 | 63 |
| 4.3 MsgBox 函数和MsgBox 语句 | 64 |
| 4.3.1 MsgBox 函数 | 64 |
| 4.3.2 MsgBox 语句 | 67 |
| 4.4 字形 | 68 |
| 4.4.1 字体类型和字号大小 | 68 |
| 4.4.2 其他属性 | 69 |
| 4.5 打印机输出 | 71 |
| 4.5.1 打印机直接输出 | 71 |
| 4.5.2 窗体输出 | 72 |
| 习题 | 72 |
| 第5章 常用标准控件的使用 | 75 |
| 5.1 焦点和Tab 键顺序 | 75 |

| | |
|--------------------------------|------------|
| 5.2 文本控件 | 76 |
| 5.2.1 标签框(Label) | 76 |
| 5.2.2 文本框(Text Box) | 77 |
| 5.3 图形控件 | 79 |
| 5.3.1 图片框(Pi ct ureBox) | 79 |
| 5.3.2 图像框(Image) | 82 |
| 5.3.3 直线和形状 | 83 |
| 5.4 命令按钮 | 85 |
| 5.5 复选框 | 89 |
| 5.6 单选按钮 | 92 |
| 5.7 列表框 | 94 |
| 5.8 组合框 | 99 |
| 5.9 滚动条 | 101 |
| 5.10 计时器 | 104 |
| 5.11 框架 | 106 |
| 习题 | 109 |
| 第6章 控制结构 | 112 |
| 6.1 选择结构 | 112 |
| 6.1.1 单行结构条件语句 | 112 |
| 6.1.2 块结构条件语句 | 114 |
| 6.1.3 IIf 函数 | 116 |
| 6.2 多分支结构 | 116 |
| 6.3 For 循环控制结构 | 117 |
| 6.4 当循环控制结构 | 119 |
| 6.5 Do 循环控制结构 | 120 |
| 6.6 多重循环 | 122 |
| 6.7 GoTo 型控制 | 124 |
| 6.7.1 GoTo 语句 | 124 |
| 6.7.2 On-GoTo 语句 | 124 |
| 习题 | 125 |
| 第7章 数组 | 128 |
| 7.1 数组的概念 | 128 |
| 7.2 数组的定义 | 128 |
| 7.2.1 下标下界固定的数组定义 | 129 |
| 7.2.2 可设定下标下界的数组定义 | 129 |
| 7.2.3 默认数组的定义 | 130 |
| 7.2.4 嵌套数组的定义 | 130 |
| 7.2.5 动态数组的定义 | 131 |
| 7.2.6 数组的清除与重新定义 | 131 |
| 7.3 数组的基本操作 | 132 |
| 7.3.1 数组元素的赋值、引用、输出和复制 | 132 |
| 7.3.2 For Each...Next 语句 | 133 |
| 7.4 数组的初始化 | 134 |
| 7.5 控件数组 | 135 |

| | |
|------------------------|-----|
| 习题 | 137 |
| 第 8 章 过程 | 140 |
| 8.1 Sub 过程的定义和建立 | 140 |
| 8.2 Sub 过程的调用 | 142 |
| 8.3 通用过程与事件过程 | 142 |
| 8.4 Function 过程的定义 | 143 |
| 8.5 Function 过程的调用 | 144 |
| 8.6 参数传送 | 144 |
| 8.6.1 按地址传送 | 145 |
| 8.6.2 按值传送 | 146 |
| 8.6.3 数组参数的传送 | 146 |
| 8.6.4 可选参数的使用 | 147 |
| 8.6.5 可变参数的使用 | 148 |
| 8.6.6 指名参数的使用 | 148 |
| 8.6.7 对象参数的使用 | 149 |
| 8.6.8 Shell 函数 | 151 |
| 习题 | 153 |
| 第 9 章 键盘与鼠标事件过程 | 157 |
| 9.1 KeyPress 事件 | 157 |
| 9.2 KeyDown 和 KeyUp 事件 | 160 |
| 9.3 鼠标事件 | 162 |
| 9.4 鼠标光标的形状 | 165 |
| 9.5 拖放 | 168 |
| 9.5.1 与拖放有关的属性、事件和方法 | 168 |
| 9.5.2 自动拖放 | 169 |
| 9.5.3 手动拖放 | 171 |
| 习题 | 172 |
| 第 10 章 菜单设计 | 175 |
| 10.1 菜单的基本概念 | 175 |
| 10.2 菜单编辑器 | 177 |
| 10.3 用菜单编辑器建立菜单 | 179 |
| 10.4 菜单项的控制 | 184 |
| 10.4.1 有效性控制 | 184 |
| 10.4.2 菜单项标记 | 186 |
| 10.4.3 键盘选择 | 188 |
| 10.5 菜单项的增减 | 189 |
| 10.6 弹出式菜单 | 193 |
| 习题 | 197 |
| 第 11 章 对话框设计 | 199 |
| 11.1 自定义对话框 | 199 |
| 11.2 通用对话框 | 202 |
| 11.2.1 通用对话框控件 | 202 |
| 11.2.2 文件对话框 | 204 |

| | | |
|--------|-------------------------------------|-----|
| 11.2.3 | 颜色对话框 | 208 |
| 11.2.4 | 字体对话框 | 210 |
| 11.2.5 | 打印对话框 | 213 |
| 习题 | | 216 |
| 第 12 章 | 多重窗体程序设计 | 218 |
| 12.1 | 多重窗体程序设计的语句、方法和属性 | 219 |
| 12.2 | 多重窗体应用程序举例 | 220 |
| 12.2.1 | 创建由相互独立的若干窗体组成的多重窗体应用程序 | 220 |
| 12.2.2 | 创建由一个父窗体和若干个子窗体组成的多重窗体应用程序 | 228 |
| 12.2.3 | 用自定义窗体对象数组变量方法自动生成 MDI 窗体的子菜单 | 236 |
| 12.3 | 多重窗体程序的存取与编译 | 238 |
| 12.4 | Visual Basic 工程结构 | 240 |
| 12.5 | 闲置循环与 DoEvents 语句 | 243 |
| 习题 | | 245 |
| 第 13 章 | 数据文件 | 246 |
| 13.1 | 文件的基本概念 | 246 |
| 13.2 | 文件操作的语句和函数 | 247 |
| 13.3 | 顺序文件 | 249 |
| 13.3.1 | 顺序文件的打开和关闭 | 249 |
| 13.3.2 | 顺序文件的写操作 | 250 |
| 13.3.3 | 顺序文件的读操作 | 252 |
| 13.3.4 | 顺序文件的应用举例 | 253 |
| 13.4 | 随机文件 | 260 |
| 13.4.1 | 随机文件的打开和关闭 | 261 |
| 13.4.2 | 随机文件的写操作 | 261 |
| 13.4.3 | 随机文件的读操作 | 262 |
| 13.4.4 | 随机文件中记录的增加与删除 | 262 |
| 13.4.5 | 用控件显示和修改随机文件 | 262 |
| 13.4.6 | 随机文件的应用举例 | 262 |
| 13.5 | 文件系统控件 | 278 |
| 13.5.1 | 驱动器列表框(DriveListBox) | 278 |
| 13.5.2 | 目录列表框(DirListBox) | 279 |
| 13.5.3 | 文件列表框(FileListBox) | 279 |
| 13.5.4 | 文件系统控件的应用举例 | 281 |
| 习题 | | 290 |
| 第 14 章 | ActiveX 控件 | 292 |
| 14.1 | ActiveX 控件的添加与删除 | 292 |
| 14.2 | ActiveX 控件的应用举例 | 293 |
| 14.3 | 开发自己的 ActiveX 控件 | 296 |
| 习题 | | 301 |
| 第 15 章 | 数据库编程技术 | 303 |
| 15.1 | 数据库的基本概念 | 303 |
| 15.2 | 用可视化数据管理器建立数据库 | 304 |

| | | |
|--------|-------------------------------|-----|
| 15.3 | 用数据窗体设计器设计用户窗体 | 309 |
| 15.4 | 使用 Data 控件访问数据库 | 311 |
| 15.5 | 使用 ADO 控件访问数据库 | 314 |
| 15.6 | 使用 DataGrid 控件访问数据库 | 318 |
| 15.7 | 使用数据窗体向导 | 321 |
| 15.8 | 使用 SQL 操作数据库 | 326 |
| 15.8.1 | SQL 的构成 | 326 |
| 15.8.2 | SQL 在 Visual Basic 中的应用 | 327 |
| 习题 | | 330 |
| 附录 | 习题参考答案 | 332 |

第 1 章 概 述

Visual Basic(简称 VB)是美国微软(Microsoft)公司推出的一种功能强大、应用广泛的编程语言。它为在 Windows 图形界面下编写各种应用程序的用户,提供了一种迅速而便捷的软件开发环境和方法。

Visual Basic 简单易学、功能强大,它为用户提供了一整套工具用来方便地开发各种应用程序。对于专业编程人员,用 Visual Basic 完全可以实现其他 Windows 编程语言所能实现的功能;对于初学人员,只需掌握一些关键词句就可以建立实用的 Windows 应用程序。所以,无论是 Windows 应用程序的专业开发人员,还是初学者,掌握了 Visual Basic 编程语言,对于自己的应用程序设计都是非常有益的。

作为入门章节,本章将对 Visual Basic 进行一些简要的介绍。内容包括 Visual Basic 的由来与发展,它有些什么特点,其功能如何,怎样来启动与退出 Visual Basic,以及 Visual Basic 的集成开发环境由哪些要素构成等。

1.1 Visual Basic 的由来与发展

微软公司为了方便用户在其图形界面的 Windows 操作系统下进行应用程序的开发,于 1991 年推出了在 Windows 环境下新一代面向对象的程序设计语言——Visual Basic 1.0 版本。随后,微软公司于 1992 年秋季推出了 Visual Basic 2.0 版本,并在 1993 年 4 月推出了 Visual Basic 3.0 版本。

随着 Windows 95 的发行,为适应它的 32 位操作系统的需要,微软公司于 1995 年 10 月推出了能开发 32 位应用程序的 Visual Basic 4.0 版本。并在 1997 年推出了具有对 Internet 支持和开发能力的 Visual Basic 5.0 版本。

1998 年秋季,微软公司又推出了功能更强大、更完善的 Visual Basic 6.0 版本。该版本在建立自定义控件、访问数据库以及对 Internet 的支持等方面都有进一步的加强、完善和提高,是 Visual Basic 的跨世纪版本。

从 Visual Basic 1.0 版到 4.0 版,Visual Basic 只有英文版,自 5.0 版本起,Visual Basic 有了中文版。

在 Visual Basic 6.0 版中,根据不同用户开发应用程序的需要,又分为学习版、专业版和企业版。这三种类型的版本是在相同的基础上建立起来的,但在功能方面有些差异,在它们集成环境中的菜单及工具条中的一些选项上也稍有不同。

本书是以 Visual Basic 6.0 中文企业版为蓝本来进行介绍的,但其内容完全适用于 Visual Basic 6.0 学习版或专业版的读者,书中所介绍的例题程序,在专业版和学习版中都可运行。我们在这里约定,本书在后面的编程介绍中,除特别指明外,一般指的都是 Visual Basic 6.0。

1.2 Visual Basic 的主要功能与特点

Visual Basic 是在原 Basic 语言的基础上开发出来的新一代面向对象的程序设计语言。

它既有原 Basic 语言易学、易用的特点,又能方便地在 Windows 操作系统中进行图形界面应用程序的开发。所以,自推出以来,深受广大编程爱好者及专业程序员的青睐。与过去传统的编程语言相比,Visual Basic 在许多方面有重要的改革和突破。这里只介绍以下一些最基本的和最容易理解的特点。

1. 简单易学的可视化图形界面设计方法

用传统的 C, Pascal 及 Basic 等计算机高级语言编写程序时,最令程序员烦恼的是编写友好的用户界面。往往为了能在屏幕上显示出一个简单的图形界面,也要去编写一大段程序语句,然后通过编译后运行才能看到其实际效果。如果对该界面的实际效果不满意,则还要返回到程序编写器中去修改。而使用 Visual Basic 则为用户编写应用程序提供了可视化的集成开发环境,编程人员不需编写大量的代码去描述用户界面的外观和位置,只需在 Visual Basic 为用户提供的工具中,自由地取出所需部件,放到窗体中的指定位置,通过鼠标的移动操作,即可直接“画”出所需的用户界面。画出的界面具有“所见即所得”的特点,即画出的界面是什么样的,则通过编译后运行的实际效果就是什么样的。

令程序员烦恼的编写用户界面的工作就这样轻而易举地被 Visual Basic 解决了。Windows 之所以比 DOS 受欢迎,就是因为它具有丰富多彩、深受用户欢迎的友好界面。

2. 面向对象的程序设计思想

用 Visual Basic 编写应用程序,一是要设计用户界面,二是要编写程序代码。Visual Basic 不但设计用户界面的方法与传统的计算机高级语言不同,而且程序代码编写的方法也与传统的计算机高级语言不同。

传统的计算机高级语言编程方法是:根据程序要实现的功能,编写出一个完整的程序。运行时,先从第一行代码开始执行,然后按照应用程序中预定的路径或规则一条一条地去执行,直到结束语句为止。在程序运行的过程中,除了需要用户输入数据时暂停等待输入外,程序会不停顿地按规则执行各项指令,直到程序结束。程序设计者必须十分周到地考虑到程序运行中的每个细节,否则程序就有可能会出现不可预见的结果,甚至陷入崩溃状态,造成死机。

而 Visual Basic 改变了编程的机制,它没有传统意义上的主程序,程序执行的基本方法是由“事件”来驱动“过程”,即面向对象的程序设计方法。这是随 Windows 图形界面的诞生而产生的一种新的程序设计思想。所谓“对象”就是一个可操作的实体,如窗体及窗体中的按钮、文本框等控件。每个对象都能响应多个不同的事件,每个事件均能驱动一段代码(事件过程),该段代码决定了对象的功能。事件可以由用户操作触发,也可以由来自操作系统或其他应用程序的消息触发,甚至由应用程序本身的消息触发。例如,单击一个按钮,则触发按钮的 Click(单击)事件,处于该事件过程中的代码就会被执行。整个应用程序就是由这样一些彼此独立的事件过程构成。而用 Visual Basic 编写应用程序,实际上就是为各个对象编写事件过程的工作。但由于各个过程要实现的功能是单一的,过程的规模一般不会太大。这样,将原来传统语言编程方法中由统一控制的、包罗万象的大程序分解为许多个独立的、小规模的过程,然后分别由各种“事件”来驱动执行,就使得程序编制的工作变得相对简单了。这就是 Visual Basic 区别于传统高级编程语言的另一个重要的特点。

3. 交互式的集成开发环境

Visual Basic 集成开发环境是交互式的。传统高级语言编程一般都要经过编码、调试和编译这三个步骤,其中每一步还需调用专门的处理软件。而 Visual Basic 的集成开发环境,则把这三个步骤的操作集中在其开发环境内统一处理,大大方便了编程者的使用。

在传统高级语言编程中,如果编写的代码有误,则在进行应用程序编译时,这种错误就会被其编译软件捕获。编程者必须查找并改正这种错误,才能再次进行编译。而 Visual Basic 采用的是交互式的在线检测方式,即在编程者输入代码时便进行解释,即时捕获并突出显示

其语法或拼写错误，使编程者能及时方便地进行修改。

Visual Basic 的编译操作非常方便快捷，只要通过菜单选择或功能键的使用就可以启动。编译后即能运行并显示其结果。

所以说，Visual Basic 中集应用程序编码、调试和编译功能于一体的开发环境，是 Visual Basic 区别于传统高级编程语言的又一个重要的特点。

4. 高度的可扩充性

Visual Basic 是一种高度可扩充的语言，它为用户扩充其功能提供了多种途径。主要有：支持其他软件商为扩充其功能而开发的外部控件，只要这些控件的文件扩展名为 .OCX，就可将其加入到 Visual Basic 系统中，以扩充其编程能力；Visual Basic 为用户提供了访问动态链接库(Dynamic Link Library，简称 DLL)的功能，编程人员可以利用其他语言，如 Visual C++等，将需要实现的功能编译成动态链接库，然后提供给 Visual Basic 调用；支持访问应用程序接口(Application Program Interface，简称 API)，应用程序接口是 Windows 环境中可供其应用程序访问和调用的一组函数集合，在微软公司的 Windows 操作系统中，包含有一大批功能强大的 API 函数，供用户编程时直接调用。Visual Basic 提供了访问和调用这些 API 函数的能力，利用这些 API 函数，可大大增强 Visual Basic 的编程能力，并可实现一些用 Visual Basic 语言本身不能实现的一些特殊功能。

除了这些基本的特点之外，Visual Basic 在数据库的访问、动态数据交换、建立用户自用的 ActiveX 控件及 Internet 组件的下载等许多方面都具有自己独到的特点。

1.3 Visual Basic 的启动与退出

不言而喻，要运行一个软件的前提是该软件已经被安装在要运行它的计算机系统上了。

微软公司发行的 Visual Basic 6.0 是 Visual Studio 6.0 成套软件中的一个组成部分。它既可以和 Visual Studio 6.0 一起安装，也可以单独安装。安装方法与大多数微软公司软件的相同，即如用户的计算机能够运行光盘中的自动加载程序，则在插入安装光盘时，安装程序将被自动加载。否则，就用鼠标双击安装光盘中的 Set up. exe 文件，使屏幕上出现 Visual Basic 6.0 的安装画面，然后按照屏幕上的安装指令行事即可完成安装操作。

1.3.1 Visual Basic 的启动

Visual Basic 的启动，通常可分为命令方式、快捷方式和运行方式。

1. 用命令方式启动

使用命令方式启动 Visual Basic，只需在 Windows 桌面上，用鼠标单击任务栏上的“开始”按钮，在弹出的一级菜单中选择“程序”项；然后在其二级菜单中选择“Microsoft Visual Basic 6.0 中文版”；最后，在其三级菜单中单击“Microsoft Visual Basic 6.0 中文版”，即可启动 Visual Basic。见图 1-1。

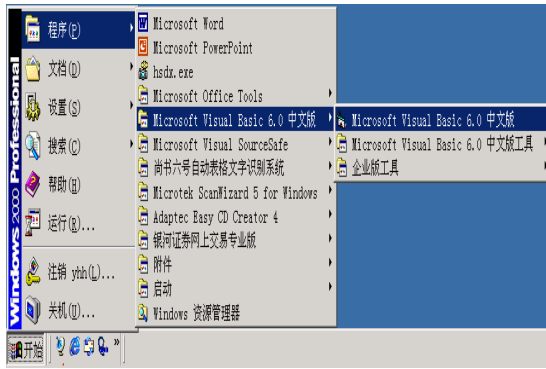


图 1-1 使用命令方式启动 Visual Basic

2. 用快捷方式启动

使用快捷方式启动 Visual Basic，只需要用鼠标双击 Windows 桌面上的 Visual Basic 快捷方式图标就行了。若您的 Windows 桌面上还没有 Visual Basic 的快捷方式图标，可以用以下方式在您的 Windows 桌面上建立起一个 Visual Basic 的快捷方式图标。

在 Windows 桌面上单击任务栏上的“开始”按钮，然后在弹出的一级菜单中选择“程序”项，再在其二级菜单中单击“Windows 资源管理器”命令，打开“Windows 资源管理器”。然后按照路径找到 Visual Basic 执行程序 VB6.EXE 所在的位置(这里是采用系统安装时设定的默认路径 C:\Program Files\Microsoft Visual Studio\VB98)。

用鼠标左键单击 VB6.EXE 程序，使条形光标覆盖在该程序名上。见图 1-2。

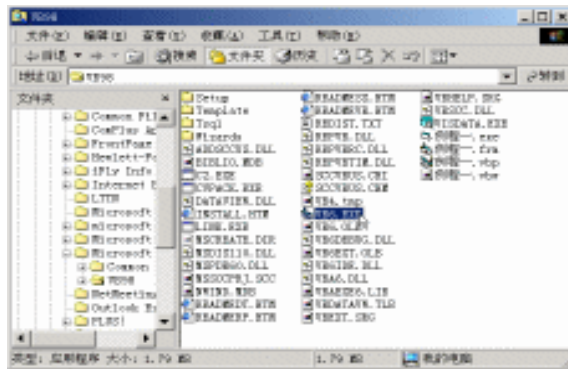


图 1-2 创建 Visual Basic 快捷方式图标步骤之一

再按住鼠标右键将 VB6.EXE 程序拖到“Windows 资源管理器”外的 Windows 桌面上，松开鼠标右键，系统弹出一个快捷方式设置菜单。见图 1-3。

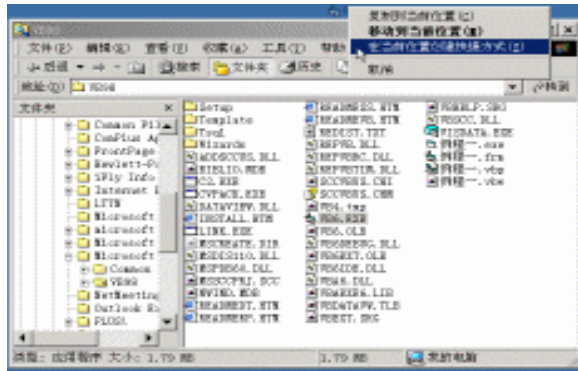


图 1-3 创建 Visual Basic 快捷方式图标步骤之二

用鼠标单击快捷方式设置菜单中的“在当前位置创建快捷方式”项，快捷方式设置菜单消失，桌面上增加了一个 VB6.EXE 快捷方式图标。见图 1-4。

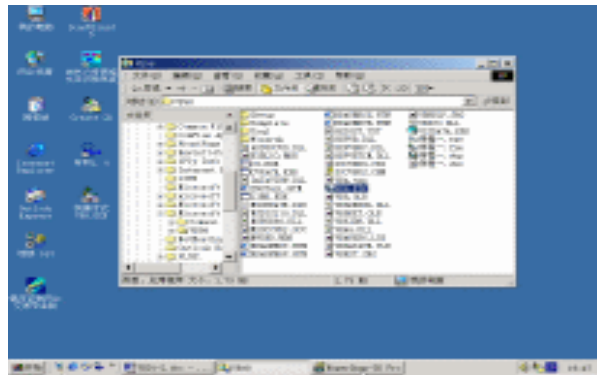


图 1-4 创建 Visual Basic 快捷方式图标步骤之三

3. 用运行方式启动

使用运行方式启动 Visual Basic 的操作，可以在“我的电脑”或“Windows 资源管理器”中，直接用鼠标双击 Visual Basic 的执行程序 VB6.EXE(这里的存放路径是 C:\Program Files\Microsoft Visual Studio\VB98)，即可达到启动 Visual Basic 的目的。

1.3.2 应用程序编制前的类型选择

启动 Visual Basic 6.0 后，系统运行时首先显示出 Visual Basic 的版权页。稍候，会打开一个如图 1-5 所示的界面，即在 Visual Basic 集成开发环境的上面出现一个“新建工程”对话框。

“新建工程”对话框是用来供用户对所编应用程序的工程类型进行选择处理的对话框。在“新建工程”对话框中有“新建”、“现存”和“最新”等三个选项卡。其中“新建”选项卡对话框显示了如下可在 Visual Basic 中使用的工程类型(即可以建立的应用程序)。

- 标准 EXE：选择该项工程类型，可以建立一个标准的 EXE 应用程序。
- ActiveX EXE：选择该项工程类型，可以在专业版或企业版中建立一个可执行的 ActiveX EXE 程序。
- ActiveX DLL：选择该项工程类型，可以在专业版或企业版中建立一个 ActiveX DLL 的动态链接库(DLL)程序。

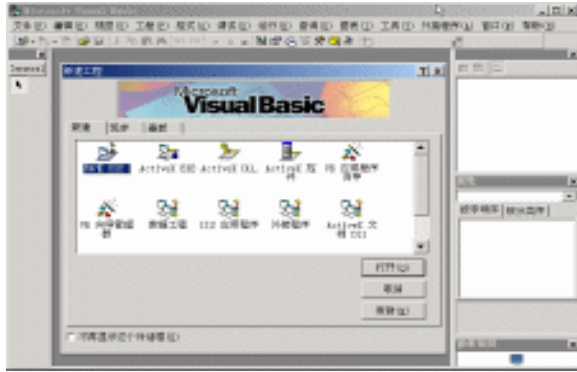


图 1-5 Visual Basic 6.0 “新建工程”对话框

- ActiveX 控件：选择该项工程类型，可以在专业版或企业版中建立一个用于开发用户自定义的 ActiveX 控件。
- VB 应用程序向导：选择该项工程类型，可以在开发环境中为开发自己的工程，直接建立新的应用程序框架。
- VB 向导管理器：选择该项工程类型，可以帮助用户建立自己的向导程序。
- 数据工程：选择该项工程类型，可以为编程人员提供开发数据报表应用程序的框架，自动打开数据环境设计器和数据报表设计器。
- IIS 应用程序：这是 Visual Basic 6.0 版新增加的工程类型，选择该项工程类型，可以用 Visual Basic 代码来编写服务器方的 Internet 应用程序，用来响应由浏览器发出的用户需求。
- 外接程序：选择该项工程类型，可以建立自己的 Visual Basic 外接程序，并在开发环境中自动打开连接设计器。
- ActiveX 文档 EXE：选择该项工程类型，可以建立一个在超链接环境(即 Web 浏览器，如 Microsoft Internet Explorer)中运行的 ActiveX EXE 程序。
- ActiveX 文档 DLL：选择该项工程类型，可以建立一个在超链接环境(即 Web 浏览器，如 Microsoft Internet Explorer)中运用的 ActiveX DLL 的动态链接库(DLL)程序。
- DHTML 应用程序：选择该项工程类型，可以在专业版或企业版中建立一个用于编写响应 HTML 页面操作的 Visual Basic 代码程序，并可将处理过程传送到服务器上的 DHTML 应用程序。
- VB 企业版控件：该选择不是用来建立应用程序，而是用来在工具箱中加入企业版控件图标的。选用该图标后，企业版控件将出现在工具箱中。

如果在图 1-5 中，用鼠标单击“现存”或“最新”选项卡，则可分别显示现有的或最新的 Visual Basic 应用程序文件名列表，用户可从中选择要打开的文件名。

如何利用该对话框来创建多种应用程序，将在以后的章节里进行讲解。若用户想在启动 Visual Basic 集成开发环境时，不显示该“新建工程”对话框，则可用鼠标在其下方的“不再显示这个对话框”的复选框小方格中单击一下，使其中出现一个“”，然后，用鼠标单击“新建工程”对话框右上角的关闭按钮，即可完成关闭操作。下一次启动时，Visual Basic 集成开发环境上面就不再出现该“新建工程”对话框，而是直接进入 Visual Basic 的集成开发环境。进入后若要使用“新建工程”对话框，仍可选择“文件”菜单中的“新建工程”项命令将其调出。

当用户在对话框中选择了—个要建立的工程类型(例如“标准 EXE”)，然后单击“打开”按钮，即可进入如图 1-6 所示的 Visual Basic 6.0 集成环境。

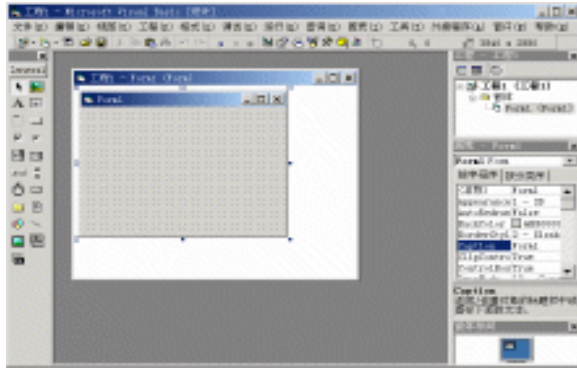


图 1-6 Visual Basic 6.0 集成开发环境界面

1.3.3 Visual Basic 的退出

进入 Visual Basic 集成开发环境后的运行与操作，将在后面介绍。这里先介绍一下如何退出 Visual Basic 系统。

退出 Visual Basic 系统可以通过菜单栏菜单的选择，也可通过系统控制菜单框的选择，还可用鼠标单击 Visual Basic 系统的关闭按钮来解决。

若采用菜单栏的菜单选择方式，则只要执行“文件”菜单中的“退出”命令(快捷键是 Alt+Q)，即可退出 Visual Basic 系统。

若采用系统控制菜单框选择方式，则可先用鼠标单击集成环境窗口左上角的系统控制菜单框，再在其弹出的下拉框中用鼠标单击“关闭”选项，就可退出 Visual Basic 系统。

若采用系统的关闭按钮方式，则只需用鼠标单击 Visual Basic 集成环境窗口右上角的关闭按钮，即可退出 Visual Basic 系统。

无论用何种方式退出 Visual Basic，如果 Visual Basic 集成开发环境中修改过的文件没有保存，系统都会弹出一个提示用户是否保存已修改过文件的对话框，在用户选择答案后，按用户要求处理后退出。退出 Visual Basic 系统的结果都是返回到 Windows 系统。

1.4 Visual Basic 开发环境的主窗口

Visual Basic 集成开发环境，是一个集成了设计、编辑、编译和调试等多种功能于一体的 Visual Basic 编程操作的工作平台。

如前所述，Visual Basic 被启动后，当用户在对话框中选择了—个要建立的工程类型后，单击“打开”按钮，即可进入 Visual Basic 的集成开发环境。

Visual Basic 集成环境由主窗口(内含标题栏、菜单栏、工具栏和工具箱)、窗体设计器窗口、工程资源管理窗口、属性窗口和窗体布局窗口等要素构成。在图 1-7 中分别用文字对这些要素进行了标示。

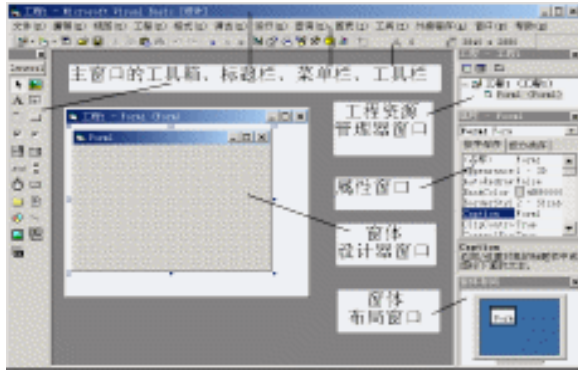


图 1-7 Visual Basic 6.0 集成开发环境

主窗口是 Visual Basic 集成开发环境的容器窗体。其他窗口的活动，一般情况下都在其范围内进行(某些窗口在浮动方式状态时，可以离开主窗口使用)。主窗口由标题栏、菜单栏和工具栏组成。

1. 标题栏

标题栏位于集成开发环境窗口的顶部，主要用来显示所编应用程序的名字及集成开发环境当前的工作状态。启动 Visual Basic 后 标题栏中显示的信息为“工程 1—Microsoft Visual Basic[设计]”。当进行所编应用程序的运行或调试时，标题栏中显示的信息就会变成“工程 1—Microsoft Visual Basic [运行]”或“工程 1—Microsoft Visual Basic [Break]”。这三个阶段依次分别称为“设计模式”、“运行模式”和“中断模式”。这些信息的处理是由 Windows 系统来完成的，不需用户进行干预。

2. 菜单栏

在标题栏的下面是集成环境的菜单栏。菜单栏列出了开发、调试和保存应用程序所需要的操作指令。Visual Basic 6.0 中文版的菜单栏共有 13 个菜单项，即文件、编辑、视图、工程、格式、调试、运行、查询、图表、工具、外接程序、窗口和帮助。每个菜单项又有若干个菜单命令，用鼠标单击某个菜单项，即可打开该菜单，弹出一个附属于该菜单的下拉菜单，可让你选择不同的菜单命令去执行。

Visual Basic 的菜单其表现形式和使用方法与 Windows 系统的其他软件相同。即当前可直接使用的菜单命令呈深色显示，当前不能直接使用的菜单命令或在使用该命令前需先选定另一项的菜单命令呈灰色显示，需要通过打开“对话框”来执行的菜单命令用命令名后面带有省略号(...)的形式来显示，需要通过弹出下拉菜单来进一步选择的菜单命令用命令名后面带有三角号()的形式来显示。

另外，为了方便菜单命令使用，很多菜单命令后都设置了用 Alt 加上一个字母的访问键或用 Ctrl 加上一个字母的快捷键(又称热键)。例如，按 Alt+F 键可以立即弹出文件项的下拉菜单，按 Ctrl+N 键可以立即执行“新建工程”命令。

因此，菜单命令除用鼠标单击选择外，还可通过访问键和快捷键来选择。其实，Windows 系统下的菜单命令(包括 Visual Basic)，还可用 Alt 键或 F10 键来先激活主菜单栏，然后通过上()、下()、左()及右() 4 个方向键选择某项菜单命令；或通过 Esc 键来退回到上一步的选择层面，选中所需的菜单项后，再按回车键，即可执行所选择的菜单命令。

3. 工具栏

在菜单栏下面的一行图标按钮是 Visual Basic 编辑应用程序的工具栏。虽然这些按钮的功能在菜单栏中都已具备，但为了让用户使用起来更简洁方便，该处又用按钮的形式来表现。单击某个命令的图标按钮，就能直接执行相应的菜单命令。

Visual Basic 6.0 提供了 4 种工具栏，包括编辑工具栏、标准工具栏、窗体设计工具栏和调试工具栏。用户还可根据需要进行自定义的工具栏。按照系统的默认规定，启动 Visual Basic 之后，主窗口上只显示标准工具栏，其他工具栏可以通过“视图”菜单中的“工具栏”命令去打开或关闭。

上述 4 种工具栏都有固定和浮动两种形式，只要把鼠标箭头移到固定形式工具栏中没有图标的地方，按住鼠标左键向下拖动，或双击工具栏左端的两条浅色竖线，就可把固定工具栏转变为浮动工具栏。反之，若双击浮动工具栏的标题条，则可使其转变为固定工具栏。

标准工具栏的固定形式位于菜单栏的下面，它以图标的形式提供了系统菜单中的一部分常用菜单命令的操作。用户只要用鼠标单击代表某个命令的图标按钮，就能直接执行相应的菜单命令。默认情况下，标准工具栏中有代表 21 种菜单命令的图标按钮，以及两个用来显示窗体或控件当前位置与大小的状态栏，如图 1-8 所示。而图 1-9 显示的则是标准工具栏的浮动形式。

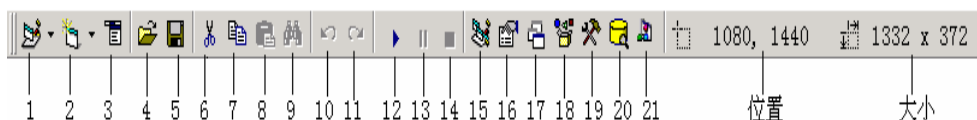


图 1-8 固定形式的标准工具栏

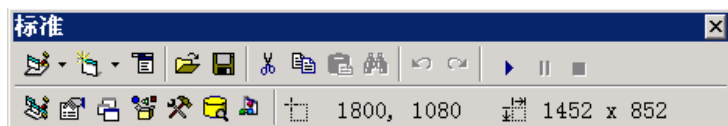


图 1-9 浮动形式的标准工具栏

下面将标准工具栏图标及其作用，用表 1-1 的形式进行一些简要的说明。

表 1-1 标准工具栏图标及作用

| 顺序号 | 图标名称 | 功 能 |
|-----|------|-----------------------------------|
| 1 | 添加工程 | 添加一个新工程，与“文件”菜单中的“添加工程”命令功能相同 |
| 2 | 添加窗体 | 在工程中添加一个新窗体，与“工程”菜单中的“添加窗体”命令功能相同 |

(续表)

| 顺序号 | 图标名称 | 功 能 |
|-----|-------|---|
| 3 | 菜单编辑器 | 打开菜单编辑对话框，与“工具”菜单中的“菜单编辑器”命令功能相同 |
| 4 | 打开工程 | 用来打开一个已经存在的 Visual Basic 工程文件，与“文件”菜单中的“打开工程”命令功能相同 |
| 5 | 保存工程 | 保存当前的 Visual Basic 工程(组)文件，与“文件”菜单中的“保存工程”命令功能相同 |
| 6 | 剪切 | 把选择的内容剪切到剪贴板，与“编辑”菜单中的“剪切”命令功能相同 |
| 7 | 复制 | 把选择的内容复制到剪贴板，与“编辑”菜单中的“复制”命令功能相同 |
| 8 | 粘贴 | 把剪贴板的内容复制到当前插入位置，与“编辑”菜单中的“粘贴”命令功能相同 |
| 9 | 查找 | 打开“查找”对话框，与“编辑”菜单中的“查找”命令功能相同 |
| 10 | 撤消 | 用来撤消当前的修改 |