

高等教育 **21世纪** 课程教材
计算机基础教育课程体系改革教材

Visual Basic

程序设计教程

吴文斗 周 兵 吴兴勇 编著

湖南教育出版社
www.hnepb.com

高等教育 21 世纪 课程教材
计算机基础教育课程体系改革教材

Visual Basic

程序设计教程

吴文斗 周 兵 吴兴勇 编著



湖南教育出版社

图书在版编目(CIP)数据

Visual Basic程序设计教程/吴文斗著.
—长沙:湖南教育出版社,2005

I. V... II. 吴... III. BASIC语言—程序设计—教材
IV. TP312

中国版本图书馆CIP数据核字(2005)第043891号

Visual Basic程序设计教程

吴文斗 周 兵 吴兴勇 编著
责任编辑:龙育群

湖南教育出版社出版发行(长沙市韶山北路643号)

网 址: <http://www.hneph.com>

电子邮箱: csgaojiao@163.com

湖南华商文化商务有限公司印刷

787×1092 16开 印张: 21 字数: 510000

2005年5月第1版 2006年1月第2版第1次印刷

ISBN7-5355-4519-X/G·4514

定价: 28.00元

本书若有印刷、装订错误,可向承印厂调换

内 容 提 要

本书是根据教育部计算机基础课程教学指导委员会提出的高等学校计算机基础课程教学基本要求和全国计算机等级考试《Visual Basic程序设计》大纲要求组织编写,以Visual Basic 6.0为背景,从初学者的角度出发,用简短的篇幅、通俗易懂的语言,以面向对象程序设计为主线,由浅入深地向读者介绍了Visual Basic程序设计中最基本的内容。全书共15章,内容包括Visual Basic程序开发环境、Visual Basic简单应用程序设计、Visual Basic程序设计基础、顺序结构程序设计、选择结构程序设计、循环结构程序设计、数组、过程与函数、多重窗体程序设计、对话框及菜单设计、数据文件、图形设计、Visual Basic数据库技术等内容。

为了更好地让学生掌握Visual Basic程序设计的测试与调试技能,掌握可视化编程环境的使用,以及各类人员自学及考试辅导,我们配套出版了《Visual Basic程序设计上机指导与习题解析》一书。本书所配教学课件及教学示例源程序可从湖南教育出版社网站(<ftp://ftp.hneph.com/ftpdownload/>)下载。

本书内容重点突出、层次分明、难点详解、循序渐进,理论联系实际。本书既可作为高等院校、职业院校、成人高校开设程序设计课程的教材,也可作为计算机等级考试的应试教材,同时还可以作为软件管理和计算机应用开发人员的阅读参考资料和培训教材。

前 言

Visual Basic(简称 VB)是 Microsoft 公司推出的一种 Windows 应用程序开发工具。它采用面向对象和事件驱动的程序设计思想,集程序设计、编辑、运行于一体的开发环境,使程序设计变得更加方便和快捷。因而,Visual Basic 是目前在 Windows 平台上应用最为广泛的应用程序开发工具之一。

Visual Basic 以其使用方便、简单易学、功能丰富的特点,得到越来越多的计算机爱好者和计算机应用技术人员青睐,并用它开发了大量的应用软件。在我国,许多高校也已相继开设了 Visual Basic 程序设计课程,以培养学生快速掌握开发基于图形界面应用程序的能力和解决实际工程问题的能力。

为帮助广大读者系统掌握 Visual Basic 程序设计的基本思想和方法,我们组织由在教·学第一线具有丰富计算机基础教学经验以及长期从事计算机开发应用的教师编写了此教程。全书在注重理论性、系统性和科学性的同时,坚持以“应用为主,兼顾原理”的原则,在语言组织上力求通俗易懂,在内容编排上由浅入深、循序渐进,系统的阐述了 Visual Basic 的编程思想、程序控制以及应用开发。为帮助读者加强对所学知识的巩固和理解,我们在教材中列举了大量实例,并在每章后附有丰富的习题,同时还配套编写了《Visual Basic 程序设计上机指导与习题解析》。

本书由吴文斗、周兵、吴兴勇、丁超、侯发忠等编写,吴文斗修改定稿,唐红丽、余颖等对本书进行了仔细的校对工作,在此表示衷心感谢。教育部计算机科学教学指导委员会委员李仁发教授对教材的编写给予了悉心指导,李勇帆教授、刘震宇副教授仔细审阅了全书并提出了许多宝贵意见,湖南教育出版社为本书的出版给予了大力支持和帮助,在此一并致谢。

由于作者水平有限,书中错误和不足之处在所难免,敬请有关专家、学者及使用本书的教师和各类人员批评指正,以便再版时修订完善。来信请寄: csgaojiao@163.com。

编 者

2005 年 4 月

目 录

第 1 章 Visual Basic 程序设计概述	1
1.1 Visual Basic 语言简介	1
1.2 Visual Basic 6.0 的安装、启动与退出	3
1.3 Visual Basic 集成开发环境	5
1.4 Visual Basic 6.0 帮助系统的使用	11
习题一	12
第 2 章 Visual Basic 对象及其操作	14
2.1 Visual Basic 程序设计中的几个基本概念	14
2.2 创建 Visual Basic 简单应用程序	16
2.3 窗体的常用属性、方法和事件	23
2.4 命令按钮、标签、文本框	28
2.5 焦点	36
2.6 Visual Basic 应用程序的结构与工作方式	37
习题二	39
第 3 章 Visual Basic 程序设计基础	42
3.1 数据类型	42
3.2 常量与变量	45
3.3 运算符与表达式	51
3.4 常用内部函数	55
3.5 Visual Basic 语句的书写格式	62
习题三	64
第 4 章 顺序程序设计结构	66
4.1 赋值语句	66
4.2 注释、暂停与程序结束语句	68
4.3 数据的输出	69
4.4 输出数据的字符控制	73
4.5 用户交互函数和过程	75
习题四	79
第 5 章 选择结构程序设计	84
5.1 IF 结构条件语句	84
5.2 多分支结构程序设计	92



5.3 选择结构的嵌套·····	95
习题五·····	97
第6章 循环结构·····	102
6.1 Do...Loop 循环结构·····	102
6.2 For...Next 循环语句·····	107
6.3 嵌套循环·····	109
6.4 GoTo 语句·····	111
6.5 循环结构应用举例·····	112
习题六·····	115
第7章 数组·····	122
7.1 数组的概念·····	122
7.2 数组的定义·····	123
7.3 数组的基本操作·····	128
7.4 For Each...Next 语句·····	132
7.5 控件数组·····	133
7.6 数组的应用·····	136
习题七·····	143
第8章 过程与函数·····	151
8.1 Sub 过程·····	151
8.2 Function 过程·····	156
8.3 过程中参数的传递·····	158
8.4 可选参数与可变参数·····	163
8.5 对象参数·····	164
8.6 过程的嵌套与递归·····	166
8.7 变量与过程的作用范围·····	171
习题八·····	174
第9章 常用标准控件·····	180
9.1 控件类型·····	180
9.2 常用标准控件·····	181
9.3 图形控件·····	195
9.4 文件系统控件·····	199
习题九·····	203
第10章 多重窗体程序设计·····	208
10.1 多重窗体的基本操作·····	208
10.2 Sub Main 过程·····	210
10.3 闲置循环与 DoEvents 语句·····	212
10.4 Visual Basic 工程管理·····	214



习题十	218
第 11 章 对话框及菜单程序设计	220
11.1 对话框程序设计	220
11.2 菜单设计	228
习题十一	237
第 12 章 键盘与鼠标事件过程	240
12.1 键盘事件	240
12.2 鼠标事件	243
12.3 拖放操作	249
习题十二	252
第 13 章 数据文件	257
13.1 文件概述	257
13.2 顺序文件的基本操作	260
13.3 随机文件的操作	267
13.4 二进制文件	271
13.5 文件操作语句	272
习题十三	273
第 14 章 图形操作	276
14.1 直线控件和形状控件	276
14.2 坐标系统	279
14.3 绘图属性和事件	284
14.4 绘图方法	286
习题十四	290
第 15 章 Visual Basic 数据库技术	292
15.1 数据库的基本概念	292
15.2 数据库管理器	295
15.3 数据控件	299
15.4 Recordset 对象的属性与方法	303
15.5 数据绑定控件	306
15.6 SQL 语言	310
15.7 ActiveX 数据对象及其使用	313
习题十五	323
参考文献	326

第 1 章 Visual Basic 程序设计概述

Visual Basic 是由微软公司推出、在 Basic 基础上发展起来的 Windows 应用程序开发工具。Visual Basic 提供了开发 Windows 应用程序最迅速、最简洁的方法。本章将以 Visual Basic 6.0 为蓝本,对 Visual Basic 语言特征、功能特点和集成开发环境等内容进行简要介绍。

1.1 Visual Basic 语言简介

1.1.1 Visual Basic 语言发展简介

Visual 意为“可视化的”,指的是一种开发图形用户界面(GUI)的方法,所以 Visual Basic 是基于 Basic 的可视化的程序设计语言。Basic 指的是 BASIC(Beginners All-Purpose Symbolic Instruction Code,初学者通用符号指令代码)语言,一种在计算技术发展历史上应用得最为广泛的语言。Visual Basic 在原有 BASIC 语言的基础上进一步发展,包含了数百条语句、函数及关键词,其中很多和 Windows GUI 有直接关系。在 Visual Basic 中,一方面继承了 BASIC 所具有的程序设计语言简单易用的特点,另一方面在其编程系统中采用了面向对象、事件驱动的编程机制,用一种巧妙的方法把 Windows 的编程复杂性封装起来,提供了一种所见即所得的可视化程序设计方法。因此,Visual Basic 被认为是大多数电脑初学者的首选入门编程语言。

1.1.2 Visual Basic 语言的版本

Microsoft 公司于 1991 年推出 Visual Basic 1.0 版,获得巨大成功。随后 Microsoft 公司又分别在 1992 年、1993 年、1995 年和 1997 年相继推出了 2.0、3.0、4.0、5.0,直到 1998 年推出 6.0 版。2001 年,Microsoft 推出了 Visual Basic .NET,将 Visual Basic 语言提升到了全新的高度。随着版本的改进,Visual Basic 已逐渐成为简单易学、功能强大的编程工具。本书主要介绍 Visual Basic 6.0 版本。

Visual Basic 6.0 包括三种版本,它们分别为学习版、专业版和企业版。这三种版本是在相同的基础上建立起来的,以满足不同层次的用户需要。

学习版(Learning Edition):可使用一组工具来创建功能完备的 Windows 应用程序,它包括所有的内部控件、网格控件及数据绑定等控件。

专业版(Professional Edition):在学习版的基础上添加了一些专门的工具和一个报表编制工具。它主要针对计算机专业开发人员(除了具有学习版的全部功能外,还包括 ActiveX 和 Internet 控件开发工具之类的高级特性)。



企业版(Enterprise Edition): 包括了用于客户/服务器应用程序的工具(除了具有专业版的全部功能外, 还包括了一些特殊的工具)。

本书使用的是 Visual Basic 6.0 中文企业版, 但其内容可用于专业版, 所有程序可在专业版中运行, 大多数程序可在学习版中运行。

1.1.3 Visual Basic 语言的特点

1. 具有面向对象的可视化设计工具

在 Visual Basic 中, 应用面向对象的程序设计方法(OOP), 把程序和数据封装起来视为一个对象, 每个对象都是可视的。程序员在设计时只需用现有工具根据界面设计要求, 直接在屏幕上“画”出窗口、菜单、按钮、滚动条等不同类型的对象, 并为每个对象设置属性。程序员的编程工作仅限于编写相关对象要完成的功能程序, 因而程序设计的效率可大大提高。

2. 事件驱动的编程机制

事件驱动是非常适合图形用户界面的编程方式。在图形用户界面的应用程序中, 是由用户的动作即事件掌握着程序运行的流向, 每个事件都能驱动一段程序的运行。程序员只需要编写响应用户动作的代码, 而各个动作之间不一定有联系。

3. 提供了易学易用的应用程序集成开发环境

在 Visual Basic 集成开发环境中, 用户可设计界面、编写代码、调试程序, 直至把应用程序编译成可执行文件在 Windows 中运行, 使用户在友好的开发环境中工作。

4. 结构化的程序设计语言

Visual Basic 沿用了结构化程序设计的思想, 具有丰富的数据类型、众多的内部函数和结构化程序设计结构, 简单易学。

5. 支持多种数据库系统的访问

利用数据控件可访问的数据库系统有: Microsoft Access、Btrieve、dBase、Microsoft FoxPro 和 Paradox 等, 也可以访问 Microsoft Excel、Lotus1-2-3 等多种电子表格。

6. 支持动态数据交换(DDE)、动态链接库(DLL)和对象的链接与嵌入(OLE)技术

动态数据交换是 Microsoft Windows 除了剪贴板和动态链接库以外, 在 Windows 内部交换数据的第三种方式。利用这项技术可使 Visual Basic 开发的应用程序与其他 Windows 应用程序之间建立数据通信。

动态链接库中存放了所有 Windows 应用程序可以共享的代码和资源, 这些代码或函数可以用多种语言写出。Visual Basic 利用这些技术可以调用任何语言产生的 DLL, 也可以调用 Windows 应用程序接口(API)函数, 以实现 SDK 所能实现的功能。

对象的链接与嵌入是 Visual Basic 访问所有对象的一种方法。利用 OLE 技术, Visual Basic 将其他应用软件作为一个对象嵌入到应用程序中进行操作, 也可以将各种基于 Windows 的应用程序嵌入到 Visual Basic 应用程序中, 实现声音、图像、动画等多媒体的功能。

7. 完备的 HELP 联机帮助功能

与 Windows 环境下的软件一样, 在 Visual Basic 中, 利用帮助菜单和 F1 功能键, 用户可以随时方便地得到所需要的帮助信息。Visual Basic 帮助窗口中显示了有关的示例代码, 通过复制、粘贴操作可获取大量的示例代码, 为用户的学习和使用提供了极大的方便。



1.2 Visual Basic 6.0 的安装、启动与退出

1.2.1 Visual Basic 6.0 的运行环境

- (1) CPU: 最低配置应不低于 486DX, 一般应选择奔腾 133 或更高主频的 CPU。
- (2) 内存: 最低不能少于 16MB。一般应选用 32MB, 以便能提高运行速度。如果用于图形处理或使用 ActiveX, 则内存最好在 32MB 或 64MB 以上。如果要处理三维动画, 则内存配置应大于 64MB 或 128MB, 并相应增大显存容量。
- (3) 磁盘空间:
 - 学习版: 典型安装需要 48MB, 完全安装需要 80MB。
 - 专业版: 典型安装需要 48MB, 完全安装需要 80MB。
 - 企业版: 典型安装需要 128MB, 完全安装需要 147MB。
- MSDN: 若使用 Visual Basic 帮助文件, 则必须安装, 至少需要 67MB。
- 另外, 可以根据实际需要安装一些附加控件。
- (4) 显示设备: VGA 或更高分辨率的显示器, 建议使用 SuperVGA。
- (5) 读取设备: 应配置 8 倍速或更高速度的 CD-ROM 驱动器。
- (6) 操作系统: Windows9X 或 WindowsNT 3.51 以上版本。

1.2.2 Visual Basic 6.0 的安装

在 CD-ROM 驱动器中插入 Visual Basic 6.0 系统 CD 盘。安装程序在 CD 盘的根目录下, 运行安装程序 Setup.exe, 即可进入“安装程序向导”, 如图 1-1 所示。

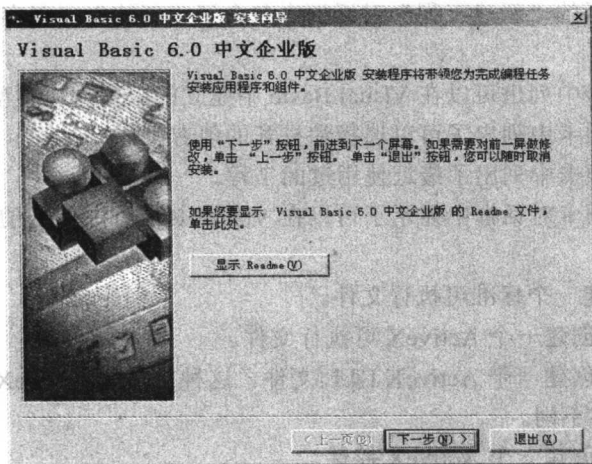


图 1-1 Visual Basic 安装向导

用户只需按照提示一步一步进行简单操作, 便可完成安装。有关 Visual Basic 的信息安装说明, 可以参阅安装盘中的 Readme 文件。

当安装好 Visual Basic 系统后, 若需要添加或删除某些部件, 可插入安装盘重新执行



Setup.exe 文件，安装程序将自动检测当前系统安装的 Visual Basic 组件，当用户单击“添加/删除”按钮后，在安装维护对话框中选定要添加的部件或撤销选定要删除的部件。

1.2.3 Visual Basic 6.0 的启动和退出

1. Visual Basic 6.0 的启动

启动 Visual Basic 6.0 和启动其他 Windows 应用程序非常相似，常用方法有：

- (1) 双击桌面上的 Visual Basic 图标。
- (2) 选择“开始”→“程序”→“Microsoft Visual Basic 6.0 中文版”→“Microsoft Visual Basic 6.0 中文版”。

启动 Visual Basic 后系统弹出如图 1-2 所示“新建工程”对话框。

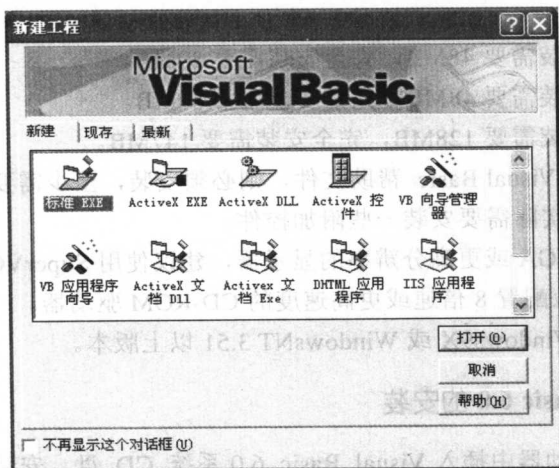


图 1-2 新建工程对话框

在如图 1-2 所示的“新建工程”对话框中包含了“新建”、“现存”和“最新”3 个选项卡：

新建：该选项卡中列出可以在 Visual Basic 中建立的应用程序类型，共有 13 种。

现存：这个选项卡中列出了可以选择和打开的现有工程。

最新：这个选项卡中列出了最近使用过的工程。

在“新建”选项卡对话框中显示了可以在 Visual Basic 中建立的应用程序类型，具体说明如下：

标准 EXE：创建一个标准可执行文件。

ActiveX EXE：创建一个 ActiveX 可执行文件。

ActiveX DLL：创建一个 ActiveX DLL 文件。这种文件与 ActiveX EXE 文件在功能上是相同的，只是包装不同。

ActiveX 控件：创建一个 ActiveX 控件。

VB 应用程序向导：这个向导可以帮助用户建立新的应用程序框架。用户在开发自己的工程时可能使用到。

VB 向导管理器：能够帮助创建自定义的向导，效果与 Visual Basic 提供的向导非常相似。

数据工程：创建一个数据工程。



IIS 应用程序：创建一个 IIS 应用程序。

外接程序：建立自定义的 Visual Basic 外接程序。

ActiveX 文档：ActiveX 文档实际上是可以在支持超级链接的容器中运行的 Visual Basic 应用程序。这个环境可以就是一个 Web 浏览器，如 Internet Explorer。

DHTML 应用程序：创建一个 DHTML 应用程序。

用户只要从对话框中选择要建立的工程类型(如：标准 EXE)，单击“打开”按钮，或直接双击所需要的工程类型，即可进入 Visual Basic 集成开发环境(IDE，Integrated Development Environment)主窗口，如图 1-3 所示。

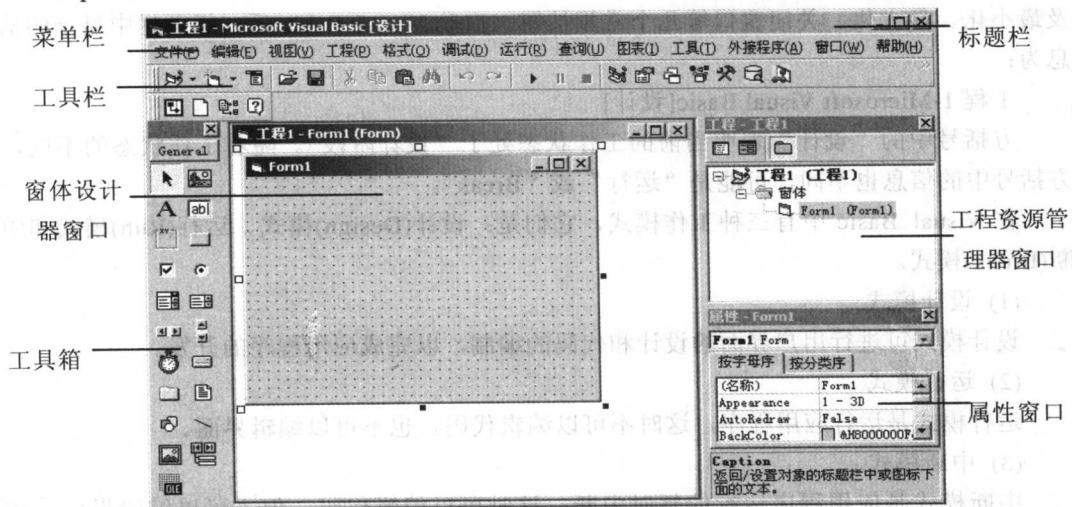


图 1-3 Visual Basic 界面组成

2. Visual Basic 的退出

以下四种方式，均可以退出 Visual Basic 应用程序。

- (1) 选择“文件”菜单中的“退出”。
- (2) 单击窗口右上角关闭按钮。
- (3) 单击窗口左上角(标题栏最左端的窗口控制菜单框)，从快捷菜单中选择“关闭”。
- (4) 按快捷键 Alt+Q。

在退出 Visual Basic 集成开发环境之前，如果用户对当前程序已进行了修改而未存盘，Visual Basic 将显示一个对话框，询问用户是否将其存盘，如果单击“是”按钮，则表示对当前程序进行保存，单击“否”，则系统会自动放弃对当前程序的保存。

1.3 Visual Basic 集成开发环境

Visual Basic 集成开发环境与其 Windows 环境下的许多应用程序相似，所不同的是，Visual Basic 是由一组独立的窗口组成，主要包括：主窗口、窗体窗口、属性窗口、工程资源管理器窗口、代码窗口、窗体布局窗口、对象浏览器窗口、工具箱窗口、立即窗口。



Visual Basic 的应用程序通过图 1-3 所示的集成开发环境进行开发。下面介绍各窗口的功能和使用方法。

1.3.1 主窗口

主窗口与 Office 家族的软件类似,它由标题栏、菜单栏和工具栏组成,含有一个 Windows 应用程序应具有的所有组成元素。主窗口主要包括如下内容:

1. 标题栏

标题栏是屏幕顶部的水平条,显示应用程序的名字。标题栏中含有当前工程的名字以及最小化、最大化、关闭窗口等几个常规按钮。启动 Visual Basic 后,标题栏中显示的信息为:

工程 1-Microsoft Visual Basic[设计]

方括号中的“设计”表明当前的工作状态处于“设计阶段”。随着工作状态的不同,方括号中的信息也不同,可能是“运行”或“Break”。

在 Visual Basic 中有三种工作模式,它们是:设计(Design)模式、运行(Run)模式和中断(Break)模式。

(1) 设计模式

设计模式可进行用户界面的设计和代码的编制,以完成应用程序的开发。

(2) 运行模式

运行模式是运行应用程序,这时不可以编辑代码,也不可以编辑界面。

(3) 中断模式

中断模式是应用程序运行的暂时中断,这时可以编辑代码,但不可以编辑界面。按 F5 键或单击“继续”按钮,程序继续运行,单击“结束”按钮程序停止运行。在此模式下会弹出“立即”窗口,在窗口内可输入简短的命令,并立即执行。

当程序运行过程中出现错误时,往往需要进行手工调试,此时要做的就是中断程序运行,随后屏幕出现调试窗口以供检查。

2. 菜单栏

菜单是提供给用户设计和调试程序时的必要部分,Visual Basic 沿用了 Windows 的菜单形式,菜单栏中含有 13 个下拉菜单。这是程序开发过程中需要的命令。

3. 工具栏

Visual Basic 提供了 4 种工具栏,包括编辑、标准、窗体编辑器和调试,用户可根据需要定义用户自己的工具栏。一般情况下,集成环境中只显示标准工具栏,其他工具栏可以通过“视图”菜单中的“工具栏”命令打开或关闭。单击某个按钮,即可执行对应的相关操作。

“标准”工具栏(如图 1-4 所示)及其各按钮的功能如表 1-1 所示。



图 1-4 标准工具栏



表 1-1 标准工具栏按钮

图标	名称	功能
	添加工程	添加一个新工程, 相当于“文件”菜单中的“添加工程”命令
	添加窗体	在工程中添加一个新窗体, 相当于“工程”菜单中的“添加窗体”命令
	菜单编辑器	显示菜单编辑器对话框
	打开工程	用于打开已有的工程文件
	保存工程	用于保存当前的工程文件
	启动	开始运行当前的工程
	中断	暂时中断当前工程的运行
	结束	结束当前工程的运行
	工程资源管理器	打开工程资源管理器窗口
	属性窗口	打开属性窗口
	窗体布局窗口	打开窗体布局窗口
	对象浏览器	打开对象浏览器
	工具箱	打开工具箱窗口
	数据视图窗口	打开数据视图窗口
	可视化部件管理器	打开可视化部件管理器

1.3.2 窗体设计器窗口

窗体设计器窗口(如图 1-5 所示)简称窗体(Form), 是应用程序最终面向用户的窗口, 对应于应用程序的运行结果。各种图形、图像、数据等都是通过窗体或窗体中的控件显示出来的。当打开一个新的工程文件时, Visual Basic 建立一个空的窗体, 并命名为 Form x(这里的 x 为 1, 2, 3...)。启动 Visual Basic 后, 窗体的名字为 Form1, 其操作区中布满了小点, 这些小点是供对齐用的。如果想清除这些小点, 或者想改变点与点之间的距离, 选择“工具”菜单中的“选项”命令(“通用”选项卡)。

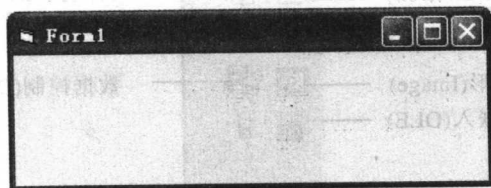


图 1-5 窗体设计器窗口

1.3.3 工程管理器(Project Explorer)窗口

应用程序是建立在工程基础上完成的, 而一个工程则是各种类型的文件的集合。这些文件包括工程文件(.Vbp)、窗体文件(.Frm)、标准模块文件(.Bas)、类模块文件(.Cls)、工程组文件(.Vbg)和资源文件(.Res)。

工程文件就是与该工程有关的所有文件和对象的清单, 这些文件和对象自动链接到工程文件上, 每次保存时, 其相关文件信息随之更新。当然, 某个工程下的对象和文件也可



供其他工程共享使用。在工程的所有对象和文件被汇集在一起并完成编码后,就可以编译工程,生成可执行文件。

工程管理器窗口(如图 1-6 所示)类似于 Windows 下的资源管理器,在这个窗口中列出了当前工程的窗体和模块,其结构用树状的层次管理方法显示。

在工程管理器窗口中有“查看代码”、“查看对象”和“切换文件夹”3 个按钮。单击“查看代码”按钮,可打开代码编辑器查看代码;单击“查看对象”按钮,可打开窗体设计器查看正在设计的窗体;单击“切换文件夹”按钮,则可隐藏或显示包含对象文件夹中的个别项目列表。

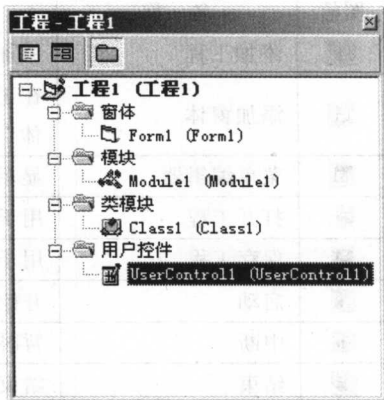


图 1-6 工程资源管理器

1.3.4 Visual Basic 6.0 的工具箱窗口

工具箱窗口(如图 1-7 所示)由工具图标组成,这些图标是 Visual Basic 应用程序的构件,称为图形对象或控件(Control),每个控件由工具箱中的一个工具图标表示。用户可以通过“视图”菜单下的“工具箱”选项进行显示和隐藏工具箱。

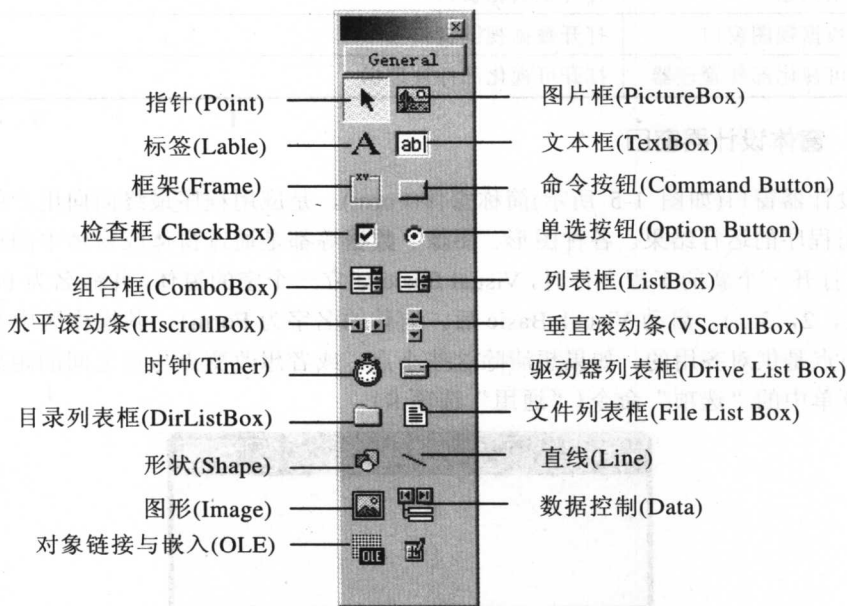


图 1-7 工具箱窗口

在 Visual Basic 中,工具箱除了已有的“通用(General)”工具外,还可以通过“工程”菜单下的“部件”选项添加系统所提供的部分控件,具体操作将在标准控件一章中详细论述。

1.3.5 属性窗口

属性窗口(如图 1-8 所示)列出对选定窗体和控件的属性设置值。属性是指对象的特征,如大小、标题或颜色等。属性显示方式分为两种,即按字母顺序和按分类顺序,分别通过单击相应的选项卡来实现。



属性窗口包括了所有窗体或控件的属性，如果要更改属性，可以通过该窗口来修改。另外，对于只读属性之外的属性，可以在程序运行过程中动态设置。属性窗口由以下几个部分组成：

(1) 标题栏：包括了窗体或控件的名称。

(2) 对象列表框：其中含有窗体或控件的名称，单击它右侧的箭头，可查看所选窗体包含的对象的列表，可以选择当前窗体或其他控件。

(3) 属性列表框：有“按字母序”和“按分类序”两个按钮选项卡，列出所选对象(控件)的属性。左边为对象的属性，右边是每个属性对应的当前设置。对于不同的对象它所列出的属性是不同的。用户可以选择某一属性，然后对该属性值进行设置或修改。

(4) 属性功能说明：当在属性列表框选择某一属性时，在该区域显示所选属性的含义。

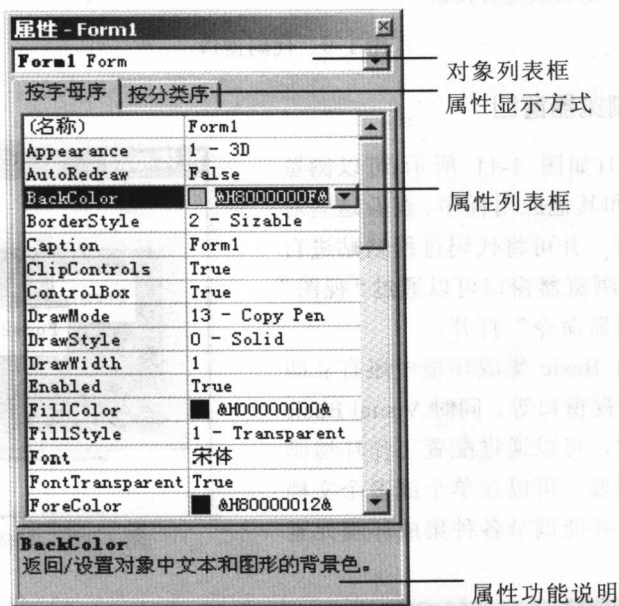


图 1-8 属性窗口

1.3.6 代码窗口

代码窗口又称代码编辑器，各种通用过程和事件过程代码均在此窗口上编写和修改。通过(1)双击窗体的任何地方；(2)右击鼠标，选择快捷菜单中的“查看代码”；(3)单击工程窗口中的“查看代码”按钮；(4)或者选择“视图”菜单中的“代码窗口”四种方法均可打开代码窗口，如图 1-9 所示。

1.3.7 窗体布局窗口

窗体布局窗口(如图 1-10 所示)中有一个表示屏幕的小图像，用来布置应用程序中各窗体的位置，使用鼠标拖动窗体布局窗口中的小窗体图标，可方便地调整程序运行时窗体显示的位置。