



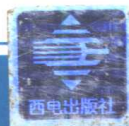
全新计算机应用技术丛书



Visual Basic

实用教程

薛亮 编著



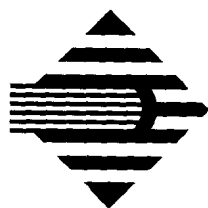
西安电子科技大学出版社

<http://www.xduph.com>

 全新计算机应用技术丛书

Visual Basic 实用教程

□ 薛 亮 编著



西安电子科技大学出版社

2002

内 容 简 介

本书介绍了 Visual Basic 集成开发环境、Visual Basic 语言基础、程序结构、子过程、常用控件及窗体、Windows API 接口、通用对话框和剪贴板的应用、菜单的设计、文件操作、图形基础、程序调试和出错处理、鼠标事件和 OLE 操作、数据库操作、数据库操作的扩展等内容。书中概念清晰，逻辑性强，层次分明，例题丰富，并附有典型的习题。

本书可作为培训类教材使用，可供初学编程的人员参考使用，也可供大专院校相关专业的师生参考使用。

图书在版编目 (CIP) 数据

Visual Basic 实用教程 / 薛亮编著.

—西安: 西安电子科技大学出版社, 2002.6

(全新计算机应用技术丛书)

ISBN 7-5606-1130-3

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

中国版本图书馆 CIP 数据核字 (2002) 第 031361 号

责任编辑 臧延新 张晓燕

出版发行 西安电子科技大学出版社 (西安市太白南路 2 号)

电 话 (029)8227828 邮 编 710071

http: //www.xduph.com E-mail: xdupfxb@pub.xaonline.com

经 销 新华书店

印 刷 西安兰翔印刷厂

版 次 2002 年 6 月第 1 版 2002 年 6 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印张 15

字 数 353 千字

印 数 1~4 000 册

定 价 18.00 元

ISBN 7-5606-1130-3 / TP · 0573

XDUP 1401001-1

*** 如有印装问题可调换 ***

本书封面贴有西安电子科技大学出版社的激光防伪标志，无标志者不得销售。

前 言

Visual Basic 是一种可视化编程语言，它将编程人员从面向过程的细节工作中解脱出来，不仅可以帮助用户方便地设计出良好的用户界面，而且基于事件的响应机制使得程序开发变得更为简洁。

本书是针对不具备程序设计知识的初学者而编写的，它具有以下特点：

(1) 从讲解计算机语言的基本知识、基本结构等基础内容出发，不要求读者具备程序设计知识。

(2) 知识最新，内容综合。本书除了介绍 Visual Basic 的基本内容外，还涉及到更广泛和更进一步的内容，而且所涉及的内容尽量做到全面、准确。

(3) 思路清晰，实例丰富。对于程序设计语言的学习，就要做到理论联系实际。本书在讲解理论的同时，辅以大量的实例，便于读者更好地理解和掌握。

(4) 举一反三，融会贯通。可视化编程是一种程序设计的新思路。本书针对某一类控件，只介绍了其中一个控件的用法，由于这一类控件的调用和实现思想都是一致的，因而读者可以在掌握一种方法的基础上举一反三，触类旁通。

(5) 习题典型，便于练习。本书每章中都附有一定的习题，主要是针对本章所涉及的知识点的具体操练而编写的。通过练习这些习题，读者可以很好地理解本章的内容。

由于作者水平有限，书中不足、疏漏之处难免，恳请广大读者提出宝贵意见和建议。

编 者

2002 年 1 月

目 录

第1章 Visual Basic 概述.....1	3.1.2 用户交互函数和过程..... 31
1.1 Visual Basic 概述.....1	3.2 程序结构..... 34
1.1.1 Visual Basic的功能特点.....1	3.2.1 选择结构..... 34
1.1.2 Visual Basic的启动和退出.....2	3.2.2 循环结构..... 38
1.1.3 Visual Basic的集成开发环境.....3	3.2.3 其它辅助控制语句..... 40
1.1.4 Visual Basic的帮助系统使用.....7	3.3 数组..... 40
1.2 Visual Basic中对象的概念.....8	3.3.1 静态数组..... 41
1.2.1 类和对象.....8	3.3.2 动态数组..... 41
1.2.2 基本控件属性.....11	3.3.3 控件数组..... 43
习题一.....14	习题三..... 45
第2章 语言基础.....15	第4章 子过程(Sub、Function、 Property 和 Event)..... 46
2.1 编码规则.....15	4.1 子过程的定义..... 46
2.2 数据类型.....16	4.2 过程的调用..... 48
2.2.1 标准数据类型.....16	4.3 参数传递..... 49
2.2.2 自定义类型.....18	4.3.1 传址与传值..... 49
2.3 变量与常量.....19	4.3.2 数组参数的传递..... 50
2.3.1 变量或常量的命名规则.....19	4.4 变量、过程的作用域..... 51
2.3.2 变量声明.....19	4.4.1 过程的作用域..... 51
2.3.3 常量.....21	4.4.2 变量的作用域..... 52
2.4 运算符.....22	4.4.3 变量的生存期..... 54
2.5 表达式.....24	4.5 常用方法..... 55
2.6 常用内部函数.....25	4.5.1 Print方法..... 55
2.6.1 数学函数.....25	4.5.2 Cls方法..... 56
2.6.2 转换函数.....26	4.5.3 Move方法..... 56
2.6.3 字符串函数.....26	4.6 用户定义类型..... 56
2.6.4 日期函数.....27	习题四..... 57
2.6.5 格式输出函数.....27	第5章 常用控件与窗体..... 58
2.6.6 Shell函数.....29	5.1 标签控件..... 58
习题二.....30	5.1.1 标签的属性..... 58
第3章 Visual Basic 程序结构.....31	5.1.2 标签接收的事件..... 58
3.1 Visual Basic基本语句.....31	5.2 命令按钮控件..... 58
3.1.1 赋值语句.....31	

5.2.1 命令按钮的属性	58
5.2.2 命令按钮接收的事件	59
5.3 文本框控件	59
5.3.1 文本框的属性	60
5.3.2 文本框响应的事件	61
5.3.3 文本框使用的方法	61
5.4 单选钮和检查框	61
5.4.1 单选钮和检查框的属性	62
5.4.2 单选钮和检查框接收的事件	62
5.5 框架	62
5.5.1 框架的属性	63
5.5.2 框架响应的事件	63
5.6 滚动条	64
5.6.1 滚动条的属性	64
5.6.2 滚动条响应的事件	65
5.7 列表框	66
5.7.1 列表框的属性	66
5.7.2 列表框使用的方法	67
5.8 组合框	69
5.9 时钟	70
5.9.1 时钟的属性	70
5.9.2 时钟响应的事件	70
5.10 窗体	72
5.10.1 窗体的属性	72
5.10.2 窗体响应的事件	73
5.10.3 窗体使用的方法	73
5.10.4 多重窗体	73
习题五	74
第 6 章 使用 Windows API	75
6.1 在 Visual Basic 中访问 Win32 API 函数	75
6.2 查看 API 函数	76
6.3 使用 API 函数例程	77
6.3.1 软件封面	77
6.3.2 注册表	79
6.3.3 其它	82
习题六	84
第 7 章 通用对话框和剪贴板的应用	85

7.1 通用对话框	85
7.1.1 通用对话框的基本属性和打开方法	85
7.1.2 “打开”对话框	86
7.1.3 “另存为”对话框	89
7.1.4 “颜色”对话框	89
7.1.5 “字体”对话框	90
7.1.6 “打印”对话框	91
7.1.7 “帮助”对话框	92
7.2 剪贴板的应用	93
7.2.1 文本剪贴	94
7.2.2 图形剪贴	95
习题七	97

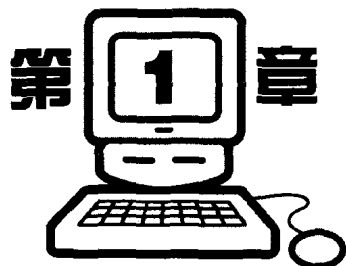
第 8 章 菜单的设计和 MDI

程序设计	98
8.1 菜单的设计	98
8.2 多文档界面(MDI)	102
8.2.1 创建 MDI 窗体及其子窗体	103
8.2.2 显示 MDI 窗体及其子窗体	103
8.2.3 维护子窗体的状态信息	105
8.2.4 用 QueryUnload 卸载 MDI 窗体	106
8.2.5 多文档界面应用程序中的“窗口”菜单	106
8.2.6 应用举例——创建一个 MDI NotePad	107
8.3 工具栏	109
8.3.1 手工创建工具栏	109
8.3.2 工具栏应用举例	110
8.3.3 协调菜单与工具栏的外观	110
8.4 对话框	111
8.4.1 设计自定义对话框	111
8.4.2 显示自定义对话框	112
习题八	112

第 9 章 文件操作

9.1 文件及其结构	113
9.2 驱动器、目录和文件列表框	114
9.3 文件系统对象模型	116
9.3.1 访问驱动器对象	118

9.3.2 访问文件夹	119	12.2.1 OLE基本术语	161
9.3.3 访问文件	120	12.2.2 OLE容器控件	162
9.4 传统的方式操作文件	125	12.2.3 OLE自动化	166
9.4.1 顺序文件	125	12.3 对象拖放	168
9.4.2 随机文件	127	12.3.1 对象拖放	168
9.4.3 二进制文件	129	12.3.2 OLE拖放	173
习题九	130	习题十二	181
第 10 章 图形基础	131	第 13 章 Visual Basic 和数据库操作	182
10.1 坐标系	131	13.1 数据库概念	182
10.1.1 坐标系统	131	13.2 数据控件	184
10.1.2 自定义坐标系	132	13.2.1 数据控件属性	184
10.2 绘图属性	133	13.2.2 数据控件的常用方法	186
10.3 图形控件	134	13.2.3 数据输入	189
10.4 图形方法	140	13.2.4 高级约束数据控件	199
习题十	144	13.3 使用对象变量访问数据库	204
第 11 章 程序调试和出错处理	145	习题十三	208
11.1 错误类型	145	第 14 章 Visual Basic 与数据库	
11.2 调试和排错	146	的扩展	209
11.2.1 调试窗口	146	14.1 开放数据库连接性(ODBC)	209
11.2.2 逐语句或逐过程运行	147	14.2 数据访问方式的演变	213
11.2.3 设置断点	148	14.2.1 常用的方法	213
11.3 出错处理	148	14.2.2 主要方法	215
习题十一	157	14.3 ADO Data控件	228
第 12 章 鼠标事件和 OLE 拖放	158	14.4 数据库管理器	231
12.1 Visual Basic的鼠标事件	158	习题十四	231
12.2 Visual Basic与OLE	161		



Visual Basic 概述

1.1 Visual Basic 概述

Visual Basic 是 Microsoft 公司在 Quick Basic 的基础上为开发 Windows 应用程序而提供的强有力的开发环境和工具，是具有很好的图形用户界面(Graphic User Interface)的程序语言。

Visual Basic 采用面向对象和事件驱动的两程序设计机制，把过程化和结构化编程集合在一起。正是由于它在应用程序开发中的图形化构思，使得开发者可以非常容易地创建一个窗口。

Visual 指的是开发图形用户界面(GUI)的方法，它不需要编写大量代码去描述界面元素的外观和位置，而只要把预先建立的对象拖放到窗体上即可，这就叫做可视化编程。

1.1.1 Visual Basic 的功能特点

Visual Basic 是面向对象的可视化编程语言中最简单、最容易使用的语言。作为一门程序设计语言，Visual Basic 程序设计具有如下的功能特点。

1. 可视化的设计平台

面向过程的程序设计存在的缺点是在设计过程中看不到程序的实际显示效果，必须在程序运行时才能观察到。如果用户对程序的界面不满意，就要返回到程序中去修改。这样反复修改，会影响编程的效率。Visual Basic 提供的可视化设计平台，把 Windows 界面设计的复杂性封装起来。程序员不必再为界面的设计而编写大量程序代码，只需按设计的要求，用 Visual Basic 中提供的控件工具在设计界面上画出各种对象，而这些界面设计代码是由 Visual Basic 自动产生的，程序员所编写的只是实现程序功能的那部分代码，从而大大提高了编程的效率。

2. 面向对象的设计方法

Visual Basic 采用面向对象的编程方法(OOP)，把程序和数据封装起来作为一个对象，并为每个对象赋予相应的属性，调用相应的方法。在设计对象时，不必编写建立和描述对象的程序代码，而是用工具画在界面上，由 Visual Basic 自动生成对象的程序代码并封装起来。



3. 事件驱动的编程机制

Visual Basic 通过对事件的响应完成具体的操作。Visual Basic 程序设计针对不同的对象，可由用户操作引发某个事件来驱动完成某种特定的功能，或由事件驱动程序调用通用过程来执行指定的操作。

4. 结构化的设计语言

同面向过程的程序设计不同的是，Visual Basic 应用程序无需具有明显的开始和结尾，而是由若干个子过程组成。这些子过程都是针对不同对象的操作。

5. 充分利用 Windows 资源

Visual Basic 通过三种技术来实现与 Windows 资源的交互，这三种技术是：动态数据交换(DDE)编程技术、对象链接与嵌入(OLE)技术以及动态链接库(DLL)技术。DDE 技术建立动态的数据交换，使得不同应用程序之间的通信成为可能；OLE 技术可以将 Windows 应用程序看作一个对象，将不同的对象连接起来，嵌入到 Visual Basic 的某个应用程序中，实现信息的集合化；DLL 技术可以将 C/C++编写的程序加入到 Visual Basic 的应用程序中，或是调用 Windows 应用程序接口(API)函数，实现 SDK 所具有的功能。

6. 开放的数据库功能与网络支持

由于应用程序设计中三层体系结构的出现，使得网络、应用程序以及数据库的连接更加紧密起来。Visual Basic 中提供了一些接口来实现同网络以及数据库的连接。Visual Basic 有很强的数据库管理功能，可以通过直接访问或建立连接的方式访问并操作后台数据库。对后台数据库的访问主要是通过 ODBC 功能实现的，从而获得了很好的开放性；而对后台数据库的操作是通过 Visual Basic 提供的简单的命令集来实现的，也可以使用结构化查询语言以及一些高级的 ActiveX 控件或 API 函数等。

1.1.2 Visual Basic 的启动和退出

在使用 Visual Basic 编写应用程序之前，首先要在计算机中安装 Visual Basic 系统软件，通常按照安装软件的导航直接进行操作就可以了。

安装完毕后，在“开始”菜单的“程序”组中将多出一个“Microsoft Visual Basic 6.0 中文版”菜单选项，这时就可以启动 Visual Basic 了。启动 Visual Basic 的步骤如下：

(1) 单击 Windows 任务栏中的“开始”按钮，从“程序”组中选择其中的“Microsoft Visual Basic 6.0 中文版”菜单选项，启动 Visual Basic 6.0；另外一种启动方法是将图标拖到桌面上，直接单击图标。

(2) 启动 Visual Basic 6.0 后，首先显示“New Project”对话框，如图 1.1 所示。

(3) 系统默认选中“New Project”选项卡中的“Standard EXE”项。双击“New Project”选项卡中的“Standard EXE”项，或直接单击“OK”按钮，将进入如图 1.2 所示的 Visual Basic 的集成开发环境。

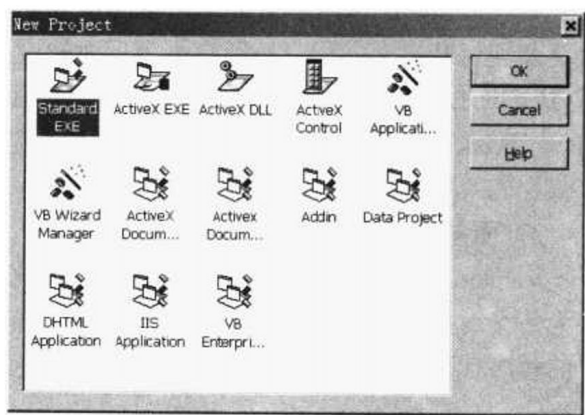


图 1.1 “New Project”对话框

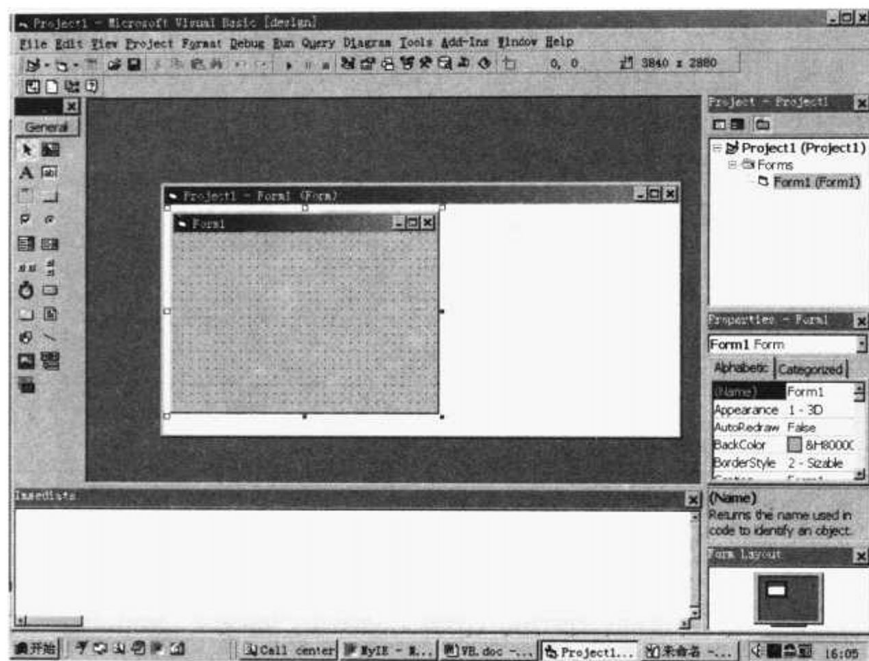


图 1.2 Visual Basic 的集成开发环境

(4) 退出 Visual Basic 6.0 可单击 Visual Basic 窗口中的“关闭”按钮，或选择“文件”菜单中的“退出”命令，Visual Basic 会自动判断用户是否修改了工程的内容，并询问用户是否保存文件或直接退出。

1.1.3 Visual Basic 的集成开发环境

软件开发通常需要经过设计和代码编辑、编译和调试、运行等几个步骤。传统的软件开发工具中，这些步骤一般需要在各自独立的支持环境下完成，而 Visual Basic 把支持软件开发的各功能都集成在一个公共的工作环境中，称之为“集成开发环境”，如图 1.2 所示。该环境集中提供了程序开发所需要的各种工具、窗口和方法，使得用户可以非常方便地在软件开发的各阶段工作中来回切换，从而提高开发效率。

Visual Basic 6.0 集成开发环境主要包括以下 8 个部分。

1. 标题栏和菜单栏

标题栏显示的是窗体控制菜单图标，当前激活的工程名称，当前工作模式以及最小化、最大化 / 还原、关闭按钮。

菜单栏起了统领程序设计所有功能模块的作用，包括程序设计过程中最常用的一些命令。菜单项包括“文件”、“编辑”、“视图”、“工程”等，单击这些菜单名即可打开下拉菜单。下拉菜单中显示了各功能子菜单，以及执行该项功能的热键和快捷键。

2. 工具栏和数字显示区

1) 工具栏

菜单栏下面就是工具栏，工具栏提供了一些常用命令的快速访问按钮。单击某个按钮，即可执行相应的操作。



Visual Basic 集成开发环境默认的工具栏是“标准”工具栏。如果对准菜单栏或工具栏并单击右键，则弹出工具栏快捷菜单，这时可进行“标准”、“编辑”和“调试”等工具栏的显示 / 隐藏的切换。工具栏可以紧贴在菜单栏之下，也可拖放到窗体的其它任何地方。

“标准”工具栏按钮的图标及其功能如表 1.1 所示。

表 1.1 “标准”工具栏按钮的图标及其功能

图 标	名 称 与 功 能
	添加标准 EXE 工程——添加新的工程到工作组中，单击其右边的箭头，将弹出一个下拉菜单，可以从中选择需要添加的工程类型
	添加窗体——添加新的窗体到工程中，单击右边的下拉箭头，将弹出一个下拉菜单，可以从中选择需要添加的窗体类型
	菜单编辑器——显示菜单编辑器对话框
	打开工程——打开已有的工程文件
	保存工程——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——打开对象浏览器对话框
	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

2) 数字显示区

数字显示区包含两个部分，左数字区显示的是对象的坐标位置(窗体工作区的左上角为坐标原点，即(0, 0)位置)，右数字区显示的是对象的高度(向下递增)和宽度，表示对象的大小。

3. 工具箱

1) 标准工具箱

Visual Basic 的标准工具箱包含了建立应用程序所需的各种控件，如图 1.3 所示。除了包含一些默认的基本控件外，Visual Basic 还提供了许多扩展的 ActiveX 控件，可以被添加到工具箱中。工具箱作为一个类的容器，可以根据具体应用程序的需要进行一些控件的添加。具体的添加方法如下：

(1) 在工具箱的空白处单击鼠标右键，弹出的快捷菜单如图 1.4 所示。在此快捷菜单中选择“Components”，或单击“Project”菜单中的“Components”子菜单，即可弹出如图 1.5 所示的“Components”对话框。



图 1.3 Visual Basic 的标准工具箱

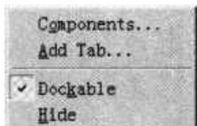


图 1.4 快捷菜单



图 1.5 “Components”对话框

(2) 在打开的“Components”对话框中，将需要的控件加上选中的标志(如图 1.5 所示)，然后单击“确定”按钮后退出，所选择的控件即可添加到工具箱中。

(3) 删除工具箱中的 ActiveX，只需在上述操作中去掉选中标志即可。

2) 选项卡

Visual Basic 6.0 中是通过定义选项卡来安排并组织控件的，设计步骤如下：

(1) 在工具箱的空白处单击鼠标右键，在弹出的快捷菜单中选择“Add Tab”，添加选项卡。

(2) 在打开的“New Tab Name”对话框中输入选项卡名称，如 Advanced，然后单击“确定”按钮，出现如图 1.6 所示的对话框。

(3) 根据具体的控件添加方法进行添加。通过单击选项卡名称可在不同的选项卡之间进行切换。

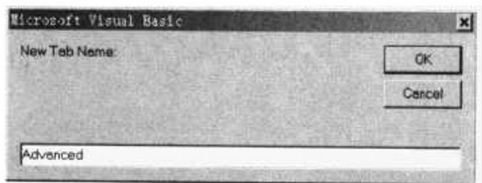


图 1.6 “添加选项卡”对话框



4. “工程管理器”窗口

在 Visual Basic 中, 所有的应用程序都是以工程为载体的。工程就是指用于创建一个应用程序的所有文件的集合。这些文件在工程管理器中是以层次的形式存在的。第一层为工程文件(vbp); 第二层为窗体文件(frm)、类模块文件(cls)、标准模块文件(bas), 在第二层还有一些资源文件(res)和 ActiveX 文件(ocx)等。

“工程管理器”窗口如图 1.7 所示, 它采用 Windows 资源管理器式的界面, 有下面三个按钮:

- (1) “查看代码”按钮, 可打开代码编辑器查看代码。
- (2) “查看对象”按钮, 可打开窗体设计器查看正在设计的窗体对象。
- (3) “切换文件夹”按钮, 可隐藏或显示包含在对象文件夹中的个别项目列表。

5. “属性”窗口

在 Visual Basic 集成开发环境的默认视图中, “属性”窗口位于工程窗口的下面。单击工具栏中的“属性窗口”按钮, 或选取“视图”菜单中的“属性窗口”子菜单, 均可打开“属性”窗口。“属性”窗口如图 1.8 所示。

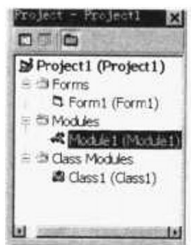


图 1.7 “工程管理器”窗口



图 1.8 “属性”窗口

“属性”窗口包含选定对象的属性列表, 对象的属性可以在“属性”窗口中进行初始化设置, 而且这些设置可以在程序的运行状态进行改变。“属性”窗口分为以下 5 部分:

- (1) 标题栏, 表示所针对表单的属性。
- (2) 对象下拉列表框, 表示当前选中的对象, 点击下拉箭头可显示窗体设计器中所有的对象。
- (3) 选项卡, 分为按字母排序和按分类排序两种排序方式。
- (4) 属性列表框, 列出当前选中对象的属性列表, 左半边为属性名称, 右半边为属性值。
- (5) 属性说明, 显示当前属性的简要说明。

6. 窗体设计器

应用程序的用户界面是存放各种可视化控件对象的容器。窗体设计器又称为“对象”窗口, 每一个应用程序中的窗体, 都有与之对应的窗体设计器。窗体的名称不能相同。

窗体上的对象可以任意移动、改变大小。可以根据用户的需要设计用户界面窗口。

在窗体的空白区域单击右键, 将弹出快捷菜单。单击相应的按钮可切换到“代码”、“菜单编辑器”、“属性”等窗口, 还可以选择“锁定控件”和“粘贴”, 如图 1.9 所示。

7. “窗体布局”窗口

“窗体布局”窗口中有一个表示屏幕的小图像, 用来显示实际运行时窗体在屏幕中的位置。可以用鼠标拖动其中的窗体小图标来调整窗体位置。“窗体布局”窗口如图 1.10 所示。

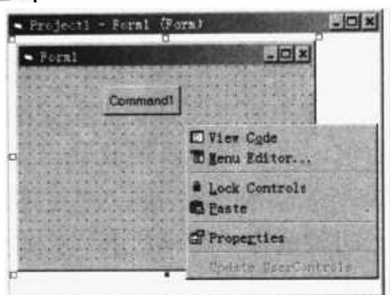


图 1.9 窗体设计器



图 1.10 “窗体布局”窗口

8. “立即”窗口

“立即”窗口可以进行一些简单值的查询，如图 1.11 所示。



图 1.11 “立即”窗口

1.1.4 Visual Basic 的帮助系统使用

使用 Visual Basic 进行程序设计时，经常会遇到一些问题。为解决这些问题，Visual Basic 6.0 为用户提供了内容丰富、使用方便的在线帮助系统。Visual Basic 帮助文件界面如图 1.12 所示。

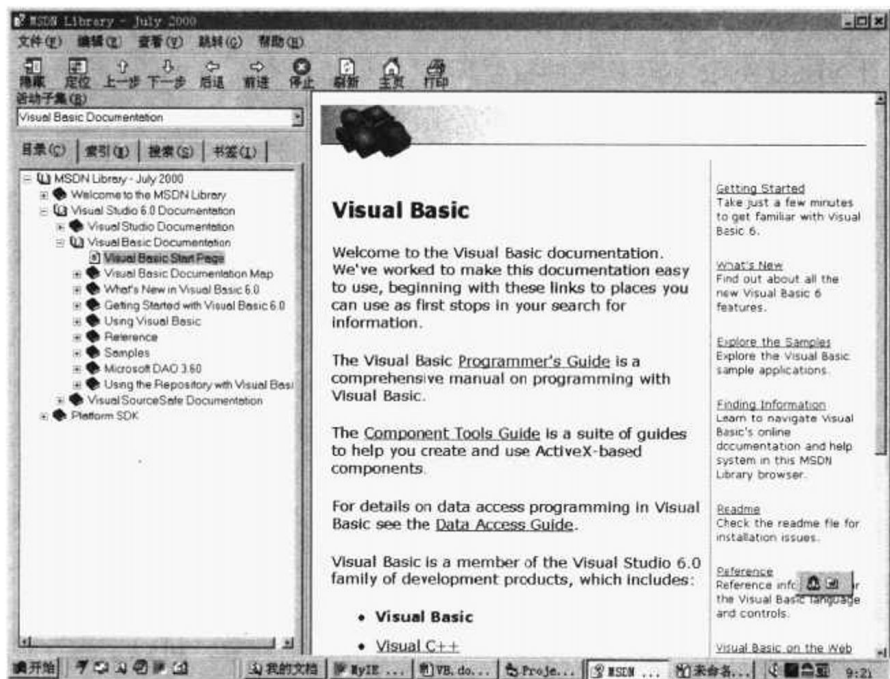


图 1.12 Visual Basic 帮助文件界面



1. MSDN Library 在线帮助

“MSDN Library”在线帮助窗口包含定位和主题两个窗格，在定位窗格中有“目录”、“索引”、“搜索”和“书签”4张选项卡，选择这4项中的主题后，即可在主题窗格中查看有关信息。选择了“搜索”选项卡后可以输入要查询的单词或短语，以便获取需要的帮助信息。输入查询的单词或短语时，可以运用一些逻辑运算符来实现连接，输入“窗体 And 按钮”，单击定位窗格中的“列出标题”按钮，意为列出同时包含“窗体”和“按钮”两个词的主题标题。若输入“窗体 Or 按钮”，单击定位窗格中的“列出标题”按钮，意为列出包含“窗体”或者包含“按钮”两个词的主题标题。

2. 上下文相关帮助

上下文相关帮助意味着无需打开“帮助”菜单，随时就可获取必要帮助。例如，可以在 Visual Basic 的代码窗口中，将鼠标插入点放在关键字“Private”上并按 F1 键，即可以调出关于“Private”的特定帮助信息。

3. 运行“帮助”中的代码

用户自己编写程序时，可以参考一些帮助中自带的实例，通过这些实例更好地掌握程序语句的运行以及 Visual Basic 中各对象属性和方法的使用。通过这些实例的注释以及运行结果，还可以更好地理解 Visual Basic 应用程序的编写方式。

1.2 Visual Basic 中对象的概念

Visual Basic 是一种面向对象的程序设计语言，它将代码和数据集成到一个独立的对象中，当运用这个对象来完成某个任务时，并不需要知道这个对象是如何工作的，只需编写一段代码来简单地给其发出一个动作即可。正确理解和掌握 Visual Basic 中对象的概念，是学习、设计 Visual Basic 应用程序的重要环节。

1.2.1 类和对象

类是面向对象可视化编程中最基本的概念之一，它是具有共同抽象的对象的集合。类定义了一个抽象模型，而程序设计时却是对实际对象的操作。类实例化后就成为对象。对象是运行时的基本实体，它既包括数据(属性)，也包括作用于对象的操作(方法)和对象的(事件)响应。在通常的面向对象程序设计中，对象是由程序员自己设计的，而在 Visual Basic 中，对象可以由系统设计好，直接供用户使用。在 Visual Basic 中也可由程序员自己设计对象(类)。Visual Basic 系统设计好的对象有窗体、各种控件、打印机及剪贴板等。用户大量使用的是窗体和控件。严格地讲，工具箱的各种控件并不是对象，而是代表了各个不同的类。通过将类实例化，可以得到真正的对象。当在窗体上画一个控件时，就将类转换为对象，即创建了一个控件对象，也简称为控件。

1. 属性、事件和方法

Visual Basic 的控件是具有自己的属性、事件和方法的对象。可以把属性看作一个对象的性质，把事件看作对象的响应，把方法看作对象的动作，它构成了对象的三要素。



1) 属性(Property)

属性可以看作对象的静态特性。程序运行时,通过对这些属性的改变,从而使对象的状态发生变化。对象中的数据就保存在属性中,所有 Visual Basic 程序中的对象都有许多属性,它们是用来描述和反映对象特征的参数。例如,控件名称(Name)、标题(Caption)、颜色(Color)、字体(FontName)及是否可见(Visible)等属性决定了对象展现给用户的界面具有什么样的外观及功能。不同的对象具有各自不同的属性,用户可通过联机帮助查阅。

对象属性的设置可以通过两种方法来做:在设计阶段利用属性框直接设置对象的属性;在程序代码中通过赋值实现,其格式为

对象.属性=属性值

例如,给一个对象名为“cmdOk”的命令按钮的 Caption 属性赋值为字符串“确定”,其在程序代码中的书写形式为

```
cmdOk.Caption="确定"
```

2) 事件(Event)及事件过程(EventProcedure)

我们把生成的对象所要完成的任务,即对象响应的动作称为事件。每个对象都有一系列预先定义好的对象事件,例如,鼠标单击(Click)、双击(DblClick)、改变(Change)、获取焦点(GetFocus)、键盘按下(KeyPress)等等。

对象与对象之间、对象与系统之间及对象与程序之间的通信都是通过事件来进行的。例如,窗体上有一个名为“cmdOk”的命令按钮对象,当鼠标指针被移动时,系统将跟踪鼠标指针的位置。当鼠标在该对象上单击一下时,系统跟踪到指针所指的对象上,并给该对象发送一个 Click 事件,同时向程序传达一个 cmdOK_Click 事件。如果我们编写了一个处理 cmdOk_Click 事件的程序代码即事件过程,系统则执行这个代码所描述的过程。执行结束以后,控制权又交还给系统,并等待下一个事件。

事件过程是指附在该对象上的程序代码,是事件的处理程序。事件过程的形式如下:

```
Sub 对象名_事件过程名[(参数列表)]  
... (事件过程代码)  
End Sub
```

例如,单击 cmdOk 命令按钮,使命令按钮的字体大小改为 20 磅,则对应的事件过程如下:

```
Sub cmdOk_Click()  
cmdOk.FontSize = 20 '设置命令按钮的字体大小为 20 磅  
End Sub
```

在 Visual Basic 中,程序设计者不必考虑对象名和事件名之间的关系,因为当建立好对象后,Visual Basic 会自动定义该对象对应的事件,用户只要直接选用即可。要对事件过程进行编程,必须进入代码窗口。



注意:

对每个对象,Visual Basic 系统事先已定义好一系列事件集,但要判定它们是否响应具体事件以及如何响应具体事件,则是编程者的责任了。例如,当用户单击某个对象,Click 事件发生时,MouseDown、MouseUp 事件也发生了。如要让系统响应 Click 事件,就应把代码写入该事件过程中,对其它事件不编程,自然也就不响应了。



3) 方法(Method)

面向对象的程序设计语言为程序设计人员提供了一种特殊的过程和函数, 这种特殊的过程和函数称为方法。在 Visual Basic 中已将一些通用的过程和函数编写好并封装起来, 作为方法供用户直接调用, 这给用户的编程带来了很大的方便。因为方法是面向对象的, 所以在调用时一定要指明对象。对象方法的调用格式为

[对象].方法 [参数名表]

若省略了对象, 表示在当前对象, 一般指窗体。例如:

Form1.Print "欢迎您使用 Visual Basic 6.0"

此语句表示使用 Print 方法在“Form1”窗体中显示字符串“欢迎您使用 Visual Basic 6.0”。

2. Visual Basic 中对象的建立和命名

1) 对象的建立

在窗体上建立对象的步骤如下:

(1) 将鼠标定位在工具箱内要制作控件对象对应的图标上, 单击左键进行选择。

(2) 将鼠标移到窗体上所需的位置处, 按住鼠标左键将窗体拖曳到所需的大小后释放鼠标。



注意: 对象的建立也可直接在工具箱双击所需的图标, 则立即在窗体出现一个默认大小的对象框。

当前建立的对象为活动的对象, 有 8 个方向的控制点。单击任一对象可选中该对象, 使该对象为活动的。

对象框的大小可以调整, 通过属性设置或用鼠标拖曳控制点来实现对对象的编辑。

选中(单击)要删除的对象, 然后按“Del”键即可删除对象。

复制对象时, 选中要复制的对象, 选择“编辑”菜单下的“复制”命令, 再选择该菜单下的“粘贴”命令。也可以通过工具栏按钮进行复制操作。

2) 对象的命名

每个对象都有自己的名字, 有了名字才能在程序代码中引用该对象。用户在“属性”窗口通过设置(名称)来给对象命名。名字必须以字母或汉字开头, 由字母、汉字、数字串组成, 长度为 40 个字符, 其中可以出现下划线(但最好不用, 以免与代码中的续行符相混)。每个窗体、控件对象都有一个 Visual Basic 提供的诸如 Form1、Form2、Text1 之类的默认名。为了便于程序的阅读和调试, 建议采用智能化的命名规则: 每个对象的名字由三个小写字母组成的前缀(指明对象的类型)和表示该对象作用的缩写字母组成。详见表 1.2。

表 1.2 Visual Basic 对象名字前缀举例

对象的类型	意 义	前缀	对象名字举例
CheckBox	检查框	chk	ChkKeyboard
ComboBox	下拉式列表框	cbo	CboStudent
CommandButton	命令按钮	cmd	CmdExit
CommonDialog	通用对话框	dig	dig 打开
Data	数据	dat	DatTelevision