



21 世纪高等院校
计算机系列课程教材

Visual Basic 程序设计

主 编 余 华 何玉香
主 审 靳润昭

 北京理工大学出版社
BEIJING INSTITUTE OF TECHNOLOGY PRESS

21 世纪高等院校计算机系列课程教材

Visual Basic 程序设计

主 编	余 华	何玉香
副主编	刘合兵	张 慧
编 者	赵 平	任艳娜
	张 浩	赵光煜
主 审	靳润昭	

 北京理工大学出版社

BEIJING INSTITUTE OF TECHNOLOGY PRESS

内 容 简 介

Visual Basic 简单易学,功能强大,深受广大计算机专业人员和非专业人员的欢迎。本书作者在多年教学经验基础上,依据学生的认知规律,以 Visual Basic 6.0 中文版为语言背景,以程序结构为主线精心组织本书内容,深入浅出地介绍了 Visual Basic 中文版的开发环境、对象和事件驱动的概念、运算符和表达式、数据输入输出、常用标准控件、基本控制结构、数组、过程调用、键盘和鼠标事件过程、菜单程序设计、对话框程序设计、多窗体程序设计、多文档界面(MDI)以及文件处理等。全书在编排上注意了由简及繁,由浅入深和循序渐进,力求通俗易懂,简洁实用,使具有 Windows 初步知识的读者都可以通过本书掌握 Visual Basic 程序设计的基本内容。

本教材知识结构及深度合理,非常适合大专院校本科、专科、成人继续教育及自学人员使用。本教材还配套出版了《Visual Basic 程序设计习题与上机指导》,以方便教师教学及学生自学。

版权专有 侵权必究

图书在版编目(CIP)数据

Visual Basic 程序设计/余华,何玉香主编. —北京:北京理工大学出版社,2005.2

(21 世纪高等院校计算机系列课程教材)

ISBN 7-5640-0391-X

I. V… II. ①余…②何… III. BASIC 语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2005)第 010487 号

出版发行 / 北京理工大学出版社

社 址 / 北京市海淀区中关村南大街 5 号

邮 编 / 100081

电 话 / (010)68914775(办公室) 68944990(发行部)

网 址 / <http://www.bitpress.com.cn>

电子邮箱 / chiefedit@bitpress.com.cn

经 销 / 全国各地新华书店

印 刷 / 北京圣瑞伦印刷厂

开 本 / 787 毫米×1092 毫米 1/16

印 张 / 15

字 数 / 355 千字

版 次 / 2005 年 2 月第 1 版 2005 年 2 月第 1 次印刷

印 数 / 1~6000 册

定 价 / 21.00 元

责任校对 / 陈玉梅

责任印制 / 刘京凤

图书出现印装质量问题,本社负责调换

前 言

Visual Basic 综合运用了 BASIC 语言的结构化特点和 GUI 可视设计工具,既具有 Windows 丰富的图形窗口工作环境,又继承了 BASIC 语言编程的简便性;既是一种可供专业设计者开发 Windows 应用程序的设计工具,又是一种可供非计算机专业的设计人员学习和掌握 Windows 编程的最简单易学的程序设计语言。Visual Basic 的出现,打破了 Windows 应用程序的开发由专业的 C 语言程序员一统天下的局面,即使非专业人员也能胜任,并可在较短时间内开发出质量高、界面好的应用程序。

Visual Basic 是在 Windows 环境下运行的编程语言,与 Windows 有着十分密切的关系。因此在学习本书之前,应适当地学习 Windows 的基础知识。

Visual Basic 内容十分丰富,作者认为,对于计算机语言初学者来说,应当把主要精力放在最基本、最常用的那些部分,待有一定基础后,再学习其他部分。本书介绍的是 Visual Basic 6.0 的基础知识,在体系结构和内容安排上注意了由简及繁、由浅入深、循序渐进、深入浅出以及理论与实践密切结合的特点。

Visual Basic 6.0 包括三种版本,即学习版、专业版和企业版。这些版本是在相同的基础上建立起来的,因此大多数应用程序可以在三个版本中通行。本书使用的是 Visual Basic 6.0 中文企业版,但其内容可用于学习版、专业版,本书中所有程序可以在学习版、专业版中运行。

全书共分 12 章,主要内容包括:Visual Basic 开发环境、对象和事件驱动的概念、常用内部函数和表达式、数据输入输出、常用标准控件、基本控制结构、数组、过程调用、键盘和鼠标事件过程、菜单程序设计、对话框程序设计、多窗体程序设计、多文档程序设计、多文档界面(MDI)、文件处理等。

为便于学习,本书每章均附有习题,并有与之配套的《Visual Basic 程序设计习题与上机指导》。

在我国,Visual Basic 正在受到越来越多的计算机专业和非专业人士的重视,希望本书的出版能给读者学习和使用 Visual Basic 带来更大的便利。感谢读者选择和使用本书,欢迎专家和广大读者对本书批评指正,提出修改意见,作者将不胜感激。

编 者

2005 年 1 月

目 录

第 1 章 概述	(1)
1.1 程序设计语言与程序设计	(1)
1.1.1 程序设计语言	(1)
1.1.2 程序设计	(2)
1.2 Visual Basic 简介	(4)
1.2.1 启动 VB	(4)
1.2.2 VB 集成开发环境浏览	(5)
1.3 标准控件简介	(10)
1.4 使用帮助	(12)
1.5 建立一个简单的应用程序	(14)
1.5.1 一个简单例子	(14)
1.5.2 程序的保存	(16)
1.6 界面、对象与事件驱动程序	(17)
练习题	(19)
第 2 章 数据类型与表达式	(20)
2.1 VB 的基本字符集和词汇集	(20)
2.1.1 字符集	(20)
2.1.2 词汇集	(20)
2.2 VB 的数据类型	(21)
2.3 常量与变量	(23)
2.3.1 常量和变量的命名规则	(23)
2.3.2 常量	(23)
2.3.3 变量	(24)
2.4 运算符和表达式	(26)
2.4.1 算术运算符与数值表达式	(26)
2.4.2 字符串运算符与字符串表达式	(27)
2.4.3 关系运算符与关系表达式	(28)
2.4.4 逻辑运算符与逻辑表达式	(28)
2.5 常用内部函数	(29)
2.5.1 语句的解释执行(立即窗口)	(29)
2.5.2 数学函数	(29)

2.5.3 字符串函数	(30)
2.5.4 随机函数	(31)
2.5.5 转换函数	(32)
练习题	(32)
第3章 VB 语言基础	(35)
3.1 结构化程序设计	(35)
3.1.1 算法及其表示	(35)
3.1.2 结构化程序设计	(37)
3.2 顺序结构	(38)
3.2.1 赋值语句	(38)
3.2.2 数据输入	(40)
3.2.3 注释、暂停与程序结束	(46)
3.2.4 Print 方法	(47)
3.3 分支结构	(50)
3.3.1 单行结构条件语句	(50)
3.3.2 块结构条件语句	(52)
3.3.3 情况语句	(55)
3.4 循环结构	(58)
3.4.1 循环概念	(58)
3.4.2 While...Wend 语句	(59)
3.4.3 Do...Loop 语句	(61)
3.4.4 For...Next 语句	(64)
3.4.5 嵌套循环与退出循环	(66)
练习题	(70)
第4章 数组	(74)
4.1 变量数组	(74)
4.1.1 固定大小的数组	(74)
4.1.2 动态数组(或称为可调数组)	(76)
4.2 数组的使用和基本操作	(77)
4.2.1 数组的使用	(77)
4.2.2 数组的基本操作	(78)
4.3 控件数组	(80)
4.4 数组实例应用	(83)
练习题	(87)
第5章 过程	(88)
5.1 过程的定义与调用	(88)
5.1.1 自定义过程的定义与调用	(88)
5.1.2 自定义函数的定义与调用	(89)
5.2 过程与变量的作用域	(90)

5.3 过程的参数	(91)
5.3.1 参数传递(地址,值)	(91)
5.3.2 数组参数	(93)
5.3.3 对象参数	(93)
5.4 事件过程	(93)
5.5 递 归	(94)
练习题	(95)
第 6 章 文件操作	(97)
6.1 文件	(97)
6.1.1 文件的结构	(98)
6.1.2 文件的种类和存取类型	(99)
6.2 顺序文件	(100)
6.2.1 顺序文件的打开与关闭	(100)
6.2.2 顺序文件的写操作	(102)
6.2.3 顺序文件的读操作	(103)
6.3 随机文件	(106)
6.4 二进制文件	(108)
6.5 与文件操作相关的语句和函数	(110)
练习题	(113)
第 7 章 窗体设计	(114)
7.1 窗体设计	(114)
7.1.1 窗体的主要属性	(114)
7.1.2 窗体的常用事件	(116)
7.2 多重窗体	(120)
7.2.1 多重窗体的管理	(120)
7.2.2 设置启动窗体	(120)
7.2.3 窗体处理	(121)
7.3 用向导生成窗体	(122)
7.4 多文档界面	(125)
练习题	(126)
第 8 章 标准控件的使用	(129)
8.1 命令按钮、图片框和图像框	(129)
8.1.1 命令按钮	(129)
8.1.2 图像框和图片框	(132)
8.2 标签与文本框	(134)
8.2.1 标签	(134)
8.2.2 文本框	(135)
8.3 单选按钮、复选框和框架	(138)
8.3.1 单选按钮(OptionButton)	(138)

8.3.2 复选框(CheckBox)	(139)
8.3.3 框架	(140)
8.4 列表框与组合框	(141)
8.4.1 列表框与组合框属性	(141)
8.4.2 常用方法	(143)
8.5 滚动条	(144)
8.5.1 滚动条的重要属性	(144)
8.5.2 举例	(145)
8.6 文件系统控件	(146)
8.6.1 文件系统控件的主要属性和事件	(146)
8.6.2 文件系统控件的作用	(147)
8.7 计时器控件	(149)
练习题	(149)
第9章 菜单与工具栏设计	(154)
9.1 菜单设计	(154)
9.1.1 菜单编辑器	(154)
9.1.2 创建菜单的步骤	(157)
9.1.3 菜单控件数组	(161)
9.1.4 弹出式菜单	(164)
9.2 工具栏设计	(168)
9.2.1 使用ToolBar控件和ImageList控件	(170)
9.2.2 使用CoolBar控件	(176)
9.3 状态栏设计	(177)
9.3.1 建立状态栏	(177)
9.3.2 运行时改变状态栏	(178)
练习题	(178)
第10章 对话框设计	(179)
10.1 用窗体定制对话框	(179)
10.1.1 创建窗体对话框	(179)
10.1.2 显示窗体对话框	(180)
10.2 用公共对话框控件实现标准对话框	(181)
10.2.1 CommonDialog控件	(181)
10.2.2 文件对话框	(183)
10.2.3 颜色对话框	(187)
10.2.4 字体对话框	(189)
10.2.5 打印对话框	(192)
10.2.6 帮助对话框	(195)
练习题	(197)
第11章 图形操作	(198)

11.1 VB 的坐标系	(198)
11.1.1 默认坐标系	(198)
11.1.2 定制坐标系	(198)
11.2 图形控件	(200)
11.2.1 Line 控件(画线工具)	(200)
11.2.2 Shape 控件(形状)	(201)
11.3 图形方法	(202)
11.3.1 Pset 方法	(202)
11.3.2 Line 方法	(204)
11.3.3 Circle 方法	(204)
11.3.4 Cls 方法、Point 方法和 PaintPicture 方法	(207)
11.4 简单动画设计	(207)
11.4.1 简单动画设计	(207)
11.4.2 应用举例	(208)
练习题	(211)
第 12 章 数据库应用	(212)
12.1 数据库的基本概念	(212)
12.2 可视化数据管理器	(213)
12.3 数据控件和数据绑定控件	(214)
12.3.1 主要属性	(215)
12.3.2 数据控件的主要事件和方法	(216)
12.3.3 记录集的主要属性和方法	(217)
12.3.4 用代码操作记录	(223)
12.4 ADO 数据控件和数据库访问对象	(224)
12.4.1 ADO 数据控件	(224)
12.4.2 数据库访问对象 DAO	(226)
12.5 结构化查询语言(SQL)简介	(227)
12.5.1 SQL 概述	(227)
12.5.2 SQL 语法	(227)
12.5.3 使用数据管理器查询和使用 SELECT 查询	(228)
练习题	(229)
参考文献	(230)

第1章 概述

Basic 语言是 20 世纪 60 年代美国 Darktouth 学院两位学者创立的计算机程序设计语言，其全称是 Beginner's All-purpose Symbolic Instruction Code(初学者通用的符号指令代码)，它是一种非常简单的语言。80 年代，为适应结构化程序设计的需要，出现了第二代的 Basic 语言版本，增加了许多数据类型和程序控制结构。其中较有影响的有 True Basic，Microsoft 公司的 Quick Basic 和 Borland 公司的 Turbo Basic 等。

Windows 操作系统的出现，为用户提供了一个直观的、图形丰富的工作环境。图形用户界面(Graphical User Interface，缩写为 GUI)使应用程序更易于学习和使用。用户只要简单地用鼠标按钮单击“菜单”中的命令选项就可以执行指定的操作，而不必键入复杂的命令行。屏幕上的多窗口可以让用户同时运行多个程序，并可以通过对话框输入所需要的信息。确实，Windows 的环境相当出色。但对程序员来说，要实现这种界面，程序的编写工作量是十分巨大的。可视化程序设计语言正是在这种背景下应运而生的。

可视化程序设计语言除了提供常规的编程环境外，还提供一套可视化设计工具，便于程序员建立图形对象，巧妙地把 Windows 编程的复杂性“封装”起来。20 世纪 90 年代，Microsoft 公司推出的 Visual Basic 语言是以结构化 Basic 语言为基础，采用事件驱动运行机制的新一代可视化程序设计语言，与传统的程序设计语言相比，它具有明显的优势，而且随着版本的改进，功能越来越强大，不但可以作为多媒体软件制作工具、实现数据库管理，而且还具有网络功能等。

我们学习 Visual Basic(本书中简称为 VB)，不但要掌握一种实用的计算机软件设计工具，而且通过学习该课程，应该对计算机的工作原理和工作方式有初步了解，掌握程序设计语言的基本语法规则，掌握可视化程序设计的基本方法，为进一步学习和应用打下良好基础。

本教材是以 Visual Basic 6.0 为基础编写的，使用其他版本的读者可结合相关的使用手册。

1.1 程序设计语言与程序设计

1.1.1 程序设计语言

在人类社会生活中，“语言”是由语音、词汇和语法构成的系统，是人与人之间用来传递信息、交流思想的工具。而“程序设计语言”(也称算法语言)是由字、词和语法规则构成的指令系统，是人与计算机交流的工具。使用程序设计语言，我们只需要告诉计算机什么时候、

什么条件下干什么，计算机根据指令一条一条地执行，并把执行结果告诉我们。

从机器的执行角度来说，程序设计语言可以分成两大类：低级语言和高级语言。低级语言是面向机器的指令系统，不同型号的中央处理器(Central Processing Unit, 缩写为CPU)有不同的指令系统。低级语言又分为机器语言和汇编语言。机器语言指令由只包含“0”和“1”的二进制符号按一定规律组成，能够被机器直接执行。这些指令记忆难度相当大，为此，人们就采用一些特定的助记符代表机器指令，从而产生了汇编语言。用汇编语言编写的程序只需要经过简单的翻译就可以被机器执行。低级语言的特点是程序执行速度快，效率高，但要求程序员了解机器的结构，程序设计难度大，非专业人员难以涉足。高级语言是由人们易于接受的、接近人类语言的描述方式构成的指令系统。它不需面向机器，构成简单，往往只有100多条词汇、若干条规则，便于记忆，易于学习，程序设计速度快。人们说的“程序设计语言”通常是指高级语言。高级语言编写的程序需要经过较复杂的编译才能成为机器指令。在不同的应用场合，往往使用不同的语言设计应用程序。

Visual Basic 语言是一种通用的高级程序设计语言。

计算机应用的领域很广泛，为了适应不同的需要，不同的程序设计语言又各具特点。例如，有适于编写系统软件的，有便于进行科学计算的，有善于数据库管理的，有适合图形设计的，又有专用于人工智能的，也有一些语言同时具备多种功能。从应用角度，我们难以对程序设计语言作严格分类。而且，随着计算机科学的发展及应用领域的迅速扩展，各种语言版本都在不断地变化，功能在不断更新、增强。每个时期都有一批语言在流行，又有一批语言在消亡。我们应该掌握程序设计语言中本质性、规律性的东西。

为了满足人们各种特定的需要，计算机工作者开发了各种工具软件。例如文字处理、表格处理、图形处理、多媒体管理以及各种系统管理工具等软件。它们是一些专用程序集。用户在界面上与计算机交互操作，就是调用它们之中的各个子程序。

然而，世界是复杂的、千变万化的，人们要求计算机做的工作也千差万别。可以说，现代社会对计算机的要求是无止境的。计算机厂家不断生产功能越来越强大的计算机系统，软件开发者也不断设计满足不同需要的应用程序。但不管计算机结构如何，它要干什么工作，人们总是用某一种形式的“语言”与计算机交流。

从这个思路出发，计算机系统可以看作是一个由不同层次语言构成的系统。人们可以用不同的语言与计算机交流。如果我们用高级语言与计算机交流，则系统提供一个翻译工具，称为编译程序或解释程序，先把人们输入计算机的指令序列翻译成机器语言的指令序列，然后由计算机的逻辑电路分解成微指令，让各部件执行相应的动作。

1.1.2 程序设计

程序设计是根据特定的问题，使用某种程序设计语言，设计出计算机执行的指令序列。程序设计是一项创造性的工作，主要完成两方面工作：

1. 数据描述

数据描述是把被处理的信息描述成计算机可以接受的数据形式：如整数、实数、字符、数组等。

信息是现实世界在大脑中的抽象反映，是通过人的感官感知出来，并且经过人脑加工而

形成的反映现实世界中事物的概念。这些概念不仅被人们所认识、理解,而且作为知识进行推理、加工和传播,从而达到认识世界、改造世界的目的。这里所说的“事物”,要比“物质”有着更广泛的含义。它不仅可以指那些看得见、摸得着的具体事物,例如汽车、房子,也可以包括那些不可触及的抽象概念,例如产量、质量、成绩等等。因此,信息可以看成为对现实世界的抽象反映。现实世界到处充满了信息,我们每时每刻接触到各种各样的信息,而且频繁地传播、加工和利用这些信息。

信息可以用人工或自动化装置进行记录、解释和处理。使用计算机进行信息处理时,这些信息必须转换成可以被机器识别的“数据”,例如数字、文字、图形、声音等。不管什么数据,计算机都以二进制形式存储和加工处理。

可见,数据是信息的载体,信息依靠数据来表达。

2. 数据处理

数据处理是指对数据进行输入、输出、整理、计算、存储、维护等一系列的活动。数据处理的目的是为了提取有用的数据成分,获得有用的资料。

程序设计的工作成果称为源程序。源程序有两种执行方式:编译执行和解释执行。编译执行是通过系统的编译程序把源程序翻译成等价的机器语言程序,称为目标程序,由系统直接执行。这些程序都可以作为文件永久保存。一旦编译成功,目标程序就可以反复高速执行。解释执行是解释程序对源程序一边分析,一边执行,并不生成目标程序文件。这种方式执行效率较低,但便于调试程序。VB 系统提供两种程序执行方式。

一般而言,编写源程序过程难免存在一些错误。这些错误分成三类:

① 编译错误。编译源程序时发现的语法错误。例如,表达式: $(A+B*(C-D))$, 缺了右括号。

② 运行错误。执行目标程序时发现的错误。例如,执行标准函数 $\text{Sqr}(x)$, 求 x 的平方根时, x 的值为负数。

③ 逻辑错误。编译和运行时均不能发现的错误。例如,由于程序员的笔误,把 x^3 写成 x^5 , 对 x 求立方变成求 x 的 5 次方了。

一个程序经常要经过反复的调试、验证才能完善,投入使用。为此,我们编写的程序应力求具有以下特点:

① 正确性。这要求程序员熟悉所使用的程序设计语言,避免语法、语义上的错误,设计简单易行的算法达到预期目的。对复杂的问题,则应考虑使用有效的程序设计方法。

② 易读性。一个程序结构清晰易读,才能便于查错,便于修改。

③ 运行高效率。指程序运行时间较短,而占用的存储空间较小。

为达到以上目标,需要我们在不断的学习和实践中提高程序设计水平。“程序”是人的智力产品,它无色、无形、无味,它正确与否,是好是坏,只有到机器上运行才能知道。所以,对比有形的产品,早期人们把程序称为软件。从理论上说,程序是永远不会损坏的。实际上,程序在整个生存周期都会根据需要进行修改、维护,都会产生错误。所有的硬件产品都允许有误差,但程序错误是不允许的,它有时甚至会产生悲剧性的后果。程序的生产和维护比硬件产品复杂得多。所以,计算机科学界期望有一套工程化的方法进行程序的开发维护。为了体现这种工程思想,程序就要伴随着一套开发、维护、使用的文档。程序加上这些相关文档称为软件。

1.2 Visual Basic 简介

1.2.1 启动 VB

VB 的安装程序自动在 Windows 的【开始】菜单中建立【Microsoft Visual Basic 6.0 中文版】程序组。它的启动遵从基本的 Windows 系统的操作：可从【开始】中执行【程序】→【Microsoft Visual Basic 6.0 中文版】程序组→【Microsoft Visual Basic 6.0 中文版】；可使用命令行，执行【开始】→【运行】命令，在【运行】对话框中键入“C:\Program Files\Microsoft Visual

Studio\VB98\Vb6.exe”（按默认盘符和路径安装），单击【确定】按钮；也可利用 Windows 建立快捷方式的功能，在桌面上创建 VB 6.0 的快捷方式，双击快捷方式图标启动。

启动 VB 6.0 后，首先弹出【新建工程】对话框，如图 1-1 所示。窗口中列出的是 VB 6.0 能够建立的应用程序类型，初学者可选择【标准 EXE】。

【新建工程】对话框中有三个选项卡：

- 【新建】：建立新工程。
- 【现存】：打开已有的工程。
- 【最新】：列出最近打开过的工程。

在【新建】选项卡中选【标准 EXE】，单击【打开】按钮，则进入 VB 6.0 的集成开发环境，如图 1-2 所示。

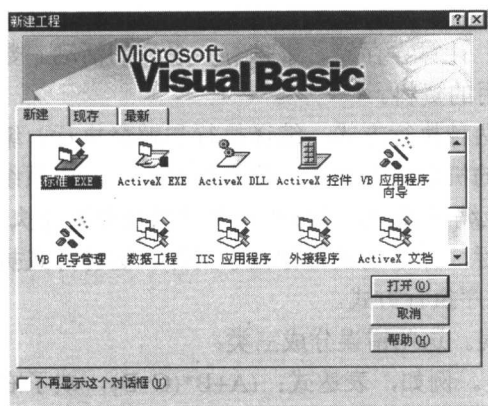


图 1-1 【新建工程】对话框

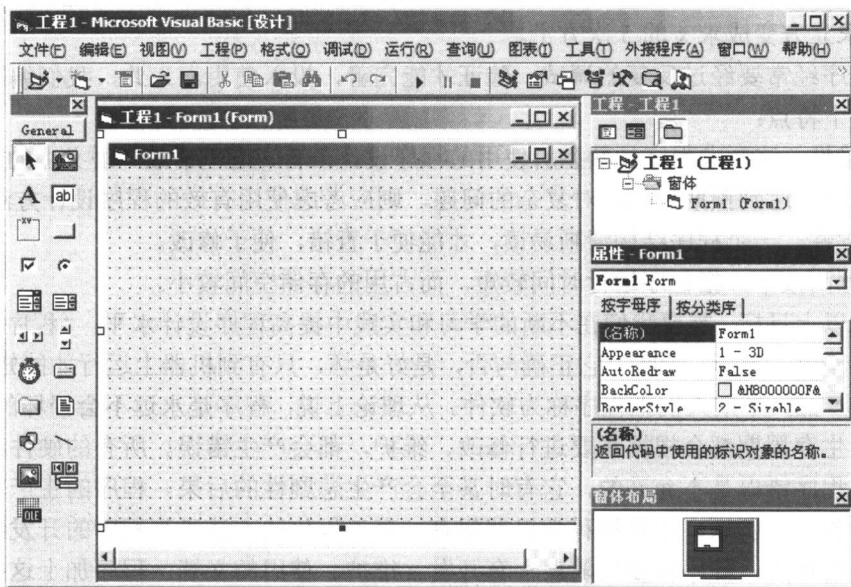


图 1-2 VB 6.0 的集成开发环境

1.2.2 VB 集成开发环境浏览

VB 6.0 的开发界面如图 1-2 所示, 整个界面由若干窗口组成, 包括 VB 6.0 的主窗口、工具箱窗口、窗体窗口、工程资源管理器窗口、属性窗口、窗体布局窗口以及代码窗口、数据视图窗口、调色板窗口等。

1. 主窗口

VB 6.0 开发界面的主窗口与 Windows 环境下其他软件的主窗口类似, 由标题栏、菜单栏、工具栏组成。其工具按钮具有提示功能, 单击右键可显示快捷菜单, 用户可自定义菜单, 在对象浏览窗口中可查看对象及相关属性。

(1) 标题栏

标题栏是 Windows 应用程序用户界面的一个共同特征, 标题栏最左端是窗口控制菜单, 最右端是最大化按钮、最小化按钮和关闭按钮, 中间是标题。启动 VB 6.0 后, 标题栏中的标题为“工程 1-Microsoft Visual Basic[设计]”, 其中“工程 1”是当前应用程序的名称, “[设计]”说明此时 VB 6.0 处于应用程序的设计模式。当进入其他模式时, 方括号中的文字会做相应的变化。

VB 6.0 有三种工作模式:

- ① 设计模式: 用 VB 6.0 开发应用程序, 此时可进行用户界面的设计和代码的编写。
- ② 运行模式: 在 VB 6.0 环境下运行应用程序, 此时不能编辑代码和界面。
- ③ 中断模式: 应用程序运行暂时中断, 此时可编写代码, 但不能编辑界面。单击【继续】按钮或按 F5 键可使程序继续运行, 单击【结束】按钮终止程序运行。

(2) 菜单栏

VB 6.0 的菜单栏包括 13 个菜单标题, 每个菜单标题都有一个下拉菜单, 这些下拉菜单包括了程序开发过程中所需要的命令。各菜单从左至右分别为:

【文件】: 用于创建、打开、保存、显示、删除、添加工程以及生成可执行文件的命令。

【编辑】: 用于程序源代码的编辑。其中的命令选项包括了各种编辑软件所具有的通用功能, 并且完全符合 Windows 的操作风格。

【视图】: 用于集成开发环境下程序源代码、控件的查看, 界面中各个窗口的显示或隐藏。

【工程】: 用于管理应用程序的工程项目, 包括对窗体、控件以及模块文件、ActiveX 控件等对象的创建、添加、删除等。

【格式】: 用于设置 VB 6.0 开发环境的格式, 如排列控件、设定栅格等。

【调试】: 用于程序调试, 如增加、编辑、跟踪变量、设置断点等。

【运行】: 用于程序启动、设置中断和停止等程序运行的命令。

【查询】: VB 6.0 新增的菜单标题, 在设计数据库应用程序时用于设计 SQL 查询, 包括运行、清除结果、校验 SQL 语法、排序等。

【图表】: VB 6.0 新增的菜单标题, 在设计数据库应用程序时对关系数据库的表进行操作的命令。

【工具】: 用于集成开发环境下工具的扩展。如增加子程序、设计菜单、设置环境参数等。

【外接程序】: 用于为工程增加或删除外接程序。

【窗口】：用于屏幕窗口的层叠、平铺等布局以及列出所有打开文档窗口。

【帮助】：帮助用户系统学习掌握 VB 6.0 的使用方法及程序设计方法。包括启动 VB 6.0 联机手册，调用联机帮助索引等。

(3) 工具栏

工具栏由多个图标按钮组成，在开发环境下提供对常用命令的快速访问。包括【标准】工具栏、【编辑】工具栏、【窗体】工具栏和【调试】工具栏。默认情况下只显示【标准】工具栏，要显示或隐藏某个工具栏，可选择【视图】菜单下的【工具栏】子菜单，从中执行相应的菜单命令，或右击【标准】工具栏，从弹出的快捷菜单中选择相应的菜单命令。

【标准】工具栏各按钮的意义说明如图 1-3 所示。

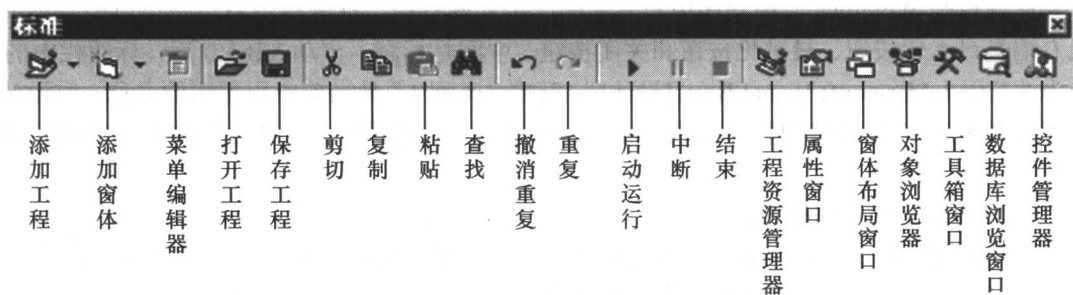


图 1-3 【标准】工具栏

2. 窗体窗口

窗体窗口如图 1-2 中间部分所示，窗体是建立 VB 应用程序的主要部分，它是应用程序界面的窗口，在窗体中可以放置诸如命令按钮、文本框、图像框、滚动条等其他控件，用户通过和这些控件交互来实现程序的功能，这些控件的外观也就是应用程序的界面。

启动 VB 6.0 时自动创建一个空窗体，设计过程中可以再添加新的窗体。每个窗体有一个惟一的名称，建立窗体时缺省的名称是 Form1、Form2 等。

窗体的顶端是与一般 Windows 应用程序一致的标题栏，默认标题是 Form1、Form2 等。标题是窗体的重要属性之一，可根据需要修改。窗体的大小也是其重要的属性，可根据需要设置和调整。

整个窗体上是用点组成的栅格，其作用是定位放入窗体中的控件，使其与栅格对齐。是否显示栅格、栅格的距离以及是否将控件与栅格对齐可自行设置。设置方法：执行【工具】→【选项】菜单命令，打开【选项】对话框，选择【通用】选项卡，如图 1-4 所示。从对话框中选中或取消要设置的选项即可。

除一般窗体外，还有 MDI (Multiple Document Interface) 多文档窗体，使用 MDI 窗体可建立多文档应用程序。

3. 工具箱窗口

工具箱提供一组工具，VB 6.0 的工具箱，由 20 个控件图标构成，利用这些工具即可在窗体上设计各种控件。

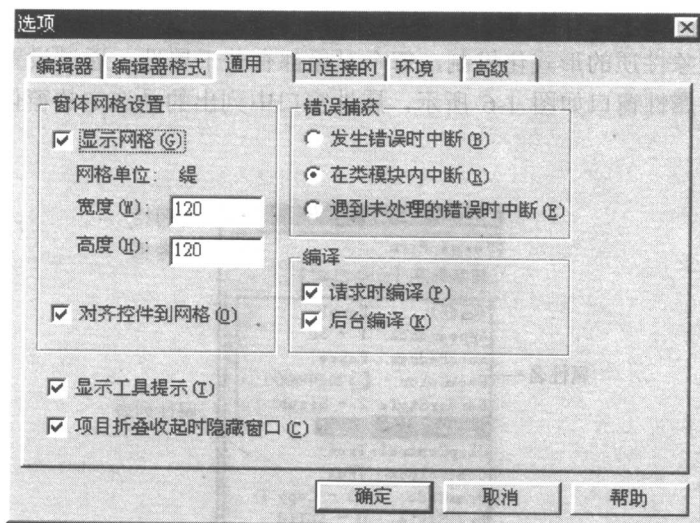


图 1-4 【选项】对话框

在设计状态下，工具箱总是显示的，若要将其隐藏，单击工具箱标题栏右端的【关闭】按钮，或右击工具箱，从弹出的快捷菜单中执行【隐藏】菜单命令。要显示工具箱，可执行【视图】→【工具箱】菜单命令，或单击【标准】工具栏上的【工具箱】按钮。在运行状态下，工具箱自动隐藏。

工具箱中的控件称为【标准控件】，除【标准控件】外，也可以把其他控件添加到工具箱中，供开发应用程序使用，或从工具箱中删除。方法如下：执行【工程】→【部件】菜单命令，打开【部件】对话框，选择【控件】选项卡，如图 1-5 所示。从列表中选定要添加或删除的控件名称前面的选择框即可。

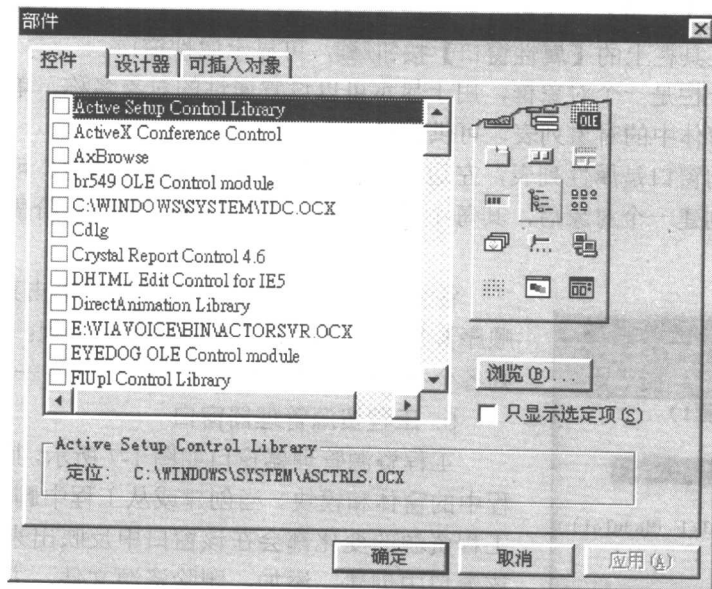


图 1-5 【部件】对话框

4. 属性窗口

属性是描述对象性质的形式化机制，每个对象都有若干属性，通过设置属性值来控制对象的外观和行为。属性窗口如图 1-6 所示，属性窗口中列出的是选定的窗体或控件的全部属性及其设置值。

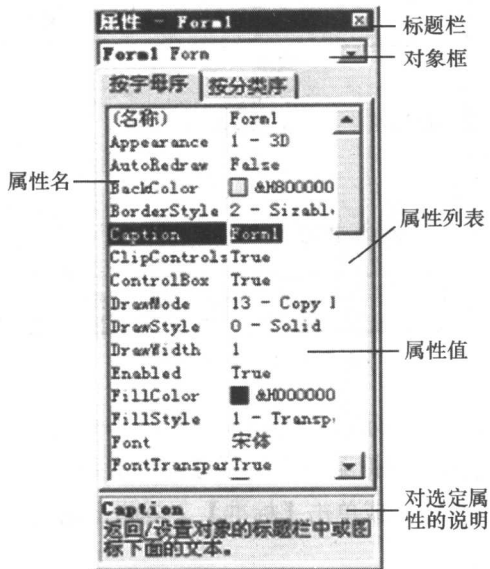


图 1-6 【属性】窗口

属性窗口最顶端是标题栏，标题“属性-Form1”中 Form1 表示当前列出的属性是关于 Form1 窗体或其中某个对象的。用鼠标拖拽它可移动位置，也可用鼠标拖动其边界改变大小。单击标题栏右端的【关闭】按钮可关闭属性窗口，执行【视图】→【属性窗口】菜单命令，或单击【标准】工具栏上的【属性窗口】按钮, 可显示属性窗口。

标题栏下面一栏是一个对象框，用于显示可以设置属性的对象名称。单击右边的下拉箭头，则列出当前窗体中的对象列表，可供选择。

对象框下面的窗口是属性列表，左边栏目列出所选对象的全部属性，可在右边栏目编辑和观察属性值。创建一个对象后，其每个属性均有一个默认值，在设计阶段可通过属性窗口直接修改。

各个属性的排列顺序可以改变：单击菜单栏中【按字母顺序】，则各个属性按字母顺序加以排列；单击【按分类序】，则各个相关的属性都分门别类地放置在一起。

5. 工程资源管理器窗口

工程资源管理器窗口如图 1-7 所示，其中列出了当前工程中的窗体和模块。当创建或从工程中删除可编辑文件时，工程资源的变化都会在该窗口中反映出来。也可以直接在该窗口中创建、添加、删除资源文件，方法是：在该窗口中单击鼠标右键，从弹出的菜单中选择【添加】子菜单，

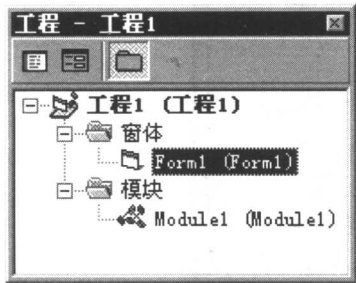


图 1-7 【工程】资源管理器窗口