

高等院校计算机基础教育规划教材



购书可获增值回报

电子教案

刘炳文 编著

Visual Basic 程序设计



机械工业出版社
CHINA MACHINE PRESS

高等院校计算机基础教育规划教材

Visual Basic 程序设计

刘炳文 编著



机械工业出版社

本书通过大量实例,深入浅出地介绍了 Visual Basic 程序开发环境、对象和事件驱动的概念、数据类型和表达式、常用内部函数、数据输入输出、常用控件、控制结构和数组、通用过程、键盘和鼠标事件过程、菜单与对话框程序设计、文件处理、多窗体程序设计、图形程序设计以及 Windows 应用程序接口(API)、数据库程序设计、多媒体程序设计、Internet 程序设计。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、简捷实用。本书每章均附有习题,便于教学和自学。

本书可作为大专院校、高职、高专和计算机培训班的教材,也可以供自学使用。

图书在版编目(CIP)数据

Visual Basic 程序设计/刘炳文编著. —北京:机械工业出版社, 2004.7
高等院校计算机基础教育规划教材
ISBN 7-111-14898-3

I. V... II. 刘... 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(2004)第068613号

机械工业出版社(北京市百万庄大街22号 邮政编码100037)

策划:胡毓坚

责任编辑:蔡岩

责任印制:李妍

北京蓝海印刷有限公司印刷·新华书店北京发行所发行

2004年8月第1版·第1次印刷

787mm×1092mm $\frac{1}{16}$ · 18印张·443千字

0001—5000册

定价:25.00元

凡购本图书,如有缺页、倒页、脱页,由本社发行部调换

本社购书热线电话:(010) 68993821、88379646

封面无防伪标均为盗版

出版说明

计算机基础教育在经济建设与社会发展中,发挥着非常重要的作用。我国高等院校十分重视计算机基础教育,在指导思想、教学设置及安排、优化知识结构等方面进行了大量的工作,其目的是为国家培养合格的高素质人才。为了满足教育的需求,机械工业出版社组织编写了这套“高等院校计算机基础教育规划教材”。

在组织编写过程中,我社聘请了各高等院校承担计算机基础教育工作的主讲教授和骨干教师,对教学经验进行了总结和提炼,并对前瞻性课题和内容进行了研讨。针对课程特点,总结出课程中出现的知识点、重点和难点,并融入教材的编写中。

本套系列教材与课程紧密结合,定位准确,注重理论教学和实践教学相结合,逻辑性强,层次分明,叙述准确而精炼,图文并茂,习题丰富,非常适合各类高等院校、高等职业技术学校及相关院校的计算机基础教育,也适合作为用各类培训班的教材或自学参考书。

前 言

Visual Basic 是一个功能强大的应用程序开发工具,在世界范围内得到了广泛的应用。它之所以受到人们的青睐的主要原因有:Windows 是当今主流操作系统,而 Visual Basic 是开发 Windows 应用程序最简单易学的语言之一;Visual Basic 引入了面向对象和事件驱动的程序设计思想,大大缩短了应用软件的开发周期;Microsoft 公司十分重视 Visual Basic 的功能扩充,不断地把最新技术运用到 Visual Basic 中,使之可以完成几乎所有 Windows 应用程序的开发任务;已有越来越多的优秀软件支持用 Visual Basic 进行二次开发。

可视化编程是程序设计的一种新概念、新方法。Microsoft 公司开发的可视化程序设计系列语言在功能和编程方法上是一致的,只要学会一种可视化程序设计语言,就可以很容易地学习另一种可视化程序设计语言。Visual Basic 不但功能强大,而且容易掌握,因此十分适合作为第一门可视化程序设计语言来学习。掌握了 Visual Basic 程序设计技术,就可以学习更深入的内容或者其他难度更大的语言。正因为如此,目前国内很多高校的计算机专业和非计算机专业相继开设了 Visual Basic 课程,作为必修课或选修课。同时,在全国及各地区组织的计算机统一考试中,几乎无一例外地都包括 Visual Basic 的考试。

本书以 Visual Basic 6.0 中文版为语言背景,以非计算机专业学生为教学对象,是在深入理解 Visual Basic 程序设计方法和长期应用实践的基础上编写而成的。它充分考虑到非计算机专业教学和初学者的需要,把通俗易懂作为重要目标。在体系结构上,根据初学者的认知规律作了细致安排,以使读者能顺利掌握所学知识;在内容上则注意了由简及繁、由浅入深、循序渐进、深入浅出以及理论与实践的密切结合。

全书共 15 章,分为两部分。第 1 部分(1~11 章)介绍了 Visual Basic 程序开发环境、对象和事件驱动的概念、数据类型和表达式、常用内部函数、数据输入输出、常用控件、控制结构和数组、通用过程、键盘和鼠标事件过程、菜单与对话框程序设计、文件处理、多窗体程序设计、图形程序设计等。第 2 部分(12~15 章)介绍了 Windows 应用程序接口(API)、数据库程序设计、多媒体程序设计以及 Internet 程序设计。

本书第 1 部分介绍的是 Visual Basic 程序设计的基础知识,是学习 Visual Basic 程序设计必须掌握的内容;而第 2 部分是 Visual Basic 较为深入的内容。在学习并掌握了第 1 部分内容的基础上,可以选学或自学第 2 部分。在实际的教学中,如果时间较紧,可以只安排第 1 部分的授课和上机实验,第 2 部分可不占用教学时间。

为了便于教学和自学,本书每章均附有习题,以方便读者进一步巩固所学知识。此外,还提供了与本书配套的《Visual Basic 程序设计题解与上机指导》以及供教师使用的电子教案。其中“题解与上机指导”提供了本书中全部习题的参考答案以及上机实验安排,而电子教案可作为教师讲授本书的课件。

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,在教学和软件开发中久盛不衰,具有举足轻重的作用。随着进一步的推广应用,学习和使用 Visual Basic 的人会越来越多。但愿本书能满足广大读者学习 Visual Basic 的强烈愿望,对 Visual Basic 的应用推广起到一些作用。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,笔者将不胜感激。

作 者

目 录

出版说明

前言

第 1 章 Visual Basic 集成开发环境与预定义对象	1
1.1 程序设计语言	1
1.1.1 面向过程与面向对象的语言	1
1.1.2 可视化与事件驱动型语言	2
1.2 启动与退出 Visual Basic	3
1.3 Visual Basic 编程环境	3
1.3.1 主窗口	4
1.3.2 窗体设计器与工程资源管理器窗口	6
1.3.3 属性窗口与工具箱窗口	7
1.4 对象	8
1.4.1 Visual Basic 中的对象	8
1.4.2 对象属性设置	9
1.5 窗体	10
1.5.1 窗体的结构与属性	10
1.5.2 窗体事件	13
1.6 控件	14
1.6.1 控件的画法	14
1.6.2 控件的基本操作	15
1.7 习题	16
第 2 章 简单程序设计	18
2.1 语句	18
2.2 命令按钮控件	19
2.3 编写简单的 Visual Basic 应用程序	20
2.3.1 程序设计	21
2.3.2 代码编辑器	25
2.4 程序的保存、装入和运行	26
2.4.1 保存程序	26
2.4.2 程序的装入	27
2.4.3 程序的运行	28
2.5 习题	29
第 3 章 数据类型、运算符与表达式	30
3.1 数据类型	30
3.1.1 基本数据类型	30
3.1.2 用户定义的数据类型	31
3.2 常量和变量	32

3.2.1 常量	32
3.2.2 变量	34
3.3 内部函数	36
3.3.1 常用内部函数	36
3.3.2 Shell 函数	38
3.4 运算符与表达式	39
3.4.1 算术运算符	39
3.4.2 关系运算符与逻辑运算符	40
3.4.3 表达式的执行顺序	41
3.5 习题	42
第4章 数据输入输出	44
4.1 数据输出——Print 方法	44
4.1.1 Print 方法	44
4.1.2 与 Print 方法有关的函数和方法	45
4.1.3 格式输出	46
4.2 数据输入——InputBox 函数	48
4.3 MsgBox 函数和 MsgBox 语句	49
4.4 字形	52
4.5 习题	55
第5章 常用控件	57
5.1 文本控件	57
5.1.1 标签	57
5.1.2 文本框	57
5.2 图形控件	59
5.2.1 图片框和图像框	59
5.2.2 图形文件的装入	60
5.2.3 直线和形状	61
5.3 复选框和单选按钮	62
5.3.1 复选框和单选按钮的属性和事件	62
5.3.2 应用举例	63
5.4 列表框和组合框	63
5.4.1 列表框	64
5.4.2 组合框	66
5.5 滚动条	68
5.6 计时器	69
5.7 框架	71
5.8 ActiveX 控件	73
5.8.1 选项卡控件	73
5.8.2 滑动器控件(Slider)	74
5.8.3 进度条(ProgressBar)	75
5.9 焦点与 Tab 顺序	77
5.9.1 设置焦点	77

5.9.2 Tab 顺序	78
5.10 习题	79
第6章 控制结构与数组	81
6.1 选择控制结构	81
6.1.1 条件语句	81
6.1.2 If 函数	82
6.2 多分支控制结构	83
6.3 循环控制结构	85
6.3.1 For 循环控制结构	85
6.3.2 当循环控制结构	86
6.3.3 Do 循环控制结构	87
6.4 数组	88
6.4.1 数组的定义	88
6.4.2 动态数组	89
6.5 数组的基本操作	90
6.5.1 数组元素的输入、输出和复制	91
6.5.2 For Each...Next 语句	92
6.6 控件数组	93
6.7 程序举例	95
6.8 习题	98
第7章 过程	102
7.1 Sub 过程	102
7.1.1 建立 Sub 过程	102
7.1.2 调用 Sub 过程	103
7.2 Function 过程	104
7.2.1 建立 Function 过程	104
7.2.2 调用 Function 过程	105
7.3 参数传送	106
7.3.1 形式参数与实际参数	106
7.3.2 引用与传值	106
7.3.3 数组参数的传送	107
7.4 对象参数	111
7.4.1 窗体参数	111
7.4.2 控件参数	111
7.5 递归	113
7.6 键盘事件过程	115
7.6.1 KeyPress 事件	115
7.6.2 KeyDown 和 KeyUp 事件	116
7.7 鼠标事件过程	117
7.8 拖放	120
7.8.1 与拖放有关的属性、事件和方法	120
7.8.2 自动拖放	122

7.8.3 手动拖放	124
7.9 习题	125
第8章 菜单与对话框程序设计	128
8.1 下拉式菜单	128
8.1.1 菜单编辑器	128
8.1.2 建立下拉式菜单	130
8.1.3 菜单项控制	134
8.2 弹出式菜单	135
8.3 对话框程序设计	137
8.3.1 文件对话框	138
8.3.2 其他对话框	140
8.4 习题	142
第9章 数据文件	145
9.1 文件概述	145
9.2 顺序文件	146
9.2.1 顺序文件的写操作	146
9.2.2 顺序文件的读操作	148
9.3 随机文件	151
9.3.1 随机文件的读写操作	151
9.3.2 程序举例	152
9.4 文件系统控件	156
9.4.1 驱动器列表框和目录列表框	156
9.4.2 文件列表框	158
9.4.3 程序举例	159
9.5 习题	162
第10章 多窗体程序与环境应用	165
10.1 建立多窗体应用程序	165
10.1.1 与多窗体程序设计有关的语句和方法	165
10.1.2 多窗体程序举例	166
10.1.3 多窗体程序的执行与保存	170
10.2 Visual Basic 工程结构	171
10.2.1 标准模块	172
10.2.2 窗体模块	172
10.2.3 Sub Main 过程	173
10.3 闲置循环与 DoEvents 语句	174
10.4 变量和过程的作用域	176
10.4.1 变量的作用域和生存期	176
10.4.2 过程的作用域	178
10.5 系统对象	178
10.5.1 App 对象	178
10.5.2 Screen 对象	179
10.5.3 Printer 对象	180

10.6 习题	181
第 11 章 图形程序设计	183
11.1 对象坐标系统	183
11.1.1 默认规格和标准规格	183
11.1.2 自定义规格	184
11.2 常用绘图方法	185
11.2.1 点与直线	185
11.2.2 矩形	189
11.2.3 圆、椭圆和弧	191
11.3 颜色	194
11.4 用 PaintPicture 方法画图	197
11.5 图形漫游	199
11.6 习题	201
第 12 章 Windows 应用程序接口(API)	203
12.1 Visual Basic 与动态链接库	203
12.1.1 动态链接库	203
12.1.2 在 Visual Basic 中使用动态链接库	203
12.2 API 文本浏览器	206
12.2.1 API 文本浏览器的使用	206
12.2.2 把声明、常量或类型复制到 Visual Basic 代码中	207
12.3 API 调用举例	209
12.3.1 调用 API 图形(图像)函数	210
12.3.2 改变窗体外观	213
12.3.3 文本输出与环境设置	217
12.4 习题	219
第 13 章 数据库程序设计	220
13.1 数据库构成及数据类型	220
13.1.1 基本概念	220
13.1.2 数据类型	221
13.2 用数据管理器建立数据库	221
13.2.1 建立新的数据库	221
13.2.2 输入数据	225
13.3 用数据控件访问数据库	226
13.3.1 一个简单例子	226
13.3.2 约束控件(Bound Control)	228
13.4 用数据控件查找、增加和删除记录	230
13.4.1 数据控件的属性、方法和事件	230
13.4.2 记录的查找、增加和删除	231
13.5 用 ADO 数据控件和 DataGrid 控件访问数据库	233
13.5.1 结构化查询语言 SQL	233
13.5.2 用 ADO 数据控件和 DataGrid 控件访问数据库	235
13.6 用数据窗体向导访问数据库	236

13.7 习题	240
第 14 章 多媒体程序设计	242
14.1 MCI 控件	242
14.1.1 MCI 控件的基本功能	242
14.1.2 MCI 命令及其使用	243
14.2 MCI 控件的属性和事件	245
14.3 MCI 控件程序举例	246
14.4 MCIWnd 控件	252
14.5 MediaPlayer 控件	255
14.6 习题	257
第 15 章 Internet 程序设计	258
15.1 Web 与 URL	258
15.1.1 什么是 Web	258
15.1.2 什么是 URL	259
15.2 WebBrowser 控件和 Internet Explorer 对象	260
15.2.1 WebBrowser 控件和 Internet Explorer 对象的属性和方法	261
15.2.2 WebBrowser 控件和 Internet Explorer 对象的事件	262
15.3 WebBrowser 控件应用举例	264
15.4 用 API 进行 Internet 程序设计	269
15.4.1 有关的 API 函数	269
15.4.2 程序举例	270
15.5 习题	275
参考文献	276

第1章 Visual Basic 集成开发环境与预定义对象

Visual Basic 是面向对象的编程语言,其程序设计是在一个高度集成的环境中进行的,它提供了很多预定义的对象。本章介绍 Visual Basic 的开发环境和最常用的两种预定义对象,即窗体和控件。

1.1 程序设计语言

“自然语言”是人与人之间用来交流思想的工具,而“程序设计语言”(或算法语言)则是人与计算机交流的工具,是由字、词和语法规则构成的指令系统,它随着计算机技术的进步而不断发展。

1.1.1 面向过程与面向对象的语言

从机器的执行角度来看,程序设计语言可以分成两大类,即低级语言和高级语言。低级语言面向机器的指令系统,分为机器语言和汇编语言两类。机器语言指令由“0”和“1”的二进制数字组成,可以被机器直接执行;而汇编语言则用一些特定的助记符来代表机器指令,其程序只需要经过简单的翻译就可以被机器执行。用低级语言编写的程序执行速度快,效率高,但程序设计难度大,非专业人员难以胜任。

高级语言由类似于人类自然语言的字、词、语句组成,易于为人们所理解。它不面向机器,结构简单,便于记忆,容易掌握,程序设计效率高,具有较好的可移植性,但运行效率低于用低级语言编写的程序。用高级语言编写的程序称为源程序,必须编译成与具体计算机相关联的机器语言程序(即目标程序)后才能在计算机上运行。所谓“程序设计语言”,通常指的是高级语言。在用高级语言编写程序时,摆脱了硬件的束缚,不必关心机器的内部结构或工作原理,而把主要精力集中在解决问题的思路和方法上。高级语言的出现是计算机技术发展的里程碑,大大提高了编程效率,从而可以开发出越来越大、功能越来越强的程序。

目前常用的高级语言大体上可以分为两类,即面向过程的语言和面向对象的语言。面向过程的程序设计语言把解题的过程看作是数据加工的过程,注重的是算法描述,因此,面向过程的高级语言又称为算法语言。常用的面向过程的语言有 BASIC、FORTRAN、PASCAL、C 等。

用面向过程的语言编写的程序可以通过两种方式执行,即解释方式和编译方式。解释方式是按照程序编排的顺序一条一条地把指令语句转换为机器代码然后执行。也就是说,每次运行程序时,每条语句都要经过转换和执行两个步骤,因此执行速度不快,而且离不开语言环境。而编译型语言则是用语言本身提供的编译(Compile)程序和连接(Link)程序把源程序编译连接成为可执行文件(.exe),这样生成的可执行文件可以脱离语言设计环境运行,而且速度较快。当前较为流行的面向过程的语言大多为编译型的,有些语言既可以通过解释方式执行,也可以通过编译方式执行。

计算机技术的进一步发展,特别是具有图形用户界面(GUI)的操作系统(如 Windows)的广泛使用,使得面向对象的程序设计思想应运而生。这种思想把现实世界看作是由不同种类对象

(Object)组成的有机整体,同一类型的对象既有共同点,又有各自不同的特性;各种类型的对象通过消息进行通信,并对消息作出相应的反应。采用了面向对象思想的程序设计语言就是面向对象的程序设计语言,而面向对象的程序设计称为 OOP(Object Oriented Programming)。目前较为流行的 OOP 语言有 C++、Java、Delphi 等,本书所介绍的 Visual Basic 也是面向对象的程序设计语言。

面向对象的程序设计语言把客观事物看作是具有属性和行为的对象,通过抽象找出同一类对象的共同属性(静态特征)和行为(动态特征),形成类;通过类的继承和多态可以方便地实现代码重用,从而可以缩短软件开发周期,使软件风格统一。因此,用面向对象的程序设计语言编写的程序能比较直接地反映问题的本来面目,使得软件研制人员可以利用人类认识事物的一般思维方法进行软件开发。

1.1.2 可视化与事件驱动型语言

Visual Basic 支持面向对象的程序设计,但它与一般的面向对象的程序设计语言(如 C++)不完全相同。在一般的面向对象程序设计语言中,对象由程序代码和数据组成,是抽象的概念;而 Visual Basic 则是应用面向对象的程序设计方法,把程序和数据封装起来作为一个对象,并为每个对象赋予应有的属性,使对象成为实在的东西。在设计对象时,不必编写建立和描述每个对象的程序代码,而是用工具画在界面上,Visual Basic 自动生成对象的程序代码并封装起来。每个对象以图形方式显示在界面上,都是可视的。

在用传统程序设计语言设计程序时,都是通过编写程序代码来设计用户界面,在设计过程中看不到界面的实际显示效果,必须编译后运行程序才能观察。如果对界面的效果不满意,还要回到程序中去修改。有时候,这种编程——编译——修改的操作可能要反复多次,大大影响了软件开发效率。Visual Basic 提供了可视化设计工具,把 Windows 界面设计的复杂性“封装”起来,开发人员不必为界面设计而编写大量程序代码。只需要按设计要求的屏幕布局,用系统提供的工具,在屏幕上画出各种“部件”,即图形对象,并设置这些图形对象的属性。Visual Basic 自动产生界面设计代码,程序设计人员只需要编写实现程序功能的那部分代码,从而可以大大提高程序设计的效率。

Visual Basic 通过事件来执行对象的操作。一个对象可能会产生多个事件,每个事件都可以通过一段程序来响应。例如,命令按钮是一个对象,当用户单击该按钮时,将产生一个“单击”(Click)事件,而在产生该事件时将执行一段程序,用来实现指定的操作。

在用 Visual Basic 设计大型应用软件时,不必建立具有明显开始和结束的程序,而是编写若干个微小的子程序,即过程,这些过程分别面向不同的对象,由用户操作引发某个事件来驱动完成某种特定的功能,或者由事件驱动程序调用通用过程来执行指定的操作。这样可以方便编程人员,提高效率。

在传统的面向过程的应用程序中,执行哪一部分代码和按何种顺序执行代码都由程序本身控制的。而在面向对象的程序设计中,编程人员要以“对象”为中心来设计模块,而不是以“过程”为中心来考虑应用程序的结构。此外,在事件驱动应用程序中,代码不是按预定的顺序执行的,而是在响应不同的事件时执行不同的代码段。

事件是可以由窗体或控件识别的操作。在响应事件时,事件驱动应用程序执行指定的代码。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。这些事件的顺序决定了代码执行的顺序,因此,在事件驱动应用程序中,每次

运行时所执行的代码和所经过的“路径”是不一样的。

Visual Basic 的窗体和每个控件都有一个预定义的事件集,当其中的某个事件发生,并且在相关联的事件过程中存在代码时,Visual Basic 将执行这些代码。

尽管 Visual Basic 中的对象能自动识别预定义的事件集,但必须通过代码判定它们是否响应具体事件以及如何响应具体事件,代码(即事件过程)与每个事件对应。为了让窗体或控件响应某个事件,必须把代码放入这个事件的事件过程中。

对象所能识别的事件类型有很多种,但多数类型为大多数对象所共有。例如,大多数对象都能识别 Click 事件,即单击事件;如果单击窗体,则执行窗体的单击事件过程中的代码;如果单击命令按钮,则执行命令按钮的单击事件过程中的代码。此外,某些事件可以在运行期间触发。例如,当在运行期间改变文本框中的文本时,将引发文本框的 Change 事件,如果 Change 事件过程中含有代码,则执行这些代码。

1.2 启动与退出 Visual Basic

开机并进入中文 Windows 后,可以用多种方法启动 Visual Basic,常用的方法是使用“开始”菜单中的“程序”命令。操作如下:

(1) 单击 Windows 环境下的【开始】按钮,弹出一个菜单,把鼠标光标移到“程序”命令上,将弹出下一个级联菜单。

(2) 把鼠标光标移到“Microsoft Visual Basic 6.0 中文版”,弹出下一个级联菜单,即 Visual Basic 6.0 程序组。

(3) 单击“Microsoft Visual Basic 6.0 中文版”,即可进入 Visual Basic 6.0 编程环境。

此外,还可以用其他方法启动 Visual Basic。例如在资源管理器中双击 Visual Basic 的启动文件 vb6.exe;使用“开始”菜单中的“运行”命令,在“运行”对话框中输入 Visual Basic 6.0 启动文件的名字(包括路径),然后单击【确定】按钮;建立启动 Visual Basic 6.0 的快捷方式等。

用上面所介绍的任何一种方法启动 Visual Basic 6.0 后,将首先显示版权屏幕,说明此程序的使用权属于谁。稍候,显示“新建工程”对话框,该对话框有“新建”、“现存”和“最新”三个选项卡,可分别用来建立新工程、显示现有的或最新的 Visual Basic 应用程序文件名列表。

“新建”选项卡对话框显示了可以在 Visual Basic 6.0 中使用的工程类型,即可以建立的应用程序,其中“标准 EXE”用来建立一个标准的 EXE 工程,本书将只讨论这种工程类型。

在对话框中选择要建立的工程类型(例如“标准 EXE”),然后单击【打开】按钮,即进入 Visual Basic 6.0 编程环境,如图 1-1 所示。

为了退出 Visual Basic,可先打开“文件”菜单,并执行其中的“退出”命令(或按<Alt>+<Q>键)。如果当前程序已修改过并且没有存盘,系统将显示一个对话框,询问用户是否将其存盘,此时选择【是】按钮则存盘,选择【否】按钮则不存盘。在上述两种情况下,都将退出 Visual Basic,回到 Windows 环境。

1.3 Visual Basic 编程环境

Visual Basic 应用程序的编写、编辑、调试、运行是在一个高度集成的环境下完成的,这个环境由若干个“窗口”组成,为了用 Visual Basic 开发应用程序,必须了解这些窗口的功能。

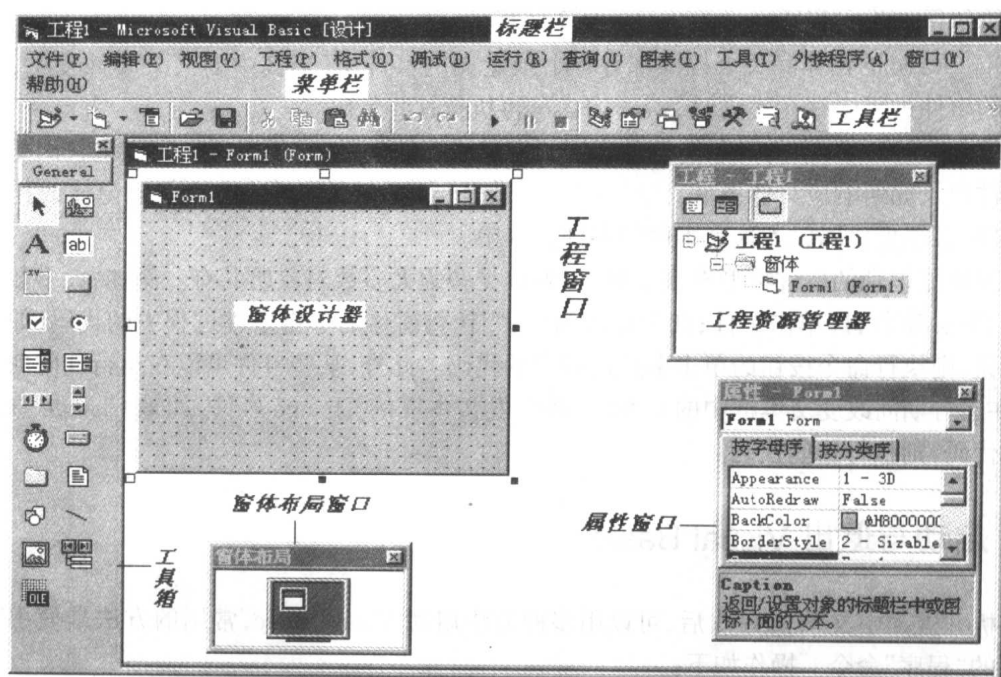


图 1-1 Visual Basic 6.0 编程环境(MDI 方式)

1.3.1 主窗口

主窗口也称设计窗口。启动 Visual Basic 后,主窗口位于集成环境的顶部,该窗口由标题栏、菜单栏和工具栏组成(参见图 1-1)。

1. 标题栏

标题栏是屏幕顶部的水平条,它显示的是应用程序(工程)的名字。用户与标题栏之间的交互关系由 Windows 处理,而不是由应用程序处理。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是“设计阶段”。随着工作状态的不同,方括号中的信息也随之改变,可能会是“运行”或“Break”,分别代表“运行阶段”或“中断阶段”。这三个阶段也分别称为“设计模式”、“运行模式”和“中断模式”。

2. 菜单栏

在标题栏的下面是集成环境的主菜单。菜单栏中的菜单命令提供了开发、调试和保存应用程序所需要的工具。Visual Basic 6.0 中文版的菜单栏共有 13 个主菜单,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。每个菜单含有若干个菜单命令,执行不同的操作。用鼠标单击某个菜单项,即可打开该菜单,然后用鼠标单击菜单中的某一条就能执行相应的菜单命令。例如,单击“文件”,就可以打开文件菜单,如图 1-2 所示。打开该菜单后,如果单击“打开工程”,就可以打开已有的工程文件;如果单击“工程另存为”,就可以保存文件,等等。

菜单中的命令分为 3 种类型,一类是可以直接执行的命令,这类命令的后面没有任何信息(例如“保存工程”);第二类在命令名后面带有省略号(例如“打开工程”),需要通过打开“对话框”来执行;第三类带有子菜单命令(见“格式”菜单),这类命令的右端有一个箭头。在用鼠标单击第

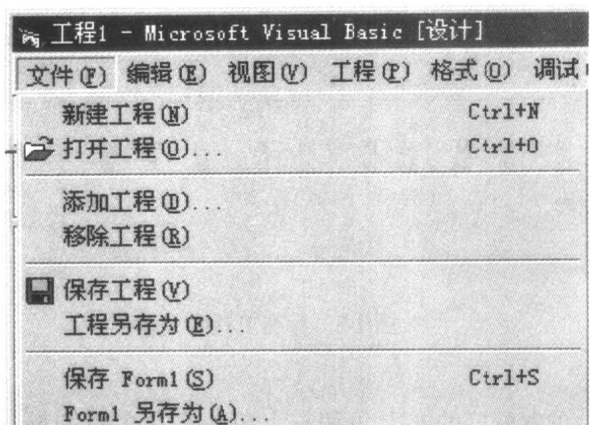


图 1-2 “文件”菜单(部分)

二类命令后,屏幕上将显示一个对话框,利用该对话框,可以执行各种有关的操作。在“文件”菜单中,“新建工程”、“保存工程”等是可以直接执行的命令,而“打开工程”、“工程另存为”等命令,则必须通过对话框来执行。此外,从“文件”菜单中可以看出,在有些命令的后面还带有其他信息,例如:

打开工程 ...	Ctrl+O
保存 Form1	Ctrl+S

其中“Ctrl+O”等称为“热键”(或快捷键)。在菜单中,热键列在相应的菜单命令之后,与菜单命令具有相同的作用。使用热键方式,不必打开菜单就能执行相应的菜单命令。例如,按〈Ctrl+O〉键,可以立即执行“打开工程”命令。注意,只有部分菜单命令能通过热键执行。

除通过鼠标和热键执行菜单命令外,也可以通过键盘执行菜单命令。当通过键盘执行菜单命令时,首先应按〈F10〉或〈Alt〉键,激活菜单栏,然后再用访问键(菜单项后面括号中的字母键)、〈→〉、〈←〉、〈↑〉或〈↓〉及回车键选择并执行指定的菜单命令。

菜单被打开后,在屏幕上显示相应的菜单命令。如果打开了不适当或不需要的菜单,或者执行菜单命令时打开了不需要的对话框,可以用〈Esc〉键关闭。

3. 工具栏

Visual Basic 6.0 提供了 4 种工具栏,包括编辑、标准、窗体编辑器和调试,并可根据需要定义用户自己的工具栏。在一般情况下,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)。每种工具栏都有固定和浮动两种形式。把鼠标光标移到固定形式工具栏中没有图标的地方,按住左按钮,向下拖动鼠标,或者双击工具栏左端的两条浅色竖线,即可把工具栏变为浮动的;而如果双击浮动工具栏的标题条,则可变为固定工具栏。

标准工具栏以图标的形式提供了部分常用菜单命令的功能。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。工具栏中有 21 个图标,代表 21 种操作,如图 1-3 所示。

在标准工具栏的右侧还有两个栏,分别用来显示窗体的当前位置和大小,其单位为 twip,1 英寸等于 1440 twip。左边一栏显示的是窗体左上角的坐标,右边一栏显示的是窗体的长×宽。

twip 是一种与屏幕分辨率无关的计量单位,无论在什么分辨率的屏幕上,如果画了一条 1440 twip 的直线,打印出来都是 1in。这种计量单位可以确保在不同的屏幕上都能保持正确的

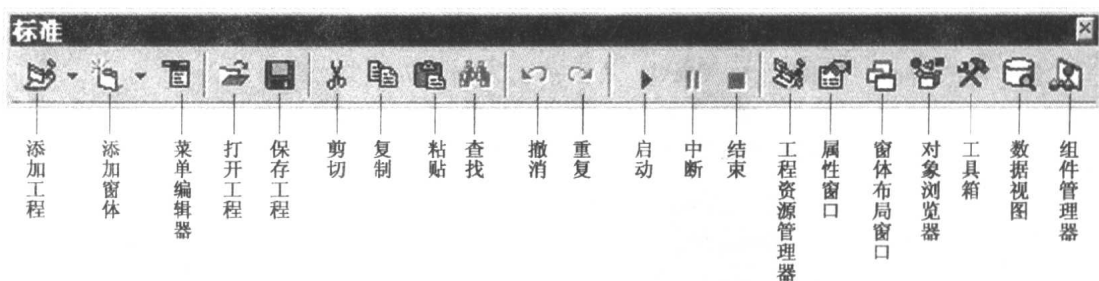


图 1-3 标准工具栏

相对位置或比例关系。

除上面几个部分外,在主窗口的左上角和右上角还有几个控制框,其作用与 Windows 下普通窗口中的控制框相同。

1.3.2 窗体设计器与工程资源管理器窗口

1. 窗体设计器窗口

窗体设计器窗口简称窗体(Form),是应用程序最终面向用户的窗口,它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 FormX(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,如图 1-4 所示,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,可通过执行“工具”菜单中的“选项”命令(“通用”选项卡)来调整。

2. 工程资源管理器窗口

在工程资源管理器窗口中,含有建立一个应用程序所需要的文件的清单。工程资源管理器窗口中的文件可以分为六类,即窗体文件(.frm)、程序模块文件(.bas)、类模块文件(.cls)、工程文件(.vbp)、工程组文件(.vbproj)和资源文件(.res),其中最常用的是窗体文件和工程文件。图 1-5 所示的是含有两个工程、多个窗体、多个程序模块和类模块的工程资源管理器窗口。

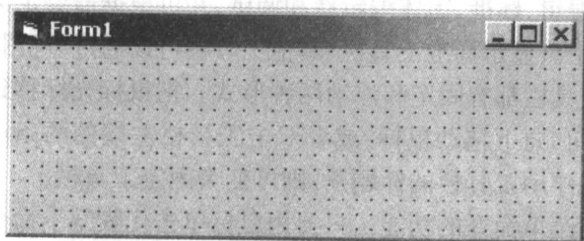


图 1-4 窗体

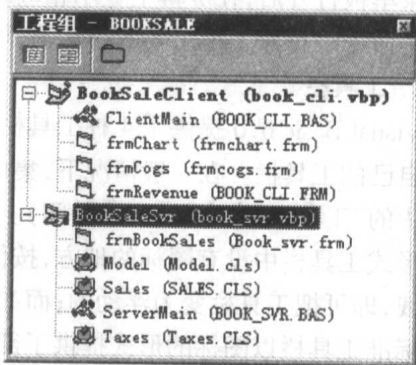


图 1-5 工程资源管理器窗口(展开)

在工程资源管理器窗口中,括号内是工程、窗体、程序模块、类模块等的存盘文件名,括号外是相应的名字(即 Name 属性)。每个工程名左侧都有一个方框,当方框内为“-”号时,该工程处于“展开”状态(见图 1-5),此时如果单击“-”号方框,则变为“折叠”状态,方框内的“-”号也变为“+”号。