

第 1 章 Visual Basic 编程环境

Visual Basic 是新一代的可视化程序设计语言,其应用程序设计是在一个集成开发环境(IDE)中进行的。本章将介绍 Visual Basic 6.0 版的集成开发环境。

1.1 可视化与事件驱动型语言

Visual Basic 是一种新型的现代程序设计语言,具有很多与传统程序设计语言不同的特点,其中最主要的特点有两个,即可视化界面设计和事件驱动的编程机制。

1.1.1 可视化界面设计

常用的高级程序设计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。

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 系列操作系统)的广泛使用,使得面向对象的程序设计思想应运而生,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。

Visual Basic 是面向对象的程序设计语言,但它与一般的面向对象的程序设计语言不完全相同。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。

用传统程序设计语言来设计程序时,主要的工作就是设计算法和编写代码,程序的功能和显示结果(包括大量的用户界面)都通过程序语句来实现。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(控件),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。也就是说,程序所需要的用户界面是用 Visual Basic 所提供的可视化设计工

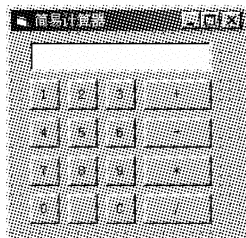


图 1.1 简易计算器面板

具“画”出来的,而不是用程序代码“写”出来的。例如,可以根据需要很容易地画出如图 1.1 所示的简易计算器面板,如果用传统的程序设计语言建立这样一个界面,则必须编写大量的程序代码,而且可能要经过反复修改才能达到设计要求。

1.1.2 事件驱动的编程机制

Visual Basic 通过事件驱动的方式来实现对象的操作,其程序不是按照预定的“路径”执行,而是在响应不同的事件时,驱动不同的事件代码,以此来控制对象的行为。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。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,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之中。

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 Visual Basic 的启动与退出

Visual Basic 6.0 可以在多种操作系统下运行,包括 Windows 95、Windows 98、Windows NT 4.0、Windows 2000 和 Windows XP。为了叙述方便,在本书中上述操作系统一律称作 Windows。此外,除非特别说明,Visual Basic 一般指的是 Visual Basic 6.0。

Visual Basic 6.0 是 Visual Studio 6.0 套装软件中的一个成员,它可以和 Visual Studio 6.0 一起安装,也可以单独安装。单独安装的 Visual Basic 6.0 中文版包括 4 张光盘,其中两张为 MSDN。安装方式不同,启动方式也略有区别。在这里,假定所使用的 Visual Basic 6.0 是单独安装的。

开机并进入中文 Windows 后,可以用多种方法启动 Visual Basic。

1. 使用“开始”菜单中的“程序”命令

(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。

2. 使用“我的电脑”

(1) 双击“我的电脑”,弹出一个窗口,然后单击 Visual Basic 6.0 所在的硬盘驱动器盘符,将打开相应的驱动器窗口。

(2) 单击驱动器窗口中的 vb60 文件夹,打开“VB60”窗口。

(3) 双击“vb6.exe”图标,即可进入 Visual Basic 6.0 编程环境。

3. 使用“开始”菜单中的“运行”命令

(1) 单击“开始”按钮,弹出一个菜单,然后单击“运行”命令,将弹出一个对话框。

(2) 在“打开”栏内输入 Visual Basic 6.0 启动文件的名字(包括路径)。例如:

```
c:\vb60\vb6.exe
```

(3) 单击“确定”按钮,即可启动 Visual Basic 6.0。

4. 建立启动 Visual Basic 6.0 的快捷方式

具体操作见有关资料。

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,说明此份程序副本的使用权属于谁。稍候,显示“新建工程”对话框,如图 1.2 所示。图中所显示的是“新建”选项卡,如果单击“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,可从中选择要打开的文件名。

“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 集成环境,如图 1.3 所示。

Visual Basic 6.0 提供了许多种工程类型,以满足不同的需要。每次启动 Visual Basic 时,都要显示“新建工程”对话框。在一般情况下,可能主要使用“标准 EXE”工程,因此,没有必要在每次启动 Visual Basic 时显示该对话框,这可以通过选择对话框(见图 1.2)左下角的“不再显示这个对话框”选项来实现(用鼠标单击小方框,使框内有“√”)。选择该选项后,再选择“标准 EXE”,然后单击“打开”按钮,进入 Visual Basic 开发环境。

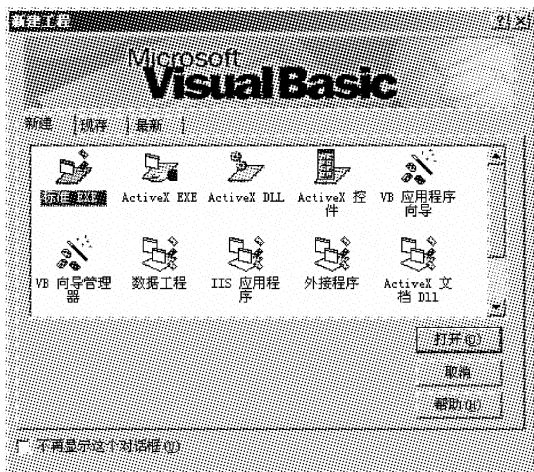


图 1.2 “新建工程”对话框(“新建”选项卡)

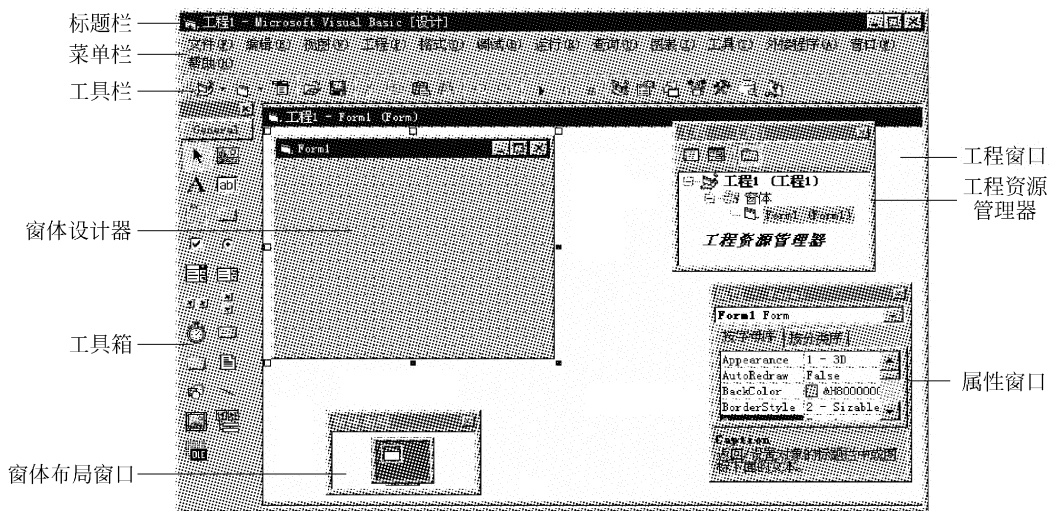


图 1.3 Visual Basic 6.0 编程环境(MDI 方式)

以后再启动 Visual Basic 时,就不再显示“新建工程”对话框,直接进入开发环境。

在默认情况下,Visual Basic 6.0 的集成开发环境为传统的 Windows MDI(多文档界面)方式(见图 1.3);此外,也可以用 SDI(单文档界面)方式启动 Visual Basic。在多数情况下,使用 SDI 方式可能会更方便。为了把编程环境变为 SDI 方式,可执行“工具”菜单中的“选项”命令,打开“选项”对话框,选择“高级”选项卡,在对话框中选择“SDI 开发环境”选项,然后单击“确定”按钮。这样设置后,退出 Visual Basic,然后重新启动,即可按 SDI 方式进入 Visual Basic 集成开发环境。

从图 1.3 中可以看出,启动 Visual Basic 后,屏幕上分为若干部分,包括标题栏、菜单栏、工具栏、工具箱、窗体设计器、工程资源管理器窗口、窗体布局窗口和属性窗口。为了能清楚地看到每个部分,这里对原来的各部分进行了缩放和重新排列。读者在启动自己

的 Visual Basic 后,所看到的各部分的排列情况可能与图 1.3 所示的有微小差别,一些窗口会重叠。实际上,和 Windows 下的窗口一样,集成开发环境中的每个窗口都可以在屏幕上移动、缩小、放大或关闭。此外,Visual Basic 保存上一次使用时屏幕上各部分最后的排列方式,并作为下一次启动 Visual Basic 后的屏幕布局。

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)。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 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(见图 1.3)。

1.3.1 标题栏和菜单栏

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的名字。用户与标题栏之间的交互关系由 Windows 来处理,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 3 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 个菜单项: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 1.4 所示。打开菜单后,如果单击“打开工程”,就可以打开已有的工程文件;而如果单击“工程另存为”,就可以保存文件;等等。在以后的叙述中,形如上面的操作记为“执行文件菜单中的‘打开工程’命令”、“执行文件菜单中的‘工程另存为’命令”。

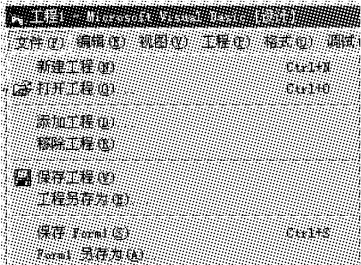


图 1.4 “文件”菜单(部分)

菜单中的命令分为 3 种类型,第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”等叫做“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。

上面介绍了通过鼠标和热键执行菜单命令的方法。除鼠标外,也可以通过键盘执行菜单命令。只有在打开菜单后,才能选择所需要的命令,执行相应的操作。Visual Basic 6.0 提供了多种打开菜单和选择菜单的方法,用户可以根据自己的兴趣或习惯选用其中的一种。

第一种方法,步骤如下:

- (1) 按 F10 或 Alt 键,激活菜单栏,此时第一个菜单项“文件”被加上一个浅色的框。
- (2) 按菜单项后面括号中的字母键,打开菜单,下拉显示该菜单项的命令。菜单被打开后,各菜单命令后面的括号内都有一个字母。
- (3) 按菜单命令后面括号中的字母键,即可执行相应的命令。

第二种方法,步骤如下:

- (1) 按 F10 或 Alt 键,激活菜单栏,此时第一个菜单项“文件”被加上一个浅色的框。
- (2) 用“→”或“←”把条形光标移到需要打开的菜单上,按回车键,打开该菜单。
- (3) 菜单被打开后,条形光标覆盖在第一个或上一次执行的菜单命令上。用“↑”或“↓”把条形光标移到所需要的命令上,按回车键即可执行条形光标所在位置的菜单命令。

第三种方法,步骤如下:

- (1) 按下 Alt 键,不要松开,接着按需要打开的菜单项后面括号中的字母键,然后松开(Alt 键接着松开),该菜单即被打开。
- (2) 按菜单命令后面括号中的字母键,即可执行指定的菜单命令。

例如,为了执行“文件”菜单中的“打开工程”命令,可以这样操作:按住 Alt 键,不要松开,接着按 F 键,先后松开 F 键和 Alt 键,再按 O 键,即可执行“文件”菜单中的“打开工程”命令。上述过程记做:Alt+F,O。

除上面 3 种方法外,有些菜单命令还可以通过热键执行。对于没有热键的菜单命令,只能通过上面 3 种方式执行。

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。

Visual Basic 应用程序的编辑、编译、连接、运行、调试及文件的打开、保存等都可以通过相应的菜单命令来实现,其用法与上面介绍的类似。

1.3.2 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。

固定形式的标准工具栏位于菜单栏的下面,即主窗口的底部,它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。工具条中有 21 个图标,代表 21 种操作,如图 1.5(a)所示。大多数图标都有与之等价的菜单命令。图 1.5(b)是浮动形式的标准工具栏。

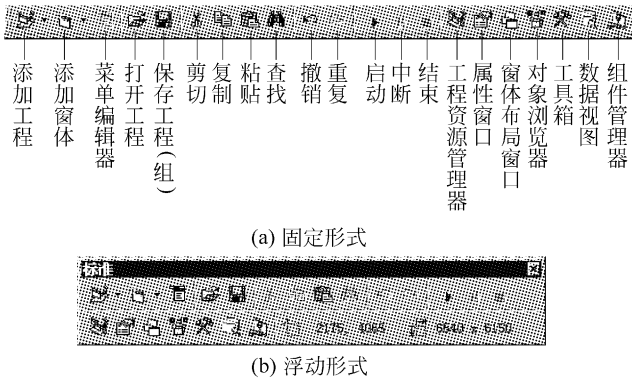


图 1.5 标准工具栏

表 1.1 列出了工具栏中各图标的作用。

表 1.1 工具栏图标

名 称	作 用
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	打开菜单编辑对话框,相当于“工具”菜单中的“菜单编辑器”命令
打开工程	用来打开一个已经存在的 Visual Basic 工程文件,相当于“文件”菜单中的“打开工程”命令
保存工程(组)	保存当前的 Visual Basic 工程(组)文件,相当于“文件”菜单中的“保存工程(组)”命令
剪切	把选择的内容剪切到剪贴板,相当于“编辑”菜单中的“剪切”命令
复制	把选择的内容复制到剪贴板,相当于“编辑”菜单中的“复制”命令

名 称	作 用
粘贴	把剪贴板的内容复制到当前插入位置,相当于“编辑”菜单中的“粘贴”命令
查找	打开“查找”对话框,相当于“编辑”菜单中的“查找”命令
撤销	撤销当前的修改
重复	对“撤销”的反操作
启动	用来运行一个应用程序,相当于“运行”菜单中的“启动”命令
中断	暂停正在运行的程序(可以单击“启动”按钮或按 Shift+F5 键继续),相当于热键 Ctrl+Break 键或“运行”菜单中的“中断”命令
结束	结束一个应用程序的运行并回到设计窗口,相当于“运行”菜单中的“结束”命令
工程资源管理器	打开工程资源管理器窗口,相当于“视图”菜单中的“工程资源管理器”命令
属性窗口	打开属性窗口,相当于“视图”菜单中的“属性窗口”命令
窗体布局窗口	打开窗体布局窗口,相当于“视图”菜单中的“窗体布局窗口”命令
对象浏览器	打开“对象浏览器”对话框,相当于“视图”菜单中的“对象浏览器”命令
工具箱	打开工具箱,相当于“视图”菜单中的“工具箱”命令
数据视图	打开数据视图窗口
组件管理器	管理系统中的组件(Component)

在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,无论在什么屏幕上,如果画了一条 1440 twip 的直线,打印出来都是 1 英寸。这种计量单位可以确保在不同的屏幕上都能保持正确的相对位置或比例关系。在 Visual Basic 中,twip 是默认单位,可以通过 Scalemode 属性改变。

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。

1.4 其他窗口

标题栏、菜单栏和工具栏所在的窗口称为主窗口。除主窗口外,Visual Basic 6.0 的编程环境中还有其他一些窗口,包括窗体设计器窗口、属性窗口、工程管理器窗口、工具箱窗口、调色板窗口、代码窗口和立即窗口等。本节介绍其中的部分窗口。

1.4.1 窗体设计器和工程资源管理器

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, 并将其命名为 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(见图 1.6), 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, 则可通过执行“工具”菜单中的“选项”命令(“通用”选项卡)来调整。

在窗体的左上角是窗体的标题, 右上角有 3 个图标, 其作用与 Windows 下普通窗口中的图标相同。

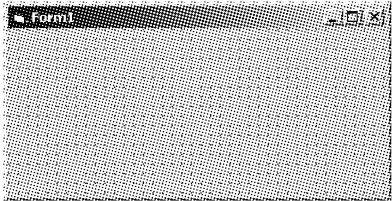


图 1.6 窗体

在设计应用程序时, 窗体就像是一块画布, 在这块画布上可以画出组成应用程序的各个构件。程序员根据程序界面的要求, 从工具箱(见 1.4.2 节)中选择所需要的工具, 并在窗体中画出来, 这样就完成了应用程序设计的第一步。

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 6 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(*.vbg)和资源文件(*.res)。图 1.7 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。

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(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 1.7), 此时如果单击“—”号方框, 则变为“折叠”状态, 方框内的“—”号变为“+”号, 如图 1.8 所示。

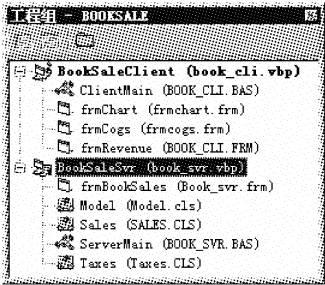


图 1.7 工程资源管理器窗口(展开)

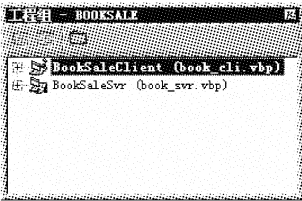


图 1.8 工程资源管理器窗口(折叠)

可以出现在工程资源管理器窗口中的文件有以下几类:

(1) 工程文件和工程组文件

工程文件的扩展名为 .vbp, 每个工程对应一个工程文件。当一个程序包括两个以上的工程时, 这些工程构成一个工程组, 工程组文件的扩展名为 .vbg。用“文件”菜单中的“新建工程”命令可以建立一个新的工程, 用“打开工程”命令可以打开一个已有的工程, 而用“添加工程”命令可以添加一个工程。

(2) 窗体文件

窗体文件的扩展名为 .frm, 每个窗体对应一个窗体文件, 窗体及其控件的属性和其他信息(包括代码)都存放在该窗体文件中。一个应用程序可以有多个窗体(最多可达 255 个), 因此就可以有多个以 .frm 为扩展名的窗体文件。

执行“工程”菜单中的“添加窗体”命令或单击工具栏中的“添加窗体”按钮可以增加一个窗体, 而执行“工程”菜单中的“移除”命令可以删除当前的窗体。每建立一个窗体, 工程管理器窗口中就增加一个窗体文件, 每个窗体都有一个不同的名字, 可以通过属性窗口设置(Name 属性), 其默认的名字为 Form x (x 为 1, 2, 3, ...), 相应的默认文件名为 Form x .frm (x 为 1, 2, 3, ...).

(3) 标准模块文件

标准模块文件也称程序模块文件, 其扩展名为 .bas, 它是为合理组织程序而设计的。标准模块是一个纯代码性质的文件, 它不属于任何一个窗体, 主要在大型应用程序中使用。

标准模块由程序代码组成, 主要用来声明全局变量和定义一些通用的过程, 可以被不同窗体的程序调用。标准模块通过“工程”菜单中的“添加模块”命令来建立。

(4) 类模块文件

Visual Basic 提供了大量预定义的类, 同时也允许用户根据需要定义自己的类, 用户通过类模块来定义自己的类, 每个类都用一个文件来保存, 其扩展名为 .cls。

(5) 资源文件

资源文件中存放的是各种“资源”, 是一种可以同时存放文本、图片、声音等多种资源的文件。资源文件由一系列独立的字符串、位图及声音文件(如 .wav 文件、.mid 文件等)组成, 其扩展名为 .res。资源文件是一个纯文本文件, 可以用简单的文字编辑器(如 NotePad)编辑。

除上面几类文件外, 在工程管理器窗口的顶部还有 3 个按钮, 分别为“查看代码”、“查看对象”和“切换文件夹”。如果单击工程资源管理器窗口中的“查看代码”按钮, 则相应文件的代码将在代码窗口中显示出来。当单击“查看对象”按钮时, Visual Basic 将显示相应的窗体。在一般情况下, 工程资源管理器窗口中的项目不显示文件夹, 如果单击“切换文件夹”按钮, 则可显示各类文件所在的文件夹, 如图 1.9 所示。如果再单击一次该按钮, 则取消文件夹显示。

用 Visual Basic 设计应用程序时, 通常先设计窗体(即界面), 然后再编写程序。设计完窗体后, 只要双击窗体的任一部位, 就可以切换到代码窗口, 与单击“查看代码”按钮的作用相同。

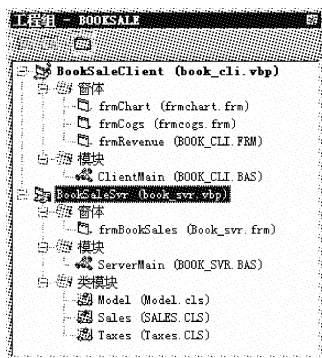


图 1.9 用“切换文件夹”按钮显示文件夹

1.4.2 属性窗口和工具箱窗口

1. 属性窗口

属性窗口主要是针对窗体和控件设置的, 在 Visual Basic 中, 窗体和控件被称为对