

第 1 章 | 概述

自从 1991 年微软公司推出 Visual Basic 1.0 以来，VB 首次以全新的可视化的编程方式给软件开发方法和软件工业带来了革命性的影响。随着 VB 版本的不断升级，在保持简单易用的特点上，功能也越来越丰富，发展到现在 Visual Basic 6.0 已经成为非常成熟稳定的开发系统，能够让数以百万计的开发人员快速建立各种应用程序，成为当前 Windows 平台上最流行的一种开发环境。但要注意的是，随着用户界面的丰富，应用程序的源代码文件也表现出多样化的特点，因此为了更好地维护应用程序，开发者需要了解一个 VB 应用程序可能包含哪些类型的文件，以及它们各自的内容和作用。

1.1 Visual Basic 语言介绍

Visual Basic 语言是一种可视化的编程语言，简称 VB，其前身是 Basic 语言，早在 20 世纪 80 年代，Basic 语言就在我国得到广泛的学习和应用。Basic 语言的英文全称为 **Beginner's All-purpose Symbolic Instruction Code**，顾名思义，它是一种适合于计算机初学者学习的程序语言，其语法规则相对简单。因为它是解释运行的，可以一行一行编写代码并执行，因此对于初学者来说比较容易理解和掌握。

随着 Windows 操作平台的普及，越来越多的计算机用户开始喜欢并逐渐习惯了这种简便的操作方式，因此编制图形化用户界面的应用程序成为开发者的首选。但是早期的程序设计语言主要以字符界面为主要的输入和输出方式，屏幕是以字符行和列（如 25 行 × 80 列）为单位的，键盘是主要的输入工具。那时候要实现一个图形化的窗口是非常困难的，一个最简单的对话框也需要编写大量的代码，就连一个按钮都需要写很多行程序才能显示出来，最后使应用程序中用于控制界面的代码往往占总体代码的一半以上。此外如果要对已完成的界面实现变动又是一件头疼的事情。于是程序员终日与界面做斗争，软件开发整体效率降低。

在广大程序员的期待之中，具有“所见即所得”特点的可视化编程工具应运而生，微软公司推出的 Visual Basic 就是其中最具有代表性的产品之一。它是一个实用的可视化编程工具，该工具提供了一个图形界面的工具箱，工具箱中包含程序常用的图形元素，这些现成的按钮、输入文本框、列表框等图形元素可以方便地被拖放到一个窗口上，开发人员可以更容易、更快速地完成用户界面的定制，设计过程中所见到的界面与最终程序运行的界面几乎完全相同。至于图形界面是如何在屏幕上显示和处理等问题完全不需要关心，于是程序员将更多的精力集中在特定应用问题的解决上，大大简化了程序员的编程工作。

Visual Basic 不仅继续保持了 Basic 语言易学易懂的特点，而且采用了事件驱动的编程模式。事件驱动编程模拟了现实生活中对象对外界刺激（或请求）的一种响应。因为整个编程机制建立在面向对象的基础上，因此采用了大量的封装技术，将许多计算机内部的技术细节

隐藏起来，使很多方面更符合人的思维，给初学者营造了一个轻松无障碍的交互式的开发环境。即使一个毫无编程经验的新手，通过几天的学习也能写出简单的应用程序，相对于其他语言和开发工具这一优势非常明显。

为了满足特定的开发需求，有三种 Visual Basic 版本可以使用：学习版、专业版和企业版，其中企业版功能最为强大。学习本书的内容最好安装专业版或企业版。

1.2 Visual Basic 可以做什么

上面讲了这么多，有的人可能会认为 Visual Basic 既然适合初学者使用，功能一定非常简单，应用也会受到限制。其实不然，从程序设计语言的角度来看，VB 提供完善的数据类型和流程控制、灵活的数组管理、功能强大的内部函数，可以实现各种格式数据的处理和文件的管理；从用户界面的角度来看，可以直接使用 VB 本身提供的大量的图形化控件和 Windows 底层的函数，而且可以方便地使用第三方提供的部件进行编程，能完成各种风格的界面定制和图形化应用；从数据库访问的角度，它有方便的交互式设计环境，丰富的数据库控件和数据库绑定控件，能处理各种不同的数据库系统中的数据。可以这么说，我们所能见到的在个人计算机上的软件几乎都可以采用 Visual Basic 来完成。

下面就一些实例来说明其具体的应用范围。

2.1 字符串替换机

我们平时经常会编写或管理一些文档，比如程序代码、论文或其他纯文本文件，如果在一篇文档的编辑中发现某个词或短语的用法不合适，希望全部替换为另外一个词语，可以通过记事本或 Word 等编辑工具所提供的替换功能来实现，但是如果有更多特殊的要求，就可以使用 VB 编写一个简单的程序来完成。我们可以为这个程序设计以下功能：

- 在程序窗口中指定一个或多个文件，或者指定某个目录下所有后缀为 .TXT 的文件；
- 在程序窗口中输入被替换的短语和替换短语；
- 选择文件后，自动读取内容，完成替换并保存（文件名不变）；
- 替换完成后，在程序窗口中显示找到并替换的次数。

编制这样的一个实用小程序会使用到 VB 提供的字符串和文件处理的功能，在界面上包含目录和文件列表、输入文本框和命令按钮等简单控件。当然还会有人提出每次只能替换一个短语不方便，希望可以事先准备一个短语替换列表，列表中包含多个替换短语，程序能一次完成，这样不必一次又一次地输入并替换，只要稍微增加一些处理，该功能同样可以实现。

通过对本书部分章节的学习后，在上述功能上再作些补充，我们就可以编写出一个功能完善的字符串替换机了。

2.2 俄罗斯方块游戏软件

我们多数人都在计算机上玩过一些游戏，有些游戏软件看似简单，但也能引人入胜，调节情绪，比如俄罗斯方块、扫雷、五子棋等。编写这样的游戏软件，需要先将规则理解透彻并转换为程序设计的算法语言，然后使用一些作图工具设计要用到的界面图形，编写程序主要是识

别有效的键盘或鼠标操作，并使用算法来控制界面图形的显示或运动。此外还可以加入一些设置选项，比如在俄罗斯方块游戏中，可以根据完成的分数来决定方块下降的速度，或者随机启用一些复杂的方块来增加难度。当然将单人游戏变成双人、三人也都难不倒 VB 的。

1.2.3 多媒体教学演示系统

VB 提供了丰富多彩的图形动画和多媒体功能，可以利用它们制作出形象生动的辅助教学软件，比如物理中的抛物线运行、碰撞试验、波的运动等，都可以运用 VB 的定时器控件和绘图功能来模拟出逼真的效果。与专门的多媒体制作平台制作的演示系统相比，VB 开发的多媒体演示程序不是一个单一死板的演示，只要程序设计得好，就是一个具有交互式的智能化的模拟系统，教师或学生可以输入不同的实验参数，从而得到不同的结果，达到直观教学的效果。

1.2.4 医疗保险信息系统

我国目前的医疗保障制度正在走向社会化，有些城市已经实行了改革，建立了一套完善的医疗保险体制，并实现了计算机管理。计算机医疗保险信息系统将个人、单位、医院和保险中心紧密地联系起来，采用数据库系统维护并管理各个部分的基本信息和所发生的医疗业务信息，而用户界面和复杂的业务逻辑则采用 VB 来实现。

比如从医院的角度来看，使用医疗保险信息系统的过程是这样的：病人在获得医生所开的处方后，在收费处可以直接出示医疗卡，根据处方项目的不同规则可以计算出本次就诊个人自付的款额，病人只需要支付这部分费用，其他款项则由医院和保险中心结清。那么为了完成这样的业务过程，我们可以这样设计应用程序：

(1) 收款员打开一个新的收费窗口，输入病人的医疗卡号，程序到数据库中读取该病人的基本信息并显示出来，如果病人的医疗账户处于冻结状态或遇到其他非法情况时，显示提示信息并拒绝使用。

(2) 收款员逐条输入处方单上的药名（或检查项目名）和数量，药品价格系统可以从数据库中读取并计算显示，确认输入完毕后，程序自动计算费用的摊销比例，并将结果显示在屏幕上。

(3) 收款员确认收款后，将本次处方记录和发生的所有分摊费用分别记录到数据库中。

这是一个典型的数据库应用程序，需要使用 VB 的数据库访问技术。据统计，计算机的应用 80%是用来解决企事业单位的信息管理，这些领域也正是 VB 大显身手的舞台。

总之以上这些实例只是 VB 应用中的冰山一角，我们对它们展开联想就会发现，VB 的确是编写 Windows 应用程序的强有力的工具。

1.3 VB6.0 的开发环境

Visual Basic 所包含的不仅仅是程序设计语言的含义，它更是一个集成开发环境（IDE），也就是说它给程序员提供了一个集界面设计、代码编辑、编译运行、跟踪调试、联机帮助以及向导工具等为一体的软件开发平台。下面介绍 VB6.0 集成开发环境的主要组成部分。

图 1-1 所示的是 VB6.0 的基本界面，该界面中的每个组成部分都是独立的小窗口，位置和大小可以自行设定，还可控制显示或隐藏。为了整个界面的简洁美观，更多的组成部分在我们需要的时候才打开。

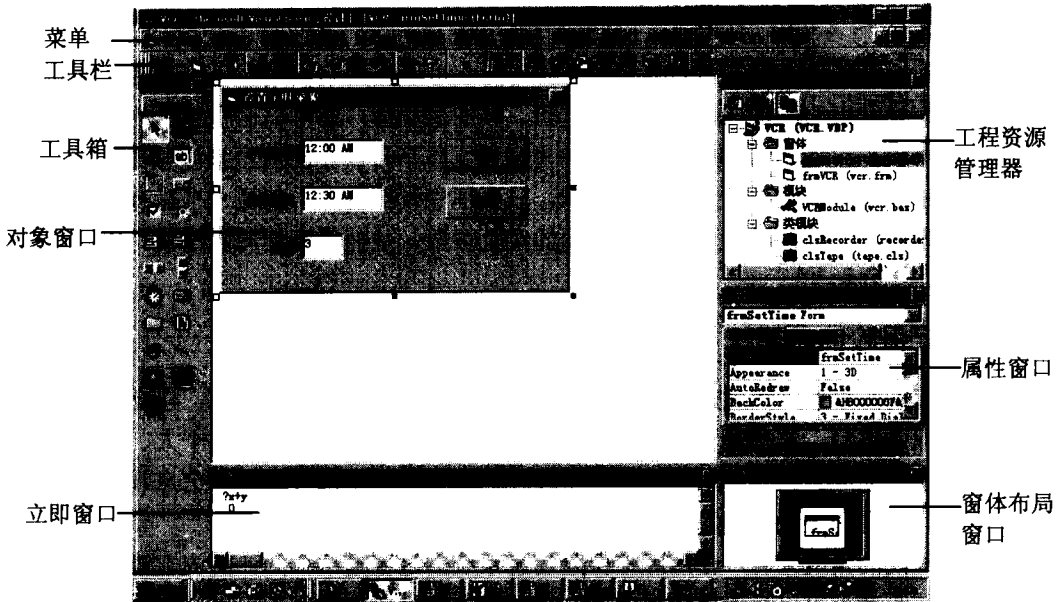


图 1-1 VB6.0 开发环境

1. 菜单和工具栏

菜单包含 VB 的所有操作命令，其中最常用的操作是以工具栏方式提供的。VB 启动时缺省显示标准的工具栏，除提供文件编辑的基本操作图标外，还有几个我们也经常会用到，如图 1-2 所示，鼠标点击图标会打开相对应的窗口。

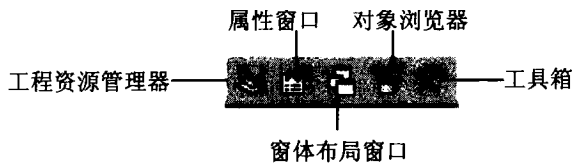


图 1-2 常用标准工具栏图标

如果有需要可以通过【视图】 【工具栏】菜单来打开或关闭其他的工具栏，如图 1-3 所示。图 1-4 是在编辑代码时常用的“编辑”工具栏。

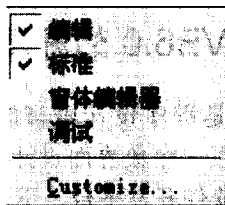


图 1-3 “工具栏”菜单

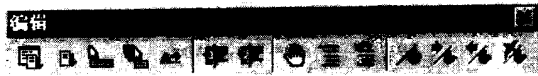


图 1-4 编辑工具栏图标

2. 工具箱

在应用程序中会使用到各种不同的图形界面元素，比如用于输入的文本框，用于选择的单选按钮或多选框，这些元素在 VB 中称为控件，不同的控件采用不同的小图标来表示，所有控件的小图标都放置在工具箱中。当设计一个窗体对象需要某个控件时，用鼠标直接单击该控件图标，使图标变为选中状态，然后在窗体的合适位置按下鼠标将控件画出。

工具箱窗口如果关闭了，可以使用工具栏按钮或【视图】 【工具箱】菜单来打开。工具箱的内容还可以扩充，通过鼠标右键的弹出式菜单【部件】可以打开一个对话框，将需要的外部控件加载进来。

3. 工程资源管理器

一个 VB 应用程序可以很简单，只有一个窗口界面或简单的代码，也可以很庞大，由很多窗口界面或代码单元组成，因此需要一个有效的机制将它们组织起来，这就是工程（Project）的概念。一个独立的应用程序就是一个工程，也可以理解为一个需要开发的项目，应用程序的各个组成部分就是工程资源，所有的工程资源由工程资源管理器窗口来实现可视化管理。

图 1-1 所示的工程是一个录像机工作的模拟程序，其中包含了多个组成部分，在工程资源管理器中我们可以清晰地看到有两个窗体、一个标准模块及两个类模块。在工程资源管理器窗口中可以打开每一个资源，也能直接添加或删除每一个资源，每个资源以独立的文件保存。因此一个完整的 VB 工程包含多个文件，初学者在保存和管理工程时应该注意不要遗漏了工程资源管理器中显示的文件，否则在需要重新修改或编译源码时会找不到相应的文件代码。

工程资源管理器可以通过工具栏按钮或【视图】 【工程资源管理器】菜单来打开。

4. 对象和代码窗口

VB 开发环境的中间部分是用来设计窗体界面和编写代码的，当用来设计窗体时称为对象窗口，而书写程序代码时称为代码窗口。对象和代码窗口采用多文档界面管理，它们可以互相覆盖，在【窗口】菜单中可以对多个窗口进行排列或选中某个窗口。

对象窗口仅仅是一个窗体界面设计器，可以定义窗体对象的大小和外观特征，安排控件对象在窗体中的显示大小和位置，然后结合属性窗口进行其他属性设置。每个对象窗口都有自己的代码窗口，当需要对某个窗体进行编程操作时，就需要打开该窗体对象的代码窗口，可以在“工程资源管理器”窗口中使用弹出菜单或执行【视图】 【代码窗口】菜单打开窗体对象的代码窗口。

代码窗口能够自动采用不同颜色显示注释和语句，还可以实现代码的自动填充，对象属性、方法和函数参数的提示，例如图 1-5 就是对象的使用提示，只要在对象后面输入了“.”运算符就能列出该对象的所有属性和方法供编程人员选用，这样大大减轻了程序员的记忆负担，加快编程速度。

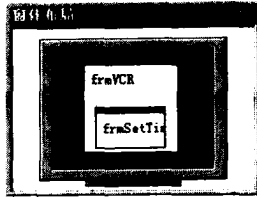


图 1-7 窗体布局窗口

7. 对象浏览器

VB 为我们提供了庞大的对象库，但任何程序员都不可能将所有对象的使用方法都牢记于心，利用对象浏览器可以显示出对象库以及工程里过程中的可用类、属性、方法、事件及常数变量，或者输入对象名或对象成员名进行搜索。

图 1-8 就是对象浏览器窗口，左边列表选中的是全局类，右边列表显示的是全局范围内可用的成员，有全局的函数、对象、常量等内容，如果左边选中的是一个控件，则右边显示该控件的所有属性、方法和事件。下方是对选中项目的一个简要描述。通过工具栏按钮或【视图】 【对象浏览器】菜单可以打开对象浏览器。

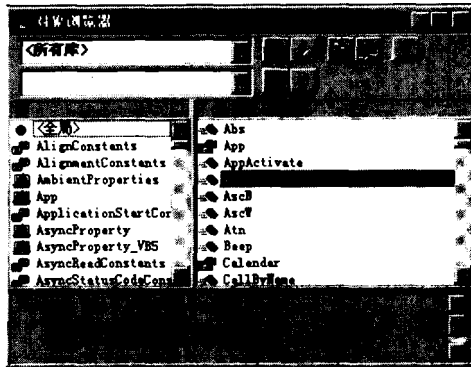


图 1-8 对象浏览器窗口

8. 立即窗口

立即窗口一般显示在屏幕的最下方，程序员可以使用它来调试程序，用法可以有两种：

- 直接在立即窗口中输入一行代码，然后按下回车键执行，立即可以看到结果的显示。

图 1-9 就是在立即窗口中输入并执行了两条语句，一条是取子串的语句，另一个是显示当前日期转化为字符串后的结果。当然如果程序进入中断运行的模式，在立即窗口中可以直接查询或更改程序中的变量值或控件的属性值。

- 在程序中加入特定的调试输出语句，程序运行过程中可以自动将输出结果显示在立即窗口中以供观察，使用的是调试对象 Debug 的 Print 方法，如 `Debug.Print"Hello"`。



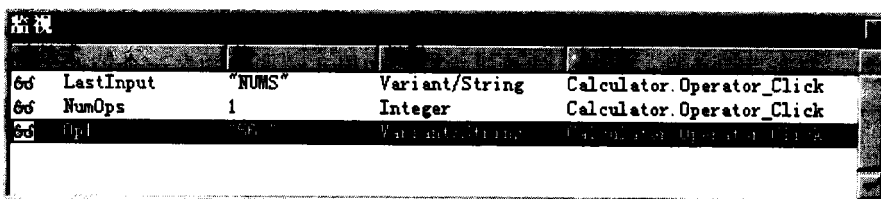
图 1-9 立即窗口

9. 本地和监视窗口

VB6.0 有很强的程序调试功能，特别是可以方便地查看或修改程序中的变量，我们可以随时监视某些重要的变量值，看它们的变化是否符合程序的逻辑，变量值的显示和编辑由本地和监视窗口来完成。

只有当运行程序处于中断模式时，本地和监视窗口才生效。这时如果立即窗口已经打开，则会自动显示出中断所在过程中的所有变量及变量值。而监视窗口在一开始是空白的，只有当我们主动添加了某些变量监视后，才会出现相应的变量及变量值。在图 1-10 所示的监视窗口中我们可以看到 LastInput、NumOps 和 Op1 三个变量的情况，不仅可以看到，甚至可以点击值所在的列，直接在监视窗口中修改变量的值。

与监视有关的调试方法我们在第 7 章有更详细的介绍。



监视	名称	值	数据类型	来源
66	LastInput	"NUMS"	Variant/String	Calculator.Operator_Click
66	NumOps	1	Integer	Calculator.Operator_Click
66	Op1	+	Variant/String	Calculator.Operator_Click

图 1-10 监视窗口

1.4 认识 VB 工程

现在对于 VB 的整个开发环境我们有了初步的了解，但在真正开始编写程序之前，对于一个程序的结构还需要一个全面的介绍。

在 VB 中编写程序都需要使用工程的概念，启动 VB 的第一步就是建立一个新的工程或者打开一个现有的工程。工程的英文原文是 Project，首先它是一个应用程序的描述，比如工程的名称就对应于应用程序的名称，其次它是一个集合管理器，用来管理一个应用程序的全部文件。那么编写一个程序究竟会出现哪些文件呢，最常见的是工程文件（.vbp）和窗体文件（.frm），也可能会需要标准模块文件（.bas）和类模块文件（.cls）这些文件也称为工程的资源，以下一一介绍。

1. 工程文件

工程文件就是描述一个工程的，该文件后缀为 .vbp，是一个纯文本文件，包含的是与该工程有关的全部文件和对象库的清单，以及有关选项的设置。需要注意的是很多初学者以为工程文件是一个集全部代码为一体的综合体，其实它并不保存任何界面或程序代码，而仅仅是一个描述文件，类似于购买一台计算机所提供的全部部件清单。

当我们需要查看已有的应用程序的源码时，我们直接打开该程序的工程文件就可以，所有的组成文件随之会在“工程资源管理器”中列出。一旦某个组成文件丢失，打开工程时就会报告错误，虽然可以继续加载该工程，但整个工程就不完整了，所以下面介绍到的组成文件是比工程文件更为重要的资源，万万不可丢失。

2. 窗体文件

只要我们编写的是有界面的应用程序，肯定需要设计窗体 (Form)，每个窗体最终分别

保存为独立的窗体文件，后缀为 **.frm**。窗体文件包含了一个窗体中的所有界面元素的描述，更重要的是与本窗体操作有关的全部代码，它也是一个纯文本文件。

一个窗体文件可以添加到另一个或多个工程中，也就是说，窗体文件是可以被多个工程共享的一种资源，当然共享情况下，如果在一个工程中修改了窗体，势必会影响到所有使用该窗体的工程。

窗体文件只保存文本信息，如果窗体上显示了图片或图标等非文本信息，则该窗体在保存时会自动产生一个包含窗体全部图片信息的二进制数据文件，后缀为 **.frx** 并且不允许编辑。该文件不是必需的，但是如果原始的图片文件丢失了，则只要 **.frx** 文件在仍然可以显示出来。

3. 标准模块文件

以下两种情况我们会使用标准模块文件：

- 一个应用程序由多个窗体组成，这些窗体可能会共用一些数据或程序代码，我们将这些公用部分抽取出来，存放于标准模块中，提供给工程中的所有窗体共用；
- 有一些程序代码与具体的窗体界面元素是无关系的，为了管理的方便有时将它们分离成为子过程或函数，保存到标准模块文件中，同时也为将来的重用提供机会。

可以看出标准模块文件只是工程的可选项目 它不含有任何可见界面的描述 其后缀为 **.bas** 与窗体模块类似，一个工程可以有多个标准模块，一个标准模块也可以用于多个工程。

4. 类模块文件

类是对于一类事物的抽象描述，比如我们描述屏幕上的窗口会指出窗口高度、宽度、填充颜色等属性，还能执行最大化、最小化、关闭等操作，不论具体的窗口对象如何，都会具有这些特征属性和操作。类模块就是用于定义某种对象特征属性和操作的模块，但与窗口对象不同的是，类模块中的对象类在计算机中不表现为可见界面，只是逻辑上的定义，比如学校班级、社会保险账户等。要编写好类模块，必须了解面向对象的基本概念和编程思想。

与标准模块一样，只有在适当的情况下，我们才会编写类模块，类模块文件后缀为 **.cls**

需要在一个工程中增加某个资源时，执行【工程】菜单中的对应菜单项即可，见图 1-11。增加的资源类型不同，则保存成文件其后缀就不同。

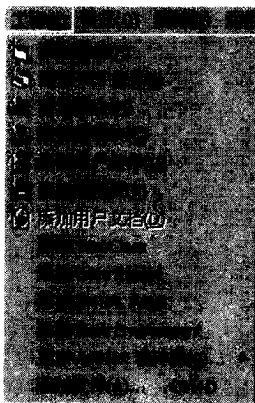


图 1-11 “工程”菜单

每个工程都会包含一堆文件，全部放在一起容易引起文件管理的混乱，因此建议编写程序时为每个不同的应用程序创建自己的文件夹，在保存文件时注意将工程文件和其他组成文件都放在同一文件夹里。

本章小结

1. VB 的来历和用途。
2. VB 集成开发环境的主要组成部分有工程资源管理器窗口、对象和代码窗口、属性窗口、窗体布局窗口、对象浏览器窗口、立即窗口等，它们分别有不同的用途。
3. VB 的一个工程就是一个应用程序，工程中可以包含一个或多个窗体模块、标准模块、类模块等资源模块，它们分别保存为不同的文件。

习题

1. 选择题

- (1) 提供控件的窗口是 ()。
A. 对象窗口 B. 对象浏览器 C. 工具箱 D. 工具栏
 - (2) 以下窗口中 () 可用来在设计时修改窗体的缺省运行位置。
A. 属性窗口 B. 立即窗口
C. 窗体布局窗口 D. 工程资源管理器窗口
 - (3) 工程文件的扩展名是 ()。
A. vbg B. vbw C. frm D. vbp
2. 用 VB 编程的优点是什么？
 3. 哪一个窗口允许在设计时设置控件属性？哪一个窗口能够列出一个程序包含的所有文件？
 4. 什么是工程？什么是工程文件？
 5. 任何一个 VB 程序肯定都包含标准模块文件吗？

第 2 章 | 面向对象编程基础

可视化编程工具的共同特点就是基于事件驱动的面向对象的程序设计，虽然 VB6.0 还不是一个完全面向对象的程序设计语言，但从一开始接触，要使用到的各种图形界面元素（控件）就是以封装成对象的方式来提供的。只有先掌握了面向对象程序设计的基本原理和事件驱动编程的机制后，才能用对象思维的方式来读懂程序，读懂 VB。而后在快速了解 VB 开发步骤的基础上，初学者便能不费吹灰之力建立起一个简单的 Windows 应用程序。

2.1 对象和控件

为了更好地理解和使用 VB 提供的图形控件，先让我们了解一个重要的概念——对象。

2.1.1 对象的概念

1. 对象是什么

计算机程序究竟是做什么的？说简单点就是为了完成我们日常工作中的业务处理和管理。而程序设计使用的是计算机语言，是一个二进制的世界，如何能将现实生活中的事物和操作反映到计算机中呢？能不能让计算机世界的思维与活动更接近于人类的思维与活动呢？多年的探索和实践总结出来的就是面向对象（Object-Oriented）的方法。

在变量、数组、语句、函数、子过程等基础之上，我们引入了新的程序元素：对象和类。以对象观点看问题，数据和操作不再是分离的，而是封装于统一体中，属性保存数据，方法完成操作。这样，对象就具有较强的独立性和自治性，不仅符合客观事物的本质，而且具有很好的模块性，为软件重用奠定了坚实的基础。

将对象的共同特征抽取出来就是类，类是模板，而对象是以类为模板创建出来的具体的实例，类与对象就像模具与成品的关系。

举一个简单的例子：没有对象概念之前，为了实现“计数器”功能，程序要提供一个计数器变量，计数器增加时变量作加 1 操作，计数器减少时变量作减 1 操作，加减的同时还要关注是否超出计数器边界值，这样写出来的代码充其量只能用于当前的程序中，而且越多地使用计数器变量，程序就越难维护。利用对象原理后可以设计一个“计数器”类：它有一个属性“value”保存计数器的计数值，对外提供三个操作：“init()”置 value 为 0，“dec()”使 value 减 1，“inc()”使 value 增 1，操作中含有边界控制。这样其他程序就可以利用计数器类来创建一个计数器对象，使用该对象的属性和操作，而不必关心内部是怎么实现的。

VB 是一个以对象为基础的程序设计语言，处处都有对象的存在，下面列出几种常用的对象类型：

- 窗体：VB 工程中的每一个窗体都是一个独立的对象。
- 控件：窗体上的控件，比如命令按钮或文本框，它们都是对象。
- 数据库：数据库是对象，查询出来的记录集是对象，包含的字段也是对象。
- 外部程序提供的对象：一个 Word 文档或 Excel 中的图表都是对象。
- 自定义对象：将某些数据和代码组织成类，然后使用，比如上述的计数器对象。

可以使用对象浏览器显示工程和对象库中的有效的类，包括控件和自定义的类。

2. 给对象命名

在计算机虚拟世界里的对象并不像现实中有血有肉，它们是摸不着的，那么如果出现了多个窗体对象，使用时该如何区分呢？从图 2-1 中所示的“属性窗口”列表中的第一行，我们能发现一个特殊的项目“(名称)”并且每个窗体和控件都有该项目。这不是一般意义的属性，里面填写的值“frmMain”代表的是窗体对象的名称。

可以这样理解：窗体是一种类，而“frmMain”是以窗体类为模板自动创建的一个实例，诸如此类，我们还可以有“frmOne”、“frmTwo”等许多的窗体对象，以及“Text1”、“Text2”、“cmdCancel”等控件对象。

给对象取名就如同给变量取名一样，在同一作用域内必须保证唯一性，但不同的是窗体或控件对象在设计时 VB 会自动提供一个缺省的名称。不过建议在编写代码之前最好给对象确定一个有意义的固定的名称，否则以后再修改可能会影响到整个工程。另外有意义的名称可以提高可读性，比较一下“text1”、“text2”和“txtCustomer”、“txtAddress”。当然如果代码中根本不访问该控件，大可不必那么费心设计名字了，比如仅用来显示静态文本的标签控件，采用系统缺省名称即可。

给对象命名除了表示数据的含义以外，通常会带有一个前缀来表示对象的类型，前缀的使用一般有约定，窗体对象前缀为 frm，如 frmMain、frmLogin、frmFind 等，命令按钮前缀为 cmd，如 cmdOpen、cmdDelete、cmdExit 等，文本框前缀为 txt。本书的例子多数采用这种对象命名方法。

3. 对象的属性

生活中的事物我们都会这样去描述：某某人的“身高”和“体重”多少，某双鞋的“鞋号”多大，什么“颜色”等，这些描述都抽取了事物的特征，在 VB 中就是对象的属性，即描述对象静态特征的一个数据项，数据内容就是属性值。

比如每个窗体对象都有诸如位置、高宽、边框等静态特征，反映到程序代码中就是 Top、Left、Width、Height 等属性项，对于窗体上的控件也一样。在设计模式下，只要打开“属性窗口”就可以看到窗体或控件的属性项和属性值的列表，其中属性值能直接设置或修改。

图 2-1 所示的属性窗口列出了窗体“frmMain”的属性。其中我们修改了窗体背景色“BackColor”属性为“0000FF”(红色)，边框“BorderStyle”属性修改为“0”(无边框)，标题“Caption”属性为“窗体对象”。

程序中要访问或修改对象的属性必须通过对象名称来指定，比如在代码中要修改窗体的标题应该这样写：

```
frmMain.Caption = "欢迎你"
```

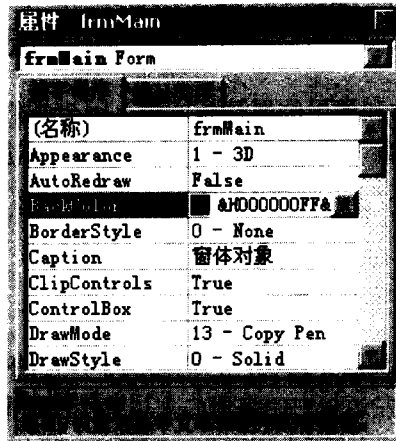


图 2-1 “对象属性”窗口

4. 对象的方法

对象为了达到某种目的必须执行的操作就是对象的方法，比如窗口要从一个位置移到另外一个位置就是通过调用窗体的“Move”方法来完成。方法其实就是该对象类内部定义的一个子过程或函数，可以有返回值，也可以没有，调用时只能通过对象来调用。比如移动窗体 frmMain 到屏幕左上角，就是使用该窗体对象的 Move 方法：

```
frmMain.Move 0, 0
```

方法能够影响属性的值，比如通过“Move”方法改变了属性“Top”和“Left”的值，实际上很多属性的改变都是利用方法来完成的，因为一个属性的改变可能会涉及对象方方面面的变化，简单的赋值语句难以实现，而在方法中可以完成更多的相关处理。

2.1.2 VB 中的控件

在 VB 中我们一般使用两类控件：

- 内部标准控件

这些控件包含在 VB 的可执行文件中，只要启动 VB 都会在工具箱中列出 比如 TextBox 和 CommandButton 等控件。它们是不能从工具箱中删除的。

- ActiveX 控件

一般是以扩展名为 .ocx 的文件独立存在，包括 VB 各个版本提供的 ActiveX 控件、第三方提供的以及自己制作的 ActiveX 控件。ActiveX 控件是对标准控件的扩充，其数量是不定的。它们不会自动在工具箱中显示，需要用到某个控件时就在工程中将该控件加载进来，该控件图标就会显示在工具箱中，如果卸载就会从工具箱中删除。

例如加载“ComDlg32.ocx”文件后，就可以使用通用对话框“CommonDialog”控件来选择文件了。因为 ActiveX 控件数量太多，本书不能一一列举，仅在部分章节介绍其中几种控件的用法。

不管是标准控件还是 ActiveX 控件，都是基于对象的概念来编程的，此外由于绝大多数控件都有可视化的界面，因此它们具有一些通用的属性和方法，下面介绍常用的属性和方法。

2.2 控件的属性和方法

本节主要介绍较为常用的和通用的属性和方法，因为它们是窗体和大部分控件所共有的，这样在后面讨论具体控件时不需再重复，而且可以进一步加深对面向对象编程的理解，对控件有较为全面的认识。

2.2.1 通用属性

1. 控件的位置和尺寸

所有可视化的控件都有下面四个属性，它们影响了控件的位置和大小：

- **Left**：返回或设置控件左边与它的容器的左边之间的距离，即控件相对于容器的 X 坐标。
- **Top**：返回或设置控件顶部和它的容器的顶边之间的距离，即控件相对于容器的 Y 坐标。
- **Width**：返回或设置控件的宽度。
- **Height**：返回或设置控件的高度。

在 **Left** 和 **Top** 属性都提到了容器，当控件包含在窗体或另一个控件之内，窗体和另一个控件称为该控件的容器，比如 **Frame** 控件和 **PictureBox** 控件都可以成为其他控件的容器。容器内的控件坐标使用的不是绝对的屏幕坐标，而是缺省以容器的左上角为坐标原点的相对坐标。随着在容器内移动控件，控件的 **Left** 和 **Top** 属性会随之改变。

以上几个属性都是 **Single** 类，在 **VB** 中缺省使用缇 (**Twip**) 为坐标单位，缇用来创建与屏幕分辨率无关的用户界面，一厘米为 567 缇。这些属性既能在“属性窗口”中设置它们的值，也能在代码中读取或修改。

在第 19 章会详细介绍 **VB** 中使用的坐标系统。

2. 控件的颜色和字体

- **BackColor**：返回或设置控件的背景颜色。
- **ForeColor**：返回或设置在控件里显示图片和文本的前景颜色。
- **Font**：返回或设置控件的文字字体。

不同控件在颜色属性的使用效果上会有所不同，比如在 **Label** 和 **Shape** 控件中，如果设置“**BackStyle**”属性值为 0 (即背景为透明模式)则会忽略“**BackColor**”属性。颜色属性在设计模式下可以利用“属性窗口”打开“颜色对话框”直接选择，在程序代码中可通常使用以下两种方法定义颜色值。

(1) 使用 **RGB** 函数

可以用 **RGB** 函数来指定任何颜色。**RGB** 函数的三个参数分别对应红、绿、蓝三种颜色，取值在 0~255 之间，0 表示亮度最低，而 255 表示亮度最高。每一种可视的颜色，都由这三种主要颜色组合产生。例如：

```
' 设定背景为黄色  
Form2.BackColor = RGB(255, 255, 0)
```

(2) 使用 VB 内部提供的颜色常量

VB 内部定义的颜色常量可以直接使用，共有 8 种主要颜色：黑色 (`vbBlack`)、红色 (`vbRed`)、绿色 (`vbGreen`)、黄色 (`vbYellow`)、蓝色 (`vbBlue`)、紫红色 (`vbMagenta`)、青色 (`vbCyan`) 和白色 (`vbWhite`)。例如：

· 设定背景为绿色

```
Form1.BackColor = vbGreen
```

有关颜色的更为详细的说明可参见第 19 章有关内容。

窗体和含有文字的控件可以设置文字的字体，同样在设计模式下可以通过“属性”窗口打开“字体”对话框直接选择，如图 2-2 所示。代码中也能通过这两种方式改变字体。

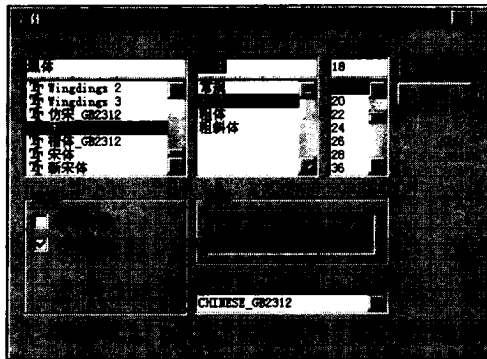


图 2-2 “字体”对话框

(3) 使用 FontName、FontSize、FontBold 等属性

控件中含有多个与字体有关的属性，分别代表不同的含义，它们是：FontName(字体名)、FontSize(字体号)、FontBold(是否加粗)、FontItalic(是否斜体)、FontUnderline(是否加下划线) 和 FontStrikethru(是否加删除线)。例如：

· 设定字体为 24 号宋体斜体

```
Form1.FontName = "黑体"
```

```
Form1.FontSize = 24
```

```
Form1.FontItalic = True
```

(4) 使用 Font 属性

该属性不是一个简单数据类型，而是对象类型，字体的各种定义由该对象的不同属性来控制。例如：

· 设定字体为 36 号黑体加粗并带下划线

```
Form1.Font.Name = "黑体"
```

```
Form1.Font.Size = 36
```

```
Form1.Font.Bold = True
```

```
Form1.Font.Underline = True
```

一个对象包含了另一个对象就是对象的嵌套，在现实中很普遍，比如一辆汽车是一个对象，它所包含的发动机可以是另一个对象。VB 及其他面向对象的程序设计都能支持这种对象嵌套的方式，实现方法是将嵌套对象定义为其他对象的一个属性，然后按层次可以一级一

级地访问到每一个被包含的对象。

3. 控件的值

一般控件都会有一个属性来存放该控件最重要或最常用的数据，此属性称为控件的值。但由于控件的特性不同，所以没有一个统一的属性名称可以适合所有控件，大多数会使用 `Caption`、`Text` 或 `Value` 这三种属性之一。

- `Caption`：窗体用此属性返回或设置窗体标题，`Frame`、`Label`、`CommandButton`、`OptionButton`、`CheckBox` 等控件使用该属性设置控件中的静态文本。
- `Text`：`Text`、`ListBox` 和 `ComboBox` 控件使用该属性设置控件的动态文本。
- `Value`：`CommandButton`、`CheckBox` 和 `OptionButton` 控件使用该属性记录控件的状态，`HScrollBar` 和 `VScrollBar` 控件使用该属性记录滚动条的当前位置。

注意：因为控件的值是最常用的属性，为了方便，这样的属性通常会作为控件的缺省属性，对于缺省属性在书写时可以省略。比如 `Form1.Caption` 可简写为 `Form1`、`Text1.Text` 可简写为 `Text1`。

4. 控件可见与可用

- `Enabled`：返回或设置一个控件是否可以使用，取值为布尔型。
- `Visible`：返回或设置一个控件是否可见，取值为布尔型。

若 `Enabled` 属性取值 `False` 时，则控件以灰色显示，表示控件无效，不能响应任何的操作，如果控件的 `Visible` 属性取值 `False` 时，在程序运行界面中看不到该控件。

图 2-3 的两个打印设置对话框中就含有这两种用法。如果“缩放”设置使用“标准”选项，则“输出纸张尺寸”和“比例”无效；如果“多页”设置使用“标准”选项，则与“海报”样式相关的选项就不可见。

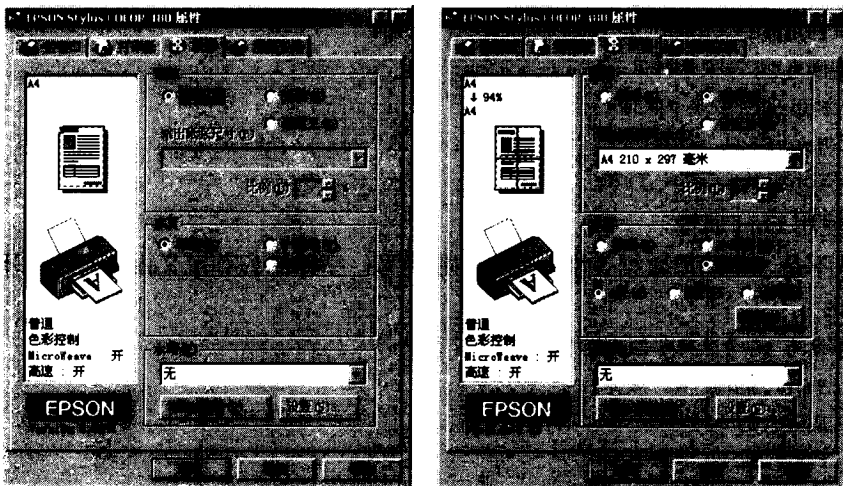


图 2-3 对象 `Enabled` 和 `Visible` 属性的外观示例

这两个属性经常用于控制用户界面的活动，不同的选项与不同的数据输入项目相关联，在程序代码中根据选项变化来控制输入项目的 `Enabled` 属性和 `Visible` 属性。

5. 控件输入的顺序

通常我们会使用鼠标的点击来完成整个界面的数据输入，输入的顺序似乎不必关心。有些情况下键盘操作效率更高，或很多人习惯操作键盘，所以程序设计还要考虑到键盘操作的方便性。采用键盘输入时常会利用【Tab】键在多个控件之间上下转移输入焦点，为了实现这一点，缺省情况下，在窗体上添加控件时 VB 会自动给每个控件分配一个【Tab】键顺序，即控件的 TabIndex 属性，通过设置该属性就能指定控件的输入顺序。

- **TabIndex**：返回或设置窗体中大部分控件的【Tab】键次序，取值为整型数字。
- **TabStop**：该值用来设置是否能使用【Tab】键将焦点从上一个控件转移到当前控件。取值为 True 时，表示该控件能被【Tab】键指定；取值为 False 时，表示利用【Tab】键不能将输入焦点移动到该控件上，即焦点会跳过该控件停到其下一个控件上。只有具备输入特性的控件有此属性。

通常界面的设计是一个反复的过程，当最后完成窗体布局后，应该检查窗体上所有控件的 TabIndex 属性是否符合键盘输入的顺序，一般输入顺序应该遵照从左至右、从上至下的习惯用法，这是判断程序易用性的一个指标。

2.2.2 通用方法

1. Move 方法

该方法可以移动控件，或改变控件的大小。使用移动方法比直接对属性进行赋值效率要高一些，比如在程序中可以这样调用 Move 方法：

• 移动窗体到屏幕左上角，设置窗体宽 600 (缇)，高 300 (缇)
Form1.Move 0,0,600,300

该方法有四个参数，依次是窗体左上角的 X 坐标、Y 坐标、窗体宽度和窗体高度，其中后三个参数是可选参数，可以省略。执行 Move 方法后其值会自动改变对象的四个属性 Left、Top、Width 和 Height。

2. SetFocus 方法

该方法可以使窗体或控件获取输入焦点。焦点是接收用户鼠标或键盘输入的关键，当对象具有焦点时，才可接收用户的输入。在 Microsoft Windows 环境中，可同时运行多个应用程序，但只有具有焦点的应用程序才有活动标题栏，才能接受用户输入；在有多个文本输入框控件的 VB 窗体中，只有具有焦点的输入框才有插入点光标。

比如在修改学生基本信息的窗口打开时，控制将插入点光标放在姓名输入框（控件 txtStudentName）中，应这样书写：txtStudentName.SetFocus

运行时使用【Tab】键可以移动输入焦点，在代码中则可以利用 SetFocus 方法设置焦点。注意当控件不可视或不可用时，即 Visible 属性或 Enabled 属性为 False 时，使用该方法会报告错误。此外有些不能接受焦点的控件不提供该方法，比如 Label 控件、Image 控件等。

2.3 控件的事件

前面介绍属性和方法只是为编程打一个基础，如何能让界面动起来呢，这就要靠编写事