

第 1 章 Visual Basic 入门

Visual Basic 是微软 (Microsoft) 公司开发的高级程序设计语言, 主要用于 Windows 操作系统下的软件开发, 目前普遍使用的版本之一是 6.0。它具有以下特点:

- ◆ 简单易学——这是 Visual Basic 继承自 BASIC 语言的特点, 不需要太多的计算机知识就可以开发出功能强大的软件, 非常适合于非计算机专业的人员使用。此外, Visual Basic 还支持面向对象技术 (Object Oriented, 简称为 OO), 这是目前软件开发采用的主要技术。因此, Visual Basic 也适合专业软件开发人员使用。
- ◆ 提供功能强大的集成开发环境——在 Visual Basic 的集成开发环境 (Integrated Development Environment, 简称为 IDE) 中, 可以方便、快捷地进行程序的创建、编辑、编译连接、调试、制作安装软件包等。为了便于编程人员查找编程信息, 该环境还提供了方便的联机帮助 (联机帮助需要单独安装)。
- ◆ 完全支持 Windows 应用程序的开发——Windows 是一个复杂的操作系统, 编写基于这个平台的程序需要了解 Windows 复杂的运行机制。在 Visual Basic 出现之前, 编制 Windows 应用程序需要非常多的专业知识; 而 Visual Basic 将 Windows 复杂的运行机制完全隐藏起来, 使用容易理解和掌握的对象来表示, 编程人员不需要了解 Windows 复杂的运行机制和编程知识, 可以很容易、快速地开发出标准的 Windows 应用程序。

要学好 Visual Basic 程序设计, 应了解三个方面的知识: 一是掌握语言本身的语法、语义及常用算法, 这是保障程序正确性的基础; 二是掌握集成开发环境的使用, 这可以保障程序开发的高效率; 三是熟悉常用对象的使用, 这可以保障程序具有强大的功能。

1.1 集成开发环境

使用 Visual Basic 编制程序的工作是在集成开发环境下完成的, 利用集成开发环境提供的各种工具, 可以快速开发出规范的、易于管理的应用程序, 因此, 必须熟练掌握集成开发环境的使用。集成开发环境提供的工具很多, 这些工具是开发程序时必不可少的。本节主要介绍集成开发环境下各种工具的用途和使用方法。

1.1.1 启动和退出 Visual Basic

1. Visual Basic 的启动

启动 Visual Basic 的操作步骤如下 (在不同的系统中路径可能稍有不同):

- (1) 在 Windows 操作系统的桌面上, 单击“开始”按钮。

- (2) 指向“程序”选项。
- (3) 指向“Microsoft Visual Basic 中文版”选项。
- (4) 单击“Microsoft Visual Basic 中文版”选项，便会启动 Visual Basic 的集成开发环境，启动完成后会显示一个工程对话框，让用户进行操作方式选择。

工程是 Visual Basic 中一个非常重要的概念（详细内容请参考 1.1.3 节）每个应用程序对应一个工程，每个工程有一个工程文件，用于组织和管理与工程有关的文件。图 1-1 是新建工程的界面，此外，该对话框还提供另外两个选项卡，可以按照不同方式打开已经存在的工程。

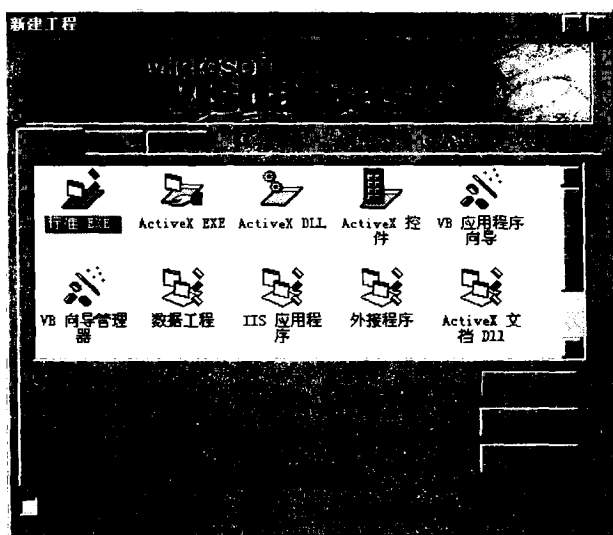


图 1-1 新建工程

Visual Basic 集成开发环境启动完成后，屏幕显示如图 1-2 所示。

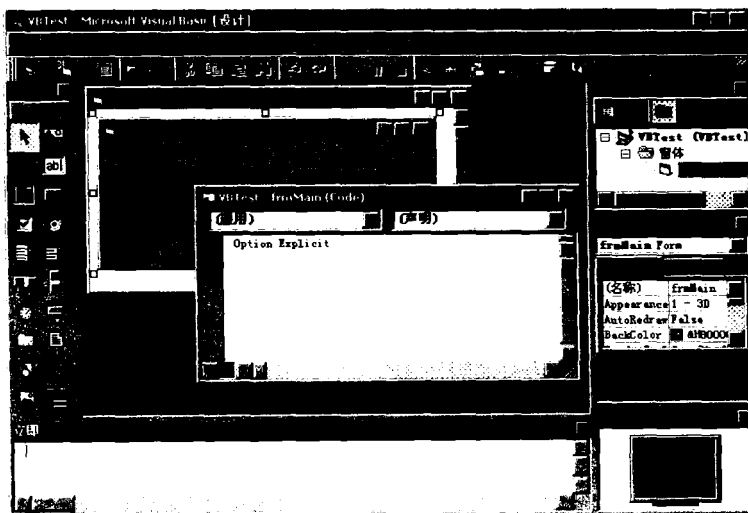


图 1-2 Visual Basic 集成开发环境

2. Visual Basic 的退出

Visual Basic 退出时就会关闭集成开发环境，退出前系统会检查目前打开的工程是否已经保存过。如果本次操作对工程做了修改但没有保存，系统会提示用户保存工程。如果该工程是新建立的，系统会显示“保存”对话框，要求将组成工程的所有文件（工程文件、窗体文件和模块文件等）保存到磁盘上。建议将每个工程的所有文件保存到一个单独的文件夹下，并且文件夹的名字用工程的名字来命名，这样便于对工程进行管理。

2 菜单栏和工具栏

Visual Basic 集成开发环境的菜单栏和工具栏如图 1-3 所示。



图 1-3 Visual Basic 集成开发环境的菜单栏和工具栏

1. 菜单栏

集成开发环境的菜单栏包含了 Visual Basic 中的大部分功能，图 1-3 的上半部分显示了 Visual Basic 集成开发环境的部分菜单。需要注意的是，Visual Basic 有 3 种版本：学习版、专业版和企业版，每种版本的菜单都有所不同。常用的菜单有：

- ◆ **文件(File)**——其选项内容较多，主要由 3 部分组成：一是工程的创建、打开、关闭、另存为等功能；二是打印设置，如窗体界面打印、源代码打印、打印机设置等；三是最近打开的工程列表，列表中的工程按使用时间由近及远排列。
- ◆ **编辑(Edit)**——包括常用的编辑选项，如查找、复制、粘贴、删除、撤销、重做等。由于集成开发环境是用于程序编制的，因此还包括一些与此有关的特殊选项，如代码行的缩进设置、显示过程、显示属性、书签操作、定位到语句、插入文件等。
- ◆ **视图(View)**——主要用于控制和管理集成开发环境中的子窗口及工具栏，如显示 / 隐藏控件工具箱、显示 / 隐藏属性窗口、设置工具栏、查看对象等。
- ◆ **工程(Project)**——用于对工程中的各种部件进行管理，如添加窗体、添加类、添加模块、删除部件、设置工程属性等。该菜单很重要，在规模较大的工程中使用频率很高。
- ◆ **格式(Format)**——在界面设计过程中，经常需要调整控件的大小、位置、间距、对齐方式等，使用“格式”菜单，可以很快捷地完成这些操作。要快速设计出美观的程序界面，熟练掌握“格式”菜单的使用是非常必要的。
- ◆ **调试(Debug)**——程序的编制不是一蹴而就的，对于规模较大的程序更是如此。编制程序时首先要保证程序的正确性，然后还要保证开发的高效性。编写程序时，出现错误是很正常的现象，关键是如何快速找到这些错误并进行改正。要快速找到程序中的错误，除了编程人员的经验外，熟练掌握软件开发环境提供的调试工具也是非常重要的。“调试”菜单提供多种调试工具，借助这些调试工具，编程人员可以实现高效的程序调试。
- ◆ **运行(Run)**——主要用于执行应用程序，检验其功能及正确性。通过“运行”菜单，可以将目前的使用模式改变为运行模式或中断模式，一般要与“调试”菜单配合使用。

- ◆ 工具(Tools)——主要包括 3 部分功能：对集成开发环境进行定制，向程序代码中添加过程和属性，激活菜单编辑器。
- ◆ 外接程序 Add-Ins)——包含两部分功能：一是打开 Visual Basic 环境下的数据库管理器；二是管理外部程序，这些外部程序对 Visual Basic 编程有一定的辅助作用，如 Windows API（应用程序编程接口）的原型说明。

2. 工具栏

在 Visual Basic 集成开发环境中，有些功能使用很频繁，例如保存文件、复制、粘贴等，为了方便用户使用这些功能，将这些功能按照类别放置在标准工具栏内，其使用方法和 Office 系列软件的工具栏相似。标准工具栏外观可参考图 1-3 的下半部分，在此只显示了其中一部分。

除了图 1-3 所描述的标准工具栏外，集成开发环境还提供其他类别的工具栏，如程序调试工具栏。用户可以根据需要对这些工具栏进行定制，包括是否显示某种工具栏以及显示位置，操作方法是：将鼠标指针指向某个工具栏，单击右键，然后从弹出的快捷菜单中选择需要的工具栏，使其显示或隐藏。

1.1.3 工程资源管理器窗口

工程是 Visual Basic 中的重要概念。使用 Visual Basic 开发的应用程序可能规模比较大，包含多个窗体、模块、自定义的类、其他文档等多个逻辑部件，每个部件对应一个文件。为了便于对这些部件进行管理，提出了工程的概念。工程资源管理器是集成开发环境中对工程及其所属文件进行管理的工具，它以树形结构图的方式将工程中所有文件按照类别组织在不同的文件夹下，查看时非常方便。在该窗口下，可以设置工程属性、快速切换要操作的文件，也可以快捷地添加和删除任何类型的文件。

由于工程的规模和功能不同，因此工程资源管理器的组成情况可能也不一样，图 1-4 显示了一个工程的组成。在进行界面设计或代码设计时，如果希望有更大的空间来进行上述操作，可以暂时将工程资源管理器窗口关闭，需要使用时可以利用“视图”菜单中的“工程资源管理器”选项打开。

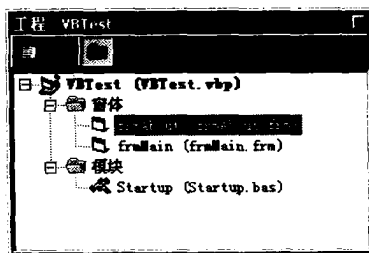


图 1-4 工程资源管理器窗口

工程资源管理器可以同时管理多个工程，这些工程组成工程组。一般情况下，工程之间存在某种逻辑上的联系。显示在工程资源管理器窗口中的工程可能有多个，但是只能有一个工程是活动的，也就是当运行程序时，运行的是处于活动状态的工程。在多个工程同时存在的情况下，编程人员可以选择运行某个工程。但在运行指定工程之前，需要先把该工程设置

为活动状态，操作步骤是：将鼠标指针移到工程名称上，单击右键，选择快捷菜单中的“设置为启动”选项。

在工程中添加、删除和保存部件的方法如下：

- ◆ 添加部件——鼠标指针指向工程名称，然后单击右键，在快捷菜单中定位到“添加”选项，显示出二级菜单，然后选择所要添加的部件类型，此时会显示“添加”对话框。“添加”对话框中包括两个选项卡：“新建”选项卡用于创建新的部件，“现存”选项卡用于将已经存在的部件添加到指定工程中，该功能可以实现多个工程间部件的共享。
- ◆ 删除部件——首先展开要删除的部件所在的工程，然后在要删除的部件项上单击右键，选择“删除”选项即可。注意，部件被删除后，只是它和工程之间不再有任何关系，但部件对应的文件并没有被删除，在需要时可以重新添加到工程中。
- ◆ 保存部件——当某些部件做过修改后，应该对修改的内容进行保存操作。保存内容的方式有两种：一是保存，用于将从未保存过的部件以文件的形式保存起来，这时会出现对话框，或者将已存在相应文件的部件再次保存，这时不会出现对话框；二是另存为，指将该部件保存到另外一个新文件中，利用这种方式可以将部件保存成多个文件，供其他工程使用。

4 窗体窗口

在 Visual Basic 程序设计中，首先要做的工作是界面设计。界面设计在一个窗口中进行，该窗口被称为窗体窗口，如图 1-5 所示。界面设计的主要步骤是：将所需的控件在窗体窗口中的窗体 Form 表面上画出来，然后调整其位置、大小、颜色、标题等，在调整过程中可以用手工方式调整某些项，也可以通过设置相关属性来调整，还可以利用“格式”菜单进行调整。

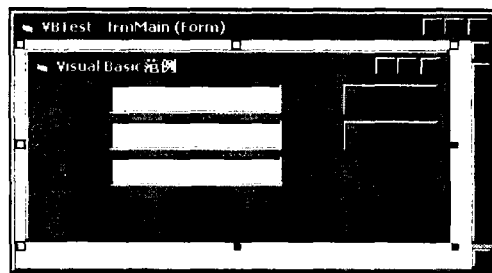


图 1-5 窗体窗口

每个窗体窗口只能容纳一个窗体；如果工程中有多个窗体，则会存在多个窗体窗口。需要对某个窗体进行界面设计时，可以在工程资源管理器中双击该窗体名称，这样包含该窗体的窗体窗口就会显示出来。窗体的界面设计完成后，应该关闭对应的窗体窗口，以保持集成开发环境的简洁。关于界面设计的内容，后面将进行详细讲解。

1.1.5 控件工具箱

标准的 Windows 应用程序以窗口为基本组织单位，在窗口表面有多种通用部件，如文本

框、下拉列表、文本标签等。在 Visual Basic 中，这些部件实际上称为控件。在进行窗体的界面设计时，主要工作就是把需要的控件按照功能要求放到合适的位置，并为之设置合适的属性。为了方便用户使用这些控件，集成开发环境提供一个窗口，专门容纳控件，称为控件工具箱 如图 1-6 所示。

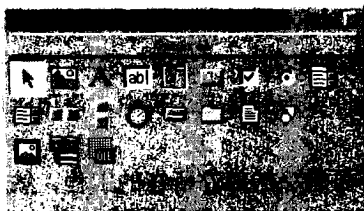


图 1-6 控件工具箱

把控件放置到窗体表面是一个很简单的过程，操作方法有两种：第一种是首先单击控件工具箱上的某个控件，然后使用鼠标拖动的方式将该控件在窗体窗口中画出来；第二种是双击控件工具箱上的某个控件，它就会自动出现在窗体中央，其大小是默认的。这两种方式的结果都是在窗体上创建一个特定类别的具体对象。

应用程序运行过程中，大部分控件都是可见的，并且这些控件的大小、位置及相关属性都可以在设计过程中改变。有些控件在界面设计时是可见的，而在程序运行时是不可见的，如 Timer 控件。这些控件在程序运行时起作用，但不会显示在窗口表面。因此，在进行界面设计时，这些控件的大小和位置是不重要的。

在编制较复杂的应用程序时，用户可能会发现它们需要的某些控件并没有包含在控件工具箱中，这时就需要手工地将控件添加到控件工具箱中，然后再使用。向工具箱添加控件的操作步骤如下：

- (1) 将鼠标指向控件窗口的任意位置，单击右键，显示快捷菜单。
- (2) 单击“部件”选项。此时，屏幕显示如图 1-7 所示的“部件”对话框，在列表框中显示可以使用的控件列表。
- (3) 将需添加的各个控件前的复选框选中，然后单击“确定”按钮，选定的控件就会出现现在工具箱中。

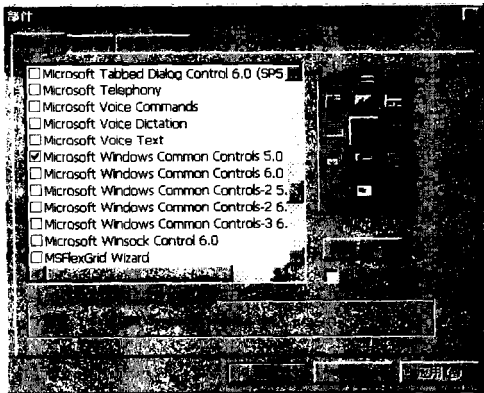


图 1-7 “部件”对话框

1.1.6 属性窗口

在 Visual Basic 中，对象的数据称为属性。属性包含两个内容：属性名称和属性值。例如，每个窗体都有一个标题，标题内容取决于窗体对象（Form）的 Caption 属性，如果其值为“VB 范例”，则窗体的标题会显示为“VB 范例”。如果将窗体的 Caption 属性的值设置为“我的第一件作品”，窗体的标题就会变成“我的第一件作品”。在该例中，Caption 是属性名称，“VB 范例”或“我的第一件作品”是属性值。

在 1.1.5 节中，我们了解到 Visual Basic 中有两种类型的对象：一种是界面设计阶段可以使用的控件，另一种是只能在代码中使用的引用对象。这两种对象都有属性，但设置方式不同：大部分控件对象的属性都可以在界面设计阶段设置，小部分属性只能在代码中设置或读取；引用对象的属性只能在代码中读取或改变。

在进行界面设计时，设置控件对象的属性是很重要的步骤。一般情况下每类控件都有多种属性，属性的设置也就要重复多次。属性设置的工作在属性窗口中进行，如图 1-8 所示。如果属性窗口没有显示出来，可通过以下步骤将打开属性窗口：打开“视图”菜单，选择“属性窗口”命令。

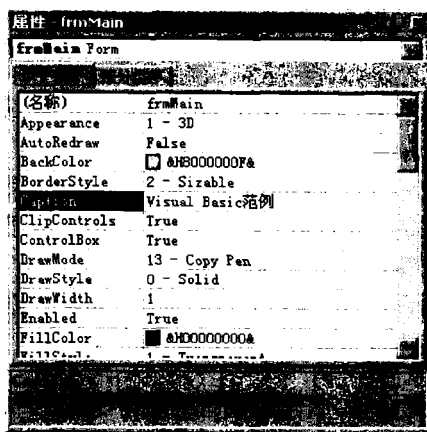


图 1-8 按属性名称的字母顺序组织的属性窗口

属性窗口由 3 部分组成：最上方是窗体及控件名称的下拉列表，列表中每项由两部分组成，左侧是控件或窗体的名称，右侧是控件或窗体的类型；中间部分是一个两列的属性表，左列是属性名称，右列是属性值；最下方是属性表中被选中属性的介绍。属性窗口所包含的内容是选中控件在设计阶段可用的属性，而另一部分只能在运行阶段使用的属性不会出现在属性窗口中。

控件的属性可能很多。因此，为便于使用，属性列表将所有属性按两种方式组织：一种按字母顺序排列，如图 1-8 所示；另一种按属性的类别组织，如图 1-9 所示。

窗体上可能有多个控件，如何设置指定控件的属性呢？有两种方法：第一种是将鼠标指针指向某个控件，单击右键，然后在快捷菜单中选择“属性”命令，属性窗口上方的下拉列表中会显示该控件的名称及类别，同时其属性显示在属性表格中；第二种是在属性窗口上方的下拉列表中选择相应的项，该项对应的控件或窗体的属性也会显示在属性表格中。

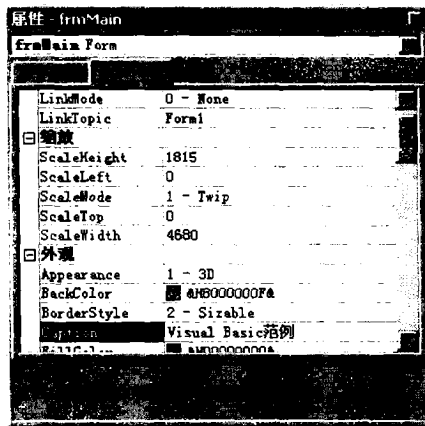


图 1-9 按属性类别组织的属性窗口

在界面设计中，只要向窗体中添加一个控件，则该控件的属性（在设计阶段可用的部分）就会显示在属性窗口中，并且每个属性都有一个系统事先规定的属性值，这种属性值称为默认值。一般情况下，大部分属性值采用默认值就可以了，但一些关键的属性值需要适当改变，例如窗体的名称、命令按钮上的文字等。要改变属性值，有三种方法：

- ◆ 第一种——直接输入设置的值：用鼠标左键单击左列的属性名称，在右列中输入其值，如窗口标题 Caption。
- ◆ 第二种——通过下拉列表框选择值：单击属性名称，如果右列出现向下的三角形，表示不用手工输入属性值，只需单击该三角形，显示下拉列表框，在列表给出的几个值中选择其一即可，如窗口状态 WindowState。
- ◆ 第三种——通过对话框设置：单击属性名称后，如果右边出现带省略号的按钮，表示该属性的值在对话框内进行设置，如字体 Font。

在设置窗体或者控件的属性时，首先应设置的是窗体或控件的名称，即 Name 属性。原因很简单，因为一个工程中可以包含多个窗体，这些窗体是通过 Name 属性来区分的。同样，一个窗体中可以包含多个相同类别的控件，例如文本框，这些文本框也是通过 Name 属性来区分的。

在添加一个窗体或控件时，它们都有一个系统默认的名称。例如，窗体名称一般是 Form1, Form2, Form3 等，命令按钮名称一般是 Command1, Command2, Command3 等。根据这些名称，我们只能推测它们的类别，但不能了解它们的功能。因此，最好不要使用这种系统默认的名称，应该按照功能的要求立即进行修改。在设置窗体或控件的 Name 属性时要遵循见名知义的原则，即根据 Name 属性值，可以很清晰地判断它的类别及功能。

Name 属性值一般由两部分组成，即类别前缀和功能描述。例如 cmdAdd, cmd 是 CommandButton 的缩写表示“命令按钮”，Add 的英文含义是“加”表示“加”，所以 cmdAdd 的含义就是“加按钮”。表 1-1 给出了常用控件的前缀及范例。

表 1-1 窗体及常用控件的前缀列表

中文名称	类别名称	前缀	范例	范例含义
窗体	Form	frm	frmMain	主窗体
命令按钮	CommandButton	cmd	cmdExit	退出按钮
文本框	TextBox	txt	txtName	姓名输入文本框
标签	Label	lbl	lblMessage	显示消息的标签
单选按钮	OptionButton	opt	optFontSize	字体大小单选按钮
复选框	CheckBox	chk	chkFontBold	粗体复选框
列表框	ListBox	lst	lstBook	书名列表
组合框	ComboBox	cbo	cboCity	城市组合框
图片框	PictureBox	pic	picPrintPreview	打印预览图片框
框架	Frame	fra	fraFontSetting	字体设置框架
定时器	Timer	tmr	tmrSendData	发送数据定时器
图像	Image	img	imgCover	图书封面的图像
驱动器	DriveListBox	drv	drvLocal	本机驱动器列表
目录列表框	DirListBox	dir	dirSource	源文件目录列表
文件列表框	FileListBox	fil	filSource	源文件列表
形状	Shape	shp	shpNote	注释信息的外框形状
直线	Line	lin	linSeprate	分隔线
水平滚动条	HScrollBar	hsb	hsbVolume	音量滚动条
垂直滚动条	VScrollBar	vsb	vsbRate	速率滚动条
菜单	Menu	mnu	mnuPaste	粘贴菜单项

1.1.7 代码窗口

代码窗口主要用于代码的编辑。工程中的每个部件都称为模块，每个模块有一个对应的代码窗口。图 1-10 是窗体 frmMain 对应的代码窗口，它由 3 部分组成：

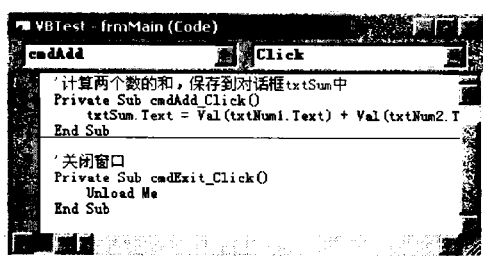


图 1-10 代码窗口

- ◆ 对象和事件列表——左侧是窗体及控件列表，包含窗体本身和窗体中的所有控件。对于控件，该表中包含的是控件的名称（由 Name 属性表示）；对于窗体，则显示的不是窗体的名称，而是固定的关键字 Form。右侧是事件列表，当从左侧列表选择一个对象时，会自动选中该对象的最常用事件，并自动生成事件过程。例如，在左侧列表选中 Form 时，右侧列表中会自动选中 Load 事件，代码编辑区域会显示事件处理过程 Form_Load。在右侧事件列表中，事件名称的字体有两种显示方式，以粗体显示的表

示其对应的事件过程已经具有了代码，以普通体显示的表示该事件过程不存在。

- ◆ 代码编辑区域——代码编辑主要在此区域内进行。该区域的内容由两部分组成：一部分是变量声明，用于声明全局或模块级的变量；另一部分是各种过程，包括普通过程、事件过程和函数过程。为了便于代码的编辑和检查，最好过程和过程之间添加几个空行，在每个过程前面加上适当的注释。
- ◆ 视图切换按钮——在一个模块中可能包含很多个函数和过程的代码，这些代码都存放在该模块对应的文件中。在编辑时，这些函数和过程的代码会按照先后顺序显示，有时需要对照不同的函数和过程进行代码编辑，有时又只想对某一个函数或过程进行编辑，而不受其他函数和过程的干扰。为此，代码窗口提供两种视图：模块视图和过程视图。在模块视图方式下 代码编辑区域会显示很多过程和函数 适用于第一种情况；在过程视图方式下，代码编辑区域只显示一个过程或函数，适用于第二种情况。两种视图的切换可通过代码窗口左下角的视图切换按钮进行，左侧为过程视图按钮，右侧为模块视图按钮。

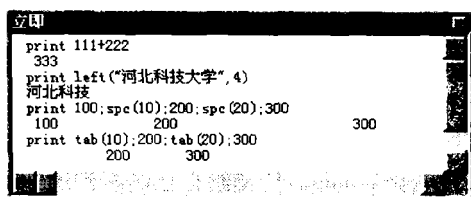
代码窗口具有以下特点：

- ◆ 通过上方的控件名称和事件名称，可以快速定位一个事件响应过程。
- ◆ 在代码编辑区中，每两个过程之间用一条线分隔开，可以使编程人员很容易了解整个程序代码的组成，避免由于代码量较大带来的代码之间的混淆问题。
- ◆ 功能完善的文本编辑功能，可以简单、高效地对代码进行复制、删除、移动以及其他操作。
- ◆ 在输入代码的过程中，代码窗口中的内容会自动按语法规则进行缩进处理，这样便于分析程序结构，便于进行调试和纠错。
- ◆ 在输入保留字时，会自动变化大小写，增加可读性。
- ◆ 在移动插入点时，工具栏上显示目前插入点的行列位置。

1.1.8 立即窗口

立即窗口是集成开发环境提供的一个辅助工具，外观如图 1-11 所示。它有两个基本功能：

- ◆ 程序调试——在程序运行过程中，编程人员可以将一些辅助性的调试信息在立即窗口中输出，以跟踪程序的执行过程，也可以在中断状态下，直接查看有关变量的内容。
- ◆ 验证语句、函数和表达式——一般情况下，计算机程序必须在代码编辑完成后开始运行，但有时编程人员需要先了解一些语句、函数的功能，或者先验证一下表达式的计算结果，而不想花大量时间将这些内容放在一个程序中运行。利用立即窗口就可以解决这个问题，在立即窗口中输入语句、函数或表达式，在确认没有语法错误的情况下按回车键，输入的内容会立即执行。



```
立即  
print 111+222  
333  
print left("河北科技大学", 4)  
河北科技  
print 100; spc(10); 200; spc(20); 300  
100          200          300  
print tab(10); 200; tab(20); 300  
          200          300
```

图 1-11 立即窗口

1.2 对象及其操作

Windows是事件驱动的操作系统，Windows会随时检测所发生的事件。当检测到发生某个事件后，Windows会将该事件的相关数据以消息的形式发送给相应的程序，由程序按照功能要求进行处理。为了能够在Windows环境下运行，应用程序必须能够识别Windows发送给自己的消息并进行必要的处理。例如，应用程序接收到一个鼠标单击消息，消息中包含的内容有：单击的是左键、右键还是中间键，单击时鼠标指针的位置，单击的同时是否按下了键盘上的功能键。应用程序接收到消息后，可以根据需要进行处理。例如，显示一个对话框，或者调用另一个应用程序，也可以不进行任何处理，由Windows进行默认的处理。

Visual Basic 是基于对象的程序设计语言。利用 Visual Basic 编程的主要工作是：将具有特定功能的对象引入程序中，并设置相应的属性，然后编制特定的事件处理过程。本节主要介绍对象的概念、窗体和文本框、标签和命令按钮等 3 种常用控件对象，其他控件将在后续章节中陆续介绍。

1.2.1 对象

1. 对象的概念

对象是程序运行时的一个相对独立的实体，包含数据和方法两部分内容。此外，大部分对象还可以接收到运行环境中的相关事件。接收到事件后，要进行相应的处理，处理过程称为事件处理。下面以命令按钮 CommandButton 这种对象为例，进一步说明数据、方法、事件和事件处理等概念。

- ◆ 数据——在 Visual Basic 中称为属性，用于描述该实体的特性，以数值、字符串或其他形式表示，一个对象可以具有多种属性。例如，窗体中的一个命令按钮，具有特定的位置、大小、标题、字体等特性，每一种特性都由不同的属性表示，位置由 Left 和 Top 两个属性来决定，如果希望它的位置是 (600, 1000) 可以将其 Left 属性的值设置为 600，Top 属性的值设置为 1000。
- ◆ 方法——指对象可以主动发出的动作，用于实现某种特定的功能，一个对象可以执行多种方法。例如，要把“退出”命令按钮 cmdExit (“cmdExit” 是命令按钮的名称，由 Name 属性标识，“退出”是命令按钮的标题，由 Caption 属性标识) 移动到 (1000, 1000) 处，除通过修改 Left 和 Top 属性外，还可以调用 Move 方法：

```
cmdExit.Move 1000,1000
```

- ◆ 事件——指外界施加到对象上的某种动作，对象是动作的被动承受者，一个对象可以承受多种事件。例如，当用户单击“退出”命令按钮时，Windows 会检测到一个鼠标单击的事件，该事件会以消息的形式传递给命令按钮，Visual Basic 会将该事件解释为名称为 Click 的事件。
- ◆ 事件处理——对象接收到 Windows 传递给自己的事件通知后，可以根据需要进行处理或不处理。在 Visual Basic 中，将进行处理的代码称为事件处理，在代码中体现为一个单独的过程，称为事件过程。在编制 Visual Basic 代码时，主要工作是针对特定事件编写相应的事件过程。例如，命令按钮 cmdExit 接收到鼠标单击事件后，希望关闭窗口体，则在代码窗口中的处理过程是：
 - 在左侧对象列表中选择 cmdExit(这是命令按钮的名称)
 - 在右侧事件列表中选择 Click(这是鼠标单击事件的名称)然后编辑区域会出现一个名为 cmdExit_Click 的事件过程。
 - 在该事件过程中添加一行代码：Unload Me。

2. 对象的使用

在 Visual Basic 中，对象分为四种：第一种是在界面设计中涉及到的对象，如窗体、文本框、命令按钮等控件；第二种是不能在界面设计时使用、只能通过特殊的过程引入并在代码中动态创建和使用的引用对象，如数据库连接对象、文件系统对象；第三种是表示系统中某种资源的对象，如打印机、屏幕；第四种是错误对象。这四种对象大都具有特定的属性和方法，一般情况下只有窗体和控件可以响应特定的事件，其他类型对象不响应事件。

- ◆ 窗体对象——Visual Basic 工程的主要组成部分是窗体，在创建了工程后，一般情况下都要先将若干窗体添加到工程中，然后再进行其他设置。创建窗体的方法是：打开“工程”菜单，选择“添加窗体”命令，系统会显示一个如图 1-12 所示的对话框。该对话框包含两种选项卡：
 - “新建”选项卡——提供多种窗体类型。从这些窗体类型中选择一种，就会有一个窗体加到当前工程；选择不同类型的窗体，可能会有不同的向导来引导编程人员进行定制。
 - “现存”选项卡——有时，Visual Basic 工程中所需窗体对应的文件已经存在，但没有加入到工程资源管理器中进行统一管理，对这种已经存在的窗体文件，可以通过该选项卡来定位这些窗体文件，然后将其添加到工程中。
- ◆ 控件对象——创建控件对象的方法很简单，可参考 1.1.5 节的控件工具箱。
- ◆ 引用对象的创建——控件工具箱不包含这种类型的对象，创建步骤是：

- (1) 打开“工程”菜单，选择“引用”命令，出现如图 1-13 所示的“引用”对话框。从列表中选择要创建的对象类型，如 Microsoft PowerPoint 9.0 Object Library 然后单击“确定”按钮，这一步骤的作用是在程序中注册这种类型。

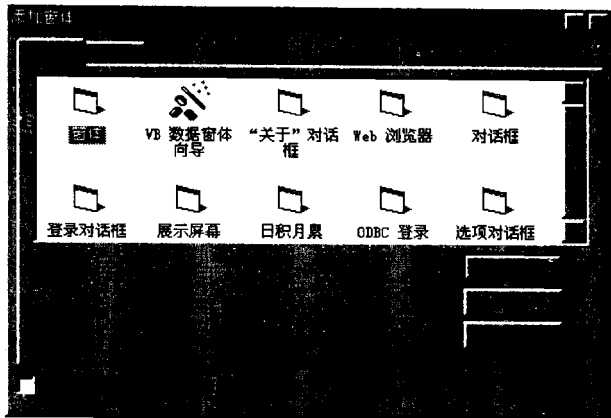


图 1-12 “添加窗体”对话框

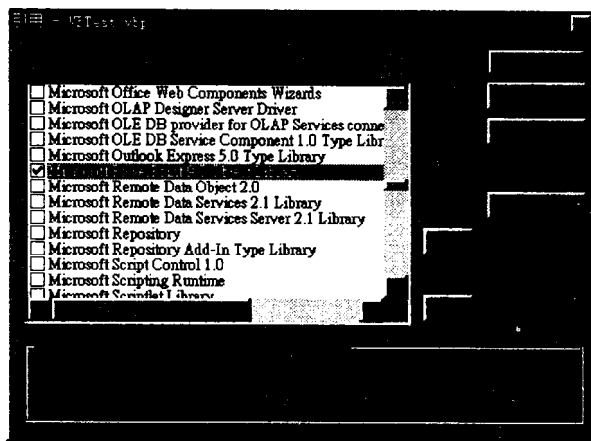


图 1-13 “引用”对话框

(2) 在代码中按如下格式创建一个该类型的变量：

```
Dim 对象名称 As New 引用类型名称
```

在本例中是：

```
Dim oPPT As New PowerPoint
```

创建该变量后，就可以使用该对象了（PowerPoint是 Microsoft PowerPoint 9.0 Object Library 的类型名称）。

- ◆ 资源类对象的使用——除上述两种对象外，还存在像打印机、屏幕、剪贴板等对象，这些对象的每个都和系统中的某种资源对应，数量有限，因此使用时不要长期占用，以保证其他程序的使用。这些对象不需要创建，可以直接使用，在程序代码中通过对象名称就可以使用相应的资源。需要注意的是，这些对象的名称是固定的，如 Printer（打印机）、Screen（屏幕）、Clipboard（剪贴板）等。
- ◆ 错误对象——这种对象只有一个，名称是固定的 Err，它的主要作用是进行程序调试。该对象有两个重要的属性：Number和 Description 前者表示出错时的错误编号，后者

表示错误的描述信息。如果没有出现错误，**Number** 属性值为 0。

- ◆ **对象命名**——在程序代码中，不同的对象通过不同的对象名称来区分，因此，每一个对象都必须有一个惟一的、区别于其他对象的名称。不同类型对象的命名方法不同：窗体和控件对象的名称就是其 **Name** 属性的值，通过设置 **Name** 属性为不同的值来命名；引用对象的名称在创建该类型变量时确定；资源类对象和错误对象不用命名，可直接使用。

1.2.2 窗体

在 **Visual Basic** 程序设计中，首先要进行的是界面设计：在工程中添加窗体，然后在窗体上安置若干控件，因此必须掌握窗体的使用。在 **Visual Basic** 工程中，每个窗体是工程中的一个模块，它单独保存在一个扩展名为 **frm** 的文件中。该文件包含两部分内容：一部分是该窗体本身的属性设置信息及包含的所有控件信息，另一部分是和该窗体有关的代码。

1. 属性

对象包含属性和方法两部分内容，属性指对象中包含的数据。在使用时有一个需要注意的问题：第一，对于窗体和控件类对象，大部分属性既可以在设计时使用，也可以在程序运行时使用；第二，对于所有对象，大部分属性的值是可以修改的，但有一小部分属性的值是只能读取不能修改的。窗体是 **Visual Basic** 中的基本对象，属性较多，下面只介绍最常用的一些属性，有些属性也适用于控件。

- ◆ **Name**——表示窗体或控件的名称。在同一个工程中，不允许有两个窗体同名。类似地，在同一个窗体中，不允许两个控件同名。该属性主要用于在代码中标识控件，并且在程序运行时不允许修改其值。注意，对于窗体，其名称只是在调用窗体时使用，而在窗体本身的代码中，一般用关键字 **Me** 表示窗体本身。向工程中添加窗体时，窗体名称的默认值是 **FormX**，**X** 是一个自动增加的数字。系统命名的窗体名称不能反映窗体的功能，因此，一般都要对窗体重新命名以反映其功能。例如，将窗体名称改变为 **frmAbout**，其可读性要明显好于 **Form1**。
- ◆ **Caption**——表示窗体标题栏的文字。该属性的值可以在设计时修改，也可以在程序运行时修改。例如，在窗体本身的 **Load** 事件中，可以设置窗体标题为“实例 1”，相关代码是：

```
Me.Caption = "实例 1"
```

该属性的默认值和 **Name** 属性的默认值相同。

- ◆ **WindowState**——设置窗体在运行时的显示方式，可以使用的值包含 0, 1, 2 三种，分别表示正常显示的窗体、最小化显示的窗体、最大化显示的窗体，默认值是 0。
- ◆ **StartPosition**——设置窗体启动时的显示位置，该属性的值在程序运行时不能修改。合法的属性值是 0~3，分别表示启动位置取决于设计阶段的位置、启动时在父窗口的中心、启动时在屏幕中心、启动时由 **Windows** 将其显示在默认位置，默认值是 0。
- ◆ **MaxButton** 与 **MinButton**——设置窗体右上角的最大化按钮和最小化按钮是否可以使用，默认值是 **True**（**True** 是一个逻辑常量，表示真），表示可用。如果希望在运行时

窗体不能被最大化 则可以设置 `MaxButton` 属性的值为 `False` (另一个逻辑常量 表示假)。

- ◆ **BorderStyle**——设置窗体的边框风格, 指定在程序运行时窗体尺寸是否可以改变。该属性在运行时是只读的, 合法属性值为 0~5, 每个值的意义如下:
 - 0——没有边框
 - 1——单边框, 窗体尺寸不可改变, 只能使用最大化和最小化按钮改变窗体大小
 - 2——可以通过拖动边框来改变窗体大小, 这是默认值
 - 3——对话框, 窗体尺寸固定, 没有最大化和最小化按钮, 不显示在任务栏上
 - 4——固定的工具窗体风格, 标题栏字体变小, 任务栏上不显示相应按钮
 - 5——不固定的工具窗体风格, 标题栏字体变小, 任务栏上不显示相应按钮
- ◆ **Picture**——指定窗体的背景图片路径, 可以在设计时指定, 也可以在代码中使用 `LoadPicture` 方法加载图片。
- ◆ **ControlBox**——指定是否显示控制按钮, 即右上角的按钮; 默认为 `True`, 可以显示。
- ◆ **Moveable**——指定窗体在显示时是否可以移动位置; 默认值是 `True`, 可以移动。

2. 方法

方法指对象本身提供的一些功能, 每个功能都有特定的名称, 方法的使用格式是:

对象名称. 方法名称 参数列表

方法一般用在事件处理过程中, 用来响应某个事件。

- ◆ **Cls**——清除程序运行时产生在窗体表面的图形和文字, 对于在设计阶段设置的控件和背景图片没有影响。在窗体上画图或用 `Print` 输出后, 如果需要清除这些内容, 可以使用该方法, 其他情况下很少使用。
- ◆ **Hide**——隐藏窗体, 但该窗体仍然存在于内存中, 需要时可以利用 `Show` 方法显示。窗体隐藏后, 窗体上的控件不能再访问, 大部分控件也不再响应事件, 但后台的通信及时钟等仍然是活动的。使用时, 一般通过 `Me.Hide` 语句实现。
- ◆ **Move**——移动窗体的位置, 同时可以改变大小。使用格式是:
窗体名称 .Move 左上角横坐标, 左上角纵坐标, 宽度, 高度
4个参数中 只有第一个是必须的 其余3个可以省略, 计量单位为默认的缇 (Twips)。
- ◆ **Print**——在窗体表面输出表达式的值。表达式的值可以通过多种控件输出, 因此尽量不要使用该方法。
- ◆ **Refresh**——强制刷新窗体中的显示内容。通常情况下, 窗体显示内容的刷新是自动完成的, 但有些情况下, 例如在一个比较大的循环中, 每执行一次循环都希望进度条能够进行相应的变化, 这时, 进度条的变化是不能自动显示出来的, 只能等循环处理完成后才会被自动刷新, 这时可以使用 `Refresh` 方法强制刷新显示, 从而可以看到一个动态的进度条的变化过程。使用该方法会导致程序运行速度变慢。
- ◆ **Show**——显示窗体。窗体加载后存在于内存中, 但不会自动显示在屏幕上, 需要使用 `Show` 方法将窗体画在屏幕上, 这是使用频率较高的一个方法。在调用 `Show` 方法时,

如果窗体还没有加载，该方法会自动将窗体加载到内存中。Show方法的使用格式是：

窗体名称.Show Style,Owner

参数说明：

- **Style**——表示窗体显示时是模态还是非模态。模态用1表示，指该窗体显示时，先于该窗体显示的同应用程序中的窗体不会获得鼠标和键盘输入；非模态用0表示，指该窗体显示时，同一应用程序中的其他窗体也可以获得鼠标和键盘输入。在显示模态窗体时，Show方法以后的代码暂不执行，直到该窗体被隐藏或卸载后才执行；对于非模态窗体，Show方法以后的代码在窗体显示期间继续执行，而不必等到该窗体被隐藏或卸载。
- **Owner**——表示被显示窗体的父窗体，一般表示为Me，该参数可以省略。

3. 事件

窗体可以接受的事件很多，本节只介绍几个常用的事件和容易混淆的事件。一般情况下，从窗体创建到删除，按时间顺序要产生以下的事件：

Initialize (初始化)→Load (加载)→Activate (激活)→其他事件→QueryUnload (查询卸载)→Unload (卸载)→Terminate (终止)

其中，Initialize和Terminate事件基本上不需要用户在代码中实现，除非有特殊要求。

- ◆ **Initialize**——窗体初始化。当窗体被创建时该事件发生，它在Load事件之前发生。如果需要在窗体加载之前进行一些重要的初始化操作，可以将代码写在该事件过程中。
- ◆ **Load**——窗体加载。在程序中，可以使用Load语句加载某个指定的窗体，也可以直接调用某个窗体的Show方法来显示窗体。窗体的加载和显示的区别在于已加载窗体的内容和代码都已经存在于内存中，但并不一定显示。如果在窗体未加载以前直接调用窗体的Show方法，则系统会自动将该窗体加载到内存中，然后显示。Load是一个非常重的事件，一般都在该事件过程中添加对窗体、控件、数据初始化的代码。注意，该事件发生时，窗体还没有显示，因此不能在Load事件过程中调用能够引起视觉变化的方法，如Print、SetFocus等，否则会产生错误。
- ◆ **Activate, Deactivate**——当应用程序中有多个窗体时，这两个事件才会发生。例如，应用程序有A和B两个窗体，目前窗体A是活动的，窗体B是不活动的，从外观看，窗体A的标题栏是蓝色，窗体B的标题栏是灰色。当使用鼠标在B上单击一下后，A变为不活动的，B变为活动的，于是产生两个事件：A由活动变为不活动，产生Deactivate事件；B由不活动变为活动，产生Activate事件。使窗体变为活动的方式有两种：通过键盘、鼠标进行操作，或者调用SetFocus（设置焦点）方法来实现。
- ◆ **Click, MouseDown, MouseUp**——这3个事件都和鼠标按键有关。Click指一次单击，即按下一个键再松开；MouseDown指鼠标按键由松开状态变为压下状态的一瞬间；MouseUp和MouseDown正好相反。这3个事件是有区别的，在Click事件过程中，不能区分鼠标的左、中、右键，而MouseUp和MouseDown事件不仅能够区分鼠标的3个键，而且还能检测这两个事件发生的同时是否还按下了键盘上的特殊键。要想精确控制鼠标，应使用后两个事件。在单击鼠标后，会发生这3个事件，发生顺序是

MouseDown ⇔ MouseUp ⇔ Click。

- ◆ **DbClick**——当在一个比较短的时间间隔内两次单击鼠标键时，会产生DbClick事件，意为 Double Click。因为双击由两次单击组成，因此，如果同一代码中存在 Click事件处理过程，则 DbClick事件的处理过程不会执行。
- ◆ **KeyPress, KeyDown, KeyUp**——当用户按了键盘上的键时产生该事件，事件响应过程格式是：

```
Form_KeyPress ( KeyAscii As Integer )
```

参数 KeyAscii是所按键的 ASCII码。若不想接受该字符，可以将参数 KeyAscii 的值设置为 0。KeyDown, KeyUp事件与 KeyPress事件相似，但它可以识别键盘上的所有键和一些组合键（如Ctrl+Alt），这两个事件过程的功能很强，但参数较多，使用较复杂。

- ◆ **Resize**——当窗体大小发生改变（包括最大化、最小化、还原以及通过拖动方式改变大小）时，发生该事件。此外，窗体首次显示时也会发生该事件。在该事件过程中，一般都要对窗体所属控件的布局 and 大小进行调整，以保持良好的界面显示。
- ◆ **QueryUnload**——当窗体即将卸载时发生该事件，事件过程格式是：

```
Form_QueryUnload(cancel As Integer, unloadmode As Integer)
```

参数 unloadmode 表示引发窗体卸载的原因；参数 cancel的值为 0时，表示允许卸载窗体，值为非 0时表示禁止卸载窗体。

- ◆ **Unload**——窗体被卸载时发生该事件，不论什么原因，只要卸载窗体都会引发该事件，它在 QueryUnload事件后发生。该事件过程格式是：

```
Form_Unload( cancel As Integer )
```

参数 cancel表示是否阻止卸载，值为 0表示不阻止，继续卸载，值为非 0表示窗体禁止卸载。一般情况下，在 Unload事件过程中进行数据保存、文件保存等操作。

- ◆ **Terminate**——窗体结束，该事件发生在 Unload事件之后。该事件发生后，窗体就会从内存中消失。因此，一般在该事件过程中添加一些善后处理的代码。

1.2.3 控件

控件是一种对象，是构造 Visual Basic 应用程序界面的基本元素，也是人机交互的基本渠道。在 Visual Basic 中，控件有两种：一种是标准控件，由 Windows 系统本身提供；另一种是第三方控件，由软件开发商提供。虽然控件种类很多，但这些控件的使用方法很相似，只要熟练掌握一些最常用控件的使用方法，就可以很快掌握其他控件的使用。本小节介绍控件的常用属性，控件的方法和能够响应的事件在后面章节中详细介绍。

- ◆ **Name**——该属性表示控件名称，用于在代码中区分每个控件。该属性在设计时可以修改，在代码中只能读取，不能修改。对控件进行命名时，一定要注意两点：一是要见名知义，即通过名称可了解该控件的功能；二是要在名称中体现类型，这可通过加前缀的方法来实现，常用前缀列表如表 1-1所示。例如 文本框控件的类型名称为 Text，假设将该控件添加到窗体设计器中时，其默认名称为 Text1，现在将其名称改为 txtPassword，前缀 txt表示控件类型是文本框，也就知道了它具有哪些属性和方法，可