# Visual Basic for Windows

# 程序设计基础

薛学勤　编著

北京航空航天大学出版社

# 前　言

由于计算机硬件技术的突飞猛进，促进了软件技术的发展，推出了各种集成化的软件系统，使用更为方便，易于掌握。

随着微机的不断普及，国内微机用户的应用水平和应用软件的开发技术也不断提高与深入。以当前国内微机的应用情况来看，MS-DOS 仍然是主流操作系统。大多数用户已在 DOS 环境下开发出各类适用于本行业的应用程序，并在多年的开发过程中对 DOS 环境有了较为深入的分析，掌握了一整套应用程序的开发技术与规律。从国际上发展趋势来看，Microsoft 公司推出的 Windows 3.1 继而 Windows 95 和 IBM 公司推出的 OS/2 将可能是今后微机的主流操作系统。这一点也已为国内用户所注意。Windows 3.1 的友善的用户界面和强有力的功能深受国内广大微机用户的欢迎与喜爱。加之各种运行在 Windows 下的应用软件更让用户爱不释手。但是在 Windows 下要自行开发适合各种专业的应用软件就不像在 MS-DOS 下那么简单了。其技术的复杂程度往往使得非软件专业人员感到望而生畏。虽然已推出有 Borland 公司的 Borland C++ for Windows 程序语言，但它也不易为非计算机专业人员所掌握，所以 Windows 带来好处的同时也带来了问题与困难。为此，Microsoft 公司及时研制、推出了 Visual Basic 程序设计系统。这是一种以事件驱动概念为基础的程序语言，它提供了一整套对象(工具)和方法。Visual Basic 使非计算机专业人员在 Windows 下开发适合本专业的应用程序变得较为容易。编者以为 Visusal Basic 是一个易学易掌握且有较大实用价值的程序系统，适合在非计算机专业的专业人员中推广应用。虽然有丰富详尽的 Visual Basic 技术手册作参考，但手册毕竟是为已掌握 Visual Basic 的用户进一步查阅用的。它是按手册的规范来编写的，不适于作为初学者的教材。基于这一认识，编者从浩繁的手册内容中取其主要之点，按照由浅入深、循序渐进的原则编写了这本书奉献给读者，希望它能成为读者掌握 Visual Basic 的一本入门书。

本书适合于初步掌握 Windows 的使用方法和稍有程序设计概念的各类工程专业人员；也可作为 Visual Basic 程序设计教材。全书 12 章。它主要讲解了 Visual Basic 诸如窗体(Form)、对象(Object)、方法(Method)和事件(Event)等基本概念；Visual Basic 程序设计环境及其应用方法；Visual Basic 系统下的各类工具及其基本属性；Visual Basic 程序语言规则；Visual Basic 的菜单设计；文件的操作；Visual Basic 的绘图功能 和 Visual Basic 程序的调试等主要内容。书中配有大量的例题。为便于读者理解、掌握书中的内容，所有例题都在 Visual Basic Version 2.0 下运行测试过。

熟悉了本书内容，就可以掌握 Visual Basic 程序设计的基本方法，有利于顺利开发应用程序。

由于编者水平有限，时间又紧，书中不妥之处敬请读者与同行不吝赐教，批评指正。

编　者

1995.6.20

# 目　录

## 第一章　概　论



## 第二章　VB 工具箱应用初步



## 第三章　Visual Basic 程序设计概念

**第四章 Visual Basic 程序设计基础**

## 第十一章 绘 图

**第十二章 调试工具与运行出错处理**



**附 录**

# 第一章 概 论

对于编写 Windows 应用程序的人来说，Microsoft 公司的 Visual Basic for Windows（可视 Basic）是一个相当出色的程序设计系统。Microsoft 公司作了极大的努力对 Visual Basic 进行了不断的改进与升级。第 3 版的 Visual Basic，就其功能、灵活性和速度已可与 C 语言媲美；而且当谈到软件生产率的指标时，Visual Basic 明显地优于 C 语言。

## 1.1 Visual Basic 的发展简史

Visual Basic 具有事件驱动（event driven）的编程机制，有新颖、易用的可视程序设计工具，它充分利用了 Windows 图形环境的优点，并且让应用程序开发人员能迅速地构造功能很复杂的应用程序。

BASIC 语言是于 1963 年由 John Kemeny 和 Thomas Kurtz 设计创建的。这种程序设计语言对于初学使用计算机编程的人来说确实是最容易学会与使用的。其特点也是很明显的：每个语句前都有行号；书写不缩进；GOTO 和 GOSUB 都以行号为控制转移的目的。可惜的是它不符合后来发展起来的结构化程序设计的概念；也是由于其上述特点，而被人们认为是不能适应于现实世界程序设计任务的“玩具”语言。

随着个人计算机发展和软件技术的进步。BASIC 也从一个慢的，非结构化的和解释型的语言发展成为一个快的、结构化的编译型语言，适用于开发范围广泛的应用程序。Microsoft 公司和其它公司都相应开发了具有先进高级功能的 BASIC 增强版。

例如，用早期 BASIC 编写的一个求全部小于 100 的质数的程序如下：

```
10 REM - PRIME NUMBER LESS THAN 100
20 N=N+1
30 IF N=100 THEN GOTO 120
40 I=1
50 I=I+1
60 J=N/I
70 IF INT(J)=J THEN GOTO 20
80 IF I>=SQR(N) THEN GOTO 100
90 GOTO 50
100 PRINT N
110 GOTO 20
120 END
```

这是一个非结构化程序设计的例子，它多处使用 GOTO 语句将程序的逻辑控制流搞得上下翻滚，非常复杂。即使这样短短几行程序，也已让人很难读懂了。Microsoft 公司随 4.01 版和比它早一些版本的 MS-DOS 之中都带有一个解释型的 BASIC——称作 GW-BASIC。它与 Microsoft 公司的早先的 BASIC 没有多大差别。尽管利用它可以进行较快的计算和完成一些

简单的任务，但它仍然不为软件人员考虑作为编写商品软件的程序语言。其理由之一，它是一个解释型的 BASIC，作为商品软件必须要将源程序提供给用户，这是所有软件商所不愿意的，而且程序运行仍然很慢。

1982 年，Microsoft 公司开发成功 Quick Basic，对 BASIC 来说是一次变革，使它成为了在 MS－DOS 环境下的正式的开发软件的程序语言。Quick Basic 语言将 GW－BASIC 的交互性、高产性与编译语言的强有力的功能与速度有效地结合在一起，取消了行号，同时增加了现代程序语言具有的诸如子程序、用户定义的结构类型等特性。此外，它提供高级图形与声音功能，使得 Quick Basic 程序员能获得比从 C、PASCAL 和 FORTRAN 语言中通常提供的更强的功能。Quick Basic 程序的另一个独到之处是它可以以解释方式运行，也可编译成独立的可执行的程序，适用于软件市场的需要。

为使用户有一个强有力的、标准化的环境，在这环境下能充分利用 Intel 公司的最新微处理器的能力，Microsoft 公司的 Windows 是一个突破。它使微机的人机界面更加友好。Windows 也是当代图形用户界面(GUI)技术的一个应用典范。Windows 为 PC 机用户提供了一个直观的、图形丰富的工作环境。图形用户界面使得应用程序变得更易于学习和使用，也使用户更为主动，更感兴趣。

虽然 Windows 环境对用户来说是相当吸引人的，但对应用程序编写人员来说就不那么简单了。它(她)们不能再用陈旧的编写"过程程序"的概念去编写程序了。必须重新学习在 Windows 下一套程序设计的新概念，其核心就是"事件驱动"的概念；要学习如何创建窗口、菜单、各种开关控制按钮、对话框……等等构件。即使是一个最简单的程序也都必须有上述构件组成。图形用户界面(GUI)使编程工作变得越来越复杂。例如，在 MS－DOS 下用任意一种程序语言编写一个在屏幕上显示一条信息的程序，最多只需 4 行语句就够了，而在 Windows 下编写同一功能的程序则其代码却可能长达 2～3 页之多。当然，Windows 环境最终对用户来说其好处是不容置疑的，而且大受欢迎。对于专业软件人员来说，自然只有顺应潮流，再学习、再提高。但是对于非计算机专业领域的其它专业人员来说，要在 Windows 下编写适用于本专业的应用程序其困难程度就远远大于在 MS－DOS 下的情形了。因此，问题也就产生了，非计算机专业领域的专业人员在 Windows 环境下怎么办？Microsoft 公司作出回答，这就是 1991 年 Microsoft 公司开发设计出的 Visual Basic，它对上述问题提出了一个有效的解决办法。

Visual Basic for Windows 是一个在 Windows 环境下的一个编程系统，它利用面向对象的程序设计概念巧妙地将 Windows 下的编程复杂性封装起来。Visual Basic 运用了 BASIC 语言和一系列的可视设计工具，它既未牺牲 Windows 的精华之处和图形工作环境，同时又提供了一套简单、易学的编程方法，对窗口、菜单、按钮、对话框、信息框、滚动条等构件的设计也提供了一整套工具，使设计人员很容易在较短的时间内，设计出十分复杂的窗口版面。

Visual Basic 是首批采用事件驱动编程机制的程序语言之一。事件驱动是一种非常适合于图形用户界面的编程方式。

从 1991 年推出 Visual Basic 1.0 版后，1992 年再度改进为 2.0 版，至今已推出 3.0 版，获得极大成功。

事件驱动机制是所有 Microsoft Windows 应用程序的核心。当用惯了这种编程方法后，那么不再会留恋老的一套过程代码的编程方式了。事件驱动代码和过程代码之间在概念上的差别并不十分大，许多在 Windows 下的 Visual Basic 应用程序的代码对 BASIC 程序员来说是很

熟悉的。下列三个例题程序展示了从BASIC(GW－BASIC)到 Microsoft Quick Basic ，最后直到 Visual Basic 的发展过程。这三个程序都在屏幕顶端(对于 Visual Basic 是在窗口顶端)显示出如下一行信息："Testing…1 2 3"

1. GW－BASIC 版本

```
10    REM: GW－BASIC Test Program
20    CLS
30    PRINT "Testing…"
40    FOR I=1 TO 3
50    PRINT I;
60    NEXT I
70    PRINT
80    END
```

2. Quick Basic 版本程序

```
Quick  Basic Test Program
CLS
PRINT "Testing…";
FOR i=1 TO 3
     PRINT i;
NEXT i
PRINT
END
```

这个程序只对BASIC 作了少量改进，如语句前不必加行号。其余相同。

3. Visual Basic 版本程序

```
' Visual Basic for Windows Test Program
Sub Form _ Click( )
    Dim i As Integer
    Print "Testing…";
    For i=1 TO 3
    Print i;
    Next i
End Sub
```

这个程序除关键字第一个字母用大写，其余字母用小写字母，如 Print 这点写书形式上的差异外，其句法是完全相同的；其最大的差别在于这一段代码放置在事件驱动的子程序 Sub Form _ Click( )之中，前后有 Sub  Form _ Click( ) 和 End Sub 两个语句分别表示子程序的开始与结束。

## 1.2 Visual Basic 的优点与特性

Visual Basic for Windows 除保留了 Quick Basic 的许多高级性能外，还为在 Windows 环境下开发应用程序作了许多强化和扩展。强化之一是图形可以直接输出到一个窗口的任何部分中，甚至输出到打印机上。

另一个强化是在 Visual Basic 中变量作用域规则的简化和改进得更易理解与记忆。

一个用 Visual Basic 编写的应用程序包括两种类型的文件：一种称为窗体文件(form)，它包含窗口和对话框的可视图例和其相应代码；另一种为模块文件(modules)，模块中只包含 Visual Basic 的源代码。在模块或窗体的通用声明段 (general—declaration section)中声明的变量或常量，可以被这一模块或窗体的各个子程序或函数来存取；在模块中作为 Global(全局的)量来声明的变量与常量可以被本项目(project)中的任意窗体或模块中的任一过程或函数子程序来存取；而在过程和函数子程序中声明的变量和常量，局部于那个过程或函数子程序，除非你声明这个变量或常量为全局的(Global)。

图 1-1 是一个简单应用程序的代码。这项目(Project 1)由一个模块文件 Module 1. bas 和一个窗体文件 Form1. frm 构成。当运行这个应用程序时，只要鼠标指针指在窗口内任意处并单击一下鼠标左键，它会在窗口的左上角显示出如下结果：

```
3.141593     6.28186
```

从这个例子中可见：常量 PI 在模块文件 Module1. bas 中声明作为 Global(全局的)，它在本项目的任何地方都是可见的；常量 TOWPI 在窗体文件 Form1. frm 中的通用声明段中声明的，它只在 Form 1. frm 的任何地方是可见的；而常量 THREEPI 是在子程序 From _ Load( ) 中声明的，它只在子程序 From _ Load( )中是可见的，在子程序 b Form _ Click( )中是不可见的。从这个例子可以看出 Visual Basic 的变量与常量的可见性(或称作用域)是十分好记，易理解的，只要看一看它们在一个项目(project)中的具体位置就可确定其可见性。至于如何使用 Visual Basic 程序设计环境来编写 VB 应用程序请参见 1. 4 节。

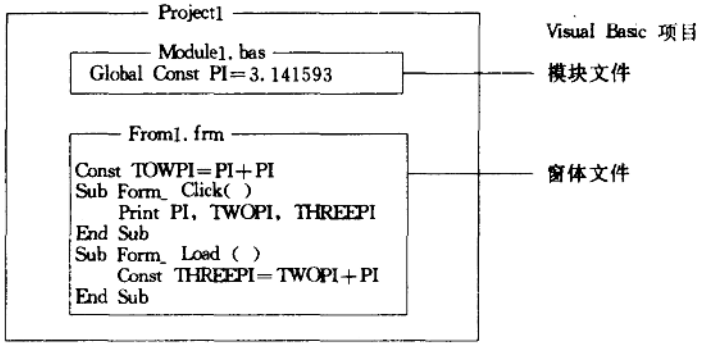


图 1-1　变量、常量作用域示例

Visual Basic 一个最大的优点是对一个有 Windows 使用经验的人来说，只需要一个较短时间的训练就可以成为一个高产的程序员。如果已经掌握 Quick Basic，那么用起 Visual Basic 来将会更顺手。众所周知，当你在 Windows 下用 C 语言学习编写应用程序时，那么需要消化和理解大量的资料。相比之下，你会特别对 Visual Basic 的直观性和交互性感兴趣，学习用 Visual Basic 在 Windows 下编写功能很强的应用程序是一条最快的途径。即使是对一个熟练的 C 语言 Windows 应用程序的开发者来说，Visual Basic 新提供的交互式图形用户界面的开发工具，他也会相当感兴趣的。

Visual Basic 的另一个优点是它的可扩展性。一般说，在 Windows 下运行 Visual Basic 的应用程序的速度是快的。在某些特殊情况下，用优化 C 语言的代码，运行速度比它更快。关于 Visual Basic 应用程序如何调用 Windows DLL(Dynamic Link Library)中提供的，用 C 语言编写的函数的技术，因超出本书编写范围而没有提及。其有关技术读者可参考相关手册。

## 1.3 Visual Basic for Windows 的安装与启动

### 1.3.1 必需的软、硬件配置

为运行 Visual Basic，必须具备下列的软、硬件。

(1)80286 或更高的 IBM 兼容机，大于 1MB 的内存，EGA 或 VGA 显示器，相应的硬盘和软盘驱动器及鼠标器；

(2)3.1 版以上的 MS—DOS 和 3.0 版以上的 Windows。

### 1.3.2 安装之前的准备工作

在安装之前应将 Visual Basic 盘做一份备份，以备不测。备份可以利用 MS—DOS 下的 Copy 或 Diskcopy 命令完成，也可以利用 Windows 下的 File Manager 的 Disk 或 File 菜单下的 Copy Disk 或 Copy 命令来完成。自然不排除用其它 PC 软件工具来备份，这里没有严格规定，随读者自便。

### 1.3.3 安装与启动

Visual Basic 的安装可以用 SETUP.EXE 程序来完成。具体步骤是：

(1)将 Visual Basic #1 盘片插入驱动器 A；

(2)在 Windows 下，从 Program Manager 或 File Manager 的 File 菜单中选择 Run 项；

(3)在提示下键入 A:setup 并按一下 ENTER；

(4)当 SETUP 程序运行时，可以按屏幕上提示的安装指示，一步一步地完成。

安装过程并不复杂。当完成安装后可以双击(注一)新程序组中的 Visual Basic 图标来启动 Visual Basic。

Visual Basic 系统的操作方法与 Windows 的操作完全相同。如果不熟悉 Windows 的操作，最好预先练习一下 Windows 的几个基本操作。

(注一)："双击"是《Visual Basic 使用手册》中 Double—Click 一词的译文。其含义是将鼠标指针(箭头)指向某一图标，并连续、快速敲击鼠标左键两次。为行文方便，使用"双击"一词。与双击相对应，对于 Click 一词就译为"单击"。其含义是将鼠标指针指向某一图标，并敲击鼠标左键一次。以后行文中的单击与双击都是这个意思。

## 1.4 Visual Basic 程序设计环境

当启动 Visual Basic 后，在屏幕上可以看到如图 1-2 所示的 Visual Basic 程序设计环境的用户界面。

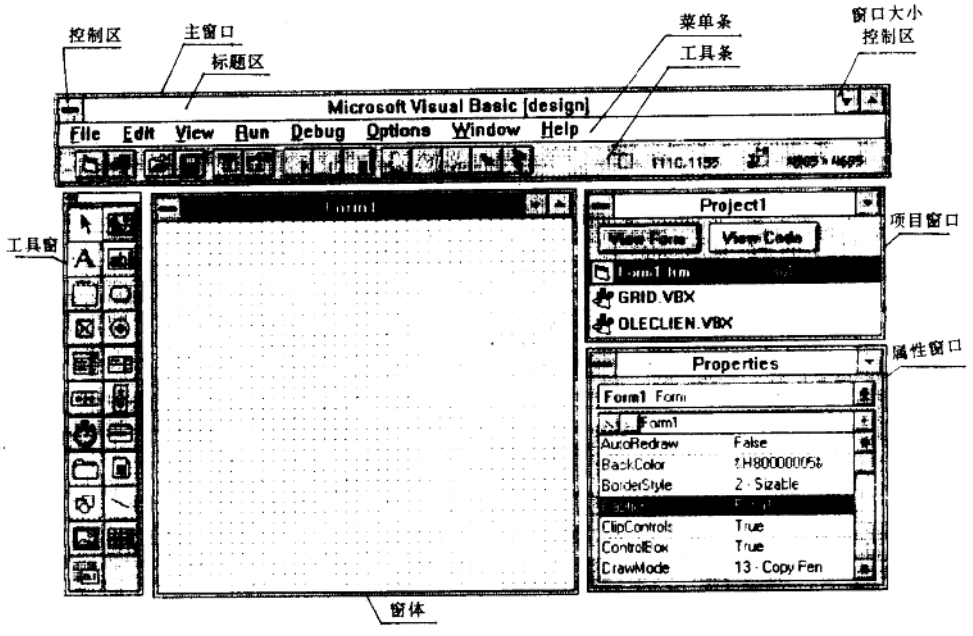


图 1-2 Visual Basic 程序设计环境的用户界面

为了清晰起见，图中有意调整了各窗口的大小，使得 VB 的用户界面的主要窗口都显示在屏幕之上。实际运行时，为使用方便，其中几个窗口是互相覆盖的。

这个界面由 VB 主窗口(Design)、工具箱窗口、项目窗口、属性窗口和窗体窗口(Form)等组成。要开发一个 VB 应用程序，首先要熟悉掌握这一环境各部分的功用与使用方法。在进入 VB 的详细讨论之前，先对这一环境作一简单的全面介绍，以便读者对它有一全局性的认识。

### 1.4.1 主窗口(Design)

这一窗口的布局完全类似于 Windows 窗口，它包括有：

1. **标题区** 它位于窗口的最顶端一行的中间部分，用来表示窗口的标题，这里具体的是：Microsoft Visual Basic[Design]。

2. **控制区** 它位于左上角用▣表示，用来控制此窗口的恢复(Restore)、移动(Move)、大小(Size)、最小化(Minimize)、最大化(Maximize)、关闭(Close)和切换(Switch To ...)等；

3. **窗口大小控制区** 它位于右上角，用图标▲▼表示，用来放大或缩小窗口。

以上三项位于该窗口的最上面一行之中。

4. **菜单条**(Menu Bar) 它位于窗口的第二行，上面列出了本窗口可使用的菜单项。它们是：File、Edit、View、Run、Debug、Options、Windows 和 Help 等八项。对于这八项菜单分别还有一整套的子菜单项。可以用鼠标指针指向所选定的某一菜单项，然后单击一下，就可以显现

相应于该菜单项的下拉菜单；也可以使用键盘的组合键，用 Alt＋相应菜单项中带有下划线的字母键，同样可以显现其对应的下拉子菜单。再用↑↓键来选择所要选用的子菜单项。

例如，鼠标指针指向 File 项并单击一下鼠标左键(或按住 Alt 键后再按一下字符 F 键)就可出现如图 1-3 的下拉菜单。这菜单将命令分成几个命令组，用横线隔开。它的主要功能是用

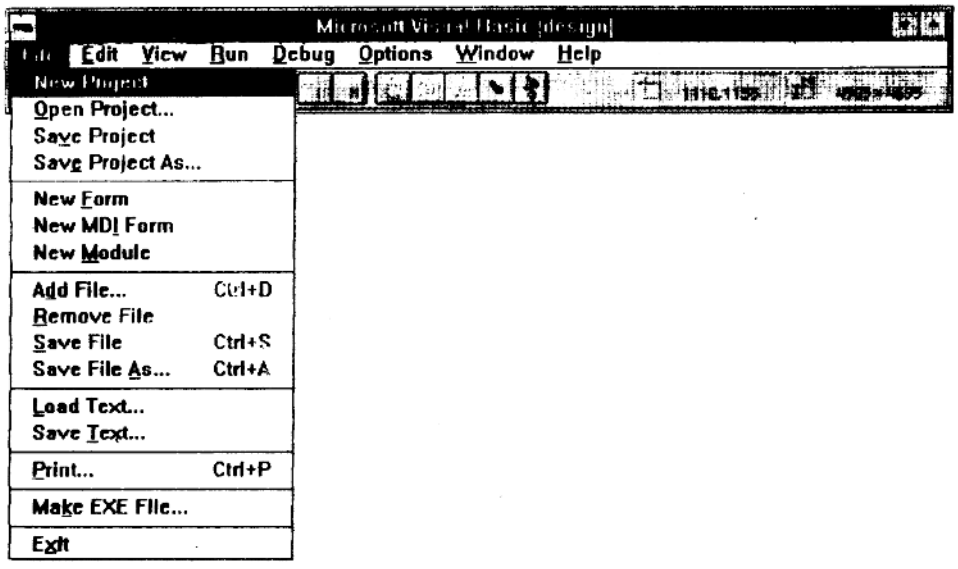


图 1-3　File 项的下拉菜单

来打开，建立、保存……窗体(Form)或项目(Project)文件、打印文件、编译源程序代码、生成.EXE文件和退出等。

用相同的操作可以拉出各菜单项的子菜单。图 1-4 就是相应于 Help 菜单项的下拉菜单。

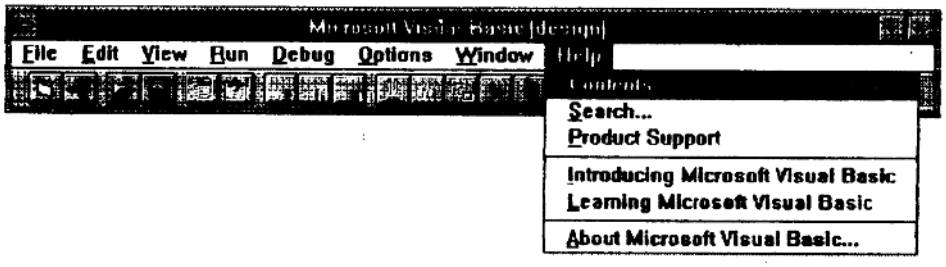


图 1-4　相应于 Help 菜单项的下拉菜单

5. **工具条**(Toolbar)

它位于设计窗口的第三行的右边部分。其左边部分是窗体左上角顶点的坐标及其长宽的尺寸。它的单位是 twip，且

$$1\ \text{twip} = 1/1440\ '' = 1/567\ \text{cm} = 1/20\ \text{点}$$

工具条向用户提供一种存取程序设计环境下菜单条中一些常用命令的简捷方法。用户只要用鼠标指针指到工具条上所选定的图标并单击一下鼠标左键，就可以执行该图标所代表的命令与动作了。图 1-5 列出工具条中各图标的名字。表 1-1 逐个列出各图标及其所对应的菜单条中的命令。


图 1-5　工具条

**表 1-1　图标及其对应的菜单命令**

| 图标 | 动作 | 等价的菜单命令 |
|---|---|---|
| | 创建一个新窗体 | File 菜单上的 New Form 命令 |
| | 创建一个新模块 | File 菜单上的 New Modoule 命令 |
| | 打开一个已存在的项目(project) | File 菜单上的 Open Project 命令 |
| | 保存当前的项目 | File 菜单上的 Save Project 命令 |
| | 显示菜单设计窗口 | Window 菜单上的 Menu Design 命令 |
| | 显示属性窗口 | Window 菜单上的 Properties 命令 |
| | 以设计模式开始一应用程序 | Run 菜单上的 Start 命令 |
| | 当程序正在运行时停止其运行 | Run 菜单上的 Break 命令 |
| | 停止执行一个应用程序并返回到设计模式 | Run 菜单上的 End 命令 |
| | 在当前行上 Toggles 断点 | Debug 菜单上的 Toggle Breakpoint 命令 |
| | 显示代码窗口中当前选择的值 | Debug 菜单上的 Instant Watch 命令 |
| | 显示激活调用的结构 | Debug 菜单上的 Calls 命令 |
| | 代码窗口中一次一个语句执行代码 | Debug 菜单上的 Single Step 命令 |
| | 代码窗口一次执行一个过程或语句 | Debug 菜单上的 Procedure Step 命令 |

### 1.4.2 工具箱窗口(Toolbox)

图 1-2 中左边所示的一个窗口,这个窗口像所有 Windows 的窗口一样,可以用其左上角的控制区来关闭(Close)或移动(Move),关闭后,可以选菜单条中的 Window 菜单项中的 Toolbox 子菜单来打开。

这一工具箱(ToolBox)为 VB 应用开发者提供了一套在设计应用界面时放置在窗体上的各种 Windows 的标准控制件(Controls)。这里先对工具箱中每一图标(icon)所表示的具体意义与动作作一简单的介绍。以后各章中,将配合实例进一步阐述它们的用途。图 1-6 说明工具箱中各图标的含义。

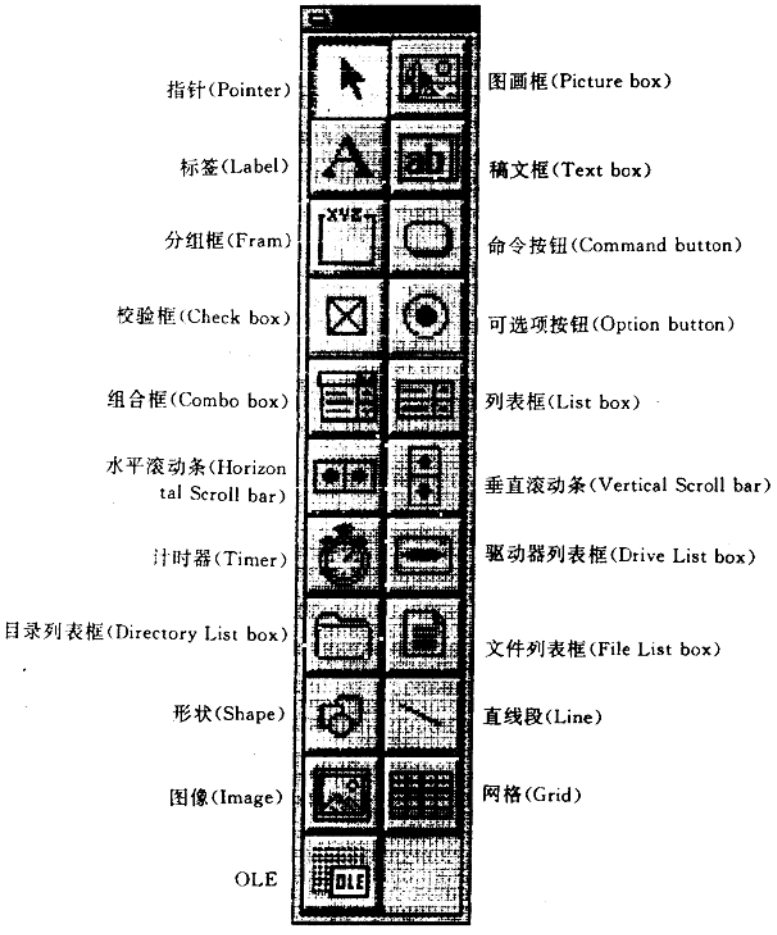


图 1-6 工具箱中各图标的含义