

Visual Basic 3.0 程序设计大全

[美] Charles Parly 著
梁念蓉 等 译
王 勇 校

- 学习和使用 Visual Basic 3.0 的最新特性,了解 Visual Basic 3.0 的程序开发环境
- 掌握 Visual Basic 3.0 的输入/输出技术与技巧
- Visual Basic 3.0 高级应用程序开发
- Visual Basic 3.0 应用程序优化及调试
- Visual Basic 3.0 高级程序设计技术,包括 DDE、OLE 及绘图技术
- 充分利用 Visual Basic 3.0 和数据库技术进行数据访问



电子工业出版社

Publishing House of Electronic Industry



Visual Basic 3.0 程序设计大全

帕利

[美] Charles Parly 著

梁念蓉 等 译
王 勇 校

电子工业出版社

1994.9

(京)新登字第055号

内 容 提 要

Visual Basic 3.0是微软公司新推出的新一代高级程序设计语言,也是目前国内较为流行的微机应用程序开发工具之一。《Visual Basic 3.0程序设计大全》适用于所有在Windows环境下使用Visual Basic 3.0进行应用程序开发的编程人员。全书共分为十九章,由浅入深地介绍了Visual Basic 3.0的最新特性、编程基础以及程序开发环境等基础性内容,以及常用命令按键及菜单、输入/输出技术、绘图和鼠标应用等常用的技术。此外,《Visual Basic 3.0程序设计大全》还使用了大量的篇幅讨论了诸如动态数据交换(DDE)、对象链接与嵌入(OLE)等众多高深的程序设计专题。可以说,无论是初学Visual Basic 3.0程序设计,还是已有Visual Basic 3.0的程序设计经验,您均会在《Visual Basic 3.0程序设计大全》中找到令您感兴趣的内容。

本书英文版由美国Far East Books出版公司出版,该书的中文出版版权已由原出版公司授予中国电子工业出版社。未经出版者同意,任何人不得以任何手段复制或抄袭本书的内容。

Visual Basic 3.0 程序设计大全

[美] Charles Parly 著

梁念蓉 等 译

王 勇 校

责任编辑 王世忠

电子工业出版社出版(北京市万寿路)

电子工业出版社发行 各地新华书店经销

北京市京林排版公司排版

北京市顺义县天竺颖华印刷印刷

*

开本:787×1092毫米 1/16 印张:29 字数:695千字

1994年9月第1版 1994年9月第1次印刷

印数:10100册 定价:70.00元

ISBN 7-5053-2634-1/TP·805

目 录

第一章 Visual Basic 中的一些新概念与新名词	1
1.1 对象(Object)	1
1.2 属性(Property)	2
1.3 Visual Basic 中的对象	2
1.4 设定属性	4
1.5 事件(Event)	8
1.6 事件程序 (Event Procedure)	8
1.7 方法 (Method)	10
1.8 如何扩充Visual Basic 中的对象	10
第二章 Visual Basic 编程基础	13
2.1 Visual Basic 应用程序的基本结构	13
2.2 过程 (Procedures)	16
2.3 过程中的代码	19
2.4 流程控制	24
2.5 变量声明	31
2.6 变量的作用范围及生命周期	33
2.7 基本数据类型	36
2.8 数组	44
2.9 用户自定义数据类型 (结构)	48
2.10 符号常数	50
第三章 Visual Basic 程序开发环境	53
3.1 主窗口	53
3.2 工具箱	54
3.3 窗体	55
3.4 专题窗口	56
3.5 设计一个简易的数字时钟	57
3.6 实用型数字时钟的设计	60
3.7 程序开发的步骤	65
3.8 专题的开发与关闭	66
3.9 环境的设置	66
3.10 实例	67
3.11 程序的调用	74
3.12 事件驱动程序的设计	76
3.13 程序的开始	78

3.14 如何控制窗体中的对象	79
3.15 实例2	79
3.16 常用的快捷键	80
第四章 命令按键与菜单	83
4.1 实例1	83
4.2 执行系统命令	84
4.3 利用程序改变按键的状态	85
4.4 菜单的建立	86
4.5 选项状态的改变	92
4.6 选项标记(check Mark)	93
4.7 菜单选项的增减	95
4.8 键盘选取	98
4.9 显示弹出菜单(pop_up menus)	100
第五章 输入方法与技巧	103
5.1 文字框(Text Box)	104
5.2 选择按钮(Option Button)	108
5.3 确认框	111
5.4 列表框及下拉式列表框	113
5.5 滚动条(Scroll Bar)	117
5.6 InputBox\$函数	119
第六章 输出方法和技巧	121
6.1 信息框(Message Box)	121
6.2 由标记输出	122
6.3 文字的字型	124
6.4 输出到窗体及图片框	125
6.5 字型的高度及宽度	126
6.6 定位(TAB)	127
6.7 制定数字的输出格式及日期(时间)格式(Format\$函数)	129
6.8 输出到打印机	130
6.9 利用网格控制项输出	131
第七章 绘图在应用程序中的应用	139
7.1 在窗体上加图片	139
7.2 操作控制项	141
7.3 设定对象的坐标系统	146
7.4 绘图功能与色彩	149
第八章 鼠标的使用	167
8.1 鼠标事件	167
8.2 利用鼠标来定位控制项	168

8.3	用鼠标绘图	168
8.4	鼠标按键及Shift参数	171
8.5	拖放	175
8.6	程序控制拖曳	179
8.7	控制项的重新定位	180
第九章	多重窗体与多重文件界面(MDI)	183
9.1	多重窗体	183
9.2	通用对话框自制控制项的使用	191
9.3	多重文件界面(MDI)	197
第十章	调试及执行时期错误处理	205
10.1	调试	205
10.2	执行时错误处理	216
第十一章	系统资源	225
11.1	选中字符串的处理	225
11.2	剪贴板 (Clipboard) 的使用	226
11.3	定时器的使用	229
11.4	用DoEvent函数写闲置循环(Idle Loops)	233
11.5	低级键盘控制	236
第十二章	文件系统控制项的使用	241
12.1	文件系统控制项	241
12.2	一个文件管理器	243
12.3	打开文件对话框	248
第十三章	文件处理	257
13.1	概述	257
13.2	顺序存取文字文件	261
13.3	随机存取文件中的记录	273
13.4	二进制文件的处理	291
第十四章	动态数据交换(DDE)	297
14.1	动态数据交换(DDE)	297
14.2	执行期间进行DDE链接	300
14.3	DDE运行控制及错误处理	310
第十五章	对象的链接与嵌入(OLE)	317
15.1	OLE 2.0概述	317
15.2	如何建立和使用对象链接和嵌入	319
15.3	OLE自动化(OLE Automation)的使用	327
15.4	OLE 2.0 的编程及错误处理	334
第十六章	数据访问	345
16.1	概述	345

16.2 利用数据控制项来访问数据库	346
16.3 通过编程操作记录	354
16.4 Microsoft Access 1.1 Engine的结构和可编程对象	361
16.5 建立一个Visual Basic 3.0 数据库	372
第十七章 应用程序的优化	383
17.1 提高计算和程序执行速度	383
17.2 提高显示速度	384
17.3 加快应用程序的开始	386
17.4 使应用程序所占用的总内存最少	387
17.5 使图画所占用的内存最小	389
17.6 减少堆栈的使用	391
第十八章 在 Visual Basic 中使用 Windows API	393
18.1 拓展 Visual Basic for Windows	393
18.2 Windows API 的基本概念	394
18.3 在Visual Basic中调用DLL	402
18.4 错误处理及注意事项	411
第十九章 VBA 简介	417
19.1 历史的回顾	417
19.2 VBA简介	419
附录 函数索引	423

第一章 Visual Basic 中的一些新概念与新名词

Visual Basic 虽然仍然沿用 Basic 的语法，但这并不意味着会用 Basic 的人就会使用 Visual Basic，因为 Visual Basic 中有许多 Basic 中所没有的新概念及新方法。

1.1 对象 (Object)

学好 Visual Basic 的秘诀就是先学会以“对象”的角度来看整个程序设计。这里所说的“对象”其实是和“面向对象的程序设计”中的对象是一样的，只是在面向对象程序设计中，必须由用户自己来设计对象。而在 Visual Basic 中，对象是已经设计好了的，可以直接使用和操作。比如，在以前，如果我们要把屏幕上的钟移动到新的位置上去，必须大费周折地先记住起始坐标位置、长和宽，再把它读入缓冲区，然后清除掉屏幕上的钟，接着再取得目的坐标，最后将钟显示在新的位置上，这样才算是最后完成了这个工作。

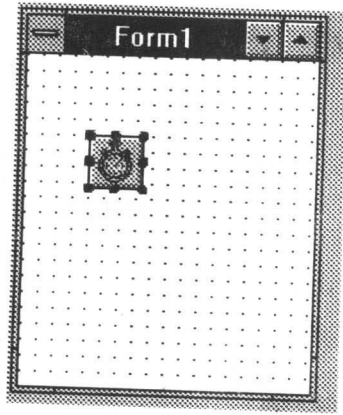


图 1-1

而任何人在家里移动墙壁上的钟时，想必没有如此麻烦。我们其实都是以“对象”的眼光来看待墙上的钟的，因此要移动它时，脑子里所浮现的命令一般都是：“将钟向左移动50公分”等等，而不会是将(150, 50)位置上，长40公分，宽25公分的钟移动到(100, 50)的地方去”这样的思想。

“对象”其实就是我们习以为常的名词(noun)。我们可以说整个世界是由许许多多的“对象”所构成的，而地球也是宇宙中的一个“对象”，而面向对象程序设计中最重要的思想之一就是，希望能够把真实世界中的事物更精确地反映到程序设计之中。

“对象”既然是“名词”，因此就会有名字，所以不同的对象不应该有相同的名字，例如：电冰箱和洗衣机，如果都叫做“铁箱子”的话，显然会造成混淆。如果发生在程序中，就会造成无法正确执行的结果。

1.2 属性(Property)

我们先看看下面一个生活中非常简单的例子：皮球和铅球都是球，“球”可以看作是一个“空的对象”，也就是说，它不含有任何属性(一种假设)。

球：

皮球 <-----> 玻璃球

而为什么会有皮球、玻璃球或者铅球呢？这是因为我们将“球”的材料这项属性设为皮、铅或者玻璃的缘故。如果我们进一步设定颜色、大小，就可以得到更多的分类后的对象。

例如：颜色 = 红，材料 = 皮，大小 = 大；

或者：颜色 = 黑，材料 = 玻璃，大小 = 小；

而“人”的属性和“球”的属性显然是不同的，例如：

空的对象1：人

人的属性：肤色 性别 高矮 体重……

空的对象2：球

球的属性：材料 颜色 大小 重量……

也就是说，不同的对象会有不同的属性，当然偶尔会有些相同的情况。而各个对象的所有属性的集合叫做“属性表”。所以，我们可以为空的对象配置各种不同的属性，以便创造出各种各样的对象来。

1.3 Visual Basic 中的对象

Visual Basic 中的对象主要分为窗体(Form)和控制项(Control)两类。

Form：即窗体，由于Windows的环境很象办公桌上的环境，所以Visual Basic 把一个可视窗口看做办公桌上的一份表格。

Control：即控制项，可以说是Windows应用程序上的一个组件，而这些组件都是由“空白对象”的各项属性的设定所调配出来的。

下面的窗体里含有三个控制项，因此总共有四个对象。

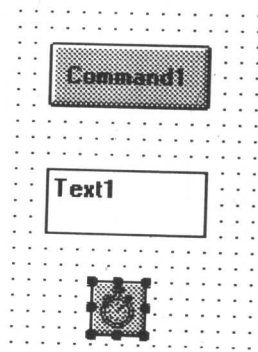


图 1-2

在开发Visual Basic 应用程序时，需要四个窗口配合在一起使用，它们分别是主窗口 (Main Window)、工具箱 (Toolbox)、窗体 (Form)、专题窗口 (Project Window) 及属性窗口 (Properties Window)。若看不到某些窗口，可以先找找看它们是否被缩成了图标，或者从 Window 菜单中把相关的窗口 (工具箱、属性窗口、或专题窗口) 调出来。

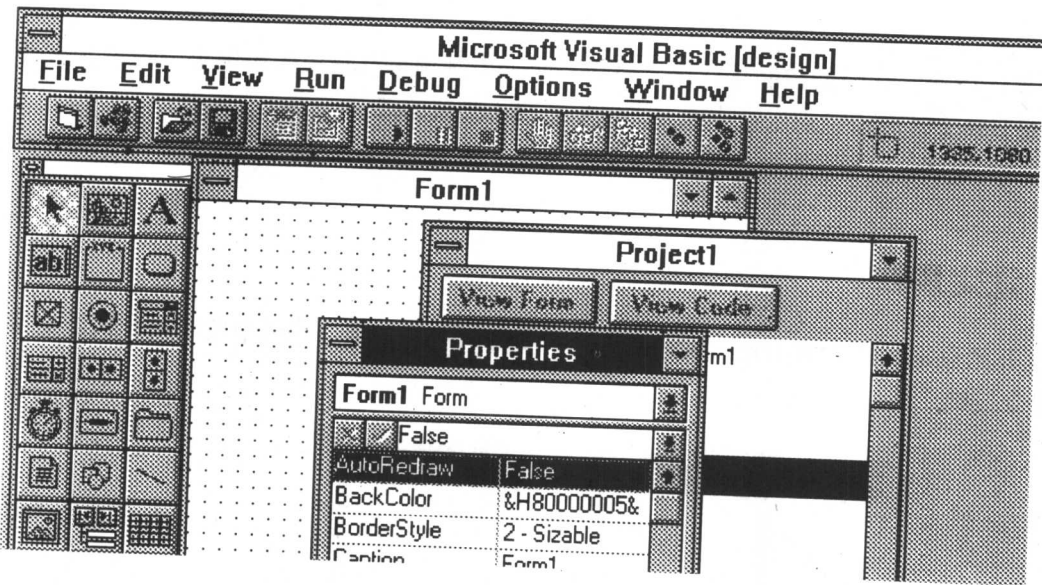


图 1-3

而在工具箱中则装满了各种各样的控制项 (也就是“空的对象”)，读者只要在工具箱中的某个控制项上双击鼠标键，该控制项就会出现在窗体的正中央，同时会变成双箭头。此时，拖曳鼠标就可以调整控制项的大小，也可以直接拖曳该控制项以调整其位置 (光标不要放在调整框上)。

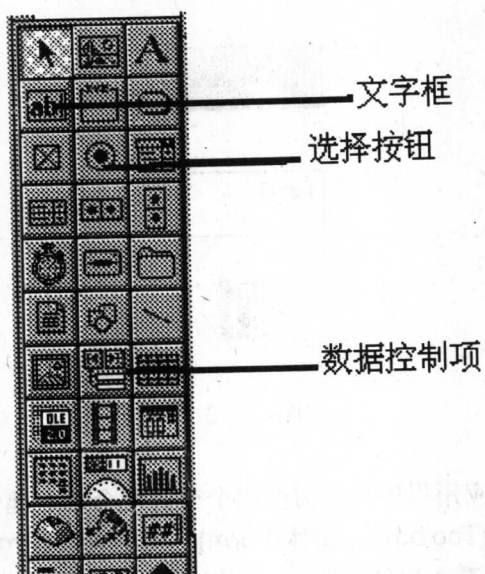


图 1-4

如果连续双击相同的控制项两次，则会有两个控制项重叠在窗体正中央，这是初次使用Visual Basic时应该特别注意的地方，以免造成不必要的迷惑。

1.4 设定属性

在开始设定属性之前，先让我们在窗体中安排两个控制项，这两个控制项分别是文字框及命令按钮。

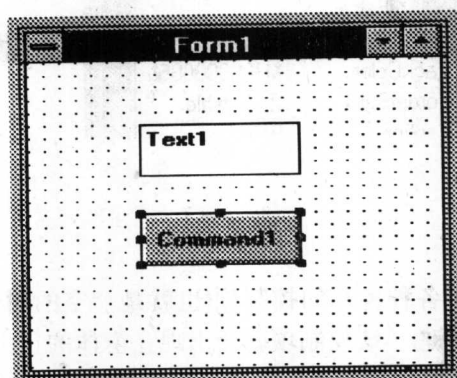


图 1-5

由于当前共有三个“对象”（窗体、文字框及命令按钮），因此可分别为这三个对象设定某些属性。在进行之前应先选定准备设定的那个对象的属性，因此必须先在该对象上单击一下（会出现大小调整框框住该对象，但窗体除外）。如果要选定窗体，则必须在它的工作区（也就是客户区，Client area，没有放控制项的空白区域）上单击一下。

由于窗体的Caption(标题)属性的缺省值为Form1，因此窗口的标题部分可以看到Form1字样，而文字框的Text属性及命令按钮的Caption属性则分别被预设为Text1及Command1（文字框没有Caption属性，而命令按钮则没有Text属性）。

所以只要重新设定各个对象的某些属性，就可以符合应用程序的特殊要求。由于每个对象的属性表都记录在“属性窗口”中，因此在选定对象之后，就可以到属性窗口中寻找caption或text属性，并在“属性设定”栏位中直接输入所需要的字符串。

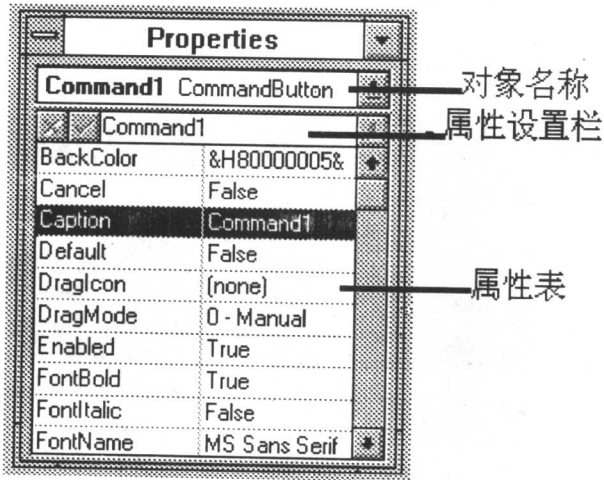


图 1—6

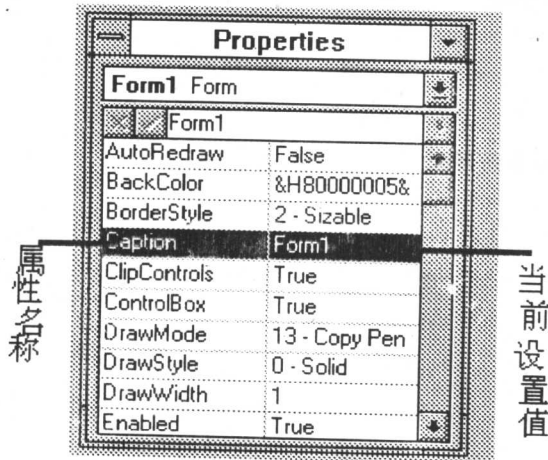


图 1—7

“对象名称”栏位显示目前被选对象的名称及其类别。如果在该栏位右边的向下箭头上单击一下，会将列表框拉下，并且显示所有的对象。

“属性设定栏”可以用来输入新的属性。当输入完毕后，可以在左边的打勾符号上单击一下，以确定新值设定完毕。若在打叉符号上单击一下，则用来取消刚刚设定的值。至于最右边的向下箭头，则用来显示所有的属性值。

“属性表”则是所有属性及目前设定值的总表，除了直接用鼠标之外，亦可按向上或向下箭头键来选取不同的属性。

例如：如果想改变窗体的背景颜色，可以先在窗体的工作区上单击一下，然后寻找 Backcolor 属性，并在“属性设定”栏位中直接输入十六进制的颜色值，或按右边的“...”按钮(三点，表示会弹出对话框)，然后从显示出来的调色板中直接选取颜色，窗体的背景颜色立刻随之改变。

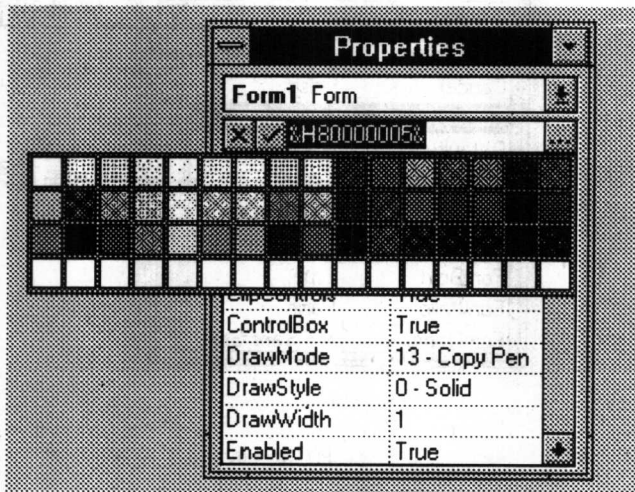


图 1-8

以下是窗体、文字框及命令按键的几个属性的缺省值：

窗体：

属性	缺省值	说明
Caption	Form1	窗体的标题
Name	Form1	窗体的名称
BackColor	&H80000005&	窗体的背景颜色

文字框:

属性	缺省值	说明
Text	Text1	文字框中的缺省字符串
Name	Text1	控制项名称

命令按键:

属性	缺省值	说明
Caption	Command1	命令按键上的文字
Name	Command1	控制项名称

其中的窗体名称及控制项名称, 都是供程序进一步控制时引用所用, 因此最好取一些描述性比较强的名称。如果能在变量名称之前加上代表对象类型的前导字, 那就更理想了, 下面是对象类型与前导字的参考表:

对象	前导字	范例
窗体(Form)	frm	frmTitle
确认框(Check box)	chk	chkBackup
下拉式列表框(Combo box)	cbo	cboLanguage
命令按键(Command button)	cmd	cmdOK
目录列表框(Driectory List box)	dir	dirHardD
磁盘列表框(Drive List box)	drv	drvFloppy
文件列表框(File list box)	grd	grdScore
水平滚动条(Horizontal scroll bar)	hsb	hsbDegree
图象框(Image)	img	imgHouse
标记(Label)	lbl	lblFileName
直线(line)	lin	linconnect
列表框(list box)	mnu	mnuGame
圆钮(option button)	opt	optChinese
ole(Ole)	ole	oleCome
图片框(Picture Box)	pic	picCat
图形(shape)	shp	shpSquare
文字框(Text Box)	txt	txtName
定时器(Timer)	tmr	tmrTimeBase
垂直滚动条(Vertical Scroll Bar)	tom Control)	CMDialog
数据控制项(Data Control)	data	data3

注意: 在Visual Basic 中的变量长度不应该超过40个字。
所有对象的属性表都在附录中, 必要时可以参阅。

1.5 事件(Event)

当我们用铁锤敲击桌子时，就是一件发生在桌子上的事件。而某些事件可能只能发生在某些对象上，例如：老师可以责备学生，但是不可能去责备桌子。

大家可能都已经了解 Windows 是属于“事件驱动结构”的。也就是说，只有在事件发生时，程序才会执行。在没有事件时，整个程序处于停滞状态。这一点和DOS程序有极大的差别。

鼠标的单击及双击应该是我们最熟悉的事件了。实际上，事件可以分为“用户事件”(User Events)及“系统事件”(System Event)两类。

典型的用户事件包括：键盘输入、鼠标的单击、鼠标的双击、拖曳…等等。而定时信号则是典型的系统事件，它是由系统本身负责产生的。每个对象究竟有多少事件可能发生在它身上，可以参考对象与事件对照表，例如，定时器对象只有Timer一项事件。

1.6 事件程序 (Event Procedure)

前面提到的老师责备学生是发生在学生身上的一个事件。但是用同样的话去责备不同的学生，所得到的反应可能是完全不同的。其中一个可能痛哭流涕，而另一个却可能是嘻皮笑脸。如果把学生当成“对象”，那么“责备”就是一个发生在对象上的“事件”。那么不同对象对同一事件的反应为什么会不一样呢？这是因为每个对象对每个事件都有一个“事件程序”的缘故，当程序内容不同时，所表现出来的反应自然也就随之不同。

事件程序的句法如下：

```
Sub 对象名称—某一事件()  
事件程序的内容  
End Sub
```

在这里，对象名称可能是窗体或者控制项名称。

所以，如果希望书桌被摸一下抽屉就打开，其事件程序可以这样写：

```
Sub 书桌—摸一下()  
(让抽屉打开)  
End Sub
```

而如下事件程序：

```
Sub 书桌一拍一下()  
(让抽屉合上)  
End sub
```

则会让书桌的抽屉被拍一下后，就合上。

当我们在**Visual Basic**中建立了许多对象后，希望某个对象在收到某个事件之后，能做出预期的反应，就只要在该对象的特定事件程序中编写相对应的程序就可以了。

如果我们先在窗体上安置一个命令按钮，然后在该命令按钮上双击鼠标，就可看到命令按钮的**Click**事件程序(里面是空的)。

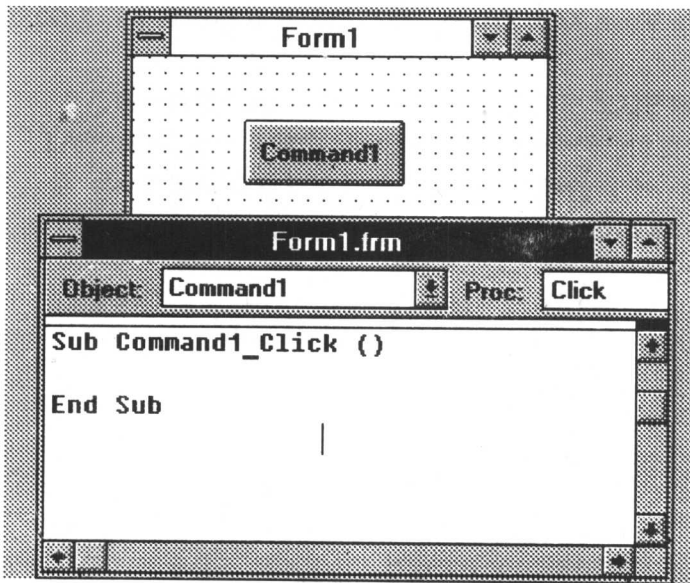


图 1-9

所以，在**Visual Basic**下开发应用程序必须把重点放在“事件程序”上，而不应该象传统**Basic**那样，一行接一行地编写程序。由于“事件程序”通常都很精简，加上其功能非常明确，因此整个应用程序是很容易维护的，同时具备很高的灵活性，这是**Visual Basic**语言的重要特征之一，可以说，**Visual Basic**是“天生结构化”的语言。

1.7 方法 (Method)

当我们设定对象的某项属性时，其实就是把某些数据放入对象之中，而对象本身所内含的函数或过程(注意，这里不是事件程序)就叫做“方法”(Method)。“方法”的内容是不可见的，我们只要用它就行了。当我们用“方法”来控制某个对象时，其实就是执行该对象的某个内部函数或者过程。

例如：`Form1.Hide`是用来隐藏Form1这个对象的，至于它是如何被隐藏的，我们并不在意。另外，`picture.Move Xpos, Ypos`则是把图片框移动到坐标为(Xpos, Ypos)的位置上去。

若想了解Visual Basic 中的对象与方法，可参阅相应的对照表。

到目前为止，我们已经介绍了Visual Basic 中的几个最重要的特征，为的是给Visual Basic 的初学者建立一个新的概念体系，这些对于学好 Visual Basic 是至关重要的。为了进一步灵活地运用这些特征，建议读者把下面的表格形式记在脑子里，将来在开始编写程序时会有很大的帮助。

对象:

属性:

事件:

方法:

1.8 如何扩充Visual Basic 中的对象

Visual Basic 中所提供的对象可以说都是属于基本应用类型的，虽然不是很多，但是只要善加调配，还是能创造出很多对象的。

但是对于象多媒体这类的应用，基本对象就显得有些力不从心了。幸好Microsoft公司还推出了Visual Basic的专业版本，其中包含了许多专业用途的对象。因此可以轻易地扩充程序的功能，这种扩充的对象其实就是所谓的“软件IC”。

由于Visual Basic 的高度扩充性，相信会有许多独立的软件开发公司也会陆续推出各种各样千奇百怪的“软件IC”来，读者也许会想写个属于自己的对象来使用，只是目前尚无法用 Visual Basic 来开发。这种扩充用的“软件IC”，非得使用SDK不可。严格地说，只靠SDK仍然不够，因为还得配合CDK(Control Development Kit)一起使用才行，而这种事情就让那些专业软件开发人员去完成好了，而我们只需买来使用就行了。就象我们要装个收音机，如果连每个电容、电阻都得自己制作，那简直是太辛苦了。

Visual Basic 是以对象为基础向上开发应用程序的语言。而C++则是从底层建立构造出对象，这种方式就象一位“万能”的包工，在盖房子之前还得伤脑筋如何制造钢筋、