

WINDOWS 技术丛书

Visual Basic

编程入门

主 编 刘盛弘 陈阿林
编 著 陈阿林 邵晓希 唐黎



重庆出版社

73.87424
C127

阅览 8 清

windows 技术丛

Visual Basic

编 程 人 门

陈阿林 邵晓希 唐 黎 编著

重 庆 出 版 社

1995 年 · 重庆

(川)新登字 010 号

责任编辑 周定国

封面设计 江 东

技术设计 聂丹英

陈阿林 邵晓希 唐 黎 编著

Visual Basic 编程入门

重 庆 出 版 社 出 版、发 行(重庆长江二路 205 号)

新 华 书 店 经 销 重 庆 印 制 一 厂 印 刷

*

开本 787×1092 1/16 印张 15.25 字数 380 千

1995 年 9 月第一版 1995 年 9 月第一版第一次印刷

印数:1—10000

*

ISBN 7-5366-3162-6/TP·19

定价:24.00 元

引言

§ 1. BASIC 语言发展简介

电子计算机对人类科学技术的进步产生了巨大的影响，而计算机编程语言“BASIC”对计算机应用创立了不可磨灭的功绩。

自60年代编程语言“BASIC”出现以来，便以它短小精干、易于学习掌握的特点，获得了广大计算机用户的喜爱，遂成为学习掌握计算机的标准语言。随着计算机科学技术的进步，各种编程语言有了很大的发展，BASIC语言也从基本BASIC发展到了80年代初IBMPC 标准配置语言GW-BASIC，以及80年代中期出现的集成化开发环境 (IDE) 的Quick BASIC、TRUE-BASIC、TURBO-BASIC等。为了更便于用户使用计算机，80年代后期计算机行业提出了图形用户界面 (GUI) 的概念，由美国APPLE计算机公司开创微机GUI和Microsoft公司推出了视窗环境WINDOWS。

GUI为计算机用户提供了一个轻松直观、图形丰富、操作简便的工作环境，使操作者更易于学习和使用。在视窗环境下，用户常仅需使用鼠标器，便能轻松地操纵机器，完成相应工作，屏幕的多窗口特征可使用户同时运行多个程序以及它们之间相互通讯。

从Microsoft在1990年5月推出WINDOWS 3.0到目前5年的时间内，WINDOWS在全世界已达到5000万套装机量，成为了GUI事实上的标准。WINDOWS环境对用户说来是方便易用，而对编程人员而言，用传统常规的编程语言便难以完成高质量应用程序的编写。为此，MS公司在1991年开发了Visual Basic编程语言。Visual Basic (可视 Basic) 是一个面向对象的集成开发系统，它继承了 Basic的简单、高效、易学功能，以Quick-Basic为它的基本子集，融入了软件开发的最新技术，如：面向对象、设计过程可视、事件驱动、动态数据交换等，使 Basic语言编程技术步入了一个更高的境界。

使用Visual Basic编程语言，可使一般程序员甚至普通用户编写出，以前只有专业水平程序员才能写出的窗口应用程序。为此，Visual Basic在1992年、1993 年连续获得程序设计工具大奖。

§ 2、VB 编写程序步骤：

程序运行,概括地讲可分成两大部份:用户界面交互和事件处理。前者在传统程序中可分为“输入—处理—输出”模式,这里,输入和输出称之为“人机接口界面”,亦称为“用户界面”,“处理”则是对应于不同输入情况或者不同事件而进行的,故可以抽象地视为事件驱动。传统编程方法为获取良好的用户界面,如:下拉菜单、水平垂直滚动条、对话框选择框等将花费大量精力,而现代编程方法,则视这类常用的人机界面元素为一个一个独立的对象,在VB中称它们为“控件”,直接提供给用户使用。

VB采用了面向对象的程序设计思想以及事件驱动工作方式,它将传统程序过程中的各个实体,抽象为一个封装好的对象,用一组数据来描述它的特征,称之为属性;并且支持一组对此对象施加的操作;正如大家使用的电视机一样,我们并不关心它的内部结构,只用其尺寸大小、黑白或彩色、频道多少等等来描述它,而仅用开关或遥控器来操作它一样。

VB提供了一组工具,将应用程序界面上所用的界面元素抽象成VB称之为“控件”,每个控件都是对象,类似WINDOWS界面的窗体也是一种对象。是对象便有其属性,不同控件属性名称和种类有相同的部分也有不同的部分,每个控件可以施以不同的操作(即方法或事件)。这样,VB的程序设计便可由以下三步完成:

设计用户界面

在屏幕窗体上,直观地建立相应的控件(它仅是用鼠标器从工具箱中选出所需控件,然后放置在窗体的某部份),窗体上的这些控件的设置便是应用程序真实运行时用户界面上的对象元素。

设置控件属性

对窗体中选取的控件设置相应的属性,如标题、层次、运行时否是显示、条件、色彩等,控件的特征可随属性的改变立即反应在正在设计的屏幕窗体上。本步骤也可在建立每一控件时完成。

编写代码

VB采用了事件驱动的编程机制,具体说是对应于传统程序中某一输入,反应在VB中便为某控件的事件产生,如对“命令键”的一次鼠标按键行为,一旦捕获到这些事件产生,VB便调用相应事件处理代码来作出反应,完成对应的动作。

事件驱动如同是对应传统编程中的“处理”过程的编程。虽然可用这种比喻来描述VB代码编写,但绝不能视其等同。事实上,事件驱动程序所编写的代码仅解决该事件产生时响应的动作程序,界面上可能有多个事件,其产生先后是无序的,将形成处理的多路开关式流程(由VB系统自身控制)。在这点上,与传统程序设计方法(如:分枝、循环、函数调用等)有着本质不同的风格。

通过以上三个步骤,便完成了应用程序的开发,由于第一、二步中设计的界面便是实际程序运行的界面,且设计直观,故称可视;第三步,与传统编程方法变化不大,是对应一个小的事件驱动。这样,程序员不必编写大型程序,而是创建一个由若干个微小程序组成的应用程序,它们由用户启动的事件激发,完成大型程序的功能。

内容简介

凡是学过Basic语言的读者，都能毫不费力地掌握 Visual Basic。然而Visual Basic（简称 VB）既保留了传统Basic语言的易学、易懂、方便的特性，又引进了可视编程的新技术和结构化事件驱动模式，使其功能更加强大，编程更加简便。此外还提供了许多高级功能，如对象的嵌入和连接OLE、动态数据交换DDE、开放数据库连接ODBC等等，以支持复杂的应用环境下的程序开发。更重要的是，VB被人们称为“可视 Basic”，使用它时，即可以在屏幕的窗体上直观地建立控件，又可以使程序的运行直观、形象、简便，只要你创建了一个若干微小程序组成的应用程序，就能完成大型程序的功能。它确实使编制程序的工作变得十分简单，一般学习过Basic语言的计算机操作员都能胜任愉快。因此，它连续几年获得了实用程序设计工具大奖。

本书是帮助读者掌握 VB 的工具性读物，是一本学习和使用VB的极好的入门教材，它以简明的语言、丰富的实例，详细而系统地讲解了 VB 的基本语句，程序设计和调试的方法，文件的建立和处理，作图语句的使用，输入与输出等等，一步步把读者引入VB编程技术的新天地，使读者用较短的时间获得 VB 编程技术的知识和技能。

作者还以其丰富的实践，为读者提供了一些可资参考的实用程序，使编程的工作变得更加轻松愉快。

目 录

第一章 Visual Basic 基本概念	1
§ 1.1 VB的工作环境安装、启动及退出	1
§ 1.2 VB的屏幕界面	2
§ 1.3 控件及对象	6
1.3.1 控件	6
1.3.2 控件的建立	7
§ 1.4. 对象(object)	9
1.4.1 对象的属性	9
1.4.2 对象事件	10
1.4.3 对象方法(METHOD)	11
§ 1.5 实例设计	12
第二章 基本语言	16
§ 2.1 VB的基本数据类型	16
§ 2.2 常量与变量	17
2.2.1 常量	17
2.2.2 变量(Variable)	19
§ 2.3 变量作用域(SCOPE OF VARIABLE)	21
2.3.1 V B 的程序结构	22
2.3.3 变量作用域	23
1. 局部变量	23
2. 窗体变量和模块变量(Form Variable and Module Variable)	23
3. 全局变量(Global Variable)	24
§ 2.4 自定义类型(USER-DEFINDE TYPES)	25
1. Type语句	25
2. Deftype语句	26
§ 2.5 数组	27
§ 2.6 基本语句、表达式和运算符	28
2.6.1 赋值语句	28
2.6.2 注释	28
2.6.3 多重语句	29
2.6.4 表达式及运算符	29
§ 2.7 结构控制语句	30
§ 2.8 函数和过程的概念	39

1. 数学函数.....	40
2. 字符串函数.....	40
3. 转换函数.....	41
4. 日期函数.....	41
§ 2.9 方法.....	42
§ 2.10 空循环和DoEvents函数.....	43
第三章 Visual Basic的控制对象	45
§ 3.1 对象概述.....	45
3.1.1 对象属性.....	45
3.1.2 对象事件和方法.....	54
§ 3.2 窗体.....	56
3.2.1 窗体属性.....	56
3.2.2 窗体事件.....	56
3.2.3 窗体方法.....	58
§ 3.3 图片框和图像框.....	59
3.3.1 属性.....	59
一、图片框属性.....	59
二、图像框属性.....	60
3.3.2 事件.....	61
3.3.3 方法.....	61
§ 3.4 标签.....	62
3.4.1 标签属性.....	62
3.4.2 标签事件.....	63
3.4.3 标签方法.....	63
§ 3.5 文本框.....	63
3.5.1 属性.....	63
3.5.2 文本框事件.....	66
3.5.3 文本框方法.....	67
§ 3.6 命令按钮、复选框和单选钮.....	67
3.6.1 按钮属性.....	67
一、命令按钮属性.....	68
二、复选框属性.....	68
三、单选按钮.....	69
3.6.2 按钮事件.....	69
3.6.3 按钮方法.....	69
§ 3.7 框架.....	69
3.7.1 框架属性.....	69

3.7.2 框架事件.....	69
3.7.3 框架方法.....	69
§ 3.8 列表框	69
3.8.1 列表框属性.....	69
3.8.2 列表框事件.....	73
3.8.3 列表框方法.....	73
§ 3.9 组合框	74
3.9.1 组合框属性.....	74
3.9.2 组合框事件.....	75
3.9.3 组合框方法.....	76
§ 3.10 水平滚动条和垂直滚动条	76
3.10.1 滚动属性.....	77
3.10.2 滚动事件.....	77
§ 3.11 计时器	77
3.11.1 计时器属性.....	78
3.11.2 计时器事件.....	79
3.11.3 计时器方法.....	79
§ 3.12 线和形状	79
3.12.1 线和形状属性.....	79
一、线的属性	79
二、形状属性	79
3.12.2 线和形状事件.....	81
3.12.3 线和形状方法.....	81
§ 3.13 驱动器列表框、目录列表框和文件列表框	81
3.13.1 属性.....	82
一、驱动器列表框属性	82
二、目录列表框属性	82
三、文件列表框属性	83
3.13.2 事件.....	84
3.13.3 方法.....	84
§ 3.14 菜单	84
3.14.1 菜单的建立.....	85
3.14.2 菜单的控制.....	90
3.14.3 动态增减菜单项.....	91
第四章 输入与输出	93
§ 4.1 InputBox 函数.....	93
§ 4.2 MsgBox 函数.....	95

§ 4.3 Shell 函数	100
§ 4.4 Print 方法	103
§ 4.5 MOVE方法	107
§ 4.6 CurrentX, CurrentY属性	108
§ 4.7 TextHeight 和 TextWidth 方法	111
§ 4.8 字体设置	114
§ 4.9 打印输出	116
4.9.1 直接输出	116
4.9.2 窗体输出	118
§ 4.10 程序代码打印	119
4.10.1 直接打印	119
4.10.2 间接打印	120
§ 4.11 对象控制顺序	121
第五章 通用过程和用户自定义函数	124
§ 5.1 通用过程	124
§ 5.2 自定义函数 (Function Procedure)	127
§ 5.3 作为参数的对象	128
第六章 程序调试	130
§ 6.1 V B 的三种模式	131
§ 6.2 调试窗口	132
6.2.1 输出变量及属性值	133
6.2.2 输出到调试窗口	134
6.2.3 输入变量值及属性值	134
6.2.4 过程, 函数调试	136
§ 6.3 调试工具	137
6.3.1 断点和Stop语句调试	137
6.3.2 即时观察调试	138
6.3.3 调用调试	140
6.3.4 单步语句调试	141
6.3.5 单步过程运行调试	142
6.3.6 跳跃运行调试	143
§ 6.4 错误处理	144
6.4.1 错误处理子程序	144
6.4.2 Err, Err!, Error¥函数	147
6.4.3 Error语句	148
第七章 文件处理	151
§ 7.1 文件的基本概念	151

7.1.1 数据、文件及文件名.....	151
7.1.2 文件的长度、记录与指针.....	152
1. 文件长度.....	152
2. 记录(Record).....	152
3. 文件指针.....	152
7.1.3 文件的打开和读/写.....	152
7.1.4 文件种类.....	153
1. 程序文件(Program File):.....	153
2. 数据文件(Data File):.....	153
§ 7.2 文件操作的界面设计.....	154
7.2.1 驱动器列表框.....	154
7.2.2 目录列表框.....	155
7.2.3 文件列表框.....	155
7.2.4 驱动器列表框、目录列表框和文件列表框的联合使用.....	156
7.2.5 其他文件有关属性.....	158
7.2.6 建立一个实例.....	160
§ 7.3 顺序文件.....	165
7.3.1 建立顺序文件.....	165
7.3.2 读文件.....	168
7.3.3 修改顺序文件.....	172
§ 7.4 随机文件(Random File).....	176
7.4.1 随机文件的建立与写/读.....	176
7.4.2 随机文件中记录的修改、删除和增加.....	179
§ 7.5 二进制文件.....	180
第八章 作 图	181
§ 8.1 坐标系统.....	181
§ 8.2 作图.....	186
8.2.1 清除(Cls).....	186
8.2.2 画点(Pset).....	186
8.2.3 画线(Line).....	188
8.2.4 画圆(Circle).....	192
8.2.5 画椭圆(Ellipses).....	193
8.2.6 画圆弧(arcs).....	194
§ 8.3 颜色.....	196
8.3.1 RGB函数.....	196
8.3.2 QBColor函数.....	197

8.3.3 颜色常数.....	199
8.3.4 颜色值.....	200
§ 8.4 图像的装入.....	200
§ 8.5 鼠标事件.....	201
8.5.1 MouseDown事件.....	201
1. 按钮参数.....	201
2. 换档参数.....	203
3. 鼠标参数.....	205
8.5.2 MouseUp事件.....	206
8.5.3 MouseMove事件.....	207
§ 8.6 拖放.....	208
8.6.1 DragMode属性.....	208
8.6.2 DragDrop 事件.....	209
8.6.3 DragIcon 属性.....	210
8.6.4 DragOver 事件.....	210
8.6.5 Drag 方法.....	212
第九章 应用实例	214
§ 9.1 整体设计.....	214
§ 9.2 菜单设计.....	214
§ 9.3 窗体设计.....	216
§ 9.4 编码.....	222

第一章 Visual Basic 基本概念

§ 1.1 VB的工作环境安装、启动及退出

1991年Microsoft公司推出VB时，提供DOS和WINDOWS两种版本，随WINDOWS 3.0/3.1的巨大成功，Microsoft公司VB的后续版本，主要针对WINDOWS，本书介绍WINDOWS版本下的Visual Basic。为适应国情，本书例子均以Microsoft公司的PWIN为工作环境，当然在中文之星，双桥WIN等汉化WINDOWS环境下，VB都能正常运行。

Visual Basic与大多数WINDOWS应用程序一样，通过VB #1盘上的安装程序SETUP.EXE进行安装。安装分为二种情况，标准安装和定制安装，前者是将包括开发工具、实例、肖像库等全部安装到硬盘，最大所需空间约13M，后者可根据用户自己的需要作部份安装，最小空间约5M硬盘空间。

安装方法可以从WINDOWS程序管理器（PROGRAM MANAGER）的FILE菜单项中选RUN功能，并将VB 1#盘插入A驱动器，在命令对话框中输入A: SETUP.EXE 并按回车，安装程序便启动。首先，出现欢迎词“Welcome”，然后要求输入本软件合法持有人的用户名及公司名，接下来提问VB安装磁盘和目录，最后进入安装程序。安装过程中按系统提示依次插入磁盘并按回车或选OK按钮，系统自动将全部文件安装到VB子目录下（见图1-1），还将相应的动态连接库（.DLL）等文件安装在WINDOWS\SYSTEM子目录中。

安装完毕后，我们将会看到在程序管理器窗口中加入了一个新的VB程序组图标。

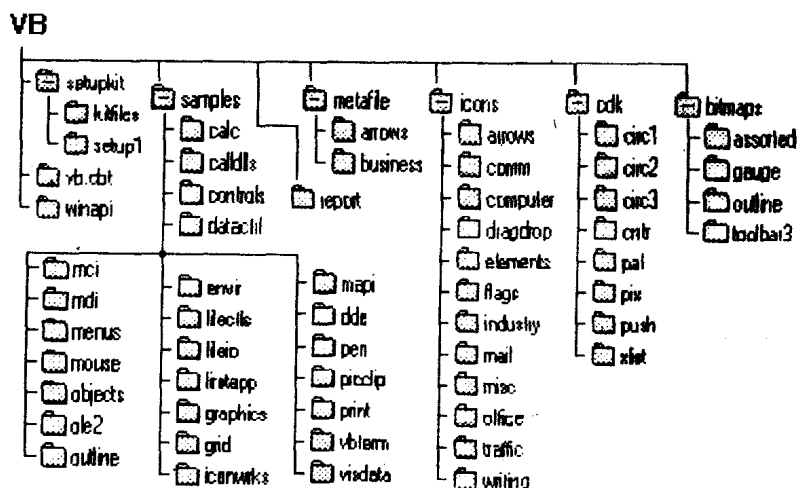


图 1-1 VB的多级目录

VB的启动方法有几种：在DOS提示符下输入WIN VB并按回车键；在WINDOWS文件管理器直接运行VB.EXE；或直接在VB窗口内双击VB的图标即可启动（VB 组图标下标有Visual Basic2.0/3.0 执行图标）。

在VB系统内调用主菜单上FILE项中的EXIT选项，便可退出VB，返回WINDOWS 环境，如果当前环境下的设计没有存盘，系统将提示是否将其存盘，然后再退出。

§ 1.2 VB的屏幕界面

VB启动后，屏幕上出现五个窗口，如图 1-2

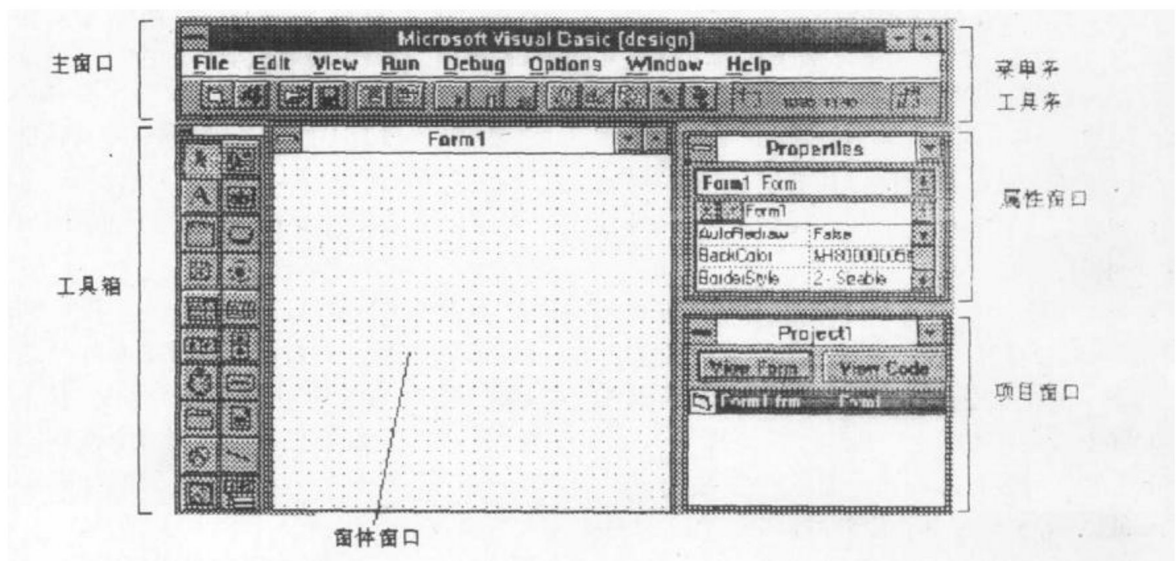


图 1-2 VB 屏幕布局

屏幕顶部是主窗口，它类似于一般典型 WINDOWS的菜单条及工具条；屏幕中间的是窗体（Form）窗口，为VB可视设计的主要区域；左边出现的是称为工具箱的窗口；右边的窗口称为属性（Properties）窗口；右边下面的窗口称为项目（Project）窗口。这些窗口并非不变，可改变它们的位置、大小、或将其关闭（在需要时再打开）。

主窗口

主窗口包括有8个下拉菜单项的菜单条和提供快捷键的工具条组成，用于管理整个系统，负责项目或程序的打开、运行、调试、保存等基本工作。

菜单项功能

FILE 项：用于打开、保存或删除窗体（FORM）文件、项目（PROJECT）文件；打印源程序、编译源程序为WINDOWS下的可执行文件（.EXE）及退出VB等操作；由于用VB建立的源程序

(.FRM、.BAK等)为非普通的文本格式,其他编辑器无法读出,必须用本选项的“Save text”将VB的源程序保存为文本格式(.TXT)和读取其他BASIC语言以文本方式建立的文件。另外本选项还可增加用户自己开发的控件(.VBX)。

EDIT 项:用于窗体或程序的编辑工作,提供了剪下(CUT)、拷贝(COPY)、粘贴(PASTE)、查找(FIND)等功能;还具有动态调整窗体网格点的距离以及进行动态数据交换(DDE)时的拷贝(COPY)和粘贴(PASTE LINK)功能

VIEW 项:调源程序窗口(CODE),以便编辑源程序;还用于工具条的开/关;以及建立普通或私有过程(PROCEDURE)。

RUN 项:以解释方式执行VB程序或停止程序执行。

DEBUG 项:提供源程序调试功能,如设置/取消断点、单步执行、或观测点等等。

OPTIONS 项:用于VB的工作环境设置。

WINDOWS 项:用来调整窗口布局,选择菜单设计窗口,还可以装入或关闭调色板、工具箱等。

HELP (帮助)项:提供完整的在线式帮助。

为了方便使用,VB设置了与菜单条中的最常用选项等价的快捷按钮组成的工具条,它由多个按钮组成,各按钮功能都可以在菜单中找到,按钮的功能如图1-3。

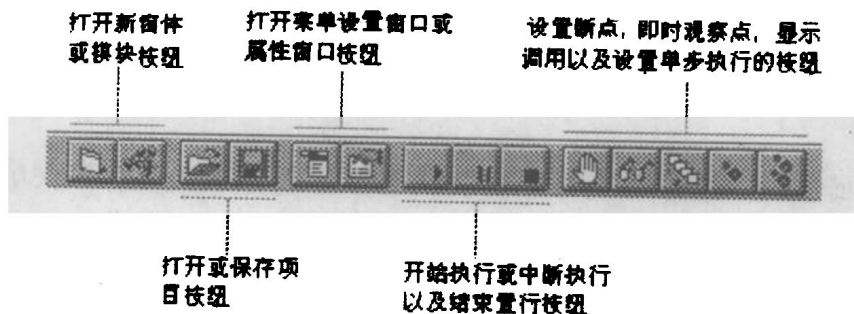


图 1-3 VB 工具条

窗体窗口

窗体(FORM)是对应于用VB开发的应用程序运行时所见的窗口显示区域,即我们平时讲的用户界面,是进行设计中重要的工具。各种图形、图像、数据等都是通过窗体中的控件显示出来的。在设计阶段它就像一块画布,设计人员依照程序界面要求,从工具箱窗口中选择要用的工具,在窗体中画出来,就完成了设计的第一步。注意,当VB启动时,它将自动地创建一个名为FORM的空窗体。

工具箱窗口

VB是面象对象编程语言。这里的对象指的就是工具箱中的工具，VB中称之为控件。这些控件是封装好的，它的属性及能够接收的事件和方法也是系统定义好的，设计人员只要在程序中使用就可以了。例如WINDOWS环境中常见的按钮在VB就是一种控件——命令按钮。如果应用程序中需要一个按钮，在设计阶段从工具箱中选择命令按钮并在窗体中画出来，就建立了一个命令按钮。程序运行时，它就可以接收用户对它的键盘或鼠标操作。

工具箱中的控件名称如图 1-4，(A)图是VB基本的控件工具箱、(B)扩充了控件的工具箱。(加入了·VBX模块，VB 2.0/3.0 使用)

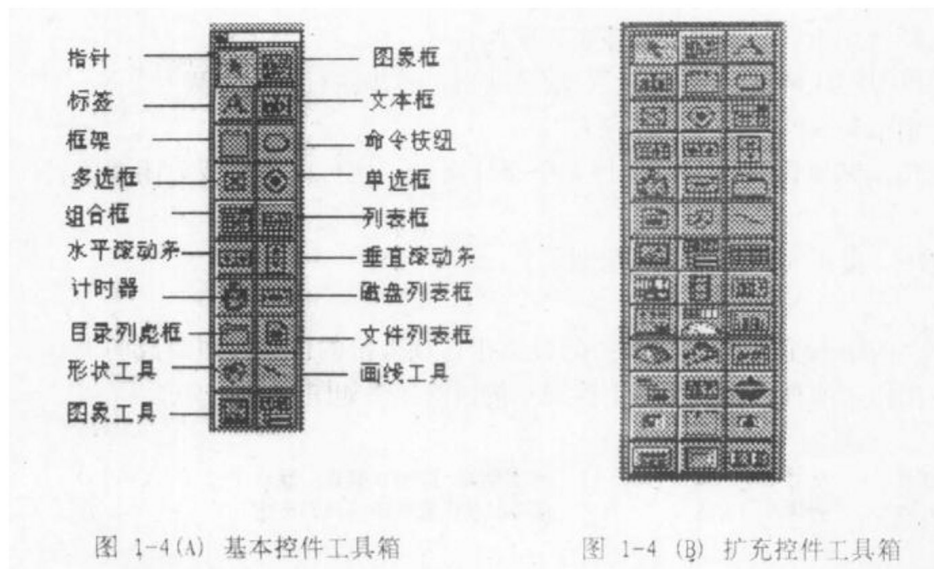


图 1-4 (A) 基本控件工具箱

图 1-4 (B) 扩充控件工具箱

VB允许程序员自己设计新的工具加入VB工具箱，也允许在工具箱中加入第三方软件开发商提供的控件工具。



图 1-5 VB 属性窗口

属性窗口

属性窗口如图 1-5。它是针对窗体中控件或窗体的。面向对象设计思想中,对象要有一组属性来刻画其特征。VB中的属性窗口就是用来设置窗体或窗体中控件属性的。一个控件就有一组属性。改变属性就可以改变控件的某些特征(外观或行为),程序员不必为控件的每一属性进行设置,许多属性可以用它的缺省值。

项目窗口

包含正在编写或正在运行的VB程序所需文件清单。VB程序文件一般有几类,即以 .FRM、.BAS、.MAK、.VBX为扩展名的文件。

.FRM文件为窗体对应的文件,该窗体的属性及其它信息都存于该文件中。一个应用程序可以有多个窗体,因此就可以有多个以 .FRM为扩展名的窗体文件。

.BAS文件是为了合理组织程序而设的,特别用于变量的声明。它不针对任何一个窗体,是一个纯代码性质的文件,在大型应用程序中是不可缺少的。它是用单击工具击中的新模块按钮产生的。

.MAK文件也称为项目文件(Project File)或用户程序文件。它连接该文件包含有 .FRM、和 .BAS、.VBX文件的信息,一旦用户选用菜单条「File」下的「Open」选项,或按打开按钮时,若不特别指明,VB将打开 .MAK文件。它连接的所有文件将全部读入内存,在项目窗口上,用户可看到读入内存的全部文件,同时当用户选用菜单条「File」下的「Save」选项或按保存按钮时,VB会将项目窗口上的全部文件,按类(如 .FRM、.BAS)和相应名称加以保存。

.VBX是扩展控件的支持文件。VB考虑到有些部门可能需要一些特殊控件,但对其它用户来说可能是不常用的。因此VB提供了用 .VBX扩展文件来增加控件的技术。它允许用户设计新的控件,并把它们加到工具箱中,这样设计时就可以使用一些比较特殊的控件。对于这些控件,VB本身没有支持文件,因此,就要求新的控件(即以VBX为扩展名的文件)提供支持。有一个新控件就有一个支持文件与之对应。图 1-6是加入了多个 .VBX模块的项目窗口。



图 1-6 Project 窗口