

引言

欢迎步入程序设计艺术的殿堂。如果你在微机上的大部分时间都是在编写文档和组织电子表格,那么你已经以用户的身份同程序打过交道了。每个程序都为你提供了各种操作,大部分时间里你是在处理已输入计算机的数据。用户同程序的交互过程和方式都是由程序员事先设计好的。交互过程是通过用户和计算机之间的对话来完成的,使用数据作为对话过程的通信媒介。

Microsoft Windows 为程序提供了一个充分展现自己的舞台,Windows 的可视资源——各种按钮,配件,以及各种指针——都是这个舞台上的道具。实际上,Microsoft Windows 这个软件本身就是一部上演的好戏,这个软件的目的也正是它的主题。

没有一个鲜明主题的戏不是一部成功的戏,与此类似,没有一个明确的目的或目标就不能构造任何计算机程序。在以往的有关程序设计的书中,作者都尽力教会读者:编写程序的具体内容,而不是教会读者如何去编写程序。

本书正是为了教会读者各种编程技术而编写的,让读者能编写自己风格的程序。程序设计技术并不只是把日常事务翻译成逻辑代码的方法,它还是一种表达工具。应用程序的用户就好比是应用程序这个舞台上的演员。然而,作为程序员,就必须设计好这样一个舞台,编排好用户和计算机之间的对话,并以某种超然的方式指导用户。

为了更好地了解程序员的作用,不妨把程序的各种组成部分——如各种模块,过程,例行程序,以及子句——想象成一部戏的各种角色。换句话说,程序的求和部分,画线部分,管理客户名的部分,以及程序中检测用户是否按了某个按钮的部分都具有不同的功能,尽管处理时是相同的。

然而,用户相当于应用程序这部戏的主角,主角和其他角色之间的对白构成了这部戏的主体。如果各个角色之间不互相对白,他们的个性就无法体现给观众,剧情也无法展开。与此类似,没有程序各个组成部分与用户之间高度的协调作用和数据交换,程序就无法运行,它的目标也不能完全达到。就如剧情混乱,不满的观众就会中途退场一样,如果程序难以使用,用户就会提早放弃它。

因此,程序员的作用类似于剧作家和舞台导演。今天,没有其他程序设计环境比 Visual Basic 更类似于舞台,其中的工作平台,各种控件从某种意义上讲,就是舞台上的道具。程序员可以在屏幕上安排这些东西。然而,软件产品本身仍是一个草本,需要细致的测试和彻底审核。

本书从程序员的角度详细阐述了程序和程序模块的开发过程,向读者介绍了开发一个完美的可执行软件产品所必需完成的工作。

综 述

首先读者需要了解开发程序的平台所具有的功能和特色。在第一部分“Visual Basic 环

境”中,读者将了解构造程序的工作区。读者一旦在新的编程环境中觉得得心应手,就可以着手集中精力学习什么是程序,Visual Basic 程序与传统的 Basic 程序或 C 程序有何不同。

第一部分

在本书第一部分,读者将看到处于不同开发阶段的示例程序和程序模块,并可从中获得开发程序的一些实践经验。在第4章“程序开发过程”中,将向读者演示一个可运行的程序,通过构造一个小的 Visual Basic 应用程序,将使读者了解到程序员要集中考虑什么,程序员要为自己设定什么类型的目标。无论采用何种程序设计语言,任何程序都不能完全证明它的源代码完美无缺。大多数优秀的程序都是程序员努力的结果,它们都经过了大量的查错和调试。

第二部分

当读者对程序设计过程感到轻松自在时,可以开始逐段检查 Visual Basic 的语言和图形元素。由于 Visual Basic 几乎无所不能,新程序员可能认为在程序设计过程中几乎不需自己设计算式。其实不然,运算逻辑和过程代数仍是程序设计的基本组成部分,因此在第二部分“值和变量”中将使用明确的名词和类比方法向读者介绍运算逻辑和过程代数。

第三部分

在第三部分“指令和指令集”中,读者将检查 Visual Basic 的“语言部分”。和英语的语句一样,Visual Basic 指令有它们自己的语法。因此,读者为了更好地了解 Visual Basic 的结构,最好使用你自己熟悉的结构。在程序设计过程中,程序员将编写程序执行的流程。本书这部分阐述了如何把 Visual Basic 程序要完成的全部工作分解为不同的模块和过程。

第四部分

开发 Windows 环境中运行的程序之技术也是一种艺术创作过程,如设计屏幕显示,对话框,以及菜单。你将发现这个过程非常类似于一位美工在设计杂志封面或绘制广告。在第四部分“设计用户界面”中读者将学会这种技术。

第五部分

Visual Basic 语言含有几十个处理图形元素和用户交互过程的术语。将在第五部分“图形对象和控制部件”中介绍它们。

第六部分

第六部分讨论的主题是“运算功能”。可以在 Visual Basic 中使用数学公式,但是使用方法不同于电子表格软件内的数学公式。在电子表格中,单元格内的公式总是不变的,因而具有可预测的结果。但是使用程序设计语言时,值(与单元格相比)是在变化的,即要处理变量的值,因而结果是不固定。

第七部分和第八部分

除了在第五部分介绍的图形控件之外,其他两种主要的数据输出是通过标准文本以及位图式图形窗格来进行的。这些输出方法在第七部分和第八部分分别介绍。第七部分介绍“文本”,第八部分介绍“位图式图形”。

第九部分

Visual Basic 解释程序提供了错误诊断和检测环境。它在程序的开发阶段就能发现程序中的错误,查错和纠错是编写和完善程序的主要工作。第九部分“错误捕获和调试”将介绍纠正错误的技术——即使你已成为一位专家,纠正程序中的错误也可能花费你一半的编程时间。

第十部分

Visual Basic 与诸如 PerForm 和 Oracle 这样的图形数据库管理系统之间的对比仍在继续。尽管所有功能都增强了,VB 为处理记录形式的数据提供的方法仍然是根植于旧的 BASIC 语言上的。第十部分“物理数据”将详细讨论这个专题。

本书的阅读对象

本书假定读者已使用过 Windows,并了解图标、按钮及下拉式菜单的工作原理。同时还假定读者已安装了 Visual Basic 系统,而且该系统可正常运行。然而,本书并不要求读者具有 BASIC 语言的知识或编程经验,除非书中有专门的声明。

本书约定

书中出现的 Visual Basic 词汇中的关键字均特别指出。Visual Basic 中的大多数关键字类似于英语句子中的主语。一条指令中若有一个关键字,这条指令就应遵循一定的语法规则。本书在一个“语法块”中说明指令的语法规则,“语法块”中含有一组关键字和其他符号化的项,用来说明关键字在程序中书写的格式,下面列出一个语法块的示例:

```
[Static] Function procedure_name ([ [ByVal] argument1 ] ( )  
    [As type], [ByVal] argument2 [ ( ) [As type] ... ]  
    instruction_block1  
    procedure_name = expression1  
[Exit Function]  
    [ instruction_block2 ]  
    [ procedure_name = expression2 ]  
End Function
```

关键字 Function 的语法块是本书中较复杂的一种。关键字在语法块中以单空格字体标识。单空格斜体字标识的项表明:该项可由其他项或指令替换。例如,语法块中第三个位置上的 *procedure_name* 表示可用任意的过程名代替它,比如用 GetHelp 也可作为过程名,它的作用是使读者一目了然。

出现在方括号内的项表明它的内容是可选的,如 [Static]。虽然 Function 语句既可含有 Static,也可不含 Static,但是两者的含义是不同的。当在语句中加入 Static 时,并不需要加入方括号。在语法描述中使用方括号是计算机书籍中的一种习惯。然而,Visual Basic3 解释程序现在认为方括号是 Visual Basic 语言的一部分(或一种语句形式)。因此在语法描述块中,需要程序员在程序中输入的方括号,都以黑体字示出。

在其他语法描述块中,若有两项或多项出现在花括号内并用竖线分隔开——如 {either | or}——这表示在程序中实际只要选择其中一项。在子句的语法示例中,省略号(…)表示此

处可以加入任意数目的指令。最后,读者要记住在 VB 指令或子句中不要输入语法描述块中的一些符号——如方括号([]),花括号({ }),省略号(…)

伪 代 码

本书用于说明 Visual Basic 语言工作方式的工具之一是伪代码(pseudocode)。伪代码部分用英语说明程序指令要实现什么功能。伪代码以斜体字示出。下面给出两个示例:第一个程序段完全用 Visual Basic 编写,第二个用伪代码解释第一个程序段的功能。

```
For C=1 to 5
    Form1.Print C
Next C
Start counting from 1 to 5.
Print the number you just counted.
Go back to where you started counting and repeat
yourself until you reach 5.
```

上面的伪代码段有点长,但是它的作用不但说明了上述 Visual Basic 源程序试图告诉解释程序什么,而且指明了该源程序的要点。

本书中的许多示例程序模块都用[I],[II],[III]罗马数字来表示它具有的难易程度。程序模块的长短并不能说明它具有的复杂度,假设作者认为书中相应的程序对读者来说有一定的难度,则使用这些罗马数字标识它们。

[I]	Level 1
[II]	Level 2
[III]	Level 3

在本书中,若遇到一条 Visual Basic 指令太长而不能在一行内写下,则在下一行的行首与上一行行首对齐以示是上一行的继续,例如:

```
Declare Function GetSystemDirectory Lib "Kernel" (ByVal lpBuffer As String,  
ByVal nSize As Integer) As Integer
```

在本书的叙述过程中引入重要的新名词,则在楷体字的注释中给出该名词的定义。

在大多数章节的末尾都有一些简短的问题。帮助读者回顾每一章的内容,检测一下读者对这些内容的理解程度。这些问题的解答在附录 B 中。

其 他 约 定

本书内将 Visual Basic 缩写成 VB,VB3 表示 Visual Basic3.0 版。

在本书中,一些关键字前面带有标点符号,通常用来表示它们在 VB 指令中具有的特殊含义。书中将采用 Load 事件,Count 属性或 Sub Form1_Click()过程这类表示形式,其中的符号.,以及()是 Visual Basic 的基本符号,使用它们有助于区分关键字的不同作用。

Microsoft Windows 运行在分辨率为 1024×768 的标准 VGA 显示器上。由于 Windows 视频显示器的驱动程序存在稍微不同的一些变种,特别是分辨率上的差别,因此用户屏幕上

显示的形式可能和本书示出的显示形式稍稍不同。

在以前的 Windows 环境中编程时,程序员需要自己输入几页源程序——文本和数字,才能得到一个可运行的程序。然而时至今日,程序员完全不用输入 Visual Basic 程序。图形对象的位置和特性可用鼠标来确定,程序员实际要做的只是在 Windows 内绘制这些控制部件。本书中列出的任何一个完整的源程序都有一个表格,表中说明了该程序内使用的图形对象的名字,还包括图形对象的属性或特性——诸如位置,大小,颜色,以及下标值——这些属性是将缺省设置修改后得到的。程序员可以根据表中列出的内容自己组装这些控制部件。

虽然 Visual Basic 自己也是一个程序,但是程序员可以使用它来辅助编程。如果读者看到“Visual Basic 程序”这个短语,但不知道它是指用 VB 语言编写的程序还是指 Visual Basic 这个系统程序,这必然会引起混乱。因此,本书把用 VB 语言编写的程序称为“Visual Basic 应用程序”,而把 Visual Basic 这个软件称为“Visual Basic 程序”。

当用户在图形环境中工作时,要进行各种操作,为了不至于混淆,对各种具体操作采用标准化的动词进行叙述。如选定菜单项(selected),选择列表框内的表项(chosen),指定窗体或窗口内的数据(indicated),单击按钮(clicked)。另外,对于在文本字段内手工键入设置、属性或文件名采用动词设定(assignment)来叙述。所选择的这些术语在整个书中都不再变动。

标准鼠标均用左键选择屏幕上的对象,然而 Microsoft Windows 不为左撇子用户考虑,使用 Control Panel(控制面板;Windows 主群中的一个应用程序——译者注)可以重新设置鼠标按键的功能,使左右按键的作用互换。因此只要用户愿意,鼠标右键均可作为左键使用,本书使用的术语:索引按钮(index button)即指用于选择对象的鼠标按钮。

文中所用的术语“光标”即指在输入正文时用于指示文本位置的闪烁的竖线或方块。指针(pointer)即指鼠标移动时屏幕上显示的箭头或符号。

目 录

第一部分 Visual Basic 环境

第 1 章 程序是什么	1
1.1 程序员的信条	1
1.2 基本术语	2
1.3 Visual Basic 解释程序	3
1.4 交互过程	4
1.5 计算环境	5
1.6 小结	6
1.7 复习题	7
第 2 章 工作区	8
2.1 Visual Basic 项目程序	8
2.2 图形对象	11
2.3 属性	14
2.4 Name 属性	14
2.5 保存已编写好的程序	16
2.6 小结	17
2.7 复习题	17
2.8 练习	17
第 3 章 语法和语言学	18
3.1 现代语法	18
3.2 表达式	19
3.3 内部函数	20
3.4 程序成分	20
3.5 其它程序成分	23
3.6 Visual Basic 符号	25
3.7 小结	27
3.8 复习题	27
第 4 章 程序开发过程	28
4.1 按程序清单输入应用程序	28
4.2 构造应用程序	29
4.3 构造 Expressor 程序	29
4.4 构造你自己的程序	31
4.5 小结	35
4.6 复习题	35

第二部分 值和变量

第5章 使用变量	36
5.1 声明变量	36
5.2 精度	39
5.3 内存保留	40
5.4 小结	41
5.5 复习题	41
5.6 练习题	41
第6章 计算公式	42
6.1 公式的作用	42
6.2 求值次序	43
6.3 使用括号	43
6.4 小结	47
6.5 复习题	47
6.6 练习题	47
第7章 逻辑和数字系统	48
7.1 布尔逻辑	48
7.2 更高级的比较	49
7.3 使用逻辑	51
7.4 布尔运算符	52
7.5 数字系统	54
7.5.1 计算机内的逻辑	54
7.5.2 二进制系统和十进制系统	54
7.5.3 进制转换过程	55
7.6 小结	56
7.7 复习题	56
第8章 字符串	58
8.1 字符串值	58
8.2 ANSI, ASCII 及其它首字母缩写词	59
8.3 小结	66
8.4 复习题	67
8.5 练习题	67
第9章 数组变量	68
9.1 数组的作用	68
9.2 为数组变量定维	68

9.3	规整数组	70
9.4	为数组定界	72
9.5	确定数组下标下界的起始值:0 或 1	72
9.6	小结	73
9.7	复习题	73
9.8	练习题	73
第 10 章	值的声明	74
10.1	结构化程序设计	74
10.2	变量的作用域	75
10.3	常数	78
10.4	复合变量	80
10.5	小结	82
10.6	复习题	82
10.7	练习题	82
 第三部分 指令与指令集 		
第 11 章	指令构成	83
11.1	语句和函数	83
11.2	程序注释	85
11.3	小结	86
11.4	复习题	86
11.5	练习题	86
第 12 章	条件子句	88
12.1	子句的构造	88
12.2	If-Then 作为双边变量	90
12.3	Select Case 语句	92
12.4	小结	95
12.5	复习题	95
12.6	练习题	96
第 13 章	循环子句	97
13.1	Do-Loop 子句	102
13.2	小结	103
13.3	复习题	104
13.4	练习题	104
第 14 章	转移指令	105
14.1	转移语句	105

14.2	程序上下文.....	108
14.3	子程序.....	108
14.4	连续转移语句.....	110
14.5	小结.....	111
14.6	复习题.....	111
14.7	练习题.....	112
第 15 章	模块分层	113
15.1	模块的起源.....	113
15.2	执行的次序.....	115
15.3	窗体模块.....	117
15.4	Sub Main()过程.....	118
15.5	DoEvents 语句	119
15.6	End 语句.....	119
15.7	小结.....	121
15.8	复习题.....	122
第 16 章	参数传递	123
16.1	参数传递次序.....	124
16.2	小结.....	126
16.3	复习题.....	126
 第四部分 设计用户交互方式 		
第 17 章	控件的标识和内容	127
17.1	图形对象分类.....	127
17.2	使用别名.....	128
17.3	控件数组索引.....	129
17.4	对象事件.....	129
17.5	用户能编辑的和不能编辑的.....	131
17.6	什么时候真正执行命令?	133
17.7	小结.....	134
17.8	复习题.....	135
第 18 章	设计与布局	136
18.1	Twip 坐标系	136
18.2	定位控件和调整控件大小.....	136
18.3	设计辅助计算.....	138
18.4	使用 Tab 键	139
18.5	监视活动性(Monitoring Activity)	140
18.6	使用颜色.....	142

18.7	小结	143
18.8	复习题	143
第 19 章	作为窗体的窗口	144
19.1	窗体的位置	144
19.2	状态间的转换	145
19.3	了解控件的名字	146
19.4	窗体剖析	147
19.5	状态确定	152
19.6	小结	154
19.7	复习题	154
第 20 章	文档窗口	155
20.1	世界里的世界	155
20.2	父窗体	155
20.3	重新排列窗口	160
20.4	小结	161
20.5	复习题	162
第 21 章	面板窗口	163
21.1	框和面板	163
21.2	输入框	165
21.3	小结	166
21.4	复习题	167
第五部分 图形对象和控件		
第 22 章	菜单和选项	168
22.1	菜单	168
22.2	菜单制作	169
22.3	在窗体中组织控件	172
22.4	在源代码中组织控件	175
22.5	小结	177
22.6	复习题	177
第 23 章	管理列表框	178
23.1	使用 Visual Basic 控件	178
23.2	列表框与组合框	178
23.3	多重选择	185
23.4	小结	187
23.5	复习题	187

23.6 练习题	188
第 24 章 存储与打印机控件	189
24.1 目录辅助	189
24.2 一个文件选择框应用程序	190
24.3 _Change 事件的分支	192
24.4 公用对话框	193
24.5 文件公用对话框	194
24.6 Print 公用对话框	195
24.7 字体公用对话框	196
24.8 小结	196
24.9 复习题	196
第 25 章 形状与直线控件	197
25.1 形状与直线	197
25.2 面向对象的图形	201
25.3 小结	201
25.4 复习题	202
第 26 章 滚动条	203
26.1 多种作用的范围探测器	203
26.2 小结	206
26.3 复习题	206
第 27 章 计时器	207
27.1 实时的尝试	207
27.2 小结	209
27.3 复习题	209
第 28 章 键盘	210
28.1 用按钮代替键	214
28.2 为按钮分配键	216
28.3 小结	217
28.4 复习题	217
第 29 章 鼠标指针设备	218
29.1 移动何时有效	221
29.2 指针形状	222
29.3 小结	223
29.4 复习题	223
第 30 章 拖放控件	224

30.1	锁和钥匙的例子.....	229
30.2	移动控件.....	232
30.3	小结.....	233
30.4	复习题.....	234
30.5	练习题.....	234
第 31 章	网格与表格	235
31.1	网格结构的分析.....	235
31.2	使网格工作.....	239
31.3	小结.....	243
31.4	复习题.....	243
第 32 章	对象变量	244
32.1	向同一个项增加几个名字标记.....	246
32.2	向对象赋予控件焦点.....	246
32.3	间接引用.....	247
32.4	任意的对象变量.....	249
32.5	小结.....	251
32.6	复习题.....	251
 第六部分 算 术 函 数 		
第 33 章	分类	252
33.1	回顾.....	253
33.2	Variants 的局限性	253
33.3	小结.....	254
33.4	复习题.....	254
第 34 章	转换	255
34.1	字符串转换.....	256
34.2	ASCII 码转换	257
34.3	小结.....	260
34.4	复习题.....	260
第 35 章	实用数学	261
35.1	其它数学函数.....	262
35.2	小结.....	263
35.3	复习题.....	263
第 36 章	字符串函数	264
36.1	String \$ (), LCase \$ () 和 UCase \$ () 函数	268

36.2	字符串匹配.....	270
36.3	小结.....	274
36.4	复习题.....	274
第 37 章	时间和日期	275
37.1	作为文本的时间和日期.....	278
37.2	变体变量如何表示日期.....	279
37.3	小结.....	279
37.4	复习题.....	280
第 38 章	随机数生成器	281
38.1	小结.....	283
38.2	复习题.....	283

第七部分 文 本

第 39 章	常规输出	284
39.1	格式化文本输出.....	287
39.2	小结.....	292
39.3	复习题.....	292
第 40 章	文本属性	293
40.1	字体特性.....	293
40.2	裁剪文本以适合控件以及裁剪控件以适合文本.....	294
40.3	True/False 值的字体属性.....	294
40.4	文本捕获技术.....	295
40.5	充分适合.....	300
40.6	选择和网格.....	301
40.7	小结.....	303
40.8	复习题.....	303
第 41 章	打印机和声音输出	304
41.1	字体选择.....	305
41.2	声音.....	305
41.3	小结.....	306
41.4	复习题.....	306

第八部分 位图式图形

第 42 章	图片和图象框	307
---------------	---------------------	------------

42.1 永久性位图.....	310
42.2 小结.....	311
42.3 复习题.....	312
第 43 章 绘图	313
43.1 坐标系统.....	313
43.2 Windows 中的窗口	314
43.3 着色方案.....	314
43.4 画线.....	318
43.5 线宽和填充.....	319
43.6 小结.....	323
43.7 复习题.....	324
43.8 练习题.....	324
第 44 章 图象的完整性和按比例缩放	325
44.1 完整性属性.....	325
44.2 重新定标.....	326
44.3 小结.....	329
44.4 复习题.....	329

第九部分 错误捕获和调试

第 45 章 登录错误	330
45.1 错误捕获例程.....	330
45.2 小结.....	333
45.3 复习题.....	333
第 46 章 调试技术	334
46.1 使用 Watch Expressions(监视表达式)	336
46.2 什么是调试.....	337
46.3 小结.....	338
46.4 复习题.....	338

第十部分 物理数据

第 47 章 顺序存取	339
47.1 以页为单位进行处理的顺序数据.....	347
47.2 小结.....	348
47.3 复习题.....	348
47.4 练习.....	348

第 48 章 随机存取和二进制存取	349
48.1 数据和记录	353
48.2 小结	354
48.3 复习题	355
第 49 章 数据文件属性	356
49.1 数据处理控制	358
49.2 未格式化的数据输入	360
49.3 系统剪贴板管理	361
49.4 小结	363
49.5 复习题	364
第 50 章 格式记录	365
50.1 启动 NameForm 应用程序	365
50.2 将尾部记录置于中间	369
50.3 排序过程	379
50.4 算法的逻辑	381
50.5 小结	381
50.6 复习题	381
附录 A ASCII/ANSI 码表	382
附录 B 复习题解答	388
附录 C 主要应用程序的全部源代码清单	400

第一部分 Visual Basic 环境

第1章 程序是什么

读者将在本章学习有关程序的概念,同时掌握 Visual Basic 和计算技术。本章向读者解释程序在计算机中的行为,并讲解高级语言程序设计的基本概念。如果读者熟悉程序设计语言的概念及其作用,可以直接阅读第2章。

1.1 程序员信条

程序设计的目标就是为一些处理过程建模或者模拟这些过程,把这些现实世界或构想中的处理过程合乎逻辑地描述为逻辑和运算过程。这些处理过程可以是财会系统的数学模型,也可以是地板铺设的几何模型,或者是游戏的娱乐模型。如果你了解处理过程的原理,那么就已知道计算机程序应该包含的内容。你必须把这个过程重构为数学模型。换句话说,你必须找到完全用数学描述处理过程每一部分的方法。

与计算机相关的内容都是逻辑的,它们是由数字和数学方法组成的。因此,构造一个记帐程序相当容易,理由是记帐过程由数字和数值方法组成的。然而,在制定地板铺设计划和园林规划的程序中,几何学则起重要作用。一些处理过程首先可能并不作为数值来考虑——例如在图纸上绘出将要种植树木的灌木区——而是用程序指令来模拟人的思考过程。

比如考虑一个可辅助医生诊断病情的程序,程序的一部分保存一系列发病的症状和病因,开药方的程序部分将是最容易实现的部分,因为这部分将调用数据库管理程序。数据库是已组织好的记录的集合(例如医生的病人名单)。数据库管理程序是维护数据库信息的程序部分。

* 数据库是已组织好的记录的集合。

* 数据库管理程序是维护数据库信息的程序。

模拟决策过程比维护病症和药方表格困难得多。如果编写这个程序的人不是一位医生,那么他应和医生进行深入的交谈,并学习诊断的概念和过程。程序员必须阅读大量的入门教材,就好像他是一名医科学生。当医生进行临床诊断时,在某些程度上依赖于他们的直觉。必须记住,程序员必须模拟没有直觉的计算机的诊断过程。这项任务不是不可能实现的,可以让程序模拟“记忆”功能,并让医生提供直觉。

假定我们要编写一个下国际象棋的程序。人们可能凭经验认为计算机下得很好,国际大师,如 Gerry Kasparov (卡斯帕罗夫)非常欣赏那些博弈程序下得如此之好(虽然卡斯帕罗夫经常能击败那些程序)。读者可能并不知道博弈程序自己并不能构想赢棋的策略,博弈程序在每一步棋之后重新评估下一步棋的优劣,它们一般在选择一步较优的看法之后就完全抛弃刚刚进行评估的结果。当你和计算机下棋时,你以人类的下棋策略对抗数学的分析过程。

程序员在编写下棋程序时的工作就是构造这个数学分析过程。编写博弈程序并不要求精通棋理,然而要求精通数学分析过程。程序员可能是一位优秀的棋手,但是他不能把自己的下棋策略编写到程序中,因为博弈程序不会思考。但是,他可以在程序中定义下棋的规则。

当构造程序时,若该程序执行的任务不是由人类来完成的话,程序员就应彻底理解这个任务处理过程的基本原理,以及控制这个过程的数学方法,而程序员并不需要精通这项任务的处理过程。换句话说,编写财会程序的人并不要求是一名会计,而编写临床诊断程序的人并不要求是医生。只需简便了解处理过程的基本原理,并在需要细节的时候再去查阅相关资料。但是,编程者应该了解使用程序的用户的职业,以便不至于使他们感到受到侮辱或被程序所激怒。

程序可应用于日常事务时,该程序也就称为应用程序。Visual Basic 是开发应用程序的工具。程序员使用它时,必须记住编程的目标即是自动化——提取现实工作的逻辑和运算部分,例如记录保存或数据分析,然后指派计算机去完成相应的一些处理工作。如果应用程序并不能为用户解决更多的工作,未能实现自动化,人们也就不想使用这个应用程序。应用程序应该能减轻人们繁重的工作,至少应该能分担人们的一部分工作,例如:字处理程序组织文档,数据库管理系统处理记录。

1.2 基本术语

* 程序是任务的符号形式。

要掌握 Visual Basic 还是需要一点努力的,本书假定读者已掌握了程序的基本概念。要达到精通的水平,这里给出一个新的,含义更广的程序的定义:程序是一项任务或工作过程的符号形式。它是用基本代码说明的计算过程,其中的基本代码可由人读懂,并由计算机去处理。

* 源代码是用程序的宿主语言编写的正文。

注意,在上面的定义中我未使用术语语言(language)。计算机程序一般是用语言编写的,Visual Basic 程序也不例外。程序的正文被称为源代码(source code)。

* 指令是程序中有完整含义的语句。

程序的源代码由顺序执行的指令组成。指令执行顺序并不要求自顶向下或者从头至尾,但是指令序列仍是存在的。因为 Visual Basic 程序的指令具有语句结构,而且程序中的指令序列以更类似于人类语言的方式向人们传达程序的实际目的,所以 Visual Basic 是一种高级程序设计语言。与此相对,机器语言和汇编语言被称为低级语言,因为它们更接近于计算机的逻辑,而不是人类思维的方式。

虽然使用程序设计语言驱动计算机,但是程序设计技术并不象是和雇员对话,或者发一条命令给雇员。与具有智能的人类不同,未配备程序的计算机并不能臆想复杂指令蕴含的含义,也不能提前处理与人们提交给它的任务有关的工作。如果计算机可以“跳出机箱”为你执行任何你要求它做的工作,那么就不需要任何软件了。

* 低级程序设计语言用机器代码直接与计算机“交谈”。

如果使用计算机的中央处理机(CPU)可识别最基本的符号编程,那么该程序则是用机器语言编写的,而且符号系统组成一个低级语言程序。作为低级语言的程序员,则可能向计