第 3 章　基本控制结构

结构化程序设计包含 3 种基本控制结构，即顺序结构、选择结构和循环结构。Visual Basic 在编写过程代码解决各种问题时，就采用了结构化程序设计的方法。

3.1　顺 序 结 构

顺序结构是最简单的程序设计结构，指一个过程中的语句，按照从上到下的顺序一条一条地顺序执行。执行过程如图 3-1 所示。在顺序结构中，程序只有一个入口和一个出口，必须依次执行每条语句后方可退出程序。

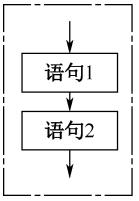

图 3-1　顺序结构流程图

3.1.1　赋值语句

赋值语句可以把指定的值赋给某个变量或某个对象的属性。

格式：[Let] <变量名> = <表达式>

说明：

（1）赋值语句中的“ Let ”关键字可以省略不写。

（2）在赋值语句中，“ = ”是赋值号，与数学意义上的等号不同。

（3）赋值语句有计算和赋值双重功能，其首先计算赋值号右边的表达式的值，然后将所得结果赋给赋值号左边的变量。

（4）变量名可以是变量或对象的属性名。

（5）表达式可以是变量、表达式（数值表达式、字符串表达式或逻辑表达式）、常量或对象的属性。

（6）赋值号左右两边的数据类型应尽量保持一致，若不一致，可按以下方式处理：

① 当表达式的值和变量均为数值型，但精度不同时，将强制把表达式的值转换成左边的精度。

例如：

```
n% = 2.8                          'n 为整型变量，2.8 将四舍五入，n 中存放结果为 3
```

② 当左侧变量为数值类型时，若右侧表达式为数值字符串，则右侧的值将自动转换成数值类型；若右侧表达式中有非数字字符串或空串，则会出错。例如：

```
n% = "111"                        'n 结果为 111
n% = "c2dd" （或 n% = ""）        '出错
```

③ 当把逻辑型赋值给数值型变量时，True 转为-1，False 转为 0；反之，当把数值型赋值给逻辑型变量时，0 转为 False，非 0 值转为 True。例如：

```
n% = True                         'n 结果为-1
```

④ 任何非字符型的值赋值给字符型变量时，都将自动转换为字符型。例如：

```
a$ = 111                          'a 结果为"111"
```

为保证程序正常执行，应该利用类型转换函数将表达式的结果转换后赋值。

在 Visual Basic 中，经常用到的赋值语句如下：

① 直接把数值赋给变量：

```
Dim a As Integer
Dim b As String
Dim Sys_Time As Date
a = 20
b = "Hello"
Sys_Time = Time()            '把当前系统时间赋给日期型变量 Sys_Time
```

② 将一个变量的值赋给另一变量：

```
a = 35
b = a
```

③ 将一个表达式的值赋给一个变量：

```
Dim a%, b%, c%
a = 1
b = 2
c = a + b
```

④ 为对象的某个属性设置属性值，一般格式如下：对象名.属性 = 属性值。

例如：

```
Command1.Caption = "确定"
```

3.1.2　程序书写规则

1．不区分大小写

为提高程序的可读性，Visual Basic 会自动对用户输入的代码进行转换。

（1）程序代码中的 VB 关键字、单个英文单词的代码，其首字母将被转换为大写，其余字母被转换成小写，如 Dim、If、Select、Integer 等；若关键字由多个英文单词组成，Visual Basic 将会把每个单词的首字母转换成大写，如 ElseIf、FontName、FillStyle 等。

（2）对于用户自己定义的变量名或过程名，以用户首次定义为准，以后输入的变量名或过程名将自动按首次定义时的形式转换。

2．语句书写自由

（1）在 Visual Basic 中，一般一行书写一条语句，在每一条语句后没有结束符。但若想在同一行中书写多条语句，可以在多条语句之间用冒号“ :”分隔，一行最多可写 255 个字符。

例如，分别为变量 a，b，c 赋值为 3，可以采用如下写法：

```
a = 3
b = 3
c = 3
```

或者使用以下语句：

```
a = 3: b = 3: c = 3
```

（2）如果一行语句过长，可以将单行语句分成若干行，只需在本行语句最后加上续

行符“ _”（空格+下画线）即可。

例如，以下语句

```
poem = "天门中断楚江开，碧水东流至此回。" & "两岸青山相对出，孤帆一片日边来。"
```

可写成：

```
poem = "天门中断楚江开，碧水东流至此回。" _
& "两岸青山相对出，孤帆一片日边来。"
```

（3）编写注释，有利于程序的阅读、调试和维护。

在 Visual Basic 中用 Rem 或单撇号“'”引导注释的内容，对程序进行注释。注释部分文字连同 Rem 或单撇号“'”都会变成绿色，在程序执行过程中，注释语句不被执行。

格式：

```
Rem <注释>
' <注释>
```

“Rem <注释>”（Rem 和注释语句之间至少要有一个空格）单独用在一行中，常作为某段程序的注释使用；“'<注释>”形式经常出现在程序的某一行代码后面，作为本行代码的注释，当然，“'<注释>”形式也可单独作为一行使用。

举例说明：

```
Rem 这是一个测试程序，测试文本框中输入的是否为数字，若不是数字，则响铃
Private Sub Text1_KeyPress(KeyAscii As Integer)
    If KeyAscii < Asc("0") Or KeyAscii > Asc("9") Then
        KeyAscii = 0              '如果输入的不是数字，则不接收这个字符
        Beep                  '铃声，发出错误信号
    End If
End Sub
```

3.1.3 数据的输入和输出

在设计程序时，除界面外，一般可将程序分为输入、处理和输出 3 部分。计算机通过对输入的数据进行处理，将处理完的数据提供给用户并输出。

1．Print 方法

Print 方法可在对象上输出信息。

格式：

```
[对象名.] Print [表达式列表] [,|;]
```

说明：

（1）对象名：可以是窗体（Form）、图片框（PictureBox）、打印机（Printer）或立即窗口（Debug）。若省略，则表示内容将在当前窗体输出。

例如：

```
Debug.Print "中国欢迎你！"    '在立即窗口中输出
Print "你好！"              '在窗体上输出
```

（2）表达式列表：可以是一个或多个表达式，表达式可以是算术表达式、字符串表达式、关系表达式或逻辑表达式。

例如：

```
a% = 10 : b% = 30
```

```
Print a                          '打印变量 a 的值
Print                            '输出一个空行
Print "abcd"                     '打印 abcd 字符串，字符串必须用双引号括起来
```

（3）当输出多个表达式时，各表达式之间用分隔符（逗号、分号或空格）分开。

若采用逗号分隔，则在输出数据项时，每隔 14 列开始一个打印区，逗号后面的内容在下一个打印区输出。

若采用分号或空格分隔，则采用紧凑格式输出数据。

若 Print 语句后面没有分隔符，则下一个输出的内容自动换行显示。

例如：

```
Dim a As Integer, b As Integer
a = 10
b = 15
Print a, b                       '变量 b 的值在下一打印区输出
Print a; b                       '变量 a 和变量 b 的值采用紧凑格式输出
Print                            '换行
Print "a", "b"
Print "a"; "b"
```

运行结果如图 3-2 所示。

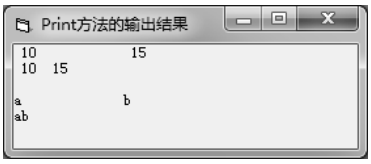

图 3-2　Print 的输出结果

（4）Print 方法具有计算和输出双重功能。对于表达式，应先计算其值，然后输出。

例如：

```
Dim a As Integer, b As Integer
a = 10
b = 15
Print a + b                      '先计算 a + b 的值，然后打印输出 25
```

（5）与 Print 方法配合使用的函数有以下几个。

① Tab()函数：对输出进行定位。

格式：

```
Tab(n)
```

其中，n 为数值表达式，其值为整数，表示下一个输出位置的列号。通常，将窗体的最左边的列号视为 1，若显示的位置超过 n，则自动下移一行，在下一行 n 的位置显示。当 n 大于行的宽度时，显示位置为“ n Mod 行宽 ”；如果 n<1，则把输出的位置移到第一列。

当一个 Print 方法中有多个 Tab()函数时，每个 Tab()函数对应一个输出项，各项之间用分号隔开。

例如：

```
Print Tab(5); "学号"; Tab(16); "姓名"; Tab(24); "性别"; Tab(32); "专业"
```

```
    Print
    Print "16074105800"; Tab(16); "张强"; Tab(24); "男"; Tab(32); "护理学"
    Print "15045105900"; Tab(16); "林放"; Tab(24); "男"; Tab(32); "预防医学"
    Print "14015105700"; Tab(16); "赵珊"; Tab(24); "女"; Tab(32); "临床医学"
```

运行结果如图 3-3 所示。

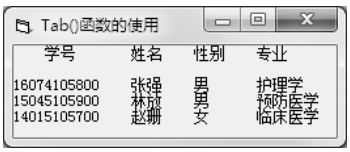

图 3-3 使用 Tab()函数

② Spc()函数：在 Print 的输出中，用 Spc()函数可以跳过指定个数的空格。

格式：

```
    Spc(n)
```

其中，n 是数值表达式，表示跳过 n 个空格，其取值为 0～32767 范围的整数。Spc()函数与输出项之间用分号隔开。

例如：

```
    Print "aaa"; Spc(3); "bbb"
```

显示结果： aaa bbb

例 3.1 随机产生一个 3 位正整数，再将产生的数与逆序数同时输出并显示。

```
Private Sub Command1_Click()
    Randomize                                        '初始化随机数生成器
    Dim x As Integer, x1%, x2%, x3%
    x = Int(Rnd() * (999 - 100 + 1) + 100) '生成[100, 999]之间的正整数
    x1 = x Mod 10                                    '分离出个位数
    x2 = (x Mod 100) \ 10                            '分离出十位数
    x3 = x \ 100                                     '分离出百位数
    Print "随机产生的数为："; x
    Print "逆序后的数为："; x1 * 100 + x2 * 10 + x3
    Print
End Sub
```

程序运行后，单击命令按钮，结果如图 3-4 所示。

图 3-4 例 3.1 运行结果

2．InputBox 函数

InputBox 是一个输入函数，其作用是弹出一个对话框，等待用户输入数据，并返回输入的内容，如图 3-5 所示。

格式：

```
    InputBox(提示[,标题][,默认值][,x 坐标位置][,y 坐标位置][,帮助,索引])
```

说明：

（1）提示：一个字符串，长度不超过 1024 个字符，作用是在对话框内显示信息，提示用户输入。在对话框中显示提示信息时，若信息一行显示不下，会自动换行显示。若要控制换行，则可使用回车 Chr(13)和换行 Chr(10)控制符，即在需要换行的位置添加

“ Chr(13) + Chr(10) ”语句，或添加“ vbCrLf ”。

图 3-5　InputBox 函数对话框

例如，提示信息输入以下内容。

```
"请输入病历号：" + Chr(13) + Chr（10）+ "输入后按回车或单击‘确定’按钮"
```

或：

```
"请输入病历号：" + vbCrLf + "输入后按回车或单击‘确定’按钮"
```

运行后，对话框中的提示信息将分两行显示。

（2）标题：一个字符串，用来显示对话框的标题，在对话框顶部标题区显示。

（3）默认值：一个字符串，显示输入到对话框中的默认信息。在程序运行显示输入对话框时，用户可选择使用已有默认信息作为输入值，也可在输入框中自行键入数据，取代默认值。若省略该参数，则运行时对话框的输入区为空白，等待用户输入。

（4）x 坐标位置，y 坐标位置：两个整数值，分别用来确定对话框左上角与屏幕左边的距离 x 和屏幕上边的距离 y，单位均为 twip。这两个参数必须同时给出或同时省略。若省略这两个参数，则对话框运行时显示在屏幕中心线向下约三分之一处。

（5）帮助，索引：帮助是一个字符串变量或字符串表达式，表示帮助文件的名称；索引是数值变量或表达式，表示相关帮助主题的帮助目录号。两个参数必须同时给出或省略，参数给出后，在对话框中会出现一个“ 帮助 ”按钮，单击该按钮或按 F1 键，可以得到有关的帮助信息。

（6）在默认情况下，InputBox 函数的返回值是一个字符串。

（7）每次执行 InputBox 函数时，只能输入一个值，若要输入多个值，则需要多次调用 InputBox 函数。

例 3.2　编写程序，根据输入的半径计算圆的面积。

```
Private Sub Command1_Click()
    Dim r As Single
    r = Val(InputBox("请输入圆的半径：", "计算圆的面积", "3"))
    Print "圆的面积为："; 3.14 * r * r
End Sub
```

程序运行后，单击命令按钮，弹出如图 3-6（a）所示的对话框，输入一个数据后，会在窗体上显示运算后圆的面积值，如图 3-6（b）所示。

3．MsgBox 函数和 MsgBox 语句

使用 Windows 进行某些操作时，若操作有误，屏幕上会弹出让用户选择操作的对话框，用来确定以后的操作。Visual Basic 中的 MsgBox 函数的功能与此类似，它可以向用户传送信息，并可通过用户的选择，作为程序继续执行的依据。

（a） （b）

图 3-6 例 3.2 运行结果

MsgBox 函数格式如下。

```
MsgBox(提示[,按钮][,标题] [,帮助,索引])
```

说明：

（1）提示：一个字符串，长度不超过 1024 个字符，是对话框内显示的信息。当提示信息过长，一行内显示不完时，将自动换行，也可使用“ Chr(13) + Chr(10) ”或 vbCrLf 强制换行。

（2）按钮：一个整数值或符号常量，控制在对话框内显示的按钮及按钮数量、图标的种类。该参数由 4 类数值相加产生，这 4 类数值或符号常量分别表示按钮的类型、显示图标的种类、默认按钮及等待模式，具体内容如表 3-1 所示。

表 3-1 MsgBox()函数“按钮”参数的取值

类 型	常 量	数 值	说 明
按钮种类	vbOKOnly	0	只显示一个 OK 按钮
	vbOKCancel	1	显示 OK 和 Cancel 按钮
	vbAbortRetryIgnore	2	显示 Abort、Retry 和 Ignore 按钮
	vbYesNoCancel	3	显示 Yes、No 和 Cancel 按钮
	vbYesNo	4	显示 Yes 和 No 按钮
	vbRetryCancel	5	显示 Retry 和 Cancel 按钮
图标	vbCritical	16	显示停止图标“ × ”
	vbQuestion	32	显示提问图标“ ？ ”
	vbExclamation	48	显示警告图标“ ！”
	vbInformation	64	显示输出信息“ i ”
默认按钮	vbDefaultButton1	0	第一个为默认按钮
	vbDefaultButton2	256	第二个为默认按钮
	vbDefaultButton3	512	第三个为默认按钮
等待模式	vbApplicationModal	0	当前应用程序挂起，直到用户对信息框做出响应才继续工作
	vbSystemModal	4096	所有应用程序挂起，直到用户对信息框做出响应才继续工作

MsgBox 函数的“ 按钮 ”参数由表 3-1 的 4 类参数组成，在组成“ 按钮 ”参数时，从“ 按钮种类 ”、“ 图标 ”、“ 默认按钮 ”和“ 等待模式 ”中各选择一个值，把这几个值加到一起，就是“ 按钮 ”参数的值。不同的组合会得到不同的结果。

例如：

```
0 + 16              '显示“ 确定 ”按钮和“ × ”图标
1 + 32 + 0          '显示“ 确定 ”、“ 取消 ”按钮和“ ？ ”图标，默认按钮为“ 确定 ”
```

在表 3-1 中，每个数值都有相对应的符号常量，其作用与数值相同。编写程序时，

也可选用符号常量，增加程序的可读性。

MsgBox 对话框弹出后，需要用户做出选择，即单击某个按钮或按回车键，否则不能执行其他操作。

（3）标题：一个字符串，用来显示对话框的标题。

（4）帮助，索引：同 InputBox 函数。

（5）MsgBox 函数的返回值是一个整数，此数值的大小与用户选择的按钮有关。具体内容如表 3-2 所示。

表 3-2　MsgBox()函数的返回值

返 回 常 量	返　回　值	说　　明
vbOK	1	选择了 OK 按钮
vbCancel	2	选择了 Cancel 按钮
vbAbort	3	选择了 Abort 按钮
vbRetry	4	选择了 Retry 按钮
vbIgnore	5	选择了 Ignore 按钮
vbYes	6	选择了 Yes 按钮
vbNo	7	选择了 No 按钮

MsgBox 共有 5 个参数，除第一个参数外，其余参数都是可选的。若省略第二个参数“ 按钮 ”（默认值为 0），则对话框内只显示一个“ 确定 ”按钮，不显示任何图标，并将该按钮设为默认按钮。若省略第三个参数“ 标题 ”，则弹出的对话框的标题为当前工程的名称，若希望对话框的标题处不显示任何文字，则需要把“ 标题 ”参数设为空串（""）。

例 3.3　编写程序，计算输入数值的平方根。

```
Private Sub Command1_Click()
    Dim n As String, a As Integer
    n = InputBox("请输入一个数值：")
    a = MsgBox("输入值是" + n + "，它的平方根是" + Str(Sqr(Val(n))) + "！", _
vbOKOnly + vbInformation, "计算数的平方根")
End Sub
```

程序运行后，单击命令按钮，运行结果如图 3-7 所示。

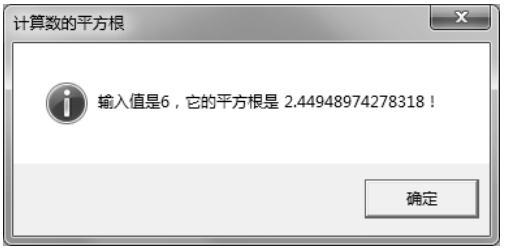

图 3-7　例 3.3 运行结果

在程序设计中，MsgBox 函数的返回值通常用来作为程序继续执行的依据，根据该对话框的返回值决定程序后面的操作。

例 3.4　编写程序，根据 MsgBox 函数的返回值决定程序的输出内容。

```
Private Sub Command1_Click()
    Dim choice As Integer
    choice = MsgBox("请问您是要去验血吗？", 4 + 32, "请选择")
    If choice = 6 Then
        MsgBox "请至 2 楼检验科！", , "提示"
    ElseIf choice = 7 Then
        MsgBox "请等待!", , "提示"
    End If
End Sub
```

程序运行后，单击命令按钮，运行结果如图 3-8 所示。

图 3-8　例 3.4 运行结果

MsgBox 语句格式如下。

```
MsgBox 提示[,按钮][,标题] [,帮助,索引]
```

此语句中各参数的含义与作用和 MsgBox 函数相同，由于 MsgBox 语句没有返回值，因此 MsgBox 语句格式常用于设置较简单的提示信息。

3.2　选 择 结 构

在处理日常问题时，常需要根据给定的条件进行分析判断，并根据判断的结果采取不同的操作。在 Visual Basic 中，这样的问题通过选择结构来解决。

在 Visual Basic 中，选择结构可以分成单分支、双分支和多分支选择结构 3 种。

3.2.1　单分支选择结构

格式一：

```
If <表达式> Then
    <语句块>
End If
```

格式二：

```
If <表达式> Then <语句>
```

说明：

（1）表达式一般为关系表达式、逻辑表达式，也可为算术表达式。对表达式的结果进行判断，结果为 True 或 False。

（2）语句块可以是一条或多条语句。

格式一是块 If 格式，以 If 开头，End If 结尾，语句块写在 Then 的下一行，End If 之间要由空格分隔。

格式二是行 If 格式，语句和 Then 写在一行，不需要 End If。

（3）单分支选择结构的执行过程如图 3-9 所示：先判断表达式的值，当表达式的值为 True（非零）时，执行语句块（或语句），否则不进行任何操作。

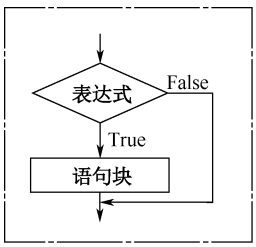

图 3-9　单分支选择结构流程图

例 3.5　已知两个数 x 和 y, 比较其大小, 并使得 x > y。(两个数的交换过程如图 3-10 所示。)

```
Private Sub Command1_Click()
    Dim x As Integer, y As Integer
    Dim t As Integer
    x = Val(InputBox("请输入第一个数"))
    y = Val(InputBox("请输入第二个数"))
    Print "交换前 x="; x, "y="; y
    If x < y Then
        t = x
        x = y
        y = t
    End If
    Print "交换后 x="; x, "y="; y
End Sub
```

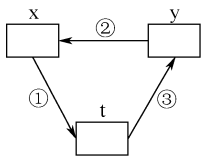

图 3-10　交换过程

3.2.2　双分支选择结构

格式一：

```
If <表达式> Then
    <语句块 1>
Else
     <语句块 2>
End If
```

格式二：

```
If <表达式> Then <语句 1> Else <语句 2>
```

双分支选择结构的执行过程如图 3-11 所示：先判断表达式的值，当表达式的值为 True（非零）时，执行语句块 1（或语句 1），否则执行语句块 2（或语句 2）。

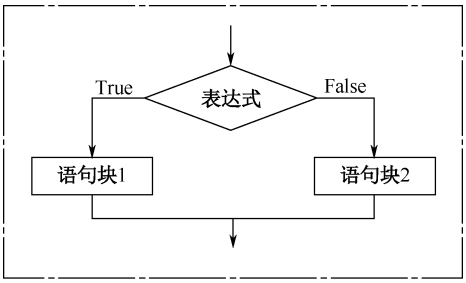

图 3-11　双分支选择结构流程图

例 3.6　判断输入的整数是奇数还是偶数。

```
Private Sub Command1_Click()
    Dim n As Integer
    n = Val(InputBox("请输入一个整数"))
    If n Mod 2 = 0 Then
        Print n; "是偶数！"
    Else
        Print n; "是奇数！"
    End If
End Sub
```

3.2.3　多分支选择结构

双分支选择结构能处理的问题相对有限，当处理的问题有多种条件时，将用到多分支选择结构。

1．If…Then…ElseIf 语句

格式：

```
If <表达式 1> Then
    <语句块 1>
ElseIf <表达式 2> Then
     <语句块 2>
     ……
[Else
     <语句块 n+1>]
End If
```

多分支选择结构执行过程如图 3-12 所示：先测试表达式 1，若表达式 1 结果为 True（或非零），则执行 Then 后面的“ 语句块 1 ”；若表达式 1 的值为 False（或 -1），则依次对每个 ElseIf 后面的表达式进行测试，一旦某个表达式的值为 True（或非零），则执行该表达式对应的“ 语句块 ”；若所有 ElseIf 子句后面的“ 表达式 ”都不为 True，则执行 Else 后面的“ 语句块 n+1 ”。在执行了一次某语句块后，程序退出 If 语句，执行 End If

后面的语句。

在 If…Then…ElseIf 结构中，ElseIf 子句的数量没有限制，可根据需要加入。结构中的 Else 子句是可选的。

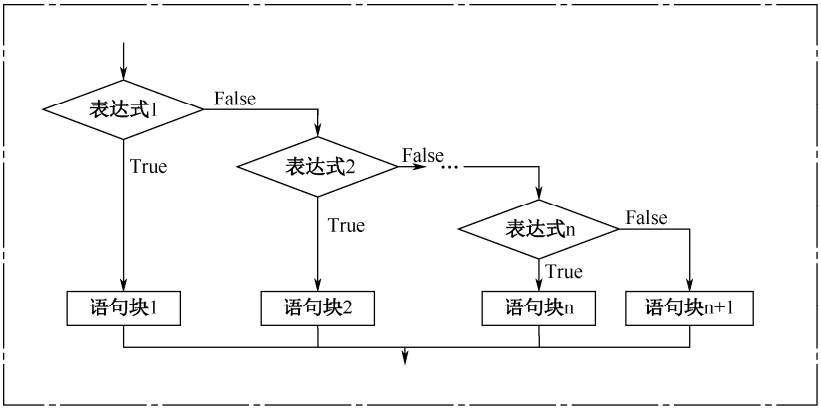

图 3-12　多分支选择结构流程图

例 3.7　编写程序，对由键盘上输入的某学生某门课程的成绩进行判断，在窗体上显示该成绩的级别。学生成绩级别如表 3-3 显示。

表 3-3　学生成绩级别

分　数　段	级　　别
90 分以上（含 90 分）	优
80～90 分（含 80 分，不含 90 分）	良
70～80 分（含 70 分，不含 80 分）	中
60～70 分（含 60 分，不含 70 分）	及格
60 分以下（不含 60 分）	差

```
Private Sub Command1_Click()
    Dim score As Single
    score = Val(InputBox("请输入 0-100 之间的数值：", "判断成绩级别"))
    If score >= 90 Then
        Print score; "分", "优"
    ElseIf score >= 80 Then
        Print score; "分", "良"
    ElseIf score >= 70 Then
        Print score; "分", "中"
    ElseIf score >= 60 Then
        Print score; "分", "及格"
    Else
        Print score; "分", "差"
    End If
End Sub
```

运行程序后，单击命令按钮，运行结果如图 3-13 所示。

图 3-13　例 3.7 运行结果

2．Select Case 语句

Select Case 语句又称情况语句，是多分支结构的另一种表现形式，它可以根据一个表达式的值，在一组相互独立的可选语句序列中挑选要执行的语句序列。

格式：

```
Select Case 变量或表达式
    Case 表达式列表 1
        <语句块 1>
    Case 表达式列表 2
        <语句块 2>
        ……
    [Case Else
        <语句块 n+1>]
End Select
```

说明：

（1）变量或表达式可以是数值表达式或字符串表达式，通常为变量或常量。

（2）语句块 1、语句块 2、…、语句块 n：由一行或多行 Visual Basic 语句组成。

（3）表达式列表 1、表达式列表 2……：称为值域，可以是下列形式之一。

① 表达式结果 1[,表达式结果 2]……

例如，Case 1,2,3。

② 表达式结果 To 表达式结果。

To 用来指定一个范围，必须把小值写在前面，大值写在后面。

例如：Case 1 To 8。

③ Is 关系运算符 数值或字符串。

关系运算符包括<，<=，>，>=，<>，=。

例如，Case Is > 7　或 Case Is > "ACD"。

不能用逻辑运算符将两个或多个简单条件组合起来，如 Case Is > 5 And Is < 9 是错误的。

该语句的执行过程如图 3-14 所示：先对变量或表达式求值，然后查找相匹配的 Case 表达式列表，若找到，则执行相应的 Case 子句后的语句块，然后退出 Select Case 语句；若没有找到，则执行 Case Else 子句后面的语句块，然后退出 Select Case 语句，执行 End Select 后面的语句。

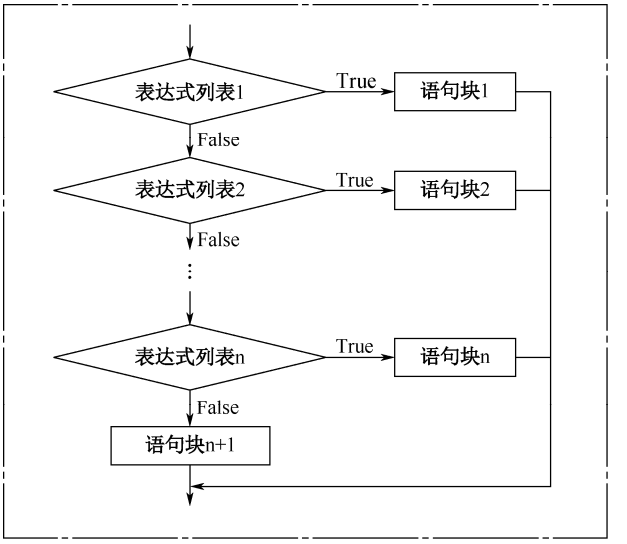

图 3-14　Select Case 语句流程图

例 3.8　将例 3.7 用 Select Case 语句实现。

```
Private Sub Command1_Click()
    Dim score As Single, x As Integer
    score = Val(InputBox("请输入 0～100 之间的数值：", "输入框"))
    x = Int(score / 10)
    Select Case x
        Case 9, 10
            Print score; "分", "优"
        Case 8
            Print score; "分", "良"
        Case 7
            Print score; "分", "中"
        Case 6
            Print score; "分", "及格"
        Case Else
            Print score; "分", "差"
    End Select
End Sub
```

3.2.4　条件函数

Visual Basic 提供了 IIf 函数，用来执行简单的条件判断操作，可以完成 If 语句的双分支选择功能。

格式：

```
result = IIf(表达式,表达式 1,表达式 2)
```

“ result ”是函数的返回值。

IIf 函数是 If…Then…Else 的一种简单表示形式。执行 IIf 函数时，先测试表达式的值，当表达式的值为 True 时，函数返回表达式 1 的值，否则返回表达式 2 的值。表达式

1、表达式 2 可以是表达式、变量或其他函数。

注意：IIf 函数中的 3 个参数都不能省略，且要求表达式 1、表达式 2 及结果变量类型保持一致。

例如，判断 score 变量中的值是否大于 60，若大于等于 60，则返回“ 及格 ”给变量 result，否则返回“ 不及格 ”给变量 result。可以使用如下语句表示：

```
result$ = IIf(score >= 60, "及格", "不及格")
```

该语句等价于：

```
If score >= 60 Then
    result$ = "及格"
Else
    result$ = "不及格"
End If
```

3.3 循 环 结 构

在实际应用中，经常遇到一些操作并不复杂，但需要反复多次处理的问题，如计算 1 + 2 + … + 1000，按人口增长率统计人口数等问题，如果采用顺序结构解决此类问题，一步一步手工输入操作显然不现实，有时也难以实现。循环结构可以方便地解决这类问题。循环结构是指在一定的条件下，有规律地重复执行一段代码的操作，其中，被重复执行的代码称为循环体，而每次循环前判断循环是否执行的条件部分称为循环条件。在 Visual Basic 中，有两种类型的循环语句：计数型循环和条件型循环。

3.3.1 For 循环

For 循环是计数型循环语句，通常用于循环次数可以预知的循环，在 For 循环中，循环次数是固定的，执行一定次数的循环后即可结束循环。

格式：

```
For 循环变量= 初值 To 终值[Step 步长]
    [循环体]
   [Exit For]
Next [循环变量]
```

例如：

```
For i = 1 To 5 Step 1
   Print "*"
Next i
```

该例可以打印 5 个“ * ”。

说明：

（1）循环变量必须为数值型变量。

（2）初值、终值、步长均是数值表达式。步长的值可以是正数或负数，但不能为 0。若步长为正数，则循环变量的初值应小于终值；若步长为负数，则循环变量的初值应大于终值；如果步长值为 1，则可以省略不写。

可以通过循环变量的初值、终值、步长来计算循环执行的次数 *n*：

$n = \mathrm{I}nt\left(\dfrac{\text{终值 - 初值}}{\text{步长}} + 1\right)$，此公式仅限于循环体内循环变量不发生改变的情况。

（3）循环体在 For…Next 之间，可以是一条或多条语句。

（4）Exit For 用于退出当前的 For 循环。

（5）Next 是循环终端语句，Next 后面的“ 循环变量 ”与当前 For 循环中的“ 循环变量 ”必须相同。

（6）For 语句的执行过程如图 3-15 所示。

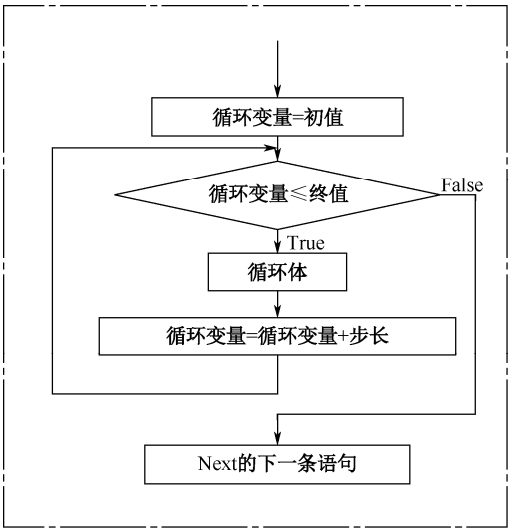

图 3-15　For 循环流程图

① 初值赋给循环变量。

② 判断循环变量的值是否超过终值，若超过，则不执行循环体，直接执行 Next 下一行的语句；否则，执行一次“ 循环体 ”。

这里所说的“ 超过 ”有两种含义，即大于或小于。当步长为正数时，检查“ 循环变量 ”的值是否大于终值；当步长为负数时，判断“ 循环变量 ”的值是否小于终值。

③ 循环变量=循环变量+步长。

④ 重复步骤②和步骤③。

（7）在未使用“ Exit For ”语句退出 For 循环后，循环变量的值应超出终值的表示范围。

（8）For 语句和 Next 语句必须成对出现，不能单独使用。

（9）For 循环可以嵌套使用，嵌套层次没有具体限制，但是，每个循环必须有一个唯一的变量名作为循环变量 ;内层循环的 Next 语句必须放在外层循环的 Next 语句之内，内外循环不得互相交叉。

（10）For 循环中的“ 循环体 ”可以不写，此时 For 循环将执行“ 空循环 ”。

（11）For…Next 循环遵循“ 先检查，后执行 ”的原则，即先检查循环变量的值是否超过终值，然后决定是否执行循环体。

在下列情况下，循环体将不被执行。

① 当步长为正数时，初值大于终值。

② 当步长为负数时，初值小于终值。

当初值等于终值时，不管步长值是正数还是负数，均执行一次循环体。

例 3.9 据 2011 年 4 月 28 日国家统计局公布的第六次人口普查结果显示，我国现有人口 13.4 亿，按人口年增长 0.57%计算，2020 年时我国人口将达到多少？

分析：解此问题可根据公式 $x = 13.4 * (1 + 0.0057)^{(2020 - 2011)}$ 计算，公式中的 x 为 2020 年时我国的人数。x 的值可以直接求得，也可利用循环求得，程序如下。

```
Private Sub Command1_Click()
    Dim n As Integer, x As Double       'x 用于存放每年人口数量
    x = 13.4
    For n = 2012 To 2020
        x = x + x * 0.0057          '每一年的人口数量都是上一年人口数*(1+0.0057)
    Next n
    Print "到 2020 年，我国人口总数为："; x; "亿"
End Sub
```

3.3.2 Do 循环

Do 循环可以用来控制循环次数未知的循环结构。如果满足指定的条件，则循环体多次执行。Do 循环属于条件型循环，它有以下两种语法格式。

格式一：

```
Do [ { While | Until } 条件]
    [循环体]
    [Exit Do]
Loop
```

格式二：

```
Do
    [循环体]
    [Exit Do]
Loop [ { While | Until } 条件]
```

Do 循环执行流程如图 3-16 所示。

说明：

（1）循环体是需要重复执行的一条或多条语句。

（2）在一般情况下，“ 条件 ”多为关系或逻辑表达式，结果为 True 或 False。

（3）Exit Do 用于退出 Do 循环。一个 Do 循环可以有一个或多个 Exit Do 语句，且 Exit Do 语句可以出现在循环体的任何地方。

（4）Do 循环可以嵌套，其规则与 For 循环相同。

（5）格式一：先判断循环条件，在条件满足时才执行循环体，否则不执行。格式二：不管循环条件是否满足，先执行一次循环体，再判断循环条件是否满足，以决定是否再次执行循环。

（6）两种格式中的“ While 条件 ”均表示当条件成立时执行循环，如果条件不成立，则退出循环，执行 Loop 后面的语句；“ Until 条件 ”表示当条件成立时退出循环，也就是说，在 Until 条件下，条件不成立时，才执行循环体。

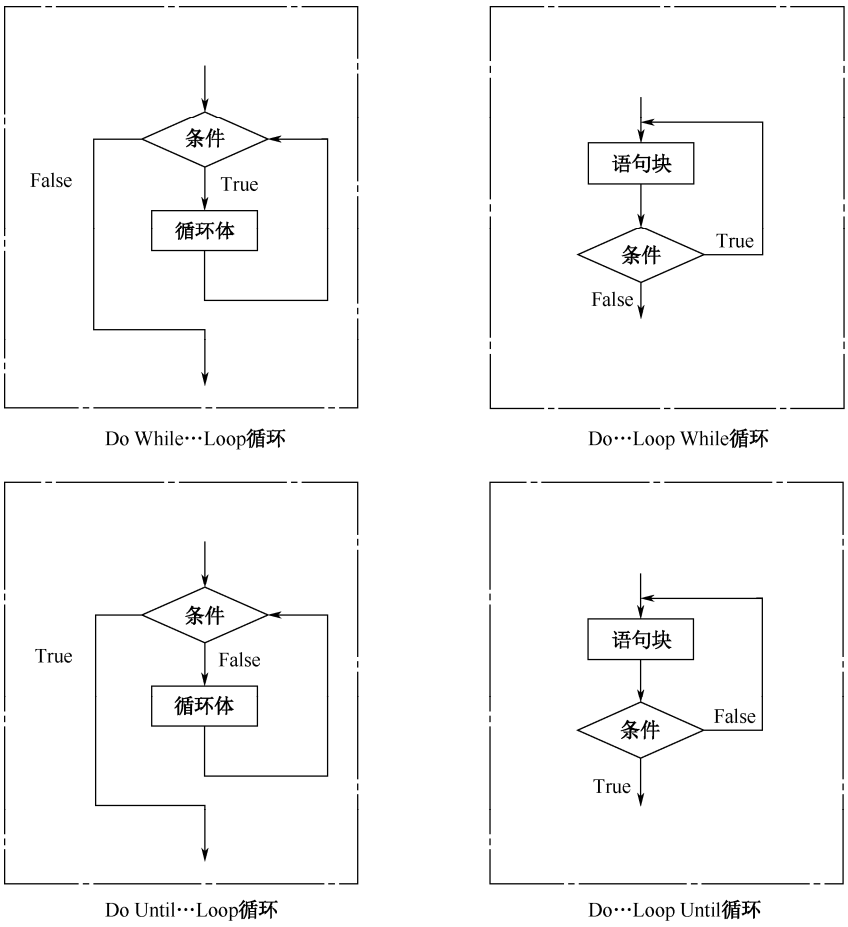

图 3-16　Do 循环流程图

例 3.10　据 2011 年 4 月 28 日国家统计局公布的第六次人口普查结果显示，我国现有人口 13.4 亿，按人口年增长 0.57%计算，多少年后我国人口将超过 20 亿？

分析：解此问题可根据公式 $20=13.4*(1+0.0057)^n$ 计算，公式中的 n 为年数。n 的值可以直接利用标准对数函数求得，也可利用循环求得，程序如下。

```
Private Sub Command1_Click()
    Dim x As Double, n As Integer
    x = 13.4
    n = 0
    Do While x <= 20
        x = x * 1.0057
        n = n + 1
    Loop
    Print n; "年后，我国人口将突破 20 亿，人口为："; x; "亿"
End Sub
```

3.3.3　While…Wend 循环

While…Wend 循环与 Do While…Loop 循环相似，它根据循环条件的真假，决定是否执行循环。