Chapter 3

第3章 Visual Basic语法基础

本章学习目标：

- 掌握数据类型的概念、特点和表示方法。
- 熟练掌握常量和变量的概念及使用方法。
- 熟练掌握 Visual Basic 提供的各种运算符的功能、优先级。
- 熟练掌握表达式的书写规则。
- 掌握常用内部函数的功能和使用方法。

通过前两章的学习，读者对 Visual Basic 有了初步的认识。读者可以参照例题，编写一些简单的应用程序。

在完成应用程序的界面设计后，需要再编写事件过程代码。只有界面设计合理、过程代码编写正确的程序，才能实现具体的功能。因此，掌握 Visual Basic 的语法及其使用方法是开发应用程序的基础和关键。

本章主要介绍 Visual Basic 的数据类型、常量与变量、运算符与表达式，以及常用内部函数的功能与应用。

3.1 数据类型

数据是程序的必要组成部分，也是程序处理的对象。Visual Basic 预定义了丰富的数据类型，不同数据类型体现了不同数据结构的特点，如表 3.1 所示。

表 3.1 Visual Basic 6.0 的常用数据类型

类型	名称	字节数	取值范围和有效位数
整型	Integer	2	精确表示－32 768～32 767 范围内的整数

续表

类型	名称	字节数	取值范围和有效位数
长整型	Long	4	精确表示－2 147 483 648～2 147 483 647 范围内的整数
单精度浮点型	Single	4	$-3.402\ 823\times10^{38}$～$-1.401\ 298\times10^{-45}$ $1.401\ 298\times10^{-45}$～$3.402\ 823\times10^{38}$ 7 位有效位数
双精度浮点型	Double	8	$-1.797\ 693\ 134\ 862\ 32\times10^{308}$～$-4.940\ 656\ 458\ 412\ 47\times10^{-324}$ $4.940\ 656\ 458\ 412\ 47\times10^{-324}$～$1.797\ 693\ 134\ 862\ 32\times10^{308}$ 15 位有效位数
货币型	Currency	8	－922 337 203 685 477.5808 ～ 922 337 203 685 477.5807
字节型	Byte	1	0～255
变长字符串	String	每个字符占 1 个字节，每个字符串最多可存放约 20 亿个字符	
定长字符串	String * size	size 是小于 65 535 的无符号整常数，为字符串长度	
逻辑型	Boolean	2	True 或 False
日期型	Date	8	100.1.1～9999.12.31
对象型	Object	4	任何对象的引用
变体型	Variant	若存放数值类型数据，占 16 个字节，最大可达 Double 的范围； 若存放字符串类型数据，字符串长度与变长字符串相同	

表 3.1 中，“名称”用以标识变量的类型，“字节数”表示该类型数据所占内存空间的大小。

在 3.2.2 节将介绍如何声明变量的类型。了解不同类型变量的取值范围和有效位数，便于我们在设计时根据实际需要正确地选择数据类型。

例如，声明变量 a 用于存放某个同学一学期各门功课的总分（一般不超过 32 767），可以声明“Dim a As Integer”，Visual Basic 处理系统会为变量 a 分配两个字节的存储空间。声明变量 b 用于存放某大学所有职工的工资总和（一般不小于 32 767），则应声明“Dim b As Long”，Visual Basic 处理系统会为变量 b 分配 4 个字节的存储空间。

又如，计算圆柱体的体积并存入变量 v，声明 v 为 Single 类型，半径和圆周率也采用 Single 类型，则结果 v 具有 7 位有效数字；如果要求计算结果具有更高的精确度，可以考虑采用 Double 类型。

不同类型的数值数据，其数值范围和有效位数的差别或是由于所占用的存储空间大小不同，或是由于存储格式不同。例如，Visual Basic 用一个字节（8 个二进制位）存储 Byte 类型的数据，其最大值为$(11111111)_2$，因此该类型数据的最大值为 255。又如，Visual Basic 用两个字节（16 个二进制位）存储 Integer 类型的数据，首位为符号位（正数为 0；负数为 1），因此其最大值为$(0111111111111111)_2$，即 32 767。

至此，读者应可理解，为什么 Long 类型数据的数值范围超过了 Integer 类型的数据。

Single类型数据占用4个字节内存，第一个二进制位表示该数的符号。因为任何一个实数都可以表示为$\pm 2^{\pm J} \times Q$的形式，其中，J为阶码，即指数，Q为尾数，即数值部分，是一个纯小数。Visual Basic将Single类型数据的后31位分成两段：一段表示J的大小与符号，一段表示Q的数值。

Double类型数据比Single类型数据多出32位：1位增加在表示J的段中，31位增加在表示Q的段中，因此Double类型数据比Single类型具有更大的数值范围、更多的有效位数。

特别要指出的是，Visual Basic的Single、Double类型数据，表示各自数值范围内的数据是有误差的。读者可以做一个尝试：将十进制数0.6转换为二进制数，会发现用二进制不可以将其精确表示。事实上，计算机不可能用无限位数来表示一个实数，误差就是这样产生的。

3.2　常量与变量

3.2.1　常量

常量是直接写在程序中的数据，常量的类型由它们的书写格式决定。

1. 数值常量

Visual Basic中的整型数、长整型数、单精度浮点数、双精度浮点数、货币型数、字节型数统称为数值型数据，在使用数值型数据时，应注意以下几点。

(1) 如果数据包含小数，则应使用Single、Double或Currency类型。其中，Single类型的有效数字为7位，Double类型的有效数字为15位，Currency类型支持15位整数和4位小数，适用于货币计算。

(2) 在Visual Basic中，数值类型数据都有一个有效的取值范围，程序中的数如果超出这个范围，就会出现“溢出”(Overflow)错误。

(3) Visual Basic中的常量一般采用十进制数，但有时也使用十六进制数(数值前加前缀&H)或八进制数(数值前加前缀&O)。

例如，赋值语句“d = &H1A2”的作用是，将418(十进制)送入变量d所在的存储单元。

又如，赋值语句“d = &O216”的作用是，将142(十进制)送入变量d所在的存储单元。

2. 字符串常量

字符串常量是用双引号括起来的一串字符，格式为"$h_1h_2h_3\cdots h_n$"。每个字符占1个字节。可以是任何合法字符，如"VB""123"、Chr (13)(回车符)、"无实数解"等。

3. 逻辑常量

逻辑常量只有两个值：真(True)和假(False)。当把数值常量转换为Boolean时，0为False，非0值为True；当把Boolean常量转换为数值时，False转换为0，True转换为－1。

4. 日期常量

日期常量用来表示日期和时间，Visual Basic 可以表示多种格式的日期和时间，输出格式由 Windows 设置的格式决定。日期数据用两个“#”把表示日期和时间的值括起来，如#08/18/2001#、#08/18/2001 08:10:38 AM#等。

5. 符号常量

当程序中多次出现某个数据时，为便于程序修改和阅读，可以给它赋予一个名字，以后用到这个值时就用名字代表，这个名字就称为符号常量。符号常量的定义格式如下。

```
Const  <符号常量名>=<常量>
```

可以在窗体模块的任何地方(通用对象声明部分或事件过程中)定义。

下面是符号常量的作用域及应用的例子。

```
Const pi =3.14159          '在通用对象声明部分声明数值符号常量
Private Sub Command1_Click()
    Const pi =3.14
    ' 事件过程与通用对象声明部分声明的符号常量同名,则事件过程内部引用的是内
    ' 部声明的值,下列 Print 语句的输出结果是 3.14 而不是 3.14159
    Print pi
End Sub
Private Sub Command2_Click()
    ' 事件过程中未声明 pi,此处 pi 是通用对象声明部分所声明的 pi,输出 3.14159。
    Print pi
End Sub
```

3.2.2 变量

常量的类型由书写格式决定，而变量的类型由类型声明决定。

1. 变量的命名规则

变量名由首字符为英文字母、不超过 255 个字符的字母、数字、下画线等组成。如 Sum、a2、x_1 都可以是 Visual Basic 的变量名。

2. 变量命名的几点说明

(1) 不能使用 Visual Basic 的关键字作为变量名。关键字是指 Visual Basic 系统中已经定义的词，如语句、函数、运算符的名称等，如 Print、If 等都不能用作变量名。

(2) 变量名不能与过程名或符号常量名相同。

(3) Visual Basic 不区分变量名的大小写，即大小写是一样的，如 X1 与 x1 是同一变量。

(4) 变量取名尽量做到“见名知义”，以提高程序的可读性。建议根据变量类型确定变量名的首字母，这是一个小写字母，称为变量的数据类型代码标识符，例如：

y：字节型　　n：整型　　l：长整型　　g：单精度浮点型

d：双精度浮点型　　s：字符串型　　t：日期型

这里关于变量名首字符如何确定，提出的只是建议，而不是规定。根据变量名的首字母并不能确定变量的类型，例如，仅从变量名为 gxl 不能确定变量 gxl 为单精度浮点型，变量的类型只能根据变量声明来确定。

3. 变量声明

在程序中用到的变量一般应声明其类型，由此决定变量的存取格式、取值范围、有效数位等。声明变量类型的方法有两种：隐含声明和强制声明。

1）隐含声明

在变量名的后面加上特定字符（后缀字符），用于规定变量类型的方法称为隐含声明。由变量后缀字符决定变量类型的具体规定如下。

（1）变量后缀字符为“%”，隐含声明该变量类型为整型。

（2）变量后缀字符为“&”，隐含声明该变量类型为长整型。

（3）变量后缀字符为“!”，隐含声明该变量类型为单精度浮点型。

（4）变量后缀字符为“#”，隐含声明该变量类型为双精度浮点型。

（5）变量后缀字符为“$”，隐含声明该变量类型为字符串型。

2）强制声明

用 Dim 语句（类型强制声明语句）可以强制声明只能在本窗体中使用的变量类型。

```
Dim yb As Byte, yc As Byte, nk As Integer, k As Long
Dim gx As Single, dy As Double, sname As String * 10
```

以上语句声明 yb、yc 为字节变量，声明 nk 为整型变量，声明 k 为长整型变量，声明 gx 为单精度浮点型变量，声明 dy 为双精度浮点型变量，声明 sname 为最多 10 个字符的定长字符串型变量。

注意：不可以将语句“Dim m As Integer, n As Integer”写作“Dim m, n As Integer”，后者实际上将 m 声明为变体类型，增加了变量 m 的内存开销。

若强制声明了变量类型，则不可再为变量名加后缀字符。一个变量如果没有声明，则 Visual Basic 将其作为变体类型变量。

好的程序设计风格是声明每一个变量的类型，一方面可以提高程序的可读性，另一方面可避免采用变体数值类型数据，以减少程序运行时的内存开销。

4. 变量的初始值

在程序中声明了变量以后，Visual Basic 自动将数值类型的变量赋初值 0，变长字符串被初始化为零长度的字符串（""），定长字符串则用空格填充，而逻辑型的变量初始化为 False。

同符号常量一样，可以在窗体模块的任何地方（通用对象声明部分或事件过程中）定义变量。

下面是变量的作用域及应用。

```
Dim sMystring as String              '在通用对象声明部分声明字符串变量
Private Sub Form_Load()
    sMystring ="欢迎使用 VB6.0"
End Sub
Private Sub Form_Click()
    Print sMystring
End Sub
```

程序运行时，单击窗体，在窗体中显示“欢迎使用 VB6.0”。如果不在通用对象声明部分声明字符串变量 sMystring，或将其放在 Form_Click 或 Form_Load 事件过程中声明，则程序运行后什么也显示不出来。

5. Static 语句

在事件过程中声明的变量称为局部变量，局部变量除了用 Dim 语句声明外，还可以使用 Static 语句来声明，用这种方法来声明的变量在程序运行过程中可以保留原来的值，即变量所占用的内存空间没有释放。当以后再次使用该变量时，可以继续使用，直到程序结束才释放变量所占用的内存空间。

下面是 Static 语句使用示例。

```
Private Sub Command1_Click()
    Static m As Integer
    Dim n As Integer
    m =a +1
    n =b +1
    Print "m="; m, "n="; n
End Sub
```

当程序运行时，连续单击 Command1 按钮三次，窗体上的输出结果如下。

```
m=1          n=1
m=2          n=1
m=3          n=1
```

m 定义为静态变量，每次调用 Command1_Click 事件过程结束时，都保留 m 的当前值，作为下一次该事件过程被调用时 m 的初值；变量 n 是局部动态变量，在每次执行 Command1_Click 事件过程时都被重新声明，自动赋初值 0。

注意：静态变量只能在过程中声明，而不能在通用对象声明部分声明。

3.2.3 要求变量声明

前面讲过，在程序中用到的变量可以用隐含声明或强制声明方式加以声明。事实上，

Visual Basic 的变量也可以不定义而直接使用，此时变量类型为变体类型。例如：

```
Private Sub Form_Click()
    x = InputBox("")
    y = InputBox("")
    Print x + y
End Sub
```

程序运行时输入 12 和 34，则窗体上显示 1234，因为 InputBox 函数的返回值为字符串类型，两个字符串“12”和“34”连接得到“1234”。

在窗体的通用对象声明部分加上命令 Option Explicit，则要求程序中的变量必须先声明才能使用。例如：

```
Option Explicit
Private Sub Form_Click()
    x = InputBox("")
    y = InputBox("")
    Print x + y
End Sub
```

程序运行时出现如图 3.1 所示的编译错误提示。

图 3.1　变量未定义

3.3　运算符与表达式

3.3.1　算术运算符

如表 3.2 所示，Visual Basic 共有 7 个算术运算符，除了负号是单目运算符外，其他都是双目运算符。

表 3.2　算术运算符

运算符	名　　称	实　　例
^	乘方	2^3 值为 8，－2^3 值为－8
*	乘法	5 * 8
/	除法	7/2
\	整除	7\2 值为 3，12.58\3.45 值为 4(两边先四舍五入再运算)
Mod	求余数	7 mod 2 值为 1，12.58 Mod 3.45 值为 1(两边先四舍五入再运算)
＋	加法	1＋2
－	减法、取负	5－8，－3

注意：整除和求余运算只能对整型数据(Byte、Integer、Long)进行，如果其两边的任一个操作数为实型(Single、Double)，则 Visual Basic 自动将其四舍五入，再用四舍五入后的值作整除或求余运算。

3.3.2 字符串运算符

字符串运算符有两个：＋和 &，均为双目运算符，用于连接两边的字符串表达式。

3.3.3 关系运算符

关系运算符也称为比较运算符，包括＜、＜＝、＞、＞＝、＝、＜＞6 种，均为双目运算符，用于比较两边的表达式是否满足条件，运算结果为 True 或 False。

3.3.4 逻辑运算符

常用的逻辑(布尔)运算符有三种，如表 3.3 所示。

关系表达式的值为 False 或 True，因此也是逻辑表达式；关系表达式用逻辑运算符正确地连接后可以构成更为复杂的逻辑表达式。

表 3.3 逻辑运算符

运算符	名称	实 例 说 明
And	与	8 Mod 2 ＝ 0 And 8 Mod 3 ＝ 0，值为 False 只有当两个表达式的值都为真(True)时，结果才为真，否则为假(False)
Or	或	8Mod 2＝0 Or 8 Mod 3＝0，值为 True 两个表达式中只要有一个为真(True)时，结果就为真；只有当两个表达式的值都为假(False)时，结果才为假(False)
Not	非	Not 1 ＞ 0，值为 False，由真变假；Not 1 ＜ 0，值为 True，由假变真

3.3.5 表达式

1. 算术表达式

常量、变量、函数是表达式，将它们加圆括号或用运算符做有意义的连接后也是表达式，书写 Visual Basic 的算术表达式应注意与数学表达式在写法上的区别。

(1) 不能漏写运算符，如 3xy 应写作 3 * x * y。

(2) Visual Basic 算术表达式中使用的括号都是小括号。

下面的例子是由数学式写出相应的 Visual Basic 算术表达式。

(1) $\dfrac{1}{1+\dfrac{1}{1+x}}$　　写作：1/(1＋1/(1＋x))

(2) $-(a^2+b^3)\cdot y^4$　　写作：-(a*a+b*b*b)*y^4

(3) $(-a^{b^c}+\sqrt{b})\cdot(a-b)^{-\frac{1}{2}}$　　写作：(-a^(b^c)+b^0.5)*(a-b)^-0.5

(4) 变量 k 是一个两位整数，求其个位数与十位数之和的算术表达式为：k Mod 10 + k \ 10。

2. 字符表达式

下面的例子是字符表达式计算。

"ABCD" & "efg"　　计算后所得表达式的值为"ABCDefg"

"杭州" & "西湖"　　计算后所得表达式的值为"杭州西湖"

字符串连接符“&”具有自动将非字符串类型的数据转换成字符串后再进行连接的功能，而“+”则不能。例如：

"xyz" & 123　　计算后所得表达式的值为"xyz123"

"xyz" + 123　　出现类型不匹配错误

3. 关系表达式

在关系表达式求值时：

(1) 数值数据比较大小，如 3 <= 5 为 True。

(2) 日期类型数据比较先后，如 #11/18/1999# > #03/05/2001# 为 False。

(3) 字符类型数据比较字符的 ASCII 码：若两端首字符相同则比较第二个字符，……，直到比较出相应字符的 ASCII 值大小或两端所有字符比较结束。

例如，"ABCd" >= "ABCD" 为 True；"ABCd" >= "cd" 为 False；"ABCd" = "ABCd"为 True。

两个字符串的“=”关系比较结果为 True，它们必定是两个完全相同的字符串。

4. 逻辑表达式

下面的例子是由条件写出相应的 Visual Basic 逻辑表达式。

(1) 条件“-3<x<3”写作逻辑表达式为：-3 < x and x < 3。

(2) 判断变量 a、b 均不为 0 的逻辑表达式为：a * b <> 0 或 a <> 0 and b <> 0。

(3) 判断变量 a、b 中必有且仅有一个为 0 的逻辑表达式为：a = 0 and b <> 0 or a <> 0 and b = 0 或 a * b = 0 and a + b <> 0。

(4) 判断整型变量 k 是正的奇数的逻辑表达式为：k > 0 and k mod 2 = 1。

(5) 判断变量 a、b、c 是否等比数列中顺序的三项，逻辑表达式为：a/b = b/c。

(6) 平面三点坐标为(x1,y1)、(x2,y2)、(x3,y3)，且 x1≠x2≠x3，判断它们是否共线的逻辑表达式为：(y3-y2)/(x3-x2) = (y2-y1)/(x2-x1)。

3.3.6　运算符优先级

算术运算符之间的运算优先级从高到低如下所示：

指数运算 ^ → 取负- → 乘、除 → 整除\ → 求余 Mod → 加、减

由此可知：指数运算优先级最高，而加、减运算优先级最低。

乘、除和加、减分别为同级运算符，同级运算从左向右进行。在表达式中加括号可以改变表达式的求值顺序。

逻辑运算符的优先级是：先 Not，次 And，后 Or。

算术运算符、关系运算符和逻辑运算符的优先级关系为：算术运算符最高，其次是关系运算符，最后是逻辑运算符。

3.4 常用内部函数

Visual Basic 的内部函数是系统预定义函数，可由用户直接调用。Visual Basic 函数的自变量必须用括号括起来，并满足一定的取值要求，这里主要介绍一些常用内部函数。

3.4.1 数学函数

下列函数的参数均为数值类型。

(1) 三角函数：Sin(x)、Cos(x)、Tan(x)、Atan(x)。

以上函数分别返回正弦值、余弦值、正切值和反正切值。

Visual Basic 没有余切函数，求 x 弧度的余切值可以表示为 1/Tan(x)。

函数 Sin、Cos、Tan 的自变量必须是弧度，如数学式 Sin30°，写作 Visual Basic 的表达式为 Sin(30 * 3.1416/180)。

其他反三角函数可以转换为等值的反正切函数，然后用 Visual Basic 的反正切函数 Atan 计算，如函数 Atan(x/sqr(1-x * x))可以求 $\sin^{-1}x$(不可以写作 Asin(x)，因为 Visual Basic 没有预定义反正弦函数 Asin)。变量名不能与过程名或符号常量名相同。

(2) Abs(x)：返回 x 的绝对值。

(3) Exp(x)：返回 e 的指定次幂，即 e^x。

(4) Log(x)：返回 x 的自然对数。

(5) Sgn(x)：符号函数。当 x>0 时，Sgn(x)的值为 1；当 x=0 时，Sgn(x)的值为 0；x<0 时，Sgn(x)的值为－1。

(6) Sqr(x)：返回 x 的平方根，如 Sqr(16)的值为 4，Sqr(1.44)的值为 1.2。

(7) Int(x)：返回不大于 x 的最大整数，如 Int(7.8)值为 7，Int(－7.8)值为－8。

(8) Fix(x)：返回 x 的整数部分，如 Fix(7.8)值为 7，Fix(－7.8)值为－7。

(9) Rnd 函数：产生 0～1 之间的随机数。

读者可以连续写出多个语句“Print Rnd”，可以看到每个语句的输出结果不同。

一般地，要得到[a,b]之间的随机整数，可用公式“Int(Rnd * (b－a + 1)) + a”。

实际上，Visual Basic 的随机函数发生器是用一个特殊的公式计算“随机数”，而上一次计算的结果作为自变量参与下一次求随机函数的计算，因此所产生的是一种“伪随机数”。

Randomize 语句：该语句的作用是初始化 Visual Basic 的随机函数发生器(为其赋初值)。

下面的例子是由以下条件写出相应的 Visual Basic 表达式。

(1) 求变量 x 绝对值的平方根,算术表达式为：Sqr(Abs(x))。

(2) 判断变量 k 的整数部分是否为两位数的逻辑表达式为：Int(Abs(k)) ＞ 9 and Int(Abs(k)) ＜ 100。

(3) 数学式$\sqrt{s(s-a)(s-b)(s-c)}$写作算术表达式为：Sqr(s * (s－a) * (s－b) * (s－c))。

(4) 将大于 0 的单精度变量 k 四舍五入至小数点后两位的表达式为：Int(k * 100 ＋ 0.5)/100。

(5) 数学式 cos25°＋ctg32°写作算术表达式为：Cos(25 * 3.14159/180) ＋ 1/Tan(32 * 3.14159/180)。

(6) 数学式 $e^{12.6}\cdot\log_e 3-8.6$ 写作算术表达式为：Exp(12.6) * Log(3)－8.6。

(7) 数学式$(e^x-\log_{10}y)\cdot\cos35°$写作算术表达式为：(Exp(x)－Log(y)/Log(10)) * Cos(35 * 3.14159/180)。

(8) N 是大于 0 的整数,求 N 的位数的表达式为：Len(Str(N))－1。

3.4.2　字符函数

(1) Ltrim(x)：返回删除字符串 x 前导空格符后的字符串。如 Ltrim("　abc")的计算结果为字符串"abc"。

Rtrim(x)：返回删除字符串 x 尾随空格符后的字符串。如 Rtrim("abc　")的计算结果为字符串"abc"。

Trim(x)：返回删除字符串 x 前导和尾随空格符后的字符串。如 Trim("　abc　")的计算结果为字符串"abc"。

(2) Left(x,n)：返回字符串 x 前 n 个字符所组成的字符串。

Right(x,n)：返回字符串 x 后 n 个字符所组成的字符串。

Mid(x,m,n)：返回字符串 x 从第 m 个字符起的 n 个字符所组成的字符串。

若 s$ ＝ "abcdefg",则函数 Left(s$,2)返回"ab",函数 Right(s$,2) 返回"fg",函数 Mid(s$,9,3) 返回空字符串,Mid(s$,2,3) 返回"bcd"。

(3) Len(x)：返回字符串 x 的长度,如果 x 不是字符串,则返回 x 所占存储空间的字节数。如函数 Len("abcdefg")的返回值为 7,而函数 Len(k%)的返回值为 2,因为 Visual Basic 用两个字节存储 Integer 类型的数据。

(4) LCase(x)和 UCase(x)：分别返回以小写字母、大写字母组成的字符串。如 LCase("abCDe") 返回"abcde",UCase("abCDe") 返回"ABCDE"。

(5) Space(n)：返回由 n 个空格字符组成的字符串。如执行语句 a$＝"abc"＋Space(5)＋"def"后,变量 a$ 中的字符串为"abc　　def",其中包括 5 个空格字符。

(6) String(n, c)：返回 n 个由字符 c 组成的字符串,c 可以是一个字符的 ASCII 码,也

可以是一个字符串，但 String 函数只取其第一个字符。例如：

```
Dim MyString
MyString = String(5, "*")    '返回 "*****"
MyString = String(5, 42)    '返回 "*****"
MyString = String(10, "ABC")    '返回 "AAAAAAAAAA"
```

(7) InStr(x, y)：字符串查找函数，返回字符串 y 在字符串 x 中首次出现的位置。如果 y 不是 x 的子串，即 y 没有出现在 x 中，则返回值为 0。如 a$ = "abcd efg cd_xy"，则函数 InStr(a$,"cd")的计算结果为 3，因为 a$ 中包含"cd"、第一次出现的位置是在 a$ 中的第三个字符；而函数 InStr(a$,"yx")的返回值为 0，因为字符串 a$ 中不存在子串"yx"。

3.4.3 转换函数

(1) Str(x)：返回把数值型数据 x 转换为字符型后的字符串。如 Str(－123.45)返回"－123.45"；大于 0 的数值转换后符号位用空格表示，如 Str(123.45)返回"123.45"。

(2) Val(x)：把一个数字字符串 x 转换为相应的数值。

如果字符串中包含非数字字符，则仅将第一个数字形式的字符串转换为相应的数值，后面的字符不做处理。如函数 Val("123.45abc678")的计算结果为数值 123.45。

注意，此处函数名 Val 中全是字母，不要将字母 l 误写作数字 1。

(3) Chr(x)：返回以 ASCII 值为 x 的字符，如 Chr(65)返回"A"。

(4) Asc(x)：返回字符串 x 首字符所对应的 ASCII 值，是函数 Chr 的逆运算。如 Asc("ABcde")的返回数值为 65。

3.4.4 日期函数

(1) Date：返回系统当前日期，如语句“Print Date”可以在窗体上输出当前日期。

(2) Time：返回系统当前时间，如语句“Print Time()”可以在窗体上输出当前时间。

(3) Now：返回系统当前日期和时间。

(4) Minute(Now)、Minute(Time)：返回系统当前时间“hh：mm：ss”中的 mm(分)值。

(5) Second(Now)、Second(Time)：返回系统当前时间“hh：mm：ss”中的 ss(秒)值。

3.4.5 测试函数

(1) IIf：可以用来执行简单的条件判断操作，是“If…Then…Else…”结构的简写版本。

格式如下：

```
result=IIf(条件, True 部分, False 部分)
```

在这里，“result”是函数的返回值，“条件”是一个逻辑表达式。当“条件”为真时，IIf 函数返回“True 部分”；而当“条件”为假时，IIf 函数返回“False 部分”。“True 部分”或“False 部分”可以是表达式、变量或其他函数。

例如：m＝IIf(a＞b，a，b)

如果 a＞b 则 m＝a；否则 m＝b。因此，m 的值等于 a 与 b 中较大的值。

注意：IIf 函数中的三个参数都不能省略，而且要求“True 部分”“False 部分”及结果变量的类型一致。

(2) IsDate/IsNull/IsNumeric/IsObject：判断表达式的值是否是指定类型(日期型/无效型/数值型/对象型)。

例如：检测变量。

```
Dim MyVar, MyCheck
MyVar="53"                          '指定值
MyCheck=IsNumeric(MyVar)            '返回 True

MyVar="Help"                        '指定值
MyCheck=IsNumeric(MyVar)            '返回 False
```

(3) IsArray：判断指定变量分量是否为数组。

例如：检测是否为数组。

```
Dim MyArray(1 To 5) As Integer, YourArray, MyCheck   '声明数组变量
YourArray =Array(1, 2, 3)           '使用数组函数
MyCheck=IsArray (MyArray)           '返回 True
MyCheck=IsArray (MyArray)           '返回 True
```

【实例 3.1】 设计一个程序，程序运行界面如图 3.2 所示。界面上有 4 个标签框、1 个文本框和 3 个单选按钮。单击“Sin 正弦值”单选按钮时，文本框中输入的角度会自动计算出其正弦值，结果显示在下面三个标签框里，其他两个三角函数计算过程类似。

图 3.2　实例 3.1 的程序运行界面

(1) 界面设计。新建一个工程，参照如图 3.2 所示的运行界面在窗体上添加 4 个标签框、1 个文本框 Text1 和 3 个单选按钮 Option1、Option2、Option3。按照如表 3.4 所示设置

窗体和各对象的属性。

表 3.4　实例 3.1 的各控件属性设置

对　　象	属　　性	设　　置	说　　明
Form1	Caption	三角函数	窗体标题
Label1	Caption	请输入角度：	提示文字
Label2	Caption	清空	标签框的内容清空
Label3	Caption	清空	标签框的内容清空
Label4	Caption	清空	标签框的内容清空
Text1	Text	清空	文本框的内容清空
Option1	Caption	Sin 正弦值	单选按钮标题
Option2	Caption	Cos 余弦值	单选按钮标题
Option3	Caption	Tan 正切值	单选按钮标题

（2）代码设计。切换到代码设计窗口，编写如下程序代码。

```
Private Sub Option1_Click()      '其中的 Format 函数是将计算结果进行格式化
    Label2.Caption = "Sin" + Text1.Text + "°"
    Label3.Caption = "="
    Label4.Caption = Format(Str(Sin(Val(Text1.Text) * 3.14159265/180)), "0.00")
End Sub
Private Sub Option2_Click()
    Label2.Caption = "Cos" + Text1.Text + "°"
    Label3.Caption = "="
    Label4.Caption = Format(Str(Cos(Val(Text1.Text) * 3.14159265/180)), "0.00")
End Sub
Private Sub Option3_Click()
    Label2.Caption = "Tan" + Text1.Text + "°"
    Label3.Caption = "="
    Label4.Caption = Format(Str(Tan(Val(Text1.Text) * 3.14159265/180)), "0.00")
End Sub
```

【实例 3.2】 设计一个程序，程序运行界面如图 3.3 所示。界面上有一个文本框、一个标签框和一个命令按钮。单击“转换成 ASCII 值”命令按钮时，文本框中输入的字符串首字符会转换成对应的 ASCII 值，结果显示在下面的标签框里。

图 3.3　实例 3.2 的程序运行界面

（1）界面设计。新建一个工程，参照如图 3.3 所示的

运行界面在窗体上添加一个文本框、一个标签框和一个命令按钮。按照如表 3.5 所示设置窗体和各对象的属性。

表 3.5　实例 3.2 的各控件属性设置

对　象	属　性	设　置	说　明
Form1	Caption	转换函数	窗体标题
Label1	Caption	清空	标签框的内容清空
Text1	Text	清空	文本框的内容清空
Command1	Caption	转换成 ASCII 值	命令按钮标题

(2) 代码设计。切换到代码设计窗口,编写如下程序代码。

```
Private Sub Command1_Click()
    Label1.Caption =Str(Asc(Text1.Text))
End Sub
```

习题 3

1. 判断题

(1) 整型变量有 Byte、Integer、Long 三种类型。

(2) Byte 类型的数据,其数值范围在－255～255 之间。

(3) Visual Basic 的 Double 类型数据可以精确表示其数值范围内的所有实数。

(4) 在逻辑运算符 Not、Or、And 中,运算优先级由高到低依次为 Not、Or、And。

(5) 关系表达式是用来比较两个数据的大小关系的,结果为逻辑值。

(6) 一个表达式中若有多种运算,在同一层括号内,计算机按函数运算→逻辑运算→关系运算→算术运算的顺序对表达式求值。

(7) 赋值语句的功能是计算表达式值并转换为相同类型数据后为变量或控件属性赋值。

(8) 用 Dim 定义数值变量时,该数值变量自动赋初值为 0。

(9) 函数 InputBox 的前三个参数分别是输入对话框的提示信息、标题和默认值。

(10) 函数 MsgBox 的前三个参数分别表示默认按钮、按钮样式和图标样式。

2. 选择题

(1) Integer 类型数据能够表示的最大整数为(　　)。

A. 2^{15}　　B. 2^{15-1}　　C. 2^{16}　　D. 2^{16-1}

(2) 货币类型数据小数点后面的有效位数最多只有(　　)。

A. 1 位　　B. 6 位　　C. 16 位　　D. 4 位

(3) 输入对话框 InputBox 的返回值的类型是(　　)。

A. 字符串　　B. 整数　　C. 浮点数　　D. 长整数

(4) 运算符"\"两边的操作数若类型不同,则先(　　)再运算。

A. 取整为 Byte 类型　　B. 取整为 Integer 类型

C. 四舍五入为整型　　D. 四舍五入为 Byte 类型

(5) Int(Rnd * 100)表示的是(　　)范围内的整数。

A. [0,100]　　B. [1,99]　　C. [0,99]　　D. [1,100]

(6) 下列程序段的输出结果是(　　)。

```
a=10: b=10000: x=Log(b)/Log(a): Print "Lg(10000)=";x
```

A. Lg(10000)=5　　B. Lg(10000)=4　　C. 4　　D. 5

(7) 返回删除字符串前导和尾随空格符后的字符串,用函数(　　)。

A. Trim　　B. Ltrim　　C. Rtrim　　D. Mid

(8) Print 语句的一个输出表达式为(　　),则输出包括日期.时间信息。

A. Date　　B. Month　　C. Time　　D. Now

(9) 语句 Print "5 * 5" 的执行结果是(　　)。

A. 25　　B. "5 * 5"　　C. 5 * 5　　D. 出现错误提示

(10) 语句"Form1.Print Tab(10);"#""的作用是在窗体当前输出行(　　)。

A. 第 10 列输出字符"#"　　B. 第 9 列输出字符"#"

C. 第 11 列输出字符"#"　　D. 输出 10 个字符"#"

3. 填空题

(1) 语句"Dim C As _____"定义的变量 C,可用于存放控件的 Caption 的值。

(2) 长整型变量(Long 类型)占用_____个字节。

(3) 表达式 Right(String(65, Asc("abc")), 3)的值是_____。

(4) 表达式 2 * 4^3 + 4 * 6/3 + 3^2 的值是_____。

(5) 表达式 16/2-2 ^ 3 * 7 Mod 9 的值是_____。

(6) 表达式 81\7 Mod 2 ^ 2 的值是_____。

(7) 已知字符串变量 x 存放"1234",表达式 Val("&H"+Left(x, Len(x)/2))的值是_____。

(8) 语句 Print Not 10>15 And 8<5+2 的输出结果为_____。

(9) 设 x 为一个两位数,将其个位和十位数交换后所得两位数的 Visual Basic 表达式是_____。

(10) 用随机函数产生一个两位整数的 Visual Basic 表达式是_____。

(11) 求 a 与 b 之积除以 c 的余数,用 Visual Basic 表达式可表示为_____。

(12) 算术式 ln(x)+sin(30°)的 Visual Basic 表达式为_____。

(13) 声明单精度常量 PI 代表 3.1415926 的语句是_____。

(14) ＃20/5/01＃表示_____类型常量。

(15) 设I为大于0的实数,写出大于I的最小整数的表达式_____。

4. 程序阅读题

```
Private Sub Form_Click()
    Static a As Integer
    Dim b As Integer
    b =a +b +1
    a =a +b
    Form1.Print "a="; a, "b="; b
End Sub
```

请写出单击窗体之后,窗体上的显示结果。

5. 程序设计题

(1) 编程,输入圆的半径,计算并输出圆的面积,按下列要求分别实现。

① 界面设计尽可能美观、大方。

② 创建一个文本框控件用于输入,单击命令按钮后通过标签控件显示计算结果。

③ 修改界面和程序:单击命令按钮后,调用InputBox函数输入数据,通过标签控件显示计算结果。

④ 新建一个文件夹,保存工程(工程文件、窗体文件等,可以用默认的名称,也可以重命名)在该文件夹中,然后退出Visual Basic。

⑤ 要求计算结果具有15位有效位数,重新打开工程,检查程序并决定是否修改。

(2) 编程,创建文本框控件Text1用于输入,单击窗体后通过标签控件Label1显示计算结果(输入数据自行确定),事件过程如下。

```
Private Sub Form_Load()
    Dim x As Single, y As Single
    x=Text1.Text
    Label1.Caption=Sin(x)
End Sub
```

① 运行该程序,体会Single类型数据有效位数不超过6位、Sin函数的自变量为弧度制等。

② 修改该程序,体会其他数学函数、字符运算函数的功能以及使用规则。

Chapter 4

第4章 控制结构

本章学习目标：

- 掌握结构化程序设计的基本思路和方法，理解顺序、选择和循环三种基本结构的功能和特点。
- 熟练掌握 if 和 select 结构的语法及使用，掌握分支控制语句编写选择结构程序的方法。
- 熟练掌握 For…Next、Do…Loop、While…Wend 语句的语法及使用，掌握利用循环控制语句编写循环结构程序的方法。

程序控制结构就是控制程序中语句执行顺序的结构，程序控制结构可以使程序段的逻辑结构清晰，层次分明，可以有效地改善局部程序段的可读性和可靠性，可以保证程序质量并提高开发效率。Visual Basic 6.0 为用户提供了三种最基本的程序控制结构：顺序结构、选择结构和循环结构。

4.1 顺序结构

顺序结构是最基本、最简单的结构，它由若干块组成，按照各块的排列顺序依次执行。其中，这里的块是指三种基本结构之一或者表达式语句等，但不包括转移语句，其执行流程图如图 4.1 所示。

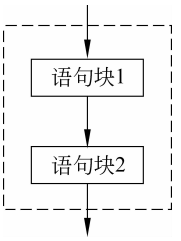

图 4.1 顺序结构

【实例 4.1】 编程，用 InputBox 函数输入球的半径 R，计算并输出球的体积。

(1) 界面设计(略)。

(2) 代码设计。

```
Private Sub Command1_Click()
```

```
    Dim R As Single, V As Single
    R =Val(InputBox("请输入球的半径:"))
    V =4 * 3.1415926535 * R ^ 3/3
    Form1.Print "半径为:" & R & "的球的体积为:" & V
End Sub
```

注意：①在顺序结构中，语句如有一定的逻辑顺序，其位置不能任意调换；②数学中圆周率π一般使用近似值来代替。

4.2 选择结构

在程序设计过程中，往往需要根据某些条件做出判断，决定选择哪些语句执行或不执行某些语句，这时候可以采用选择结构，选择结构也称分支结构。Visual Basic 语言中为我们提供了 If 结构和 Select 结构。其中，If 结构又分为单分支结构、双分支结构和多分支结构。

4.2.1 单分支 If 结构

单分支 If 结构的形式为：

```
If <表达式>Then
    <语句块>
End If
```

或

```
If <表达式>Then <语句>
```

功能：先计算表达式的值，若表达式值为 True，则执行语句，否则不执行。单分支 If 结构的执行流程图如图 4.2 所示。

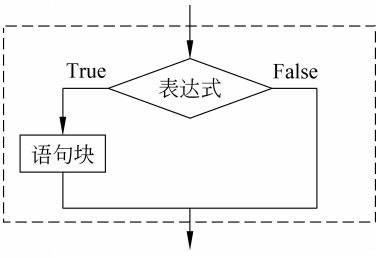

图 4.2 单分支 If 结构

【实例 4.2】 编程，输入 x、y，仅当 x＜y 时交换 x、y 值，然后输出 x、y 的值（在 Text 控件中输入，输出到 Label 控件）。

（1）界面设计。建立文本框控件 Text1 和 Text2、标签控件 Label1。

（2）代码设计。编制事件过程 Form_Click 如下（单击窗体响应）。

```
Private Sub Form_Click()
    Dim x as Single,y as Single,Temp as Single
```