第3章 数据类型、运算符与表达式(常用内部函数)

Visual Basic 应用程序包括两部分内容,即界面和程序代码。其中,程序代码的基本组成单位是语句(指令),而语句是由不同的“基本元素”构成的,包括数据类型、常量、变量、内部函数、运算符和表达式等。本章将重点介绍其数据类型、运算符、表达式及常用的内部函数。

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。为了对数据进行快速处理和有效地利用存储空间,所有高级语言都对数据进行分类处理,不同类型数据的操作方式和取值范围不同,所占存储空间的大小也不同。Visual Basic 数据类型分为基本数据类型和自定义数据类型。

3.1.1 基本数据类型

基本数据类型是系统定义的数据类型。Visual Basic 6.0 提供的基本数据类型主要有数值型和字符型数据,此外还提供了字节、货币、对象、日期、逻辑和变体数据类型,如表 3-1 所示。

表 3-1 Visual Basic 标准数据类型

数据类型	关键字	类型符	前缀	占字节数	取值范围
字符型	String	$	str	与字符串长度有关	定长字符串:0～65 535 变长字符串:0～2.0×10^{10}个字符
字节型	Byte	无	bty	1	0～255
整型	Integer	%	int	2	－32 768～32 767
长整型	Long	&	lng	4	－2 147 483 648～2 147 483 647
单精度型	Single	!	sng	4	负数:－3.402 823E＋38～－1.401 298E－45 正数:1.401 298E－45～3.402 823E＋38
双精度型	Double	#	dbl	8	负数:－1.797 693 134 862 33E＋308～－4.940 656 458 412 47E－324 正数:4.940 656 458 412 47E－324～1.797 693 134 862 33E＋308

续表

数据类型	关键字	类型符	前缀	占字节数	取值范围
货币型	Currency	@	cur	8	－922 337 203 685 477.580 8～ 922 337 203 685 477.580 7
逻辑型	Boolean	无	bln	2	True 或 False
日期型	Date	无	dtm	8	1/1/100～12/31/9999
对象型	Object	无	obj	4	任何对象
变体型	Variant	无	vnt	按需分配	上述有效范围之一

1. 数值型数据

在 Visual Basic 中，数值型数据分为整型数和实型数两大类。

1) 整型数

整型数是不带小数点和指数符号的数，在机器内部以二进制补码形式表示。根据表示数的范围不同，可以分为整型和长整型。

(1) 整数(Integer，类型符为“%”)：整型数在内存中占 2B(16b)，其取值范围为－32 768～ 32 767。也可以在一个整数后加上类型符“%”来表示一个 Integer 类型的整数，如：234%。

(2) 长整数(Long)：长整型数在内存中占 4B(32b)，其取值范围为－2 147 483 648～2 147 483 647。也可以在一个数据后面加上类型符“&”来表示一个 Long 类型的整数，如：34 567&。

2) 实型数

实型数也称浮点数，是带小数部分的数值。浮点数由三部分组成：符号、指数及尾数。根据表示数的范围不同，可以分为单精度型和双精度型数。其中，双精度数表示数的精度比单精度数要高，表示的数的范围也比单精度数大。

(1) 单精度数(Single，类型符为“!”)：单精度数在内存中占 4B(32b)，其中，符号占 1b，指数占 8b，其余 23b 表示尾数。单精度可以精确到 7 位十进制数。其负数的取值范围为－3.402 823E＋38～－1.401 298E－45，正数的取值范围为 1.401 298E－45～3.402 823E＋38。指数用“E”(或“e”)表示。

(2) 双精度数(double，类型符为“#”)：双精度数在内存中占 8B(64b)，其中，符号占 1b，指数占 11b，其余 52b 表示尾数。双精度可以精确到 15 位或 16 位十进制数。其负数的取值范围为－1.797 693 134 862 33E＋308～－4.940 656 458 412 47E－324，正数的取值范围为 4.940 656 458 412 47E－324～ 1.797 693 134 862 33E＋308。双精度浮点数的指数可用“D”(或“d”)表示，VB 会自动转换成 E。

2. 字符型数据

字符串(String)是用双引号界定的一个字符序列，由 ASCII 字符(除双引号和回车符之外)、汉字及其他可打印字符组成。

Visual Basic 的字符串有两种，即可变长度字符串和固定长度字符串。可变长度字符串是指在程序运行期间字符串的长度不固定，固定长度字符串是指在程序运行期间长度保持

不变的字符串。

3. 布尔型数据

布尔型数据(Boolean)也称为逻辑型,用两字节存储,它只取两种值,即 True(真)或 False(假)。当布尔型数据转换为整型数据时,True 转换为－1,False 转换为 0;当其他类型数据转换为布尔型时,非 0 数转换为 True,0 转换为 False。

4. 日期型数据

日期型数据(Date)用于表示日期和时间,在内存中占用 8B,以浮点数形式存储,可以表示的日期范围从 100 年 1 月 1 日至 9999 年 12 月 31 日,而时间可以从 0:00:00 至 23:59:59。

5. 货币型数据

货币型数据(Currency)主要用来表示货币值,在内存中占 8B(64b),精确到小数点后 4 位(小数点前 15 位),在小数点后 4 位以后的数字将被舍去,属于定点数,即小数点位置固定的数。其取值范围为:－922 337 203 685 477.580 8～922 337 203 685 477.5807。

6. 对象型数据

对象型数据(Object)可用来引用当前应用程序中或其他应用程序中的对象,用 4 字节存储。

7. 变体型数据

变体型数据(Variant)是一种特殊的数据类型,是所有未定义类型变量的默认类型。如果程序中的变量未定义类型,VB 将视之为变体类型。它可以表示任意值,包括数值、字符串、日期时间等。

3.1.2 自定义数据类型

如果单个基本数据不能满足用户的需要,VB 允许用户利用“Type”关键字来自定义所需的数据类型。自定义数据类型是由已存在的若干个数据类型组合而成的,常常把这种结构称为“记录”。例如,一个学生的基本信息包括“学号”“姓名”“性别”“年龄”“入学成绩”等数据,可以用 Visual Basic 所提供的 Type 语句让用户自己定义数据类型,它的形式如下。

```
Type 数据类型名
    元素名 1 As 类型
    元素名 2 As 类型
        …
        …
    元素名 n As 类型
End Type
```

```
例: Type StudType
        Xh As String
        Xm As String
        Xb As String
        Nl As Integer
        Score As Single
    End Type
```

一旦定义好了新数据类型，就可以用这种数据类型来定义变量。例如：

```
Dim Student As StudType
```

定义了一个名为 Student 的变量，属于 StudType 类型。

定义了这个变量后，就可以使用它来访问该变量中的各个元素了，其形式为：

```
变量名.元素名
```

例如：

```
Student.Nl                    '表示访问 Student 变量中的 Nl 元素
```

说明：自定义类型的作用域默认是 Public，那么默认时，必须将自定义类型的定义放在标准模块(.BAS 文件对应的模块)中。如果有自定义类型的定义放在窗体模块(.frm 文件对应的模块)中，则必须在定义前加上 Private。

3.2　常量与变量

根据程序的需要，可以将数据分为不同的数据类型。在程序中，不同类型的数据既可以以常量的形式出现，也可以以变量的形式出现。常量在程序执行期间是不发生变化的，而变量的值是可变的，它代表内存中指定的存储单元。

3.2.1　常量

Visual Basic 中常量分为三种：普通常量、符号常量和系统常量。

1. 普通常量

普通常量也称为直接常量，可从字面形式上判断其类型。Visual Basic 中常用的直接常量有：整型常量、实型常量、字符串常量、布尔常量和日期常量。

1) 整型常量

整型常量就是整常数。在 Visual Basic 中，经常使用的整型常数有三种进制，分别是十进制、八进制和十六进制。

(1) 十进制整数

十进制整数的数码为 0～9，其取值范围为－32 728～32 727，如：15、－890。

(2) 八进制整数

八进制整数必须以 &或 &O(字母 O)作为八进制数的前缀，数码的取值为 0～7，其取值范围为 &O0～&O177777，如：&O15(或 O15)表示八进制数，其对应的十进制数是 13。

(3) 十六进制整数

十六进制整数必须以 &H(或 &h)作为十六进制数的前缀，数码的取值为 0～9，A～F(或 a～f)，其取值范围为 &H0～&HFFFF，如：&H15 表示十六进制数，其对应的十进制数是 21。

说明：上面整数表示的是整型(Integer)，如果要表示长整型(Long)，则在数的后面加

类型符“&”，如：15&、&O15&、&H15& 分别表示十进制、八进制和十六进制长整型常数15，$(15)_8$，$(15)_{16}$。

2）实型常量

实型常量也称为浮点数。在 Visual Basic 中，浮点数分为单精度数和双精度数。实型常量有以下两种表示形式。

（1）十进制小数形式

它由正负号（+、-）、数字（0～9）和小数点（.）或类型符号（!、#）组成，其中，“!”表示单精度数，“#”表示双精度数。

例如：0.235、.235、235.0、235!、235# 等都是十进制小数形式。系统默认的实型常量都是双精度类型，即 235.0 和 235# 是等价的常量。除非特别用单精度类型符“!”加以说明的，如 235!。

（2）指数形式

它由符号、指数及尾数组成，其指数可用“D”或“E”表示。当指数为正数时，正号也可以省略。

例如：2.35E+2（或 2.35E2）和 2.35D+2 相当于 235 或 2.35×10^2。

同一个实数的指数表示形式有很多种，如：235.0 可以表示为 2.35E+2、0.235E+3、235E+1 或 0.0235E+5。一般将 2.35×10^2 称为规范化的指数形式。

3）字符串常量

字符串常量是一个用双引号括起来的字符序列，如" Visual Basic "、" x+y "、" 123 "、" "等。其中，" "表示空字符串，即双引号之间没有任何字符。

如果字符串中有双引号，如 abc " 123，则用连续两个双引号表示，即：" abc " " 123"。

4）布尔常量

布尔型数据又称为逻辑型，它只有两种可能的取值，即 True（逻辑真）或 False（逻辑假）。

5）日期常量

任何在字面上可以被认作日期和时间的字符串，只要用两个“#”括起来，都可以作为日期型常量。

例如：#05/16/2013#、#September 2,2013#、#05/16/2013 12:30:00#、# 9:00:00 AM #。

2. 符号常量

在 Visual Basic 中，可以定义符号常量，用来代替数值或字符串。这样做可以提高程序的可读性和可维护性。

符号常量说明的一般格式：

```
Const 常量名[As 类型|类型符号] = 常量表达式
```

说明：

（1）常量名：常量名的命名规则与变量名相同（见 3.2.2 节）。为便于与一般变量区别，符号常量名常常采用大写字母。

(2)[AS类型|类型符号]：用来说明常量的数据类型。方括号表示该项可以省略，若省略该项，则由右边常量表达式值的数据类型决定。

(3) 常量表达式：可以是数值常量、字符串常量以及由这些常量与运算符组成的表达式。在一行可以定义多个符号常量，各常量之间用逗号隔开。

例如：

```
Const PI# = 3.1415926535
Const PI As Double = 3.1415926535
Const PI2 = 2 * PI                    '声明 PI2 是符号常量,值为 2 * 3.1415926535
Const MAX As Integer = 256,MYSTR As string = "happy"
Private Const DATATODAY As Date = #10/8/2009#
```

注意：第一个和第二个定义形式等价。另外，如果符号常量只在过程或某个窗体模块中使用，则在定义时应加上关键字 Private(可以省略)。如果符号常量在多个模块中使用，则必须在标准模块中定义，并在开头加上关键字 Public。

3. 系统常量

在 Visual Basic 的对象库中，提供了应用程序和控件的系统常量。例如，在“对象浏览器”中的 Visual Basic(VB)、Visual Basic for Application(VBA)等对象库中列举了 Visual Basic 的常量，其他提供对象库的应用程序如 Microsoft Excel、Microsoft Project 以及每个 ActiveX 控件的对象库等也提供了常量。这些常量可与应用程序的对象、方法和属性一块使用。

系统内部定义的常量名使用两个字符的前缀，如 Visual Basic(VB)、Visual Basic for Application(VBA)对象库中的常量名的前缀是“vb”。

系统常量的使用，提高了程序的可读性和可维护性。同时，由于常量值在 Visual Basic 的后续版本中可能还会改变，因此使用系统常量还可保持程序的兼容性。

例如，vbCrLf 表示“回车换行”符，vbRed 表示红色，vbKeyReturn 表示 Enter 键等，再如窗体状态属性 WindowState 可接受的常量如表 3-2 所示。

表 3-2　WindowState 常量

常　量	值	描　述
vbNormal	0	正常
vbMinimized	1	极小化
vbMaximized	2	极大化

在程序中使用语句“Form1. WindowState＝vbMaximized”，将窗口极大化，显然要比使用语句“Form1. WindowState＝＝2”易于阅读和理解。

3.2.2　变量

在程序的运行过程中，变量用来临时存储数据，每个变量都有其名字和相应的数据类型。变量的名字称为变量名，通过变量名来引用该变量，而数据类型则决定了该变量的存储方式和内存中占据存储单元的大小。

1. 变量的命名规则

Visual Basic 变量用一个标识符来命名，标识符的命名遵循以下规则。

(1) 必须以字母或汉字开头，由字母、汉字、数字和下画线组成的字符串。

(2) 不能使用 Visual Basic 中的关键字，如 If、While、End 等。

(3) 不区分大小写，如 MAX 和 max 是同一个变量。

(4) 变量名最长为 255 个字符，且字符之间必须并排书写，不能出现上下标。

2. 变量的声明

在程序中使用变量前，一般必须先声明变量名及其数据类型，系统根据所做的声明为变量分配相应的存储单元。在 Visual Basic 6.0 中可以不事先声明变量，而直接引用。因此 Visual Basic 中声明变量可分为显式声明和隐式声明两种。

1) 显式声明

程序中使用 Dim 语句声明的变量，就是显式声明。

Dim 语句的一般格式：

```
Dim 变量名 [As 类型]
Dim 变量名 [类型符]
```

注意：

(1) 变量名：是用户定义的标识符，应遵循变量命名规则。

(2) [As 类型]和[类型符]：都是用来说明变量的数据类型。其中，类型可以是 Integer、Single、String 等，类型符可以是%、!、$等。方括号表示可以省略，若省略该项，所声明的变量默认为 Variant 型。

(3) 一条 Dim 语句可同时定义多个变量，但每个变量必须有自己的类型说明，类型说明不能共用，变量声明之间用逗号分隔。

例如：

```
Dim x As Integer, y As Single, z
```

等价于：

```
Dim x%, y!, z
```

在上例中，x 定义为整型，y 定义为单精度型，z 没有声明其数据类型，则属于 Variant 型。

字符串变量的存储空间与字符串的长度有关。在 Visual Basic 中，根据其存放字符串长度是否固定，有以下两种定义方式。

```
Dim 字符串变量 As String
Dim 字符串变量 $
Dim 字符串变量 As String * 字符数
```

注意：

(1) 第一种和第二种方式定义的是变长字符串，最多可存放 2MB 个字符；第二种方式定义的是定长字符串，存放的最多字符数由“*”后的“字符数”决定。

(2) 对于定长字符串,若赋值的字符数少于定义的字符数,则右端补空格;反之,将多余部分截取。

(3) 在 Visual Basic 中,一个汉字与一个西文字符一样,都算作一个字符,不过一个汉字在内存中占两字节。

例如:

```
Dim str1 As String                '声明变长字符串变量
Dim x$                            '声明变长字符串变量
Dim str2 As String * 10           '声明定长字符串变量
```

2) 隐式声明

在 Visual Basic 中,如果一个变量未经 Dim 语句声明便直接使用,称为隐式声明。使用时,系统会以该名字自动创建一个变量,并默认为变体类型。这样做给初学者带来了方便,但是正因为这一点方便,可能给程序带来不易发生的错误,同时降低程序的执行效率。

例如,下面是一个很简单的程序,其使用的变量 a,b,Sum 都没有事先定义。

```
Private Sub Form_Click()
  Sum = 0
  a = 10: b = 20
  Sum = a + b
  Print "Sum = "; Sun
End Sub
```

则程序的运行结果为:Sum=0 而不是 Sum=30。这是因为把 Print "Sum=";Sum 错误地写成:Print "Sum=";Sun,导致程序运行结果错误,但却不会给出任何错误信息。因为程序中的 a,b,Sum 变量并没有声明,Visual Basic 系统分辨不出 Sun 是写错了,还是新的变量。

为了避免上述麻烦,应该是"先声明变量,后使用变量",从而提高程序的执行效率。要做到这一点,必须要求对变量做强制显式声明,其方法有以下两种。

(1) 在各种模块的声明部分中添加如下语句:

```
Option Explicit
```

(2) 也可以选择"工具"→"选项"菜单命令,在弹出的"选项"对话框中打开"编辑器"选项卡,再勾选"要求变量声明"复选框,如图 3-1 所示。这样就可以在新模块中自动插入 Option Explicit 语句。

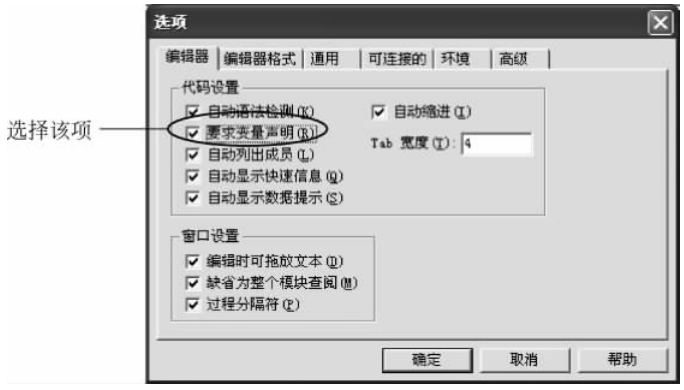

图 3-1　变量的显式声明

3. 变量的默认值

当变量被声明后，如果没有给这个变量赋初值，Visual Basic 会自动给该变量赋予一个默认值。对于不同类型的变量，默认值如表 3-3 所示。

表 3-3 WindowState 变量

变 量 类 型	默认值(初值)
变长字符串	空字符串""
定长字符串	空格字符串，其长度等于定长字符串的字符个数
数值型	0(或 0.0)
逻辑型	False
日期型	#0:00:00#
对象型	Nothing
变体类型	Empty

3.2.3 变量的作用域

变量的作用域指的是变量的有效范围，即变量的“可见性”。声明变量的位置不同以及声明时使用的关键字不同，所声明变量的有效范围也不一样。Visual Basic 中变量有三种作用域：过程级、窗体/模块级和全局级。

1. 过程级变量

过程级变量也称为局部变量，只在所定义的过程内有效，在其他过程中无效。某个过程的执行只对该过程内的变量产生作用，对其他过程中相同名字的局部变量没有任何影响。因此，在不同的过程中可以定义相同名字的局部变量，它们之间没有任何关系。

2. 窗体/模块级变量

只在该窗体模块或标准模块的各个过程中使用，其他模块中的代码不能引用。一般是在窗体或模块的通用声明部分使用 Dim 语句或 Private 语句定义的变量。

3. 全局变量

在整个工程的所有模块中均有效。一般在窗体或模块的通用部分用 Public 语句定义的变量是全局变量。如果是在模块中定义的全局变量，则可在任何过程中通过变量名直接访问；如果是在窗体中定义的全局变量，在其他模块中引用时，须在变量前加该窗体名，即定义该变量的窗体名.变量名。

三种变量作用域如表 3-4 所示。

表 3-4　不同作用域变量的使用规则

作用范围	局部变量	窗体/模块级变量	全局变量	
			窗　体	标准模块
声明方式	Dim，Static	Dim，Private	Public	
声明位置	在过程中	窗体/模块的“通用声明”段	窗体/模块“通用声明”段	
能否在本模块的其他过程存取	不能	能	能	
能否被其他模块存取	不能	不能	能，但须在变量名前加窗体名	能

3.3　运算符与表达式

运算符是描述各种不同运算关系的符号。被运算的对象，即数据，称为运算量或操作数。运算符和操作数组合成表达式，实现对数据的加工。在 Visual Basic 中有 4 种运算符：算术运算符、字符串运算符、关系运算符和逻辑运算符。各运算符的优先顺序如表 3-5 所示。

表 3-5　Visual Basic 的运算符

运算符种类	优　先　级	运算符(按优先顺序排列)
算术运算符	1	^、*、/、\、Mod、+、-
字符运算符	2	+、&
关系运算符	3	=、>、>=、<、<=、<>、Like、Is
逻辑运算符	4	Not、And、Or、Xor、Eqr、Imp

3.3.1　算术运算符与算术表达式

算术运算符是常用的运算符，用来执行简单的算术运算。Visual Basic 提供了 8 个算术运算符，表 3-6 给出了各算术运算符及其优先级。

表 3-6　Visual Basic 的算术运算符

运　算　符	含　义	优　先　级	实　例	结　果
^	乘方	1	3^2	9
-	负号	2	-3+2	-1
*	乘	3	5*3	15
/	除		5/2	2.5
\	整除	4	5\2	2
Mod	取余数	5	5 Mod 2	1
+	加	6	2+3	5
-	减		3-1	1

在 8 个算术运算符中，除取负(-)是单目运算符外，其他均为双目运算符(需要两个操作数)。加(+)、减(-)、乘(*)、除(/)、取负(-)等几个运算符的含义与数学中的基本相

同。下面介绍其他几个运算符的操作。

1. 指数运算

指数运算用来计算乘方和方根，其运算符为“^”。计算 a ^ b 时，若左操作数 a 为正实数，则右操作数 b 可为任意数值。若左操作数 a 为负实数，则右操作数 b 必须是整数。

例如：

```
10 ^ 2        '结果为 100          8 ^ ( - 1/3)      '结果为 0.5
( - 8)^2      '结果为 64           ( - 8)^(1/3)      '错误
```

2. 整除运算

整除运算符是“\”，其操作数一般为整型数。当操作数为带有小数的实型数据时，则先将操作数四舍五入为整数，然后再做整除运算。

例如：

```
10\2     '结果为 5            10\2.5     '结果为 3
```

3. 取模运算

取模运算符是“mod”，其操作数一般为整型数。当操作数为带有小数的实型数据时，则先将操作数四舍五入为整数，然后再做求余运算，而且求余结果的符号始终与第一个操作数的符号相同。

例如：

```
 10 Mod 3     '结果为 1       10 Mod 3.5     '结果为 2
 - 5 Mod 2    '结果为 - 1      5 Mod  - 2    '结果为 1
```

4. 算术运算符的优先级

Visual Basic 中，算术运算符的优先级在表 3-6 中给出了。当一个表达式中含有多种算术运算符时，按级别由高到低进行，同级运算符从左到右运算。如果表达式中含有括号，则先计算括号内表达式的值；有多层括号时，从内层括号到外层括号计算。例如：

```
5 + 2 * 10 mod 10\9 / 3  + 2 ^ 2          '结果为 11
```

3.3.2 字符串运算符与字符串表达式

字符串运算符有两个：“&”和“+”，它们都是将两个字符串依次连接起来，生成一个新的字符串。由字符串运算符与操作数组成的表达式称为字符串表达式。

例如：

```
"Abc" & "123"                          '结果为 Abc123
"100"  +  "123"                        '结果为 100123
```

在字符串变量后使用“&”时应注意，变量与运算符“&”间要加一个空格。这是因为符号“&”还是长整型的类型定义符，当变量与符号“&”接在一起时，Visual Basic 先把它作为

类型定义符处理。

此外，还需注意连接符“&”和“+”的区别。

“&”：连接符两旁的操作数不管是字符型还是数值型，进行连接操作前，系统先将操作数转换成字符型，再进行连接。

“+”：连接符两旁的操作数均为字符型，才能完成字符串的连接操作，此时和“&”连接符完全等价。若一个为数字字符型操作数，另一个为数值型，则自动将数字字符转换为数值型，然后进行算术运算；若一个为非数字字符型，另一个为数值型，则出错。

例如：

```
"100" + 123                    '结果为 223
"Abc" + 123                    '出错
```

3.3.3　关系运算符与关系表达式

关系运算符都是双目运算，用来比较两个操作数之间的关系。由关系运算符与操作数组成的式子称为关系表达式。关系表达式的运算结果是一个逻辑值，若关系成立，则结果为True；若关系不成立，则结果为 False。Visual Basic 提供了 9 种关系运算符，表 3-7 给出了各关系运算符及其优先级。

表 3-7　Visual Basic 的关系运算符

运算符	含　义	实　例	结　果
=	等于	"ABCD" = "ABR"	False
>	大于	"ABCD"> "ABR"	False
>=	大于或等于	"bc" >= "abcd"	True
<	小于	23<3	False
<=	小于或等于	"23" <= "3"	True
<>	不等于	"ABC"<> "abc"	True
Like	字符串匹配	" This " Like " * is"	True
Is	对象引用比较		

对于关系运算符需注意以下规则。

(1) “=”表示关系运算中的等于，注意与赋值运算符“=”含义不同，详见 4.2.1 节。

(2) 当两个操作数均为数值型时，按其大小比较。对于单精度数或双精度数进行比较时，应特别小心，运算可能会给出非常接近但不相等的结果。

例如：

```
1.0/3.0 * 3.0 = 1.0            '结果为 False
```

在数学上显然是一个恒等式，但在计算机上执行时可能会给出假(False)。因此应避免对两个浮点数做“相等”或“不相等”的判断。上式可改写为：

```
Abs(1.0/3.0 * 3.0 - 1.0)< 1E - 5        'Abs 是求绝对值
```

只要它们的差小于一个很小的数(这里是 10 的 -5 次方)，就认为 1.0/3.0 * 3.0 与 1.0 相等。

（3）当两个操作数均为字符型时，则按字符的 ASCII 码值从左到右逐一比较，即首先比较两个字符串的第一个字符，其 ASCII 码值大的字符串大，如果第一个字符相同，则比较第二个字符，以此类推，直到出现不同的字符为止。

例如：

```
"How" > "Hello",等价于"o" > "e"的比较。结果为: True
```

（4）不同类型数据可以进行比较。当数值型数据与可转换为数值型的数据比较时，如 23＞“123”，则按数值大小比较，结果为 False。当数值型数据与不能转换成数值型的字符型比较时，如 23＞“ABC”，就会出现错误。

（5）Like 运算符用来比较字符串表达式和 SQL 表达式中的样式，主要用于数据库查询。Is 运算符用于两个对象变量引用比较，也可以在 Select Case 语句中使用。

（6）关系运算符的优先级相同，运算顺序是从左向右。

3.3.4 逻辑运算符与表达式

逻辑运算也称为布尔运算。用逻辑运算符连接两个或多个关系式，组成一个布尔表达式。Visual Basic 中提供了 6 种逻辑运算符，表 3-8 给出了各逻辑运算符及其优先级。

表 3-8 Visual Basic 的逻辑运算符

运算符	含义	优先级	说　明	实例	结果
Not	取反	1	当操作数为假时，结果为真 当操作数为真时，结果为假	Not F	T
And	与	2	两个操作数均为真，结果才为真	T And F T And T	F T
Or	或	3	两个操作数有一个为真，结果为真	T Or F F Or F	T F
Xor	异或		两个操作数不相同，即一真一假时，结果才为真	T Xor F T Xor T	T F
Eqv	等价	4	两个操作数相同时，结果才为真	T Eqv F F Eqv F	F T
Imp	蕴含	5	第一个操作数为真，第二个操作数为假时，结果才为假，其余结果均为真	T Imp F T Imp T	F T

对于逻辑运算符需注意以下规则。

（1）数学中判断 X 是否在区间[a，b]时，习惯上写成 a≤X≤b，但在 Visual Basic 中不能写成：

```
a <= X <= b
```

应写成：

```
a <= X And X <= b
```

（2）算术运算中的操作数也可以是逻辑型，若是逻辑型，系统会自动转换成数值型（True 为－1，False 为 0）后再运算。

例如：

```
30 - True                          '结果为 31
10 - False + "3"                   '结果为 13
```

以上介绍了 Visual Basic 中常用的运算符和表达式。为了检验各个表达式的操作，可以编写事件过程，如 Form_Click 或 Command1_Click。但是这样做比较烦琐，因为必须执行事件过程才能看到结果。为此，Visual Basic 提供了命令行解释程序(Command Line Interpreter，CLI)，可以通过命令行直接显示其表达式的执行结果，这种方式称为直接方式。

在“立即”窗口中可以输入命令，命令行解释程序对输入的命令进行解释，并立即响应，与 DOS 下命令行的执行情况类似。

例如：

```
x = 2: y = 4   <CR>
print x + y   <CR>
6
```

其中，<CR>为回车键。其“立即”窗口运行情况如图 3-2 所示。

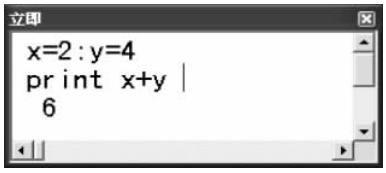

图 3-2　“立即”窗口运行情况

3.4　常用内部函数

在 Visual Basic 中，有内部函数和用户自定义函数两类。用户自定义函数是用户自己根据需要定义的函数。内部函数也称为标准函数或库函数，它们是 Visual Basic 系统为实现一些特定的功能而设置的函数，可以在程序中直接调用。本章将介绍常用的内部函数。

3.4.1　数学函数

数学函数用于各种数学运算，包括三角函数、求平方根、绝对值及对数、指数等。常用的数学函数如表 3-9 所示。表中，x 是一个数值表达式。

表 3-9　常用数学函数

函　数	功　能	实　例	结　果
Abs(x)	返回 x 的绝对值	Abs(－10)	10
Sqr(x)	返回 x 的平方根	Sqr(25)	5
Exp(x)	求 e 的 x 次方，即 e^x	Exp(2)	7.389
Log(x)	求自然对数 lnx	Log(10)	2.303
Sin(x)	返回 x 的正弦值	Sin(0)	0
Cos(x)	返回 x 的余弦值	Cos(0)	1
Tan(x)	返回 x 的正切值	Tan(0)	0
Atn(x)	返回 x 的反正切值	Atn(0)	0
Sgn(x)	符号函数。当 x 为正数时，返回值为 1；当 x 为负数时，返回值为－1；当 x 为 0 时，返回值为 0	Sgn(－9)	－1
Rnd[(x)]	随机函数	Rnd	0～1 的数

说明：

(1) 三角函数的自变量 x 是一个数值表达式。其中，Sin、Cos 和 Tan 的自变量是以弧度为单位的角度，而 Atn 函数的自变量是正切值，它返回正切值为 x 的角度，以弧度为单位。在一般情况下，自变量以角度给出，可以用下面的公式转换为弧度：

$$1(\text{度})=\pi/180=3.14159/180(\text{弧度})$$

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

(2) 随机函数 Rnd[(x)]。Rnd 函数可以不要参数，其括号也可省略。返回[0,1)(即包括 0，但不包括 1)的双精度随机数。若要产生 1～100 的随机数，则可通过下面的表达式实现。

```
Int(Rnd * 100) + 1                    '包括 1 和 100,Int 表示取整
Int(Rnd * 99) + 1                     '包括 1,不包括 100,包括 99
```

产生[N,M]区间的随机数，Visual Basic 可表示为：Int(Rnd*(M－N＋1))＋N。

调用 Rnd 函数之前，使用 Randomize 语句可产生不相同的随机数序列，避免同一序列的随机数反复出现。其格式为：

```
Randomize[n]
```

这里的 n 是一个整型数，作为随机数发生器的种子。若省略 n，则根据系统时钟获得种子。

例如，下段程序每次运行，都会产生不同序列的 10 个[1,100]的随机整数。

```
Randomize
For i = 1 To 10
   Print Int(Rnd * 100) + 1
Next i
Print
```

说明：如果程序中没有 Randomize 语句，则每次都产生相同序列的 10 个[1,100]的随机整数。

3.4.2 字符串函数

字符串函数主要用于各种字符串处理。Visual Basic 提供了大量的字符函数，具有很强的字符串处理能力。常用的字符串函数如表 3-10 所示。

表 3-10 常用字符串函数

函　数	功　能	实　例	结　果
Ltrim(s)	删除字符串 s 前导空格	Ltrim("　abc　")	"abc　"
Rtrim(s)	删除字符串 s 尾部空格	Rtrim("　abc　")	"　abc"
Trim(s)	删除字符串 s 前导和尾部空格	Trim("　abc　")	"abc"
Left(s,n)	取字符串 s 前 n 个字符	Left("abcef",3)	"abc"
Right(s,n)	取字符串 s 后 n 个字符	Right("abcef",3)	"cef"
Mid(s,m,n)	取字符串 s 从第 m 个字符开始的 n 个字符	Mid("abcde", 2, 3)	"bcd"
Len(s)	求字符串 s 的长度	Len("abcde")	5

续表

函　数	功　能	实　例	结　果
Lcase(s)	大写字母转换成小写字母	Lcase("ABcd")	"abcd"
Ucase(s)	小写字母转换成大写字母	Ucase("ABcd")	"ABCD"
Space(n)	生成有 n 个空格的字符串	Space(3)	"　　"
Instr(s1,s2)	求字符串 s2 在字符串 s1 中首次出现的位置;如果 s2 没有出现在 s1 中,则返回值为 0	InStr("ABCDEF", "CD")	3
String(n,s)	生成由字符串 s 的首字符构成的长度为 n 的新字符串	String(3, "abc")	"aaa"

3.4.3　转换函数

转换函数用于数据类型或形式的转换,包括整型、实型、字符串之间以及数值与 ASCII 字符之间的转换。常用的转换函数如表 3-11 所示。表中的 x 和 n 是数值表达式,s 是字符串表达式。

说明:

(1) Chr 函数可以得到那些非显示的控制字符。例如:

```
Chr(13)      '回车符          Chr(13) + Chr(10)      '回车换行
Chr(7)       '响铃 Beep       Chr(8)                 '退格符
```

(2) Str(x)返回把数值型数据 x 转换为字符型后的字符串,字符串的第一位一定是空格(x 是正数)或是符号(x 是负数),小数点组后的"0"将被去掉。

(3) 假设 x=12.3467,如果要求 x 保留小数点后三位,有几种方法实现?

① Fix(x * 1000+0.5)/1000;

② Int(x * 1000+0.5)/1000;

③ Round(x,3)。

表 3-11　常用类型转换函数

函　数	功　能	实　例	结　果
Fix(x)	返回 x 整数部分	Fix(-4.5)	-4
		Fix(4.5)	4
Int(x)	返回不大于 x 的最大整数	Int(-4.5)	-5
		Int(4.5)	4
Chr(x)	将 x 的值转换成对应的字符	Chr(97)	"a"
Asc(s$)	返回字符串 s 中首字符的 ASCII 码值	Asc("a")	97
Val(s$)	将数字字符串 s 转换成数值	Val("123abc")	123
		Val("abc123")	0
Str(x)	将 x 的值转换成字符串	Str(1.45)	" 1.45"
Hex(x)	将十进制数转换成十六进制数	Hex(100)	64
Oct(x)	将十进制数转换成八进制数	Oct(100)	144
Round(x,n)	按四舍五入原则对 x 保留 n 位小数。省略 n 时,则对 x 四舍五入取整	Round(3.176,2)	3.18
		Round(-3.6)	-4

3.4.4 日期和时间函数

日期和时间函数用来返回系统当前的日期和时间。常见的日期和时间函数如表 3-12 所示。表中的 d 为日期常量或变量，t 为日期/时间常量或变量。

表 3-12 常用日期和时间函数

函　数	功　能	实　例	结　果
Now	返回系统日期/时间	Now	2013-6-3 9:34:33
Day(d)	返回当前的日期	Day(Now)	3
WeekDay(d)	返回当前的星期	WeekDay(Now)	2(代表星期一)
Month(d)	返回当前的月份	Month(Now)	6
Year(d)	返回当前的年份	Year(Now)	2013
Hour(d)	返回当前小时	Hour(Now)	9
Minute(d)	返回当前分钟	Minute(Now)	34
Second(d)	返回当前秒	Second(Now)	33
Time	返回当前时间	Time	9:34:33

3.4.5 其他函数

1. Format 函数

格式输出函数可以使数值、日期或字符串按指定的格式输出，返回类型为字符型。函数的格式为：

```
Format(数值表达式,格式字符串)
```

该函数的功能是：按“格式字符串”指定的格式输出“数值表达式”的值。如果省略“格式字符串”，则 Format 函数的功能与 Str 函数基本相同，唯一的差别是，当把正数转换成字符串时，Str 函数在字符串前面留有一个空格，而 Format 函数则不留空格。

用 Format 函数可以使数值按“格式字符串”指定的格式输出，包括在输出字符串前加 $、字符串前或后补 0 及加千分位分隔逗点等。“格式字符串”是一个字符串常量或变量，它由专门的格式说明字符组成，如表 3-13 所示。当格式字符串为常量时，必须放在双引号中。

表 3-13 格式说明符号

格式字符	含　义	实　例	结　果
#	数字；不在前面或后面补 0	Format(125, "#####")	125
0	数字；在前面或后面补 0	Format(125, "00000")	00125
.	显示小数点	Format(125.25, "0000.000")	0125.250
		Format(125.25, "#####.###")	125.25
,	千位分隔逗点	Format(12512.25, "##,##.##")	12,512.25
%	百分比符号	Format(0.257, "00.0%")	25.7%
$	美元符号	Format(358.9, "$ ###0.00")	$358.90
+−	正、负号	Format(358.9, "+###0.00")	+358.90
		Format(−358.9, "−###0.00")	−−358.90
E+	指数符号	Format(358.9, "0.00E+00")	3.59E+02
E−		Format(358.9, "0.00E−00")	3.59E−02

2. Shell 函数

在 Visual Basic 中,除了可以调用内部函数外,还可以调用 Windows 下的应用程序,这一功能通过 Shell 函数实现。Shell 函数的调用格式如下:

```
Shell(命令字符串[,窗口类型])
```

其中,"命令字符串"是要执行的应用程序的文件名(包括路径)。它必须是可执行文件,其扩展名为.com、.exe、.bat 或.pif,其他文件不能用 Shell 函数执行。"窗口类型"是执行应用程序时的窗口的大小,有 6 个系统常量,如表 3-14 所示。

表 3-14　"窗口类型"系统常量

常　量	值	窗 口 类 型
vbHide	0	窗口被隐藏,焦点移到隐式窗口
vbNormalFocus	1	窗口具有焦点,并还原到原来的大小和位置
vbMinimizedFocus	2	窗口会以一个具有焦点的图标来显示
vbMaximizedFocus	3	窗口是一个具有焦点的最大化窗口
vbNormalNoFocus	4	窗口被还原到最近使用的大小和位置,而当前活动的窗口仍然保持活动
vbMinimizedNoFocus	5	窗口以一个图标来显示,而当前活动的窗口仍然保持活动

Shell 函数调用某个应用程序并成功地执行后,返回一个任务标识,它是执行程序的唯一标识。例如:

```
ProID = Shell("c:\Windows\notepad.exe",1)
```

该语句调用记事本 notepad.exe,使记事本程序启动后具有正常窗口,并成为当前窗口。同时把 ID 返回给 ProID。注意,在具体输入程序时,ID 不能省略。

习题

一、选择题

1. 以下可以作为 Visual Basic 变量名的是(　　)。

   A. A#A　　B. countA　　C. 3A　　D. ?A8

2. Visual Basic 的合法直接常量有(　　)。

   A. π　　B. %100　　C. True　　D. &H12ag

3. 下面说法不正确的是(　　)。

   A. 整型关键字为 Integer,类型符为 &

   B. 日期型关键字为 Date,无类型符

   C. 单精度型关键字为 Single,类型符为!

   D. 字符型关键字为 String,类型符为 $

4. 下列数据类型中,占用内存最小的是(　　)。

   A. Boolean　　B. Byte　　C. Integer　　D. Single

5. 下列日期型数据正确的是(　　)。

A. @January 10,2012@　　B. ＃January 10,2012＃

C. “January 10,2012”　　D. &January 10,2012&

6. 下列运算结果正确的是(　　)。

A. 10/3＝3　　B. 9 mod 4＝2　　C. "20" ＋ "12"　　D. 10\3＝3

7. Visual Basic 表达式 3\3＊3/3 mod 3 的值为(　　)。

A. －1　　B. 1　　C. －3　　D. 3

8. 设 a＝1,b＝5,c＝1,执行语句 Print a＝b＝c 后窗体上显示的是(　　)。

A. True　　B. False　　C. 10　　D. 出错信息

9. 下列程序运行结果是(　　)。

```
a = 35: b = - 25: i = Not a = b: Print i
```

A. －45　　B. True　　C. 0　　D. False

10. “x 是小于 10 的非负数”,用 Visual Basic 表达式表示正确的是(　　)。

A. 0＜＝x＜＝10　　B. 0＜＝x＜10

C. 0＜＝x and x＜10　　D. 0＜＝x or x＜10

11. 表达式(5 ＋ 6) ＞ 9 And ("23" ＜ "3")的值为(　　)。

A. True　　B. False　　C. 1　　D. 0

12. Int 函数用于取整,它返回不大于自变量的最大整数,但也可用于作四舍五入运算。要把 567.345 保留两位小数而将第三位四舍五入,应使用(　　)表达式。

A. Int(x＊100＋0.5)　　B. Int(x＊100)/100

C. Int(x＊100＋0.5)/100　　D. Int(x＊100)

13. 函数 Int(Rnd＊100)是在(　　)范围内的整数。

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

14. 设 a％＝20,b＄＝"30",则下列输出结果是"2030"的语句是(　　)。

A. Print str(a)　　B. Print "a" ＋ b　　C. Print a ＋ b　　D. Print a & b

15. 用于从字符串左边截取字符的函数是(　　)。

A. Ltrim( )　　B. Trim( )　　C. Left( )　　D. Instr( )

16. 下列语句的输出结果为(　　)。

```
Print Format $ (5689.36, "000,000.000")
```

A. 5,689.36　　B. 5,689.360　　C. 5,689.3　　D. 005,689.360

二、填空题

1. Visual Basic 在同一行可以书写多条语句,语句间用_____分隔;单行语句可分若干行书写,在本行后加续行符_____。

2. 长整型的关键字和类型符分别为_____、_____。

3. 双精度型的关键字和类型符分别为_____、_____。

4. 在 Visual Basic 中,定义全局变量的关键字为_____,且变量应在_____的变量声明区中定义,定义局部变量通常使用_____、_____或_____,其中,定义静态变量

的关键字为_____。

5. Int(－4.5),Int(4.5),Fix(－4.5),Fix(4.5)的值分别是_____、_____、_____、_____。

6. 表达式 15－8＞6 And 3＊4＜5 和 Right("abcde",3)的值分别是_____、_____。

7. String(4, "abc")和 Mid("abcde",2,3)的值分别是_____、_____。

8. 表示字符变量 s 是字母(不区分大小写)的逻辑表达式为_____。

9. UCase(Mid("china",2,3))的结果是_____。

10. 表示 x 是 5 的倍数或是 9 的倍数的逻辑表达式为_____。

三、程序题

1. 下列程序的输出结果是_____。

```
Private Sub Form_Click()
  x = 542
  x = Int(x / 100)
  y = x Mod 10
  y = x = y
  Print x; y
End Sub
```

2. 下列程序的输出结果是_____。

```
Private Sub Form_Click()
  x = Str(10.4) + "理工大学"
  y = Right(x, 2)
  Print Val(x) & y
End Sub
```

3. 编写程序,在文本框中输入一个三位数,单击窗体后,在窗体中打印输出该数的个位数、十位数和百位数。