

# 第 3 章 程序设计基础

任何一个由高级语言编写的应用程序所表达的内容均有两个重要的方面,一是数据,二是程序控制。其中数据是程序的处理对象,由所创建的数据类型决定其结构、存储方式及运算规则;程序控制则是程序流程控制,也是对数据进行处理算法。由此可将程序抽象地表示为:程序=算法+数据结构。

本章将介绍 Visual Basic 应用程序的基本内容、数据类型、基本的语句成分,如常量、变量、函数、表达式等。

## 3.1 数据类型

数据是客观事物属性反映的记录,也可以说客观事物属性的记录是用“数据”来表达的。数据形式通常有三种:数值型数据,即对客观事物进行定量记录的符号,如体重、年龄、价格的多少等;字符型数据,即对客观事物进行定性记录的符号,如姓名、单位、地址的标志等;特殊型数据,如声音、视频、图像等。在 Visual Basic 系统中,为了使数据处理更加方便,又将数据细化分为若干种类型。

### 3.1.1 标准数据类型

在 Visual Basic 系统中,常用标准数据类型分为数值型、字符型、货币型、日期型、布尔型、对象型、变体型、字节型。

#### 1. 数值型

数值型(numeric)数据是由数字(0~9)、小数点和正负号组成,是可以参加算术运算的数据。

数值型又分为不带小数点的整型和带小数点的浮点型,其中整型又分为整型(Integer)和长整型(Long);浮点型又分为单精度型(Single)和双精度型(Double)。

例如:12378、78 为整型,123.56、1.23e7、1.23e-2 为浮点型。

#### 2. 字符型

字符型(String)是由双引号(" ")括起来的一个符号序列。字符型数据又分定长字符型和不定长字符型,定长字符型可表示成(String \* 常数)。

例如:“中国人民”、“清华大学出版社”、“1304567821”、“100001”。

### 3. 货币型

货币型(Currency)数据是数值型数据的一种特殊形式,是为表示货币的多少及对货币进行计算而设置的,这种类型的数据小数点前最多有 15 位数,小数点后只保留 4 位数。

小数位超过 4 个字符的数据,系统会按四舍五入原则自动截取。

### 4. 日期型

日期型(Date)是由双井号(##)括起来用于表示时间的数据。它可以是单独日期的数据,也可以是单独时间的数据,也可以是日期和时间数据的组合。例如:

```
#2014    1    1 3:10:50 p#
#01/01/14#, #2014    1    1#, #01/01/2014#
#23:10:50#, #3:10:50 p#
```

### 5. 布尔型

布尔型(Boolean)数据是描述客观事物真假的的数据,用于表示逻辑判断结果,它只有真(True)和假(False)两个值。

布尔型可以转换成整型数据,即 True 为 1;False 为 0。

其他类型的数据可以转换成布尔型,即非 0 为 True;0 为 False。

### 6. 对象型

对象型(Object)是用于存储 OLE 对象的数据类型,OLE 对象可以是电子表格、文档、图片等。

### 7. 变体型

变体型(Variant)是一种可变的数据类型。在 3.2.2 节中将介绍它的用法。

### 8. 字节型

字节型(Byte)是以 1 个字节的无符号二进制数存储的数据类型。

表 3-1 列出了以上数据类型的相关信息。

表 3-1 常用标准数据类型

数据类型	类型符号	占用字节	取值范围
整型(Integer)	%	2	-32 768~32 767
长整型(Long)	&	4	-2 147 483 648~2 147 483 647
单精度型(Single)	!	4	负数: -3.402 823E38~-1.401 298E-45 正数: 1.401 298E-45~3.402 823E38
双精度型(Double)	#	8	负数: -1.797 693 134 862 32E308~-4.940 656 458 412 47E-324 正数: 4.940 656 458 412 47E-324~1.797 693 134 862 32E308
字符型(String)	\$	不定	0~65 400 个字符(定长字符型)
货币型(Currency)	@	8	-922 337 203 685 477.5808~922 337 203 685 477.5807

续表

数据类型	类型符号	占用字节	取值范围
日期型(Date)	无	8	100-01-01~9999-12-31
布尔型(Boolean)	无	2	True 或 False
对象型(Object)	无	4	任何引用的对象
变体型(Variant)	无	不定	由最终的数据类型而定
字节型(Byte)	无	1	0~255

### 3.1.2 自定义数据类型

Visual Basic 除了为用户提供了标准数据类型之外,还允许用户自定义数据类型,这种数据类型可包含一个或多个标准数据类型的数据元素。由于这种数据类型与其他高级语言的记录类型相同,因此,又把这种用户自定义数据类型称为“记录类型”。

定义自定义数据类型语句格式如下:

```

Type 数据类型名
    数据元素名 [(下标)] As 类型名
    数据元素名 [(下标)] As 类型名
    ...
End Type

```

为了处理数据的方便,常常需要把一些数据定义成自定义数据类型。

例如,为了表示学生的自然情况,如学号、姓名、性别、出生日期、身高等数据,可将数据定义成如下结构:

```

Type Student
    Number As String * 8
    Name As String * 3
    Sex As String * 1
    Day As Date
    Stature As Single
End Type

```

需要注意:用户自定义数据类型中的元素若是字符型,必须是定长字符型。

## 3.2 数据存储

在 Visual Basic 中,通常把常量、变量、数组、对象作为存储容器,从而实现对数据进行输入、输出和加工处理的操作。

### 3.2.1 常量

常量是在程序中可直接引用的实际值,其值在程序运行中不变。在 Visual Basic 中,有文字常量、符号常量和系统常量。

#### 1. 文字常量

文字常量实际上就是常数,数据类型的不同决定了常量的表现也不同。

例如:  $-123.56768 + 3.256767E3$  为数值型常量;"ABCDE", "中国人民解放军"为字符型常量; #04/12/14#, #2014/02/19 10:01:01# 为日期常量。

#### 2. 符号常量

符号常量是命名的数据项,其类型取决于<表达式>值的类型。

定义符号常量语句格式如下:

```
Const 常量名 [As 类型|类型符号]=<表达式>  
[,常量名 [As 类型|类型符号]=<表达式>]
```

例如:

```
Const I1%=14135    或    Const I1 As Integer=14135  
Const S1%=32,PI As Single=3.14159,S2%=S1+50
```

#### 3. 系统常量

系统常量是 Visual Basic 系统预先定义好的,用户可直接引用。例如 vbRed、vbOK 和 vbYes。

### 3.2.2 变量

变量(variable)在程序运行过程中其值可以改变。这里所讲的是一般意义上的简单变量(又称内存变量)。在 Visual Basic 程序中,每一个变量都必须有一个名称,用以标识该内存单元的存储位置,用户可以通过变量标识符使用内存单元存取数据;变量是内存中的临时单元,这就决定了它可以用来在程序的执行过程中保留中间结果与最后结果,或用来保留对数据进行某种分析处理后得到的结果;在给变量命名时,一定还要定义变量的类型,变量的类型决定了变量存取数据的类型,也决定了变量能参与哪些运算。

#### 1. 变量的声明

在程序中使用变量,就要给变量定义名称及类型,这就是对变量进行声明。

##### (1) 显式声明

声明局部变量语句格式如下:

```
Dim | Static 变量名 [AS 类型/类型符]  
[,变量名 [AS 类型/类型符]]
```

例如:

```
Dim I As integer
Dim Sum As single
Dim I%, Sum!
```

### (2) 隐式声明

未进行显式声明而通过赋值语句直接使用,或省略了[As 类型/类型符]短语的变量,其类型为变体(Variant)类型。

### (3) 强制声明

在 Visual Basic 程序的开始处,若出现“系统环境可设置”,或写入下面语句:

```
Option Explicit
```

程序中的所有变量必须进行显式说明。

对于初学者来说,建议写入 Option Explicit 语句,约束你养成一个良好的语言习惯,避免一些不必要的麻烦,并为学习其他语言提供便利。

## 2. 变量的作用域

变量的作用域就是变量在程序中的有效范围。

在 Visual Basic 程序中,能否正确使用变量,搞清变量的作用域是非常重要的。一旦变量的作用域被确定,使用时就要特别注意它的作用范围。当程序运行时,各对象间的数据传递就是依靠变量来完成的,变量的作用范围定义不当,对象间的数据传递就将导致失败。变量的作用域是一个不可忽视的问题,特别是基于面向对象程序设计理念进行应用系统开发时尤为重要。

在 Visual Basic 程序中,通常将变量的作用域分为局部级变量,窗体、模块级变量和全局级变量三类。

### (1) 局部变量

在事件过程、通用过程中用 Dim 和 Static 语句声明的变量叫局部变量。其作用域只在其所声明的事件过程、通用过程内部。

例如:

```
Private Sub Cmddisplay_Click()
Dim i As Double
Static k As Integer
.....
.....
End Sub
```

上面的过程定义 i、k 两个局部变量,它们的有效范围只在 Cmddisplay\_Click() 事件过程内。

### (2) 窗体变量和模块变量

Visual Basic 应用程序是由窗体模块、标准模块和类模块三种模块组成的,本书中会

涉及窗体模块和标准模块两种模块。其中：

- 窗体模块包括事件过程、通用过程和声明部分。
- 标准模块包括通用过程和声明部分。

一个窗体模块可以包含若干个事件过程和通用过程,同样,标准模块可以包含若干个通用过程。若使变量在一个窗体模块的多个事件过程和通用过程,或在标准模块多个通用过程中有效,可以将其声明为窗体变量或模块变量。

在窗体模块的声明部分用 Dim 和 Private 语句声明的变量叫窗体变量。

在标准模块的声明部分用 Dim 和 Private 语句声明的变量叫模块变量。

例如：

```
Dim i As Double
Private k As Integer
```

### (3) 全局变量

全局变量也叫全程变量,它是可以在整个程序的任何模块、任何过程中使用的变量。

在窗体模块的声明部分和标准模块的声明部分用 Public 语句声明的变量叫全局变量。

例如：

```
Public i As Double
Public k As Integer
```

## 3. 重名变量

在一个 Visual Basic 应用程序中,不能有重名的全局变量;但是局部变量可以和全局变量重名。在某一个事件过程、通用过程中,定义的局部变量若与全局变量重名,局部变量有效,全局变量被屏蔽,而当事件过程、通用过程结束后,这些局部变量从内存中释放(由 Static 定义的局部变量例外),而全局变量则在程序运行完毕时将从内存中释放。

### 4. Dim 与 Static 的区别

Dim 定义的局部变量称为自动变量。

Static 定义的局部变量称为静态变量。

用 Dim 定义的局部变量,当执行事件过程、通用过程时,变量有效,当事件过程、通用过程结束后,变量从内存中释放。

用 Static 定义的局部变量,当执行事件过程、通用过程时,变量有效,当事件过程、通用过程结束后,变量不从内存中释放,仍将保留原有的值,一旦重复使用事件过程、通用过程,其值可再次使用。

例如：

```
Private Sub Cmdtest_Click()
Dim sum As Integer
sum=sum+1
Print sum
```

```
End Sub
```

每次调用上面的过程, sum 的值都是 1。

```
Private Sub Cmdtest_Click()
Static sum As Integer
sum=sum+1
Print sum
End Sub
```

每次调用上面的过程, sum 的值都要加 1, sum 是一个自然数据的累加器。

### 3.3 内部函数

内部函数是 Visual Basic 系统为用户提供的标准过程。使用这些内部函数, 可以使某些特定的操作更加简便。在使用内部函数时, 要了解函数的功能、函数的书写格式、函数参数及函数结果的类型及表现形式。

根据每个内部函数的功能, 可将内部函数大致分为如下几类: 数学函数、字符函数、转换函数、日期函数、测试函数、颜色函数、路径函数等。本节仅介绍常用的内部函数的格式与功能, 其他内部函数的格式与功能见附录 E。

#### 1. 数学函数

常用的数学函数见表 3-2。

表 3-2 常用数学函数的功能及实例

函 数	功 能	例 子	函 数 值
Abs(N)	绝对值	ABS(-3)	3
Cos(N)	余弦	Cos(45 * 3.14/180)	0.707
Exp(N)	e 指数	Exp(2)	7.389
Int(N)	返回参数的整数部分	Int(1234.5678)	1234
Log(N)	自然对数	Log(2.732)	1
Rnd(N)	返回一个包含随机数	Rnd	0~1 之间的数
Sgn(N)	返回一个正负号或 0	Sgn(5)	1
Sin(N)	正弦	Sin(45 * 3.14/180)	0.7068
Sqr(N)	平方根	Sqr(25)	5
Tan(N)	正切	Tan(45 * 3.14/180)	0.9992

**注意:** N 可以是数值型常量、数值型变量、数学函数和算术表达式, 而且数学函数的返回值仍是数值型常量。

下面举例说明在立即窗口可进行的操作。

**例 3-1** 输出  $e^5$  的值。

操作命令如下：

```
?Exp (5) ✓
```

结果是：

```
148.413159102577
```

**例 3-2** 已知  $x=12, y=8$ , 计算  $\sqrt{x^2+y^2}$  的值。

操作命令如下：

```
x=12 ✓
```

```
y=8 ✓
```

```
?sqr(x*x+y*y) ✓
```

结果是：

```
14.422205101856
```

**例 3-3** 已知  $x=56.8, y=78.9$ , 计算  $x>y$  的值, 只取整数部分。

操作命令如下：

```
x=56.8 ✓
```

```
y=78.9 ✓
```

```
?Int(x*y) ✓
```

结果是：

```
4481
```

## 2. 字符函数

常用的字符函数见表 3-3。

表 3-3 常用字符函数的功能及实例

函 数	功 能	例 子	函 数值
Instr(C1,C2)	在 C1 中查找 C2 的位置	Instr("ABCDE","DE")	4
Lcase\$(C)	将 C 中的字母转换为小写	Lcase\$("ABcdE")	"abcde"
Left\$(C,N)	取 C 左边 N 个字符	Left\$("ABCDE",3)	"ABC"
Len(C)	测试 C 的长度	Len("ABCDE")	5
LTrim\$(C)	删除左边的空格	RTrim\$(" AAA "+" BBB ")	"AA BB"
Mid\$(C,M,N)	从第 N 个字符起,取 C 中 M 个字符	Mid\$("ABCDE",2,2) Mid\$("ABCDE",3,1)	"BC" "C"
Right\$(C,N)	取 C 右边 N 个字符	Right\$("ABCDE",3)	"CDE"
RTrim\$(C)	删除 C 右边的空格	RTrim\$("AA"+"BB")	"AA BB"



续表

函 数	功 能	例 子	函 数值
Space\$(N)	产生 N 个数的空格字符	Space(5)	" "
Trim\$(C)	删除 C 首尾两端的空格	Trim\$("AA"+"BB")	"AA BB"
Ucase\$(C)	将 C 中的字母转换为大写	Ucase\$("abcde")	"ABCDE"

**注意：**N 可以是数值型常量、数值型变量、数学函数和算术表达式，C 可以是字符型常量、字符型变量、字符函数和字符表达式，而且字符函数中，函数名后跟(\$)的返回值仍是字符型常量。

在立即窗口可作如例 3-4、例 3-5 和例 3-6 所示操作。

**例 3-4** 已知 X="清华大学 出版社 "，去掉 X 的尾部空格。

操作命令如下：

```
?Trim$("清华大学 出版社 ") ✓
```

结果是：

清华大学 出版社

**例 3-5** 已知 X="清华大学出版社"，测试 X 的长度。

操作命令如下：

```
?Len("清华大学出版社") ✓
```

结果是：

7

**例 3-6** 已知 X="清华大学出版社"，从 X 中截取"出版社"。

操作命令如下：

```
?mid$("清华大学出版社",5,3) ✓
```

结果是：

出版社

### 3. 转换函数

常用的转换函数见表 3-4。

表 3-4 常用转换函数的功能及实例

函 数	功 能	例 子	函 数值
Asc(C)	返回 C 的第一个字符的 ASCII 码	Asc("A")	65
Chr(N)	返回 ASCII 码 N 对应的字符	Chr(97)	"a"
Str(N)	将 N 转换成 C 类型	Str(100010)	"100010"
Val(C)	将 C 转换成 N 类型	Val("123.567")	123.567

#### 4. 日期函数

常用的日期函数见表 3-5。

表 3-5 常用日期函数的功能

函 数	功 能
Date	返回当前系统日期(含年月日)
DateAdd(C,N,date)	返回当前日期增加 N 个增量的日期
DateDiff(C,date1,date2)	返回 date1,date2 间隔的时间
Day(Date)	返回当前日期
Hour(Time)	返回当前小时
Minute(Time)	返回当前分钟
Month(Date)	返回当前月份
Now	返回当前日期和时间(含年月日、时分秒)
Second(Time)	返回当前秒
Time	返回当前时间(含时分秒)
Weekday	返回当前星期
Year(Date)	返回当前年份

**注意：**N 可以是数值型常量、数值型变量、数值型函数和算术表达式,C 是专门的字符串(YYYY—年、Q—季、M—月、WW—星期、D—日、H—时、N—分、S—秒)。

在立即窗口可作如例 3-7~例 3-12 所示操作。

**例 3-7** 若系统时间为 2014-2-25 13:35:08,输出当前日期,当前日期时间的值。

操作命令如下:

```
? Date, Now ✓
```

结果是:

```
2014-2-25      2014-2-25 13:35:08
```

**例 3-8** 若系统时间为 2014-2-25 13:35:08,输出当前日期及年、月、日的值。

操作命令如下:

```
? Date, Year(Date), Month(Date), Day(Date) ✓
```

结果是:

```
2014-2-25      2004          2          25
```

**例 3-9** 若系统时间为 2014-2-25 14:03:40,输出当前时间及时、分、秒的值。

操作命令如下:

```
? Time, Hour(Time), Minute(Time), Second(Time) ✓
```

结果是:

14:03:40          14                  3                  40

**例 3-10** 若系统时间为 2014-2-25 14:03:40,输出当前时间一月后、一周后、一天后的日期。

操作命令如下:

```
?;DateAdd("M",1,date);DateAdd("ww",1,date);DateAdd("D",1,date) ✓
```

结果是:

2014-3-22 2014-3-1 2014-2-23

**例 3-11** 输出 2014-2-25 与 2014-7-30 相隔的天数。

操作命令如下:

```
?DateDiff("D",#2014-2-25#,#2014-7-30#) ✓
```

结果是:

155

**例 3-12** 输出当前时间(2014-2-22 14:54:40)与 2018-1-1 相隔的天数、小时数。

操作命令如下:

```
?DateDiff("D",Now,#2018-1-1#),DateDiff("H",Now,#2018-1-1#) ✓
```

结果是:

1409                  33801

## 5. 测试函数

常用的测试函数见表 3-6。

表 3-6 常用测试函数的功能

函 数	功 能
IsArray(E)	测试 E 是否为数组
IsDate(E)	测试 E 是否为日期类型
IsNumeric(E)	测试 E 是否为数值类型
IsNull(E)	测试 E 是否包含有效数据
IsError(E)	测试 E 是否为一个程序错误数据
Eof()	测试文件指针是否到了文件尾

**注意:** E 为各种类型的表达式,测试函数的结果为布尔型数据。

## 6. 其他函数

### (1) 颜色函数

#### ① QBColor 函数

QBColor 函数格式如下：

##### **QBColor (N)**

功能：通过 N(颜色代码)的值产生一种颜色。

颜色代码与颜色对应关系见表 3-7。

表 3-7 颜色代码与颜色对应关系

颜色代码	颜色	颜色代码	颜色
0	黑	8	灰
1	蓝	9	亮蓝
2	绿	10	亮绿
3	青	11	亮青
4	红	12	亮红
5	洋红	13	亮洋红
6	黄	14	亮黄
7	白	15	亮白

#### ② RGB 函数

RGB 函数格式如下：

##### **RGB (N1, N2, N3)**

功能：通过 N1, N2, N3(红、绿、蓝)三种基本颜色代码产生一种颜色,其中 N1、N2 和 N3 的取值范围为 0~255 之间的整数。

例如：

RGB(255, 0, 0)产生的颜色是“红”色；

RGB(0, 0, 255)产生的颜色是“蓝”色；

RGB(100, 100, 100)产生的颜色是“深灰”色。

### (2) Format 函数

Format 函数格式如下：

##### **Format (E, fmt)**

功能：是根据 fmt(格式字符串)的指定格式输出 E(表达式)的值,表达式可以是数值、日期或字符串型表达式。

它一般用于 Print 方法中,其形式为 print Format \$(表达式[, "格式字符串"])。

常用的格式字符串见表 3-8。

表 3-8 常用格式字符串的功能及实例

格式字符	功 能	例 子	效 果
0	显示一个数字,数据位数不足自动补0	Format(123.45,"0000.00")	0123.45
#	显示一个数字	Format(123.45,"###.##")	123.45
%	数字乘以100,并在数字尾部加上%	Format(0.45,"#%")	45%
.	固定小数点的位置	Format(1.4567,"#.###")	1.46
,	千位分隔符	Format(1234567,"#,###,###")	1,234,567
-,+,\$,()	原样输出	Format(12345,"\$#,###,###")	\$12,345
<	将字符串的字符转换成小写	Format("Hello","<")	"hello"
>	将字符串的字符转换成大写	Format("hello",">")	"HELLO"
@	显示一串字符,数据位数不足前面补空格	Format("Hello","@@@@@")	" Hello"
&	显示一串字符	Format("Hello","&&&&&&&")	"Hello"
M/D/YY	月/日/年格式	Format(#2004-03-05#,"M/D/YY")	3-5-04
YYYY/M/D	年/月/日格式	Format(#2004-03-05#,"YYYY/M/D")	2004-3-5
H:MM:SS	时/分/秒格式	Format(Time,"H:MM:SS")	18:42:47

### 3.4 表 达 式

表达式是由变量、常量、函数、运算符和圆括号组成的式子。在 Visual Basic 中,根据运算符的不同,将表达式分为算术表达式、字符表达式、关系表达式、逻辑表达式。

#### 1. 算术表达式

算术表达式是由算术运算符和数值型常量、数值型变量、返回数值型数据的函数组成,其运算结果仍是数值型常数。

算术运算符及表达式的实例见表表 3-9。

表 3-9 算术运算符及实例

运 算 符	功 能	例 子	表 达 式 值
^	幂	5^2	25
取负	—	-5^2	-25
*,/	乘、除	36*4/9	16
\	整除	25\2	12
Mod	模运算(取余)	97 Mod 12	1
+, -	加,减	3+8-6	5

在进行算术表达式计算时,要遵循以下优先顺序:先括号,在同一括号内先乘方(^),再乘除(\*、/),再模运算(Mod),后加减(+、-)。

## 2. 字符表达式

字符表达式由字符运算符和字符型常量、字符型变量、返回字符型数据的函数组成,其结果是字符常数或逻辑型常数。

字符运算符及表达式的实例见见表 3-10。

表 3-10 字符运算符及表达式实例

运算符	功 能	例 子	表达式值
+	连接两个字符型数据	"计算机"+"软件"	"计算机软件"
&.	连接两个字符型数据	"计算机"&."软件"	"计算机软件"

+ 和 &. 均可完成字符串连接运算。不同的是前者既可以做加法运算又可以做字符串连接运算;后者则只能做字符串连接运算。

**例 3-13** 已知  $X=12345$ ,  $Y="12345"$ , 计算  $X+Y$ 、 $X \&. Y$  的值。

操作命令如下:

```
X=12345 ✓
Y="12345" ✓
? X+Y, X & Y ✓
```

结果是:

```
24690      1234512345
```

为了避免混淆,进行字符串连接运算一般都使用 &. 作运算符,而且要注意 &. 的前后要加空格。

## 3. 关系表达式

关系表达式可由关系运算符和字符表达式、算术表达式组成,其运算结果为逻辑型常量。关系运算是运算符两边同类型元素的比较,关系成立结果为真(True);反之结果为假(False)。

关系运算符及表达式实例见表 3-11。

表 3-11 关系运算符及表达式

运 算 符	功 能	例 子	表 达 式 值
<	小于	$3 * 5 < 20$	True
>	大于	$3 > 1$	True
=	等于	$3 * 6 = 20$	False
<>, ><	不等于	$4 < > -5, 4 > < -5$	True

续表

运算符	功能	例子	表达式值
<=	小于或等于	$3 * 2 <= 6$	True
>=	大于或等于	$6 + 8 >= 15$	False
Like	字符串是否匹配	"ABC" Like "ABC"	True

Like 的主要功能虽然是测试字符串是否完全匹配,但是在字符串中可以引用通配符,使其灵活性大大增强。

由 Like 组成的关系表达式中,可以使用的通配符如下:

- (1) \* 代表一个或多个字符;
- (2) ? 代表一个字符;
- (3) # 代表一个数字。

**例 3-14** 已知 X="清华大学计算机学院",计算(X Like "清华大学\*")的值。

操作命令如下:

```
X="清华大学计算机学院" ✓
?X Like "清华大学*" ✓
```

结果是:

True

**例 3-15** 已知:姓名="王立品",计算(姓名 Like "王??")的值。

操作命令如下:

```
姓名="王立品" ✓
?姓名 Like "王??" ✓
```

结果是:

True

#### 4. 逻辑表达式

逻辑表达式可由逻辑运算符和逻辑型常量、逻辑型变量、返回逻辑型数据的函数和关系表达式组成,其运算结果仍是逻辑型常量。

逻辑运算符及表达式实例见表 3-12。

表 3-12 逻辑运算符及表达式实例

运算符	功能	例子	表达式值
NOT	非	NOT $3 + 5 > 6$	False
AND	与	$3 + 5 > 6$ AND $4 * 5 = 20$	True
OR	或	$6 * 8 <= 45$ OR $4 < 6$	True

续表

运算符	功能	例子	表达式值
Xor	异或	$3 > 2$ Xor $3 < 4$	False
Eqv	等价	$7 > 6$ Eqv $7 < 8$	True
Imp	蕴含	$7 > 6$ Imp $7 > 8$	False

逻辑表达式在运算过程中所遵循的运算规则见表 3-13。

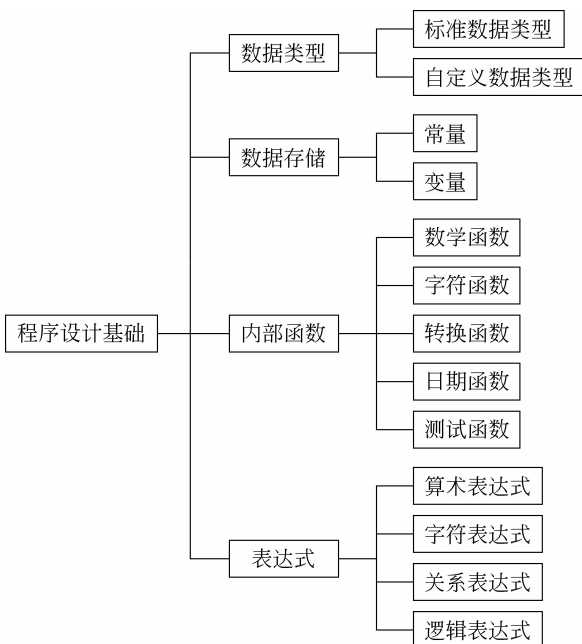
表 3-13 逻辑表达式运算规则

A	B	Not A	A and B	A or B	A Xor B	A Eqv B	A Imp B
True	True	False	True	True	False	True	True
True	False	False	False	True	True	False	False
False	True	True	False	True	True	False	True
False	False	True	False	False	False	True	True

进行逻辑表达式计算值时要遵循以下优先顺序：括号、Not、And、Or、Xor、Eqv、Imp。

以上各种类型的表达式的运算规则是：在同一个表达式中，如果只有一种类型的运算，则按各自的优先度进行运算；如果有两种或两种以上类型的运算，则按照函数运算、算术运算、字符运算、关系运算、逻辑运算的顺序进行运算。

### 本章的知识点结构





## 习 题

1. 回答下列问题:
  - (1) 常用的标准数据类型有哪些?
  - (2) 变量类型有哪些,类型符是什么?
  - (3) 在 Visual Basic 中,有几种类型表达式?
  - (4) 变量定义语句有几个,功能有什么不同?
  - (5) Option Explicit 语句的作用是什么?
  - (6) 变量的作用域分为几类,有什么不同?
  - (7) 举例说明在不同的作用域中,重名变量如何使用?
  - (8) Dim 与 Static 有什么区别?
  - (9) 表达式是由什么构成的?
  - (10) 计算逻辑表达式值时要遵循什么优先顺序?
2. 下列哪些是合法常量,属于什么类型?
  - (1) 12.678 (2) #2008 12 01# (3) 123% (4) "123" (5) False
  - (6) "2008/12/01" (7) 01/02/2014 (8) "AB"+"CD"
3. 指出下列变量的类型。
  - (1) Dim a1 As Single (2) Dim a1
  - (3) Dim a1 As Integer (4) Dim a1[KG- \* 3]\$
  - (5) Dim a1, a2 As Integer (6) Dim a1 #
4. 计算下列函数值。
  - (1) Sqr(4+3 \* 7)
  - (2) Int(123.456)
  - (3) Abs(-12345)
  - (4) Mid[KG- \* 3]\$("abcdABCD", 5, 4)
  - (5) Len("清华大学出版社 ABCD")
  - (6) Asc("M")
  - (7) Asc(Chr(100))
  - (8) DateDiff("D", #2014-1-1, #2014-10-1#)
  - (9) IsDate(#11/20/2014#)
  - (10) IsNumeric("ABC")
5. 已知 na=100,nb=56,sa \$="Visual Basic",da=#3/15/2014 8:15:03 PM#,  
sb \$="程序设计",la=True,计算下列表达式的值。
  - (1) (na+nb)/Sqr(na)
  - (2) Mid(sa+sb, 8, 7)
  - (3) Right(sa \$, 5)+Space(5)+Left(sb \$, 2)
  - (4) sb & Str(na) & " 分"

- (5) Year(da) &. Month(da) &. Day(da)
  - (6) Hour(da) &. ":" &. Minute(da) &. ":" &. Second(da)
  - (7) da+20
  - (8) na+nb>200 And Sqr(na)>10 Or la
  - (9) Len(sa)=12 And Not la And na=100
  - (10) Hour(da)>Int(na/10) And na>nb
6. 将下列代数式写成 Visual Basic 的算术表达式。

(1)  $\sin^2(\sqrt{20+a(\sqrt[4]{ab+1})})$

(2)  $15abc + (abc \sqrt[3]{a+b+c})$

(3)  $|\sqrt{x^2-y^2}| \frac{\sin 45^\circ}{\frac{x}{y}}$

(4)  $\frac{x+y}{xy-\sqrt{1-a^2}}$

(5)  $9e^{\sqrt[4]{a^5}} \ln a^2$

# 第 4 章 窗体及相关操作

窗体模块、标准模块以及窗体容纳的基本控件是 Visual Basic 应用程序必不可少的基本元素,本章将介绍标准模块、窗体模块及窗体基本控件的设计。

## 4.1 输入输出操作

一个完整的 Visual Basic 应用程序,通常可分为数据输入、数据处理、数据输出三部分。本节介绍几个常用的输入输出操作的语句及方法。有关利用控件进行输入输出操作的方法,将在介绍控件时再专门介绍。

### 4.1.1 赋值语句

赋值语句的格式如下:

<变量>=<表达式>

[<对象>].属性=<表达式>

功能:先计算的<表达式>,再将其值赋给变量或指定对象的属性。

例如:

```
I=I+1
```

```
Lbl1.Caption="快乐、轻松学 Visual Basic"
```

注意事项:

- (1) 只能给一个变量或对象的一个属性赋值;
- (2) <表达式>可以是变量、常量、函数和表达式;
- (3) =与代数式中的等号不同,它是赋值号,在代数式中 $I=I+1$ 是错误的,在 Visual Basic 中其意义是将 $I+1$ 的值赋给 $I$ ;
- (4) 变量或对象的属性引用,不能是常量、符号常量和表达式。

例如:

```
10=X+Y
```

' 赋值号左边是常量,该语句是错误的。

```
X+Y=20
```

' 赋值号左边是表达式 该语句是错误的。

## 4.1.2 Print 方法

### 1. Print 方法简介

Print 方法的格式如下：

```
[<对象名>.]Print[<表达式表>][,|;]
```

功能：在指定的对象上输出<表达式表>中各元素的值。

注意事项：

(1) [<对象名>.] 可以是窗体名(Form)、图片框名(PictureBox)，在立即窗口可省略[<对象名>.]或写 Debug，在窗体中，若省略对象，则表示在当前窗体上输出。

例如：

```
Debug.Print 3+5  
Print "快乐、轻松学 Visual Basic"  
Frm1.Print Sqr(3*3)  
Pic1.Print Exp(5)
```

(2) <表达式表>是一个或多个表达式，省略此项则输出一个空行。

(3) <表达式表>中多个表达式可用逗号(,)或分号(;)隔开。其中“;”是紧凑格式，“,”是标准格式；最后一个表达式后有“,”或“;”则不换行，没有“,”或“;”则换行。

### 2. 与 Print 方法相关的函数

(1) Tab()函数

Tab()函数格式如下：

**Tab(N)**

功能：把光标移到由 N 确定的位置。

例如：

```
Pic1.Print Tab(10); Exp(5)
```

(2) Spc()函数

Spc()函数格式如下：

**Spc(N)**

功能：光标移到 N 个空格之后的位置。

例如：

```
Pic1.Print 3+5; Spc(3); Exp(5)
```

### 3. Cls 方法

Cls 方法的格式如下：

```
[<对象名>.]Cls
```