

第3章 VB 语言基础

Visual Basic 是在 BASIC 语言的基础上发展起来的，它兼容原来版本的数据类型和语法，对其中的某些语句和函数的功能作了修改或扩展，并根据语言的可视性要求增加了一些新的功能。本章将介绍构成 Visual Basic 应用程序的元素，包括数据类型、常量、变量、运算符、表达式、函数、VB 的基本语句等基本内容。

3.1 数据类型

任务 1：正确认识 and 区分给定数据的数据类型。

【任务的提出】

数据是程序的必要组成部分，也是程序处理的对象。数值、文字、图像等都是数据，它们具有不同的类型，不同的数据类型表示它的取值范围及能进行的运算不同。因此正确使用数据类型是学习 VB 的前提。本任务包括以下内容：

- (1) VB 的基本数据类型。
- (2) 用户自定义类型。

3.1.1 编码基础

VB 和任何程序设计语言一样，编写代码也都有一定的书写规则，其主要规定如下：

1. VB 代码不区分字母的大小写

为了提高程序的可读性，VB 对用户程序代码进行自动转换：

- (1) 对于 VB 中的关键字，把构成关键字的每个单词的首字母大写，其余字母均小写。
- (2) 对于用户自定义的变量、过程名，VB 以第一次定义的为准，以后输入的自动向首次定义的格式转换。

2. 语句书写自由

- (1) 在同一行上可以书写多条语句，语句间用冒号“:”分隔。
- (2) 单行语句可以分若干行书写，在本行后加入续行符（下划线“_”）。
- (3) 一行允许多达 255 个字符。

3. 注释有利于程序的维护和调试

(1) 注释以 Rem 开头，但一般用撇号“'”引导注释内容，用撇号引导的注释可以直接出现在语句的后面。

(2) 也可以使用“编辑”工具栏的“设置注释块”、“解除注释块”按钮，使选中的若干行语句（或文字）增加注释或取消注释十分方便。

3.1.2 基本数据类型

所谓基本数据类型是指 VB 内置的、最基本的数据类型，由它们可构造出其他数据类型。表 3-1 列出了 Visual Basic 所允许使用的基本数据类型。

表 3-1 Visual Basic 基本数据类型

数据类型	占内存大小	数据有效范围	类型声明符
Integer（整型）	2 字节	-32,768~32,767	%
Long（长整型）	4 字节	-2,147,483,648~2,147,483,647	&
Single（单精度实型）	4 字节	负数: -3.402823E38~-1.401298E-45 正数: 1.401298E-45~3.402823E38	!
Double（双精度实型）	8 字节	负数: -1.79769313486232D308~ -4.94065645841247D-324 正数: 4.94065645841247D-324~ 1.79769313486232D308	#
Currency（货币型）	8 字节	-922,337,203,685,477.5808~ 922,337,203,685,477.5807	@
String（字符串型）	字符串长度	0~65,535 个字符	\$
Byte（字节型）	1 字节	0~255	无
Boolean（布尔型）	2 字节	True 或 False	无
Date（日期型）	8 字节	100 年 1 月 1 日~9999 年 12 月 31 日	无
Object（对象型）	4 字节	任何对象引用	无
Variant（通用型）	可以表示任何数据类型，也称变体型		

对于任何一种数据类型，应注意它的三个方面：

(1) 数据类型的英文名称。因为以后在声明一个变量时，使用的就是它的英文，如：Dim x As Single。

(2) 对应数据占用内存的大小。这决定了该数据类型的范围。

例如：单精度实型和双精度实型数据所占的内存大小不同，因此数据的有效范围也不一样。

(3) 类型说明符。它的作用有助于识别数据类型。例如：

10%（整型） 25&（长整型） 12.5!（单精度型） 3.14#（双精度型）

"VB"\$（字符串型） 323@（货币型） #07/28/2007#（日期型）

VB 提供了一个函数 VarType()，可以测定数据类型。可以在窗体的单击事件中编写代码，如图 3-1 所示。

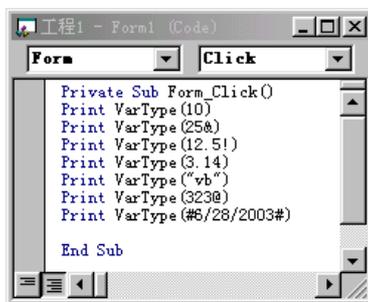


图 3-1 VarType 使用示例

运行后，在窗体上显示 2、3、4、5、8、6、7。这是 VarType()函数的返回值。其函数的

返回值的意义见表 3-2。

表 3-2 VarType()函数的返回值及相应的数据类型

VarType()函数返回值	代表的数据类型	VarType()函数返回值	代表的数据类型
0	空	5	双精度
1	NULL	6	货币型
2	整型	7	日期型
3	长整型	8	字符串型
4	单精度		

3.1.3 用户自定义类型

有时会感到以上基本数据类型不能满足实际编程需求。可以将不同类型的数据组合成一个有机的整体,以便于使用。这样一个整体是由若干不同类型的、互相有联系的数据项组成的。它类似 C 语言中的结构体、Pascal 语言中的记录类型,称之为“记录”。VB 提供了 Type 语句来定义这样的数据类型,其格式如下:

```
Type 自定义类型名
    数据类型元素名 1 As 类型名 1
    数据类型元素名 2 As 类型名 2
    .....
```

End Type

例如,定义一个名为 Student 的类型,其中包括有学号、姓名、成绩等信息:

```
Type student
    no as integer
    name as string
    grade as single
End Type
```

在定义了 student 类型后,就可以定义 student 类型的变量了,例如:

```
Dim stu1 As student
```

此语句定义了 student 类型的变量 stu1,它包括有三个成员:stu1.no、stu1.name、stu1.grade。

【新的任务】

通过本任务的学习,了解了 VB 有哪些基本的数据类型。从数据的存储形式来分,数据又可分为常量和变量。无论是常量还是变量都有某一固定的数据类型。

现在新的任务是:学习并掌握数据的表现形式——常量和变量。

3.2 常量与变量

任务 2: 熟悉并掌握常量的表现形式。

【任务的提出】

在 VB 程序设计中,经常会发现代码中包含一些常数值,也就是在程序的运行过程中其值

是不会改变的量。本任务包括以下内容:

- (1) 直接常量。
- (2) 符号常量。

3.2.1 直接常量

直接常量是指以数据实际的值的形式给出的数据,如 12.3 是一个数值常量,“ABC”是一个字符串常量,#07/20/2007#是一个日期常量。VB 的直接常量可根据数据类型的不同,分为数值型、字符串型、日期/时间型和布尔型。

1. 数值型常量

指由数字、小数点、正负号和特定符号构成的量。有四种表示方式:即整数型、长整型、货币型和浮点型,它们各自的取值范围见表 3-1。

整数型:有三种形式,即十进制、十六进制和八进制。十进制整数如 234、-4 等;十六进制整数由一个或几个十六进制数字(0~9 及 A~F 或 a~f)组成,前面冠以&H(或&h),如&H36、&H21F 等;八进制由一个或几个八进制数字(0~7)组成,前面冠以&O(或&o),如&O35、&O147 等。

长整型:也有三种形式,大体同整数型,只不过十六进制和八进制的数后要以&结尾,如&H36&、&H21F&、&O35&、&O147&等。

货币型:是为计算货币而设置的定点数,规定最多保留小数点右边 4 位和小数点左边 15 位。表示形式在数字后加@符号,如 123.45@。

浮点型:也称实数,分为单精度浮点数和双精度浮点数,后者比前者的精度更高。通常以指数形式(科学记数法)来表示,由尾数、指数符号和指数三部分组成,其中尾数本身也是一个浮点数,指数符号为 E(单精度)或 D(双精度),指数为整数。例如在 2.35E-8(2.35×10^{-8})和 43D6(43×10^{-6})中,2.35 和 43 是尾数,-8 和 6 是指数。

2. 字符串常量

字符串常量是用一对双引号括起来的若干个合法字符,汉字等非英文字符也是构成字符串常量的合法字符。如:“china”、“10/01/1997”、“x=”、“水利学院”等。应注意的是:字符串常量是指双括号中的字符串,而不包括双引号本身,双引号只是字符串常量的起止界限。字符串常量的长度不能超过 65,535 个字符。例如"4567445"、"ABC#\$&"等。

注意: ""表示空字符串,即不包含任何字符的字符串,而" "表示含有一个空格的字符串。

3. 日期/时间型

用来表示某一具体的时间或日期,它的分界符是“#”号,例如#7/14/2003#、#10:34:08#、#January 1,2007#等。日期和时间的显示会根据计算机中的短日期格式和时间格式来显示。

4. 布尔型

通常也称为逻辑型,其取值只有两个,分别为 True 和 False。

VB 在判断常量类型时会存在多义性,如 3.15 可能是单精度,也可能是双精度或货币型。默认情况下,VB 选择占用内存最小的表示法,即 3.15 通常被作为单精度处理。为了明确表明某一类型,可以在常数的后面加上类型说明符。

3.2.2 符号常量

在程序中往往多次用到同一个常量,为避免重复书写该常量,VB 提供了符号常量。也就

是说，用一个符号来代替一个常量。使用符号常量可以增强程序的可读性和可维护性。

符号常量必须先定义后使用，其定义格式如下：

[Private | Public] Const 符号常量名 [As 数据类型] =表达式

其中：中括号[]内的内容可以省略，Private | Public 表示常量的作用域，参见 3.2.5 节的内容。符号常量名的命名规则同变量的命名规则。表达式是由运算量和运算符构成的式子。

例如：

```
Const MAX as integer=999
Const MYSTRING="HELP"
```



符号常量有点像变量，但不能像对变量那样修改符号常量，也不能对符号常量赋以新值。

3.2.3 内部常量

内部常量也称系统常数，由应用程序和控件提供，通常都带有表示对象库的前缀，如 VB 和 VBA 等对象库的常量都以“vb”开头。例如：

```
Form1.BackColor=vbBlue          'vbBlue 是颜色内部常量，表示蓝色
Form1.WindowState=vbMaximized  '作用是将窗口最大化，其中的 vbMaximized 就是系统常量。
```

系统预定义的常量可通过对象浏览器查看。在对象浏览器的 Visual Basic、Visual Basic For Application 等对象库中列举了 Visual Basic 的常量。

【新的任务】

通过本任务的学习，了解了 VB 常量的表现形式。在程序设计中，运算结果往往是变化的，这样就不能以固定值的形式出现，而要以一个可变的量出现。变量也是内存中的一个临时存储单元，它可以存储该类型不同的值，以变量名的形式体现。

现在新的任务是：

- (1) 变量的声明。
- (2) 变量的作用域。

3.2.4 变量的声明

声明（或定义）一个变量，就是给变量分配一个内存空间。使用变量前，一般必须先声明变量名和其类型，以决定系统为它分配的存储单元。在 VB 中，可以用名字表示内存位置，因此一个有名称的内存位置称为变量。存储数据内容的数据类型则决定了该数据的存储方式。该变量的名字应遵循以下规则：

- ① 名字只能由字母、数字和下划线组成。
- ② 名字的第一个字符必须是英文字母，最后一个字符可以是类型说明符。
- ③ 名字的有效字符为 255 个。
- ④ 不能用 VB 的保留字作为变量名，但可以把保留字嵌入变量名中；同时，变量名也不能是末尾带有类型说明符的保留字。

在 VB 中，变量名以及过程名、符号常量名、记录类型名、控件名等都统称为标识符，它们的命名都必须遵循上述规则。同时，在 VB 中不区分名字中的大小写，Hello、HELLO、hello 指的是同一个名称。

任何变量都属于一定的数据类型,在 VB 中可以通过以下方式来定义(或声明)一个或多个变量:

(1) 用 Dim 和 Static 定义变量名称和类型。

- 使用 Dim 定义变量

Dim 变量名 As 数据类型

例如:

```
Dim max as Single           Dim sum(2,5) as Long
Dim name as String        Dim count as Integer, aver as Single
```

如果没有指明数据类型,则所创建的变量为变体类型。也就是说,你给变量什么类型的值,它就存放什么类型的值。变体型变量可以存放各种类型的数据。例如:

```
Dim x
x=4
x="hellp"
```

- 使用 Static 定义变量

Static 变量名 As 数据类型

使用 Static 定义的变量为静态变量。它与 Dim 定义的变量不同之处在于:在执行一个过程结束时,过程中所用到的 Static 变量的值会保留,下次再调用此过程时,变量的初值是上次调用结束时被保留的值;而 Dim 定义的变量在过程结束时不保留,每次调用时需要重新初始化。

例如,在窗体的单击事件中编写如下代码:

```
Private Sub Form_Click()
    Dim x As Integer
    Static y As Integer
    x = x + 1
    y = y + 1
    Print x, y
End Sub
```

当单击该窗体 3 次时,上述过程即被执行了 3 次, x 的值始终是 1, 而 y 是静态变量,能保留上次过程的值,因此 y 值的变化是 1、2、3。

(2) 用类型声明符定义变量类型。

使用类型声明符(见表 3-1)隐含定义变量的数据类型。例如, n% 是一个整型变量, s\$ 是一个字符串变量, d# 是一个双精度浮点型。要注意一点的是:对于局部变量可以用此种方式,如果没有类型说明符,VB 将该变量指定为通用型变量。但是 VB 也提供了强制对变量进行显示声明的措施,即需要用 Dim 等关键词对变量定义。其操作是通过“工具”菜单中的“选项”命令,在打开的“选项”对话框中选择“要求变量声明”,如图 3-2 所示。

这样设置后,每次建立新文件,VB 就将语句 Option Explicit 自动加到全局变量或模块级变量的声明部分,在此情况下运行含有类型声明符定义变量的程序时,会出现提示“变量未定义”。

3.2.5 变量的作用域

变量的作用域指的是变量的有效范围。由于变量的有效范围不同,所以也就有了不同级别的变量之分。为了更好地说明变量的作用域,首先应该了解一个应用程序即一个工程文件包括哪些部分。一般应用程序通常包括窗体文件(*.frm 文件)、模块文件(*.bas 文件)、类模块文件(*.cls 文件)等,如图 3-3 所示。

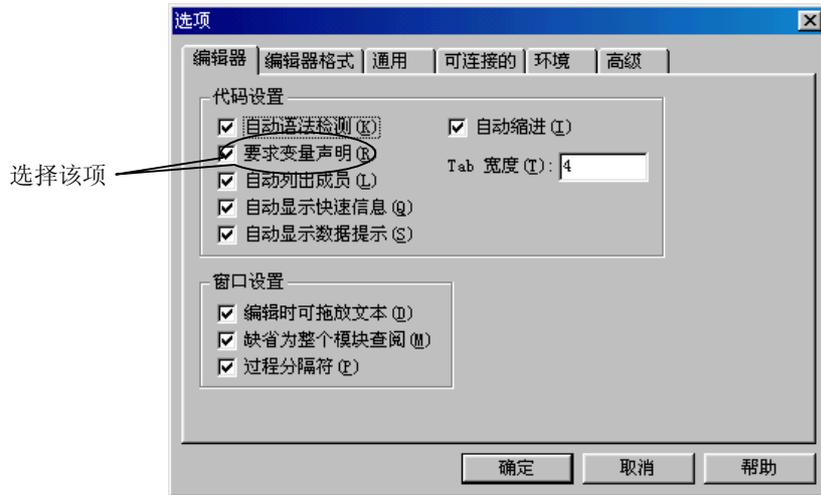


图 3-2 强制变量声明

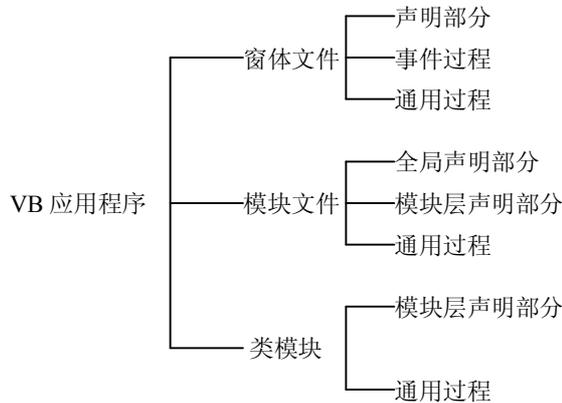


图 3-3 VB 应用程序一般框架

一个窗体文件可以包括事件过程（如 1 个命令按钮的单击事件过程 Command1_Click）、通用过程（如 1 个 Function 过程）以及变量声明部分。这些部分连同窗体一起存入窗体文件（*.frm）中。模块、类模块等文件也包括通用过程和变量声明部分。

变量的作用域分为：局部变量、窗体/模块级变量和全局变量，见表 3-3。

表 3-3 不同作用范围的三种变量声明及使用规则

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/模块的“通用声明”段	
能否被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，但在变量名前加窗体名	能

1. 局部变量

在一个过程内部（例如 Command1_Click）定义的变量称为局部变量。这个变量只能在定

义它的过程内使用。一个窗体可以包括很多过程,在不同过程中定义的变量可以不同名,也可以同名,因为它们是互相独立的,例如:

```
Private Sub Form1_click( )
    Dim Count1 As Integer
    .....
End Sub

Private Sub Command1_Click( )
    Dim Count1 As integer
    Static Sum1 As Single
    .....
End Sub
```

在 Form1_Click 过程中定义了局部变量 Count1, 在 Command1_Click 过程中定义了 Sum1 和 Count1, 两个同名变量 Count1 没有任何关系, 各自在自己的过程范围内起作用, 出了过程, 该变量就不存在了。

2. 窗体和模块级变量

如果一个窗体中的不同过程要使用同一个变量, 这就需要在该窗体或模块内的过程外面定义一个变量, 它在整个窗体或模块中有效, 即其作用域为整个窗体或模块, 本窗体或本模块内的所有过程都能访问它, 这就是窗体或模块级变量。

定义这一级变量要先进入程序代码窗口, 单击左侧下拉列表框右端的下三角按钮, 并选择“通用”, 再单击右侧下拉列表框右端的下三角按钮并选择“声明”, 如图 3-4 所示。



图 3-4 代码编辑器变量声明区域

现在就可以定义变量了。在窗体定义的所有变量都属于这类变量, 这一级变量只在本窗体或本模块中有效。如果在窗体中以 Private 或 Dim 定义变量, 其他窗体或模块不能引用此变量。例如在窗体 Form1 中有如下的变量定义:

```
Private a as integer
```

则变量 a 在窗体 Form1 内的所有过程中均可使用。

3. 全局变量

全局变量可以被程序中的任何一个模块和窗体访问。全局变量可在窗体的通用部分或在模块文件 (*.bas) 中的声明部分用 Public 或 Global 关键字声明。它们的一般格式如下:

Public |Global 变量名 As 数据类型

如果以 Public 定义这个变量, 则允许其他窗体或模块引用它, 但必须指出其所在的对象,

例如在窗体 Form1 中有如下的变量定义：

```
Public a as integer
```

在另一个窗体或模块中可以引用此变量，但必须用 Form1.a，而不能直接使用 a。

通过上面的介绍，可以知道：

- 在过程中只能定义局部变量
- 在窗体中的声明部分可以定义窗体级变量
- 在模块中的声明部分可以用 Private 或 Dim 定义模块级变量
- 在模块中的声明部分可以用 Public 或 Global 定义全局变量

【新的任务】

通过本任务的学习，了解了 VB 变量的声明方法和变量的作用域。在程序设计中，数据要经过一定的运算，以产生想要的结果。

现在新的任务是：学习并掌握 VB 提供的各类运算符和表达式。

3.3 运算符与表达式

任务 3：将相应的数学关系式转换成 VB 表达式。

【任务的提出】

运算是数据的加工。最基本的运算形式常常可以用一些简洁的符号来描述，这些符号称为运算符或操作符。被运算的对象，称为运算量或操作数。表达式就是通过运算符和括号将若干个运算量（如常量、变量或函数）连接起来的式子，其中单个的常量、变量或函数也可以看成是表达式的一种特殊形式。每个表达式都具有一个确定的值，该值具有确定的数据类型。例如， $x+3$ ，“AB”+“CD”， $y>4$ ，6， $xy!$ 等都是表达式。

本任务包括以下内容：

- (1) 算术运算符和字符串连接运算符。
- (2) 关系运算符。
- (3) 逻辑运算符。
- (4) 表达式。

3.3.1 算术运算符

VB 提供的算术运算符如表 3-4 所示。

表 3-4 VB 算术运算符

运算符	含义	举例	说明
+	加	25+30 结果为 55	
+	字符连接	“foot”+“ball”结果为“football”	
-	减	50-10 结果为 40	
-	取负	-15 结果为-15	单目运算，15 取负
*	乘	4*6 结果为 24	
/	除	9/2 结果为 4.5	

续表

运算符	含义	举例	说明
\	整除	7\3 结果为 2	结果取商的整数部分
Mod	取余	7 Mod 3 结果为 1	两数相除之后的余数
^	求幂	3^4 结果为 81	
&	字符连接	“VB”&“程序” 结果为“VB 程序”	两个字符串连接

说明:

(1) “\”是整除运算符。如果参加运算的两个量是整数,则取其商的整除部分,如 $5\ 2$,得 2。如果参加运算的两个量(除数和被除数)是实数,则先按四舍五入原则变成整数,然后取其商的整除部分。如 $4.6\ 2.4$,先变成 $5\ 2$,得 2。

(2) “Mod”是求余数运算符。如果参加运算的两个量是整数,则进行整数相除后取其整数,如 $15\ \text{Mod}\ 2$,得 1。如果参加运算的两个数是实数,则先按四舍五入原则变成整数,然后进行整数相除再取其整数。如 $18.749\ \text{mod}\ 2.378$,先变成 $19\ \text{mod}\ 2$,得余数 1。

(3) “^”是乘方运算符。 2^3 就是算术中的 2^3 。如 2.5^3 ,得 15.625。 2.5^{-3} ,得 0.064。

(4) “+”和“&”是连接运算符,也称字符运算符。在 VB 中,“+”和“&”可以用来连接字符串,生成一个较长的字符串。两者的区别是:“&”作为专门连接字符串的运算符,会自动将非字符串类型的数据转换成字符串后再进行连接,而“+”则不能自动转换。例如: $12\ \&\ "ab"$,连接后的结果为“12ab”。而如果使用 $12\ +\ "ab"$ 则出错。因此,为了避免与算术运算中的加法相混淆,一般用“&”,而不用“+”来连接两个字符串。

(5) 运算优先级。在一个表达式中可能包括多个运算符,这就要确定各运算符的优先顺序问题。算术运算符的优先级别如图 3-5 所示(由高到低排列),运算示例见图 3-6。



图 3-5 算术运算符优先级

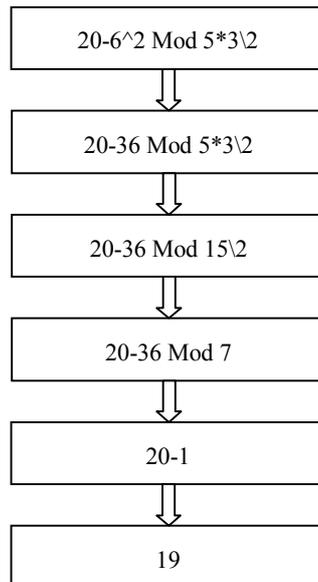


图 3-6 算术表达式运算过程

3.3.2 关系运算符

关系运算也称比较运算,关系表达式的值是一个逻辑值,逻辑值只有 True (“真”)和 False (“假”)两个。例如 $2+3 \geq 5$, $x < 60$, $"ab" < "abc"$ 等就是关系表达式。VB 提供了 8 个关系运算符,它们的优先级相同,如表 3-5 所示。

表 3-5 VB 关系运算符

运算符	含义	举例	说明
=	等于	$45=45$ 结果是 True	
>	大于	$50>60$ 结果是 False	
>=	大于等于	$7>14/2$ 结果是 True	先执行 $14/2$ 运算再进行关系运算
<	小于	$"cat">"bat"$ 结果是 True	"c"大于"b"所以结果为真
<=	小于等于	$16-2 \leq 14$ 结果是 True	
<>	不等于	$30 \neq 30$ 结果是 False	
Like	字符串匹配	$"abc" \text{ Like } "a*"$ 结果是 True	*为通配符
Is	对象比较	Form1 Is Form2 结果是 False	Form1 和 Form2 均为对象

关系运算符可以对数值、字符串和日期进行比较。两个数值型的量相比较时,按其值大小确定,如“ $3 > 2$ ”其值为“真”;两个字符串型的量相比较时,从左到右地逐个比较字符,其大小以字符的 ASCII 码值为标准,如“VB” $<$ “VC”结果为 True;日期型数据的比较,后面的日期大于前面的日期,如 $\#03/22/2003\# > \#07/21/2003\#$ 为 False。

Like 和 Is 运算符有特定的比较功能。Like 运算符用来比较字符串表达式,可以使用诸如“?”、“*”和“#”等这些通配符,主要用于数据库查询;Is 运算符用来比较两个对象的引用变量,此外它还在 Select Case 语句中使用。

3.3.3 逻辑运算符

逻辑运算也称布尔运算。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。VB 的逻辑运算符有: And、Or、Not、Xor、Eqv、Imp 等 6 种,如表 3-6 所示。

表 3-6 VB 逻辑运算符

运算符	含义	举例	说明
Not	逻辑非	$\text{Not}("a" < "b")$ 结果为假	Not 真, 结果为假
And	逻辑与	$3 > 4 \text{ And } 3 < 6$ 结果为假	假 And 真, 结果为假
Or	逻辑或	$5 < 6 \text{ Or } 5 > 1$ 结果为真	真 Or 真, 结果为真
Xor	逻辑异或	$5 < 6 \text{ Xor } 5 < 1$ 结果为真	真 Xor 假, 结果为真
Eqv	逻辑等于	$"a" < "b" \text{ Eqv } "c" > "d"$ 结果为假	真 Eqv 假, 结果为假
Imp	逻辑蕴含	$10 > 3 \text{ Imp } 10 < 15$ 结果为假	真 Imp 假, 前真后假为假

逻辑运算符的优先顺序为: Not \rightarrow And \rightarrow Or \rightarrow Xor \rightarrow Eqv \rightarrow Imp。也就是说,逻辑非(Not)级别最高,逻辑蕴含(Imp)级别最低。各个运算符的运算规则见表 3-7。

表 3-7 逻辑运算真值表

a	b	a And b	a Or b	Not a	a Xor b	a Eqv b	a Imp b
True	True	True	True	False	False	True	True
True	False	False	True	False	True	False	False
False	True	False	True	True	True	False	True
False	False	False	False	True	False	True	True

3.3.4 表达式

1. 表达式组成

由变量、常量、运算符和圆括号按一定的规则组成。表达式通过运算有一个结果，运算结果的类型由数据和运算符共同决定。

2. 表达式书写规则

- (1) 乘号不能省略。如 x 乘以 y 应写成 x*y。
- (2) 括号必须成对出现，且均使用圆括号。
- (3) 表达式从左到右在同一基准上书写，无高低、大小。

如：数学表达式 $(xy)^4$ ，写成 VB 表达式为： $(x*y)^4$ 。

一个表达式可能含有多种运算，计算机按一定的顺序对表达式求值，一般顺序是首先算术运算，其次是关系运算，最后是逻辑运算，括号的级别最高。例如： $4*2<2$ Or $2^3>9$ 先进行算术运算 $(8<2$ Or $8>9)$ ，然后求出关系运算值(False Or False)，最后算出逻辑运算值(False)。

3. 优先级

不同的运算符有不同的优先级，具体关系如下：

算术运算符>关系运算符>逻辑运算符

【新的任务】

通过本任务的学习，了解了 VB 表达式的类型及运算规则。在表达式中常含有函数，VB 提供了大量的函数供用户调用。

现在新的任务是：学习和掌握 VB 提供的常用内部函数。

3.4 常用内部函数

任务 4：熟悉并掌握常用 VB 系统函数。

【任务的提出】

函数是一种特定的运算，是一段程序。每一个函数都有函数名，调用时返回函数值。在 Visual Basic 中，有两类函数：内部函数和用户自定义函数。用户自定义函数是用户根据需要定义的函数，也就是后面所要说的 Function 过程。内部函数也称标准函数或系统函数，VB 本身提供了大量的内部函数。在这些函数中，有些是通用的，有些与某种操作有关。

本任务包括以下内容：

- (1) 数学函数。
- (2) 转换函数。

- (3) 字符串函数。
- (4) 日期与时间函数。
- (5) 随机数函数。

3.4.1 数学函数

常见的数学函数如表 3-8 所示。

表 3-8 常用数学函数

函数	说明	示例	返回值
Sin(x)	返回自变量 x 的正弦值	Sin(0)	0
Cos(x)	返回自变量 x 的余弦值	Cos(0)	1
Tan(x)	返回自变量 x 的正切值	Tan(0)	0
Atn(x)	返回自变量 x 的反正切值	Atn(0)	0
Abs(x)	返回自变量 x 的绝对值	Abs(-4)	4
Exp(x)	返回以 e 为底, 以 x 为指数的值, 即 e 的 x 次方 e^x	Exp(1)	2.7183
Sqr(x)	返回 x 的平方根	Sqr(9)	3
Sgn(x)	返回数的符号值: 当 x 为负数时, 函数返回-1; 当 x 为 0 时, 函数返回 0; 当 x 为正数时, 函数返回 1	Sgn(4)	1
Log(x)	返回以 e 为底的自然对数	Log(2.7183)	1
Round(x,[n])	对 x 进行四舍五入, 保留的小数位数由 n 指定	Round(5.346,2)	5.35

上述三角函数 (Sin、Cos 和 Tan) 的自变量 x 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。

3.4.2 转换函数

转换函数就是强制地将一个表达式转换成某种特定的数据类型, 常用的 VB 转换函数如表 3-9 所示。

表 3-9 常用转换函数

函数	说明	示例	返回值
Int(x)	求不大于自变量 x 的最大数	Int(-4.2)	-5
Fix(x)	去掉 1 个浮点数的小数部分, 保留其整数部分	Fix(-4.7)	-4
Asc(x\$)	返回字符串 x\$ 中第一个字符的 ASCII 码	Asc("abc")	97
Chr(x)	把 ASCII 码值为 x 的整数转换为相应的字符	Chr(110)或 Chr\$(110)	"n"
Str(x)	把 x 的值转换为一个字符串	Str(23.4)或 Str\$(23.4)	"23.4"
Val(x)	将数字字符串 x 转换为数值	Val("45AB")	45
Cint(x)	把 x 的小数部分四舍五入, 转换为整数	Cint(-4.7)	-5
Lcase(x)	将 x 转换为小写字母	Lcase("VisualBasic")	"visualbasic"
Ucase(x)	将 x 转换为大写字母	Ucase("VisualBasic")	"VISUALBASIC"

说明: Val(x)函数在遇到不为数字的第一个字符时, 停止转换, 如 Val("4k2")的结果为 4, 但可以识别进制符号, 如&O 和&H, 并且若 x 中含有空格、制表符和换行符时, 这些符号将从 x 中被去掉。例如, Val("&O35")和 Val("3 4"+chr(13)+"3")的值为 29 和 343。

3.4.3 字符串函数

VB 提供了丰富的字符串操作函数, 从而极大地提高了 VB 对字符的处理能力, 常见的字符串函数如表 3-10 所示。字符串函数名后加“\$”符号表示函数的返回值是字符串类型的, 但也可省略。

表 3-10 常用字符串函数

函数	说明	示例	返回值
Ltrim\$(字符串)	去掉字符串左边的空白字符	Ltrim\$("□□OK")	"OK"
Rtrim\$(字符串)	去掉字符串右边的空白字符	Rtrim\$("OK□□")	"OK"
Left\$(字符串,n)	取字符串左部的 n 个字符	Left\$("ABCD",2)	"AB"
Right\$(字符串,n)	取字符串右部的 n 个字符	Right\$("ABCD",2)	"CD"
Mid\$(字符串,p,n)	从位置 p 开始取字符串的 n 个字符	MID\$("ABCD",2,2)	"BC"
Len(字符串)	测试字符串的长度	Len("Visual Basic")	12
String\$(n,字符串)	返回由 n 个字符组成的字符串	String\$(4, "*")	"****"
Space\$(n)	返回 n 个空格	Space\$(3)	"□□□"
String\$(N,C)	返回由 C 中首字符组成的 N 个字符串	String\$(3, "ABCDEF")	"AAA"
Instr([n], 字符串 1, 字符串 2)	返回字符串 2 在字符串 1 中的位置	Instr("Computer","pu")	4

3.4.4 日期与时间函数

常见的日期与时间函数如表 3-11 所示。

表 3-11 常用日期与时间函数

函数	说明	示例	返回值
Date\$[]([])	返回系统当前日期	Date	2003-6-20
Time\$[]([])	返回系统当前时间	time	9:30:26 AM
Now	返回系统当前日期和时间	Now	2003-6-20 9:30:26 AM
Year(N)	返回指定日期 (N) 的年份	Year("2003-05-22")	2003
Month(N)	返回指定日期 (N) 的月份	Month("2003-05-22")	5
Day(N)	返回指定日期 (N) 的某一日	Day("2003-05-22")	22
Weekday(N)	返回指定日期 (N) 的星期	Weekday("2003-05-22")	5

Date 和 Time 及 Now 的返回值格式, 可以在本机 Windows 的“区域设置”中进行设置。参数 N 是能够表示日期的通用型、数值表达式、字符串表达式或它们的组合。

除了上述日期函数外, 还有两个函数比较有用。

(1) DateAdd 函数

形式: DateAdd(要增减日期形式, 增减量, 要增减的日期变量), 增减日期形式见表 3-12。

作用：对要增减的日期变量按日期形式作增减。

例如：DateAdd("ww",2,#2/13/2007#)，它表示在指定的日期上加 2 周，所以最终结果为 #2/27/2007#。

(2) DateDiff 函数

形式：DateDiff (要间隔日期形式，日期 1，日期 2)，间隔日期形式见表 3-12。

作用：两个指定的日期按日期形式求相差的日期。

例如：DateDiff("m", #2/14/2007#, #3/8/2007#)，它表示两个日期之间相差几个月，结果为 1。

表 3-12 日期形式

日期形式	yyyy	q	m	y	d	w	ww	h	n	s
意义	年	季	月	一年的天数	日	一周的日数	星期	时	分	秒

3.4.5 随机函数

格式：Rnd[(N)]

功能：产生一个小于 1 但大于等于 0 的单精度数，参数 N 决定了生成随机数的方式，一般可以省略。为了生成某个范围的随机整数，可以使用如下的公式：

$$\text{Int}((U-L+1)*\text{Rnd}+L)$$

其中：U 是产生随机整数范围的上限，L 是下限。例如产生 [10~20] 之间的随机整数，可以使用下面的表达式：Int((20-10+1)*Rnd+10)。

利用随机函数，还可以随机产生英文字母，这是因为英文字母的 ASCII 也是整数（大写字母的范围是 [65, 90]，小写字母的范围是 [97, 122]）。

如产生随机的小写英文字母的表达式为：Chr(Int((122-97+1)*Rnd+97))。

另外，当一个应用程序不断地重复使用随机函数时，同一序列的随机数会反复出现。用 Randomize 语句可以消除这种情况，它可以初始化随机数生成器，其格式为：

Randomize [(x)]

其中：x 是一个整型数，它是随机数发生器的“种子数”，可以省略。

【新的任务】

通过本任务的学习，了解了 VB 常用的系统函数。

VB 是一种程序设计语言，语言中：由笔画构成字，由字构成常用的词，由字和词按语法规则构成句，再由句构成表达某种思想的文章。计算机语言也与此类似，VB 的程序设计也相当于写文章，单个程序也是由各种语句组成。程序设计一般分三步：“输入”→“处理”→“输出”，即用户通过键盘等输入原始数据，然后将编制的程序进行处理，最后将处理结果输出到指定地点。

现在新的任务是：学习并掌握基本输入输出语句。

3.5 VB 基本输入输出语句

任务 5：将表达式的值赋给相应的变量，并以合适的形式输出。

【任务的提出】

Visual Basic 的输入、输出有着十分丰富的内容和形式，掌握输入输出是编程的入门，本

节将介绍 VB 的基本输入输出方式。本任务包括以下内容:

- (1) 基本赋值语句。
- (2) 输入对话框。
- (3) Print 方法。
- (4) 输出消息对话框。

3.5.1 语句及约定

1. 语句

(1) 语句概念

程序由语句组成, Visual Basic 中的语句是执行具体操作的指令, 每个语句以回车键(Enter)结束。程序语句是 Visual Basic 关键字、属性、函数、运算符以及能够生成 Visual Basic 编辑器可识别指令的符号的有意义组合。如 END 就是结束语句, STOP 是暂停语句等。早期 BASIC 中的某些语句(如 PRINT 输出、CLS 清屏)在 VB 中被改为方法, 而依附于对象, 大部分语句仍保留为语句(如赋值、REM 或'注释、END 结束等)。书写语句时不分大小写。

(2) 复合语句行

一般情况下, 一行写一句。所谓复合语句行是指在一行写多条语句, 此时应在语句间加语句分隔符“:”。一个语句行的长度不能超过 1023 个字符。如:

```
Text1.text="Hello" : Red=255 : Text1.BackColor=Red
```

(3) 语句断行

当一条语句太长时, 可将该语句分成两行或多行来写, 称为语句断行。语句断行必须在语句末尾使用语句续行符“_”。在使用续行符时, 在它前面至少要加一个分隔符(空格、逗号、Tab 键等), 如:

```
str="我正在学" & _  
"习 VB! "
```



注意

使用续行符时, 不能把程序中完整的标识符拆开。如下面做法是错误的: Dim
str As St_
ring

(4) 几个简单语句

END——结束程序。

STOP——暂停执行。一般在调试程序时使用。

REM 或'<注释内容>——注释语句。以对程序或语句功能或变量作用等作说明, 主要是为了增强程序的可读性。程序执行时, 注释内容被忽略。其中, REM 语句为行注释, 即同一程序行的 REM 前不能有其他语句; 撇号“'”注释可以作为行注释使用, 与 REM 一样, 但更多的是作为行末注释使用, 即它可跟在一个可执行语句的后面, 同一行的撇号“'”后的内容全部被认为是注释内容。

2. 本书约定

为便于描述, 本书的语法格式描述约定如下:

<>: 尖括号表示其内容为必需的语法成份。

[]: 中括号表示其内容可缺省。

|: 竖号表示或者，即两者任选其一。

3.5.2 赋值语句

格式: <变量名>=<表达式>

功能: 先计算赋值号 (=) 右边表达式的值，再将其值赋给左边的变量或对象的属性。

赋值语句是最常用的语句，在 VB 中，经常可以用到两种赋值语句。

1. 给变量赋值

在赋值语句中，将常量赋给一个变量是最简单的形式，也可以将一个表达式的值赋给一个变量，也可以通过该语句在变量之间传递值。例如：

```
Dim x As Integer
Dim y ,z As String
x=34*sqr(4)
x=x+1
y="欢迎使用VB"
z=y
```

但要注意的是：

- 赋值语句中的“=”不同于数学中的“等于号”，它是“赋值号”。
- 赋值号的左侧只能是一个变量，而不能是常数或表达式。例如，10=x 是错误的。
- 赋值号左侧的变量和右侧的常数或表达式的类型必须保持兼容，即类型一致或能够自动转换，否则将会出现错误。例如：

```
Dim x As Integer
x="ABC"      '类型不一致，错误赋值
```

【例 3.1】利用中间变量将变量 A、B 的值交换。

分析: 要交换两个变量的值，可以利用中间变量作为暂存单元来进行。选择中间变量 T，先将 A 的值暂存到 T 中，再将变量 B 的值送入变量 A，最后将 T 的值送入变量 B。

代码如下：

```
Private Sub Form1_Click( )
Dim A As Integer,B As Integer,T As Integer
    A=10
    B=20
    T=A: A=B: B=T      '交换变量 A 和 B 的值
End Sub
```

2. 给对象的属性赋值

在 VB 程序中，用赋值语句为对象的属性设置值，它的一般格式为：

对象名.属性=属性值

例如，为窗体 Form1 的 Caption 属性设置值：Form1.Caption="VB 学习系统"。

3. 给对象赋值

在 VB 程序中，用赋值语句可以把一个对象的引用赋值给另外一个同类型的对象变量，它的一般格式为：

Set 对象名=对象的引用

通常情况下，“对象的引用”就是另外一个对象名。

3.5.3 输入对话框

在 Visual Basic 中，输入对话框用于在程序的运行过程中接收用户由键盘输入的数据，输

入对话框由 `InputBox` 函数产生。`InputBox` 函数的语法格式为:

`InputBox("提示信息", "标题", "默认值", x 坐标, y 坐标)`

函数作用: 显示一个输入对话框, 并可利用该对话框在程序运行过程中接收用户由键盘输入的数据。

例如, 下述代码将显示如图 3-7 所示的输入提示对话框。

```
x=InputBox("请输入成绩: "+chr(13)+"(A,B,C)", "成绩输入框", "B")
```



图 3-7 输入对话框

上述函数 `Chr(13)` 返回 ASCII 值为 13 的“回车”, 这里起到换行的作用。

(1) 函数的参数说明。

- "提示信息": 是一个字符串, 用来表示在输入对话框中显示的文本内容, 该参数项不可省略。
- "标题": 为输入对话框的标题, 如图 3-7 中的“成绩输入框”。该参数项可省略, 省略该项后输入对话框的标题显示为应用程序名称。要注意的是, 如果该参数项后面还有其他参数, 则省略该项后其位置上的逗号应保留。
- "默认值": 为输入对话框的默认输入内容, 在输入对话框刚产生时, 该项的值自动显示在对话框中的文本框内。该参数项也可省略, 省略该项表示默认值为空。
- x 坐标和 y 坐标: 用来指定输入对话框在屏幕上的显示位置, 坐标值是相对于屏幕左上角的值。该项可以省略, 若省略该项则输入对话框默认显示在屏幕中央。



如果省略了某个可选参数, 并且该参数项后还有其他参数, 则必须在省略的参数项位置添加逗号分界符。

(2) 为了保存输入的数据, 一般将该函数应用在一个变量赋值语句中, 使用格式为:

变量 = `InputBox("提示信息", "标题", "默认值", x 坐标, y 坐标)`

执行该语句后, 系统首先显示一个输入对话框并等待用户输入数据, 在对话框中包含“确定”和“取消”两个按钮, 用户输入数据并单击“确定”按钮 (或直接按 `Enter` 键) 后, 函数将返回用户输入的数据并将该数据赋给指定的变量; 如果用户单击“取消”按钮, 则函数返回一个 0 长度的字符串。

(3) `InputBox` 函数的返回值为字符型。如果要产生数值型数据的输入, 应使用 `Val` 转换函数对 `InputBox` 函数的返回值进行类型转换。例如, 如果要输入一个学生的成绩, 可以使用以下语句:

```
intGrade=Val(InputBox("请输入一个学生的成绩: ", "输入成绩"))
```

(4) 每执行一次 `InputBox` 函数只能输入一个值, 如果需要输入多个值须多次调用 `InputBox` 函数。



注意

赋值语句是静态地给变量赋值，常用于程序代码中，可为各种数据类型；InputBox 函数是动态地给变量赋值，常用于程序运行时输入，它只能返回字符型数据。

3.5.4 Print 方法

Print 方法可以在窗体上显示文本字符串和表达式的值，并可在其他图形对象或打印机上输出信息。其格式为：

[<对象名称>].Print [表达式列表][,;]

说明：

(1) “对象名称”可以是窗体 (Form)、图片框 (PictureBox) 或打印机 (Printer)。如果省略对象名称，则在当前窗体上直接输出。例如：

```
Picture1.Print "Visual Basic 天地"
Print "Visual Basic"
```

前者把字符串“Visual Basic 天地”在图片框 Picture1 上显示出来，后者则把字符串直接输出到当前窗体。

(2) “表达式列表”是一个或多个表达式，可以是数值表达式或字符串。对于数值表达式，将输出表达式的值，字符串照原样输出。如果省略“表达式列表”，则输出一个空行，相当于换行。输出数据时，数值数据的前面有一个符号位，后面有一个空格，而字符串前后都没有空格。

例如：

```
Print 2
Print
Print "Hello!"
```

输出结果为：

```
2
Hello!
```

(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) 隔开。如果使用逗号分隔符，则按标准输出 (分区输出) 格式显示，此时，以 14 个字符宽度为单位将输出行分为若干区段，逗号后面的表达式在下一个区段输出。如果使用分号或空格作为分隔符，则按紧凑格式输出，即各输出项之间无间隔地连续输出。

例如：

```
Print 2, 3, "Hello!"
Print 2, 3; "Hello"; "OK"
```

输出结果为：

```
2      3      Hello!
3HelloOK
```

【例 3.2】 使用 Print 方法在窗体中直接输出字符串或数据表达式的值。

步骤：① 新建一个窗体，在窗体的单击事件中编写如下代码：

```
Private Sub Form1_Click()
    a = 10
    b = 3.5
    Print a, 2 * b,
    Print a * b
    Print "HELLO!"
    Print "a=";10
```

```
Print "欢迎使用" & "Visual Basic"
End Sub
```

② 运行程序，其结果如图 3-8 所示。

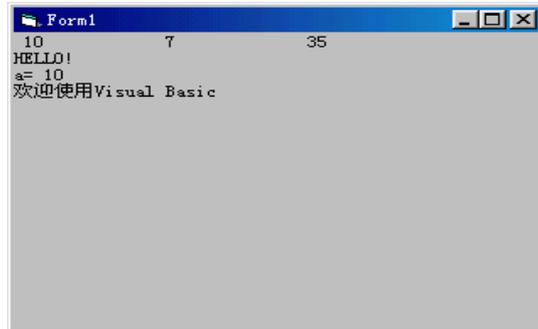


图 3-8 Print 使用示例

3.5.5 输出消息对话框

在 Windows 应用程序中，经常使用输出消息对话框（简称消息框）显示一些警告信息、提示信息、确认信息等。例如，在 Windows 系统中删除文件时，系统显示的“确认文件删除”对话框即为一种消息框。在 VB 中，可以使用 MsgBox 函数产生消息框。

MsgBox 函数的语法格式为：

变量=MsgBox("输出信息" [,按钮类型][, "标题"])

函数作用：显示一个消息框，用于输出信息，并要求用户对输出的信息进行确认等操作。函数返回值为一整型数据。

例如，在立即窗口中输入代码为：

```
x=MsgBox("请输入数据！" , 3+48+0, "消息框")
```

则显示结果如图 3-9 所示。



图 3-9 显示对话框

函数说明：

(1) 函数的参数。

- "输出信息"：为字符串表达式，作为显示在对话框中的消息，如图 3-9 中的输出信息“请输入数据！”。该参数项不可省略。



小技巧

如果“输出信息”包含多个行，则可在各行之间用回车符 Chr(13)、换行符 Chr(10) 或回车换行符的组合 Chr(13) & Chr(10) 进行分隔。

- 按钮类型：该参数项为一数值表达式，指定消息框中所显示的按钮的数目及形式、使用的图标样式、缺省按钮以及消息框的强制回应等。该参数项可省略，如果省略，则该参数项的缺省值为 0。

该参数项由四组设置值相加形成，各组设置值及含义见表 3-13。

表 3-13 “按钮类型”参数项的设置值及含义

类型	内部常数	数值	描述
按钮类型与数目	VbOKOnly	0	显示“确定”按钮
	VbOKCancel	1	显示“确定”及“取消”按钮
	VbAbortRetryIgnore	2	显示“终止”、“重试”及“忽略”按钮
	VbYesNoCancel	3	显示“是”、“否”及“取消”按钮
	VbYesNo	4	显示“是”及“否”按钮
	VbRetryCancel	5	显示“重试”及“取消”按钮
图标样式	VbCritical	16	显示错误信息图标 (❌)
	VbQuestion	32	显示询问信息图标 (❓)
	VbExclamation	48	显示警告信息图标 (⚠️)
	VbInformation	64	显示普通信息图标 (ℹ️)
缺省按钮	VbDefaultButton1	0	第一个按钮是缺省按钮
	VbDefaultButton2	256	第二个按钮是缺省按钮
	VbDefaultButton3	512	第三个按钮是缺省按钮
	VbDefaultButton4	768	第四个按钮是缺省按钮
消息框的强制 返回方式	VbApplicationModal	0	应用程序强制返回；应用程序一直被挂起，直到用户对消息框作出响应才继续工作
	VbSystemModal	4096	系统强制返回；全部应用程序都被挂起，直到用户对消息框作出响应才继续工作

消息框的按钮类型取值既可由四组设置值的数值相加形成，也可由四组设置值的内部常数相加形成。例如，产生图 3-9 所示消息框的 Visual Basic 语句也可写成以下形式：

```
x=MsgBox("请输入数据!", VbYesNoCancel + VbExclamation + vbDefaultButton1, "消息框")
```

- “标题”：指定在消息框标题栏中显示的字符串表达式。该参数项可省略，如果省略该项，则将应用程序名显示在消息对话框的标题栏中。

(2) MsgBox 函数的返回值。该值为整数，表示的是用户在消息框中选择了哪一个按钮，在进行程序设计时经常根据该返回值决定程序的流程。MsgBox 函数的返回值与消息框中常见的按钮对应关系见表 3-14。

表 3-14 MsgBox 函数的返回值及含义

返回值	内部常数	说明
1	vbOk	用户在消息框中单击了“确定”按钮
2	vbCancel	用户在消息框中单击了“取消”按钮
3	vbAbort	用户在消息框中单击了“终止”按钮
4	vbRetry	用户在消息框中单击了“重试”按钮
5	vbIgnore	用户在消息框中单击了“忽略”按钮
6	vbYes	用户在消息框中单击了“是”按钮
7	vbNo	用户在消息框中单击了“否”按钮

另外, MsgBox 函数也可以写成语句的形式, 格式为:

MsgBox <信息内容> [,<对话框类型> [,<对话框标题>]]

MsgBox 语句没有返回值, 常用于较简单的信息提示, 如图 3-10 所示。



图 3-10 MsgBox "OK!" 语句运行结果

【新的任务】

通过本任务, 主要学习了利用赋值语句给变量 (控件属性也可看成变量) 赋值, 利用 InputBox 函数产生的输入对话框给变量赋值 (字符型), 赋值时要注意数据类型匹配。学习了利用 Print 方法输出数据, Print 具有计算功能, 可跟多个输出项, 各项之间用逗号 (标准格式) 和分号 (紧凑格式) 控制输出位置。MsgBox 既可作为函数返回值也可作为语句产生消息框。通过本章的学习, 掌握 VB 的基本语言元素和基本输入输出语句, 为程序设计打下了基础。

现在新的任务是: 进一步学习程序的分支结构、循环结构及程序调试等内容。

习题三

一、填空题

1. 写出下列常量的数据类型:

45 _____ “45” _____ 4.5D+2 _____ 4.5E2 _____

2. 指出下列变量的类型:

min _____ max! _____ i% _____ str\$ _____ Count% _____ Area# _____

3. 指出下列合法的变量名是 _____。

Integer, _Student, 4r, \$test, Use, β, 变量, Book/No, Stu.No

4. VB 中, 字符串常量的分界符是_____ , 日期/时间型常量的分界符是_____。

5. 执行下列语句, 输出的结果是_____。

```
a$= "Good"
b$= "Morning"
Print a$+chr(13)+b$
```

6. 窗体的单击事件中有如下代码:

```
Private Sub Form1_Click( )
    Static x as integer
    x=x+1
    Print x
End Sub
```

运行该程序, 单击窗体两次, 窗体上显示的内容是_____。

7. 用 VB 语言的表达式正确描述下列命题。

a 小于 b 或小于 c _____

a 和 b 中有一个小于 c _____

a 是奇数 _____

a 不能被 b 整除 _____

8. InputBox 函数返回值的数据类型为_____ ; MsgBox 函数返回值的数据类型为_____。

二、单项选择题

1. 下列是合法变量的是 ()。

A. Filename B. A(A+B) C. 254D D. Print

2. 下列是 VB 所允许的数是 ()。

A. $10^{(1.25)}$ B. D32 C. 12E D. ± 2.5

3. 数“8.6787E+8”写成普通的十进制是 ()。

A. 86787000 B. 867870000 C. 8678700 D. 8678700000

4. 语句 Print 5*5\5/5 的输出结果是 ()。

A. 5 B. 25 C. 0 D. 1

5. 表达式 $4+5\backslash 6*7/8 \text{ Mod } 9$ 的值是 ()。

A. 4 B. 5 C. 6 D. 7

6. 执行以下程序段后, 变量 c\$ 的值为 ()。

```
a$="Visual Basic Programming"
b$="Quick"
c$=b$ & Ucase(Mid$(a$,7,6)) & Right$(a$,11)
```

A. Visual BASIC Programming B. Quick BASIC Programming
C. QUICK Basic Programming D. QUICK BASIC Programming

7. 执行下面的语句后, 所产生的信息框的标题是 ()。

```
a = MsgBox("AAA", , "BBBB")
```

A. BBBB B. AAA C. 空 D. 出错, 不能产生消息框

8. 下列哪一个函数能将 3.6 转为 4? ()。

A. int(3.6) B. fix(3.6) C. cint(3.6) D. round(3.6,1)

三、应用题

1. 整型与长整型、单精度与双精度的区别在哪里?
2. 何为常量和变量? 什么情况下宜用常量? 什么情况下宜用变量?
3. 写出下列函数的值。

- | | | |
|-----------------------|------------------|----------------------------|
| ① ABS(-2) | ② INT(17.8) | ③ SQR(36) |
| ④ INT(SGN(-3.8)) | ⑤ EXP(FIX(0.23)) | ⑥ LEN(" I am a student! ") |
| ⑦ LEFT\$(" ABCD ", 2) | ⑧ STR(24.5) | ⑨ CHR(87) |

4. 将下列数值表达式写成 VB 表达式。

- | | | |
|----------------------------|---|---|
| ① $a + \frac{b}{2c}$ | ② $3x \cdot 4\cos x$ | ③ $\sqrt{b^2 - 4ac}$ |
| ④ $8e^x \cdot \ln 2$ | ⑤ $\frac{-b + \sqrt{b^2 - 4ac}}{2a}$ | ⑥ $(\cos 2x + \sin 2x) \div \sqrt{n + m}$ |
| ⑦ $\frac{x^2 + y^2}{2a^2}$ | ⑧ $\frac{\sin x + \cos x}{2} + \frac{\sin x - \cos x}{2}$ | ⑨ $\sqrt[3]{x} \cdot \sqrt[4]{y}$ |

5. 设 $a=3$, $b=5$, $c=-1$, $d=7$, 写出下列关系表达式与逻辑表达式的值。

- | | | |
|--|---------------------------------|------------------------------------|
| ① $a+b>c+d$ | ② $a\leq d-c$ | ③ $a>0 \text{ AND } c>0$ |
| ④ $\text{NOT } b+d=12$ | ⑤ $a+c<b \text{ OR } b+c\leq d$ | ⑥ $(a-b>=c) \text{ AND } (a+b>=d)$ |
| ⑦ $\text{NOT } a\leq c \text{ OR } 4*c=b^2 \text{ AND } d\leq a+c$ | | |

6. 编写窗体的单击事件代码, 求一任意三角形的面积, 三角形的三条边 a 、 b 、 c 通过 `InputBox` 函数输入, 计算的结果放入变量 s 中, 并以消息框的形式输出。

注: 任意三角形的面积公式 $s = \sqrt{t(t-a)(t-b)(t-c)}$, $t = (a+b+c)/2$

7. 编写一段代码, 求给定一个圆的周长和面积。要求:
 - (1) 圆的半径 r 利用 `InputBox` 函数从键盘任意输入($r>0$)。
 - (2) 圆周率定义为符号常量。
 - (3) 利用赋值语句将求出的周长和面积赋给变量 l 和 s 。
 - (4) 用 `Print` 方法输出详细结果。
8. 编写窗体的单击事件代码, 在窗体上显示如下的图形:

```

A
AAA
AAAAA
AAAAAAA
```