第 3 章　数据类型及其运算

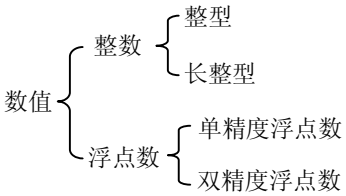
本章主要介绍构成 Visual Basic 应用程序的基本元素，包括数据类型、常量、变量、内部函数、运算符和表达式等。

3.1　数据类型

在各种程序设计语言中，数据类型的规定和处理方法是各不相同的。VB 不但提供了丰富的标准数据类型，还可以有用户自定义所需的数据类型。基本数据类型主要有数值型和字符串型，此外还提供了字节、货币、对象、日期、布尔和变体数据类型。

基本数据类型

1. 数值



整数是不带小数点和指数符号的数，整数又分为整型和长整型。

整型（Integer）：占 2 个字节，其取值范围为-32768～32767。

长整型（Long）：占 4 个字节，其取值范围为-2147483648～2147483647。

浮点数也称实型数或实数，是带有小数部分的数值。它由三部分组成：符号、指数及尾数。单精度型和双精度型的指数分别用 E（或 e）和 D（或 d）来表示。例如：

123.45E4 或 123.45e+4 （单精度型，相当于 123.45 乘以 10 的 4 次幂）

123.45678D4 或 123.45678d+4 （双精度型，相当于 123.45678 乘以 10 的 4 次幂）

在以上的例子中，123.45 或 123.45678 是尾数部分，e+4（也可以写成 E4 或 e4）和 d+4（也可以是 D4 或 d4）是指数部分。

单精度型（Single）：占 4 个字节，单精度型的数据可以精确到 7 位十进制数。其负数的取值范围为-3.402823E+38～-1.40129E-45，正数的取值范围为 1.40129E-45～3.402823E+38。

双精度型（Double）：占 8 个字节，双精度型的数据可以精确到 15 位或 16 位十进制数。其负数的取值范围为-1.797693134862316D+308～-4.94065D-324，正数的取值范围为 4.94065D-324～1.797693134862316D+308。

2. 字符串（String）

字符串类型用来定义一个字符串序列，由 ASCII 字符组成。在 Visual Basic 中，字符串是放在双引号内的若干个字符，其中长度为 0（即不含任何字符）的字符串称为空字符串。

字符串通常放在引号中，例如：

"Hi"
"Visual Basic 6.0"
""（空字符串）

3. 货币（Currency）

货币数据类型是为计算货币而设置的数据类型。在内存中用 8 个字节（64 位）存储，精确到小数点后 4 位（小数点前 15 位），在小数点后第 4 位以后的数字将被舍去。其取值的有效范围为-922337203685477.5808～922337203685477.5807。其类型声明符为@。浮点数中的小数点是“浮动”的，即小数点可以出现在数的任何位置，而货币类型数据的小数点是固定的，因此称为定点数据类型。

4. 日期（Date）

日期型数据存储为 IEEE64 位（8 个字节）浮点数值形式。它可以表示的日期范围为公元 100 年 1 月 1 日到 9999 年 12 月 31 日，而时间从 0:00:00 到 23:59:59。任何可辨认的文本日期都可以赋值给日期变量。日期文字须以数字符号（#）括起来，例如：

#January 1, 2001#

日期型数据用来表示日期信息，其格式为 mm/dd/yyyy 或 mm-dd-yyyy，取值范围为 1/1/100 到 12/31/9999。

注意：在有些 Visual Basic 版本中，输出年份时通常只输出后两位，例如“1997”输出时为“97”。对于 2000 年以后的年份，其输出为“00”、“01”等。因此，在输出 2000 以后的年份时，应该做适当的处理（如前面加上“20”）。

5. 布尔（Boolean）

布尔型数据是一个逻辑数据，用 2 个字节存储，它只取两种值，即 True（真）和 False（假）。

6. 字节（Byte）

字节实际上是一种数值类型，占 1 个字节，其取值范围为 0～255。

7. 对象（Object）

对象型数据用来表示图形、OLE 对象或其他对象，用 4 个字节存储。

8. 变体（Variant）

变体数据类型是一种可变的数据类型，可以表示任何值，包括数值、字符串、日期/时间等。

以上我们介绍了 Visual Basic 中的基本数据类型。表 3-1 列出了这些数据类型的名称、存储空间、取值范围和它们的类型声明符。

表 3-1　Visual Basic 常用的基本数据类型

类型	类型声明符	存储空间（字节）	值的有效范围
String（字符串）	$	1	0～65535
Integer（整型）	%	2	-32768～32767
Long（长整型）	&	4	-2147483648～2147483647
Single（单精度浮点型）	!	8	负数：-3.402823E+38～-1.40129E-45 正数：1.40129E-45～3.402823E+38
Double（双精度浮点型）	#	8	负数：-1.797693134862316D+308～-4.94065D-324 正数：4.94065D-324～1.797693134862316D+308

续表

类型	类型声明符	存储空间（字节）	值的有效范围
Currency（货币）	@	8	-922337203685477.5808～922337203685477.5807
Date（日期类型）	无	8	1/1/100～12/31/9999
Byte（字节）	无	1	0～255
Boolean（布尔）	无	2	True 或 False
Object（对象）	无	4	任何对象引用
Variant（变体类型）	无	按需分配	上述有效范围之一

3.2 变量和常量

计算机在处理数据时，必须将其放入内存。在高级语言中，需要将存放数据的内存单元命名，可以通过内存单元名来访问其中的数据。被命名的内存单元，就是变量或常量。

3.2.1 变量

在计算机的高级语言中，一般将一个有名称的内存位置称之为变量，即变量代表着计算机内存中指定的存储单元，必须通过某种方式去访问它，才能执行指定的操作。Visual Basic 也不例外，也通过变量来存储数据。为了让计算机为变量留出所需要的空间，在使用变量前要对其进行命名和声明。习惯上，变量的命名称之为定义，变量类型的说明（即变量的存储方式的说明）称之为声明。在大多数情况下，变量的命名与变量类型的声明是在同一条语句中完成的。也就是说，用该语句来定义一个变量或者声明一个变量，其意义是一致的。

使用变量有三个步骤：

（1）声明变量。告诉程序变量的名称和类型。

（2）给变量赋值。赋予变量一个要保存的值。

（3）使用变量。在程序中获得变量中所存储的值。

在对变量进行命名与类型说明时，还必须对变量使用时的有效范围加以说明。在 Visual Basic 中，根据变量有效范围的不同，可以将变量分为局部变量、窗体模块级变量和标准模块级全局变量。以下，我们将具体介绍变量的命名规则。

为了使用的需要，每个变量都有一个名字以及与其相对应的数据类型，以便用户通过名字来引用该变量。数据类型可以决定该变量的存储方式。

在 Visual Basic 中，变量的命名规则如下：

（1）变量名必须以英文字母或汉字开头，最后一个字符可以是类型说明符。

（2）变量名所用的字符只能由字母、数字和下划线组成，不能含有标点和空格。

（3）变量名的有效字符为 255 个。

（4）不能用 Visual Basic 的保留字作为变量名，但是可以把保留字嵌入变量名中；变量名也不能是末尾带有类型说明符的保留字。例如，变量 Print 和 Print$ 是非法的，但 Print_sequence 是合法的。

在 Visual Basic 中，变量名以及过程名、符号常量名、记录类型名、元素名等都称为名字，它们的命名规则必须遵循上述规则。

Visual Basic 不区分变量名和其他名字中字母的大小写，Hello、HeLLO 和 hello 指的是同一个名字。换句话说，就是在定义一个变量后，只要字符相同，不管其大小写，指的都是该变量。为了便于阅读，每个单词的第一个字母一般使用大写，如 PrintDocument。

例 3.1　判断下列变量是否合法。

5ax	错误，不能以数字开头
Int　al	错误，不能出现空格
String	错误，不能使用保留字
Y-z	错误，不允许出现减号
S*t	错误，不允许出现乘号
Asd_er	正确

3.2.2　变量的显式声明和隐式声明

一个变量被命名之后，就要通过变量的声明来说明该变量的存储方式，以便系统将其值存储到计算机的内存中。在 Visual Basic 中，变量的声明可以用专用语句显式声明，也可以采用默认方式隐式声明。

1. 变量的显式声明

用语句声明又可以分为以下几种方式。

（1）用 Dim 语句声明变量。

Dim 语句用来在标准模块、窗体模块或过程中声明变量。Dim 语句声明变量的格式为：

Dim 变量名 As 数据类型

例如：

```
Dim i as integer
```

一条 Dim 语句可以同时声明多个变量，但是每个变量都有其自己独立的数据类型声明，即 Dim 语句可以共用，但是数据类型不能共用。例如：

```
Dim i as integer,s as single          等价于
Dim i%,s!
```

其中，i 为整型变量，s 为单精度型浮点数变量。

（2）用 Static 语句声明变量。

Static 语句用来在过程中声明静态变量。Static 语句声明变量的格式为：

Static 变量名 As 数据类型

使用 Static 声明的变量称为静态变量。它与 Dim 语句声明的变量不同之处在于：执行一个过程结束时，过程中用 Static 声明的变量的值会被保存下来，下次再调用该过程时，变量的初值是上次调用该过程结束时被保留的值；而用 Dim 语句声明的变量在过程结束时，变量的值不被保留，每次调用时都会被重新初始化。以下，我们将通过一个示例程序来说明 Static 声明与 Dim 声明的区别。

1）运行 VB，在窗体上添加一个 CommandButton 控件。

2）打开代码编辑器，添加 Command1 控件的 Click 事件过程和代码，如下所示：

```
Private Sub Command1_Click()
    Static I As Integer
    Dim J As Integer
    Print Tab(24); "I="; I, "J="; J
    I = I + 1
    J = J + 1
End Sub
```

分析以上程序代码，当第一次调用该过程时，是第一次单击“Command1”按钮，此时系统对 I 和 J 赋以默认值，均为 0。第一个过程结束后，由于变量 I 为静态变量，其值并不释放，被保留；而变量 J 的值被释放，不保留。第二次单击“Command1”按钮，即再次调用控件的 Click 过程，变量 I 的初始值变为 1，变量 J 的初始值仍为 0。按照这个规律，第 n 次调用程序时，I 的初始值为 n-1，而变量 J 的初始值仍为 0。

按 F5 键运行程序，连续单击 10 次“Command1”按钮，如图 3-1 所示。屏幕上显示了变量 I 和 J 数值的变化过程，可以看到，在第十次单击“Command1”按钮时，I 的初始值已经变为 9，而 J 值仍 0。

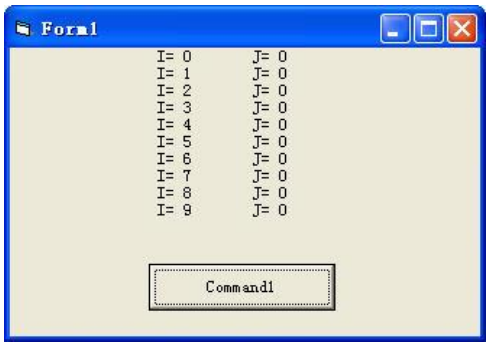


图 3-1　Static 语句与 Dim 语句声明的变量区别示例

（3）用 Public 语句声明变量。

Public 语句用来在标准模块中定义全局变量或数组。Public 语句声明变量的格式为：

Public 变量名 As 数据类型

使用 Public 声明的变量，工程中的所有模块都可以对其引用。如果一个过程或函数使它的值发生了改变，则使用它的其他过程和函数也会受到相应的影响。

以上我们介绍了 Visual Baisc 中定义变量的三种方法。在使用这些方法时，应该注意以下几点：

- 如果一个变量未被显式定义，末尾也没有类型说明符，则隐含地说明为变体（Variant）类型变量。
- 在实际应用中，应该根据需要设置变量的类型。能够使用整型变量的就不要用浮点型或货币型变量；如果要求的精度不高，尽量使用单精度变量。这样不但能够节省内存空间，还能提高处理速度。
- 用类型说明符定义的变量，在使用时可以省略类型说明符。例如，Dim 定义的语句：Dim aStr$

定义了一个字符串变量 aStr$，则可以使用 aStr$来引用这个变量，也可以用 aStr 来引用这个变量。

2. 变量的隐式声明

在 Visual Basic 中，使用变量不要求对变量都事先声明，不加声明的变量默认为变体（Variant）类型。也可以使用类型声明符（%、&、#、!、@、$）来隐含声明变量的数据类型。例如，I%是一个整型变量，C$是一个字符型变量。在程序中不经声明而使用变量，称为变量的隐式声明。隐式声明一般只适用于局部变量，模块级变量和全局变量必须在代码窗口中用 Dim 或 Public 语句显式声明。

对于初学者来说，变量的隐式声明将使其所编写的程序更加难以阅读和理解。因此，为了使程序具有较好的可读性及利于调试，建议尽量避免使用变量的隐式声明。

3.2.3　用户定义的数据类型

在 Visual Basic 中，除了上述的基本数据类型外，还为用户提供了一种自定义的数据类型。自定义数据类型虽然不能产生新的数据类型，但是可以用它来产生现有数据类型的复合类型。它可以将若干个基本数据类型组合起来成为一个整体，以利于引用。自定义数据类型使用 Type 语句来定义。它的典型形式为：

```
Type   数据类型名
       数据类型元素名 As 类型名
       数据类型元素名 As 类型名
       ……
End Type
```

其中“数据类型名”是要定义的数据类型的名字，其命名规则与变量的命名规则相同；“数据类型元素名”也遵守同样的规则；“类型名”可以是任何基本数据类型，也可以是用户定义的类型。

例如，定义一个学生信息的自定义类型：

```
Type   StudentInfo
       Name As String
       StudentNumber As Integer
       Class As String
End Type
```

这里 StudentInfo 是一个用户自定义的类型，它由三个元素组成：Name、StudentNumber 和 Class。其中 Name 和 Class 是字符串型，StudentNumber 是整型。

要引用 StudentInfo 类型变量中的某个元素，可以使用以下形式：

```
变量名.元素名
```

例如可以通过下面的形式分别引用该学生的名字、学号和班级信息：

```
Dim Student as StudentInfo
Student.Name              'Student 变量中的学生名字
Student.StudentNumber     'Student 变量中的学生学号
Student.Class             'Student 变量中学生所在的班级
```

3.2.4　常量

Visual Basic 的常量分为字符串常量、数值常量和符号常量。其中数值常量又分为整数型、长整数型、浮点数和货币型数等四种表示方式。以下将逐一介绍。

1. 字符串常量

字符串常量由字符组成，可以是除双引号和回车符之外的任何 ASCII 字符，其长度不能超过 65535 个字符。例如：

```
"$33322.00"
"学习 VB 编程"
```

2. 数值常量

数值常量共有四种表示方式，即整数型、长整数型、货币型和浮点数。

（1）整型常量。

整型常量有十进制、十六进制和八进制三种表现形式。

十进制整型常量是由一个或几个十进制数字（0～9）组成，可以带有正号或负号，其取值范围为-32768～32767。例如，122 和 221。

十六进制整型常量是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），其取值范围为-&HFFFF～&HFFFF。例如，&H23A。

八进制整型常量是由八进制数字（0～7）组成。前面冠以&（或&0），以&结尾。其取值范围为-&177777&～&177777&。例如，&01234&。

（2）长整型常量。

长整型常量也有十进制、十六进制和八进制三种表现形式。

十进制长整型常量的组成与十进制整型的相同，但是取值范围不同，其取值范围为-2147483648～2147483647。例如，1234567 或-2345678。

十六进制长整型常量是由十六进制数字组成。前面冠以&H（或&h），以&结尾。其取值范围（绝对值）为&H0&～&HFFFFFFFF&。例如，&H1234ABC&。

八进制长整型常量是由八进制数字组成。前面冠以&（或&0），以&结尾。其取值范围（绝对值）为&00&～&037777777777&。例如，&01234&。

（3）浮点型常量。

浮点型常量可以分为单精度浮点数和双精度浮点数。浮点数由尾数、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”和“e”（单精度）、“D”或“d”（双精度），指数必须是整数。其取值范围见表 3-1。指数符号的含义为“乘以 10 的 N 次幂”。例如：

```
4.35E-8
6.243D4
```

其中，4.35 和 6.243 为尾数，E 和 D 为指数符号，它们分别表示 4.35 乘以 10 的-8 次幂（即 4.35×10^{-8}）和 6.243 乘以 10 的 4 次幂（6.243×10^{4}）。

（4）货币型常量。

货币型常量是货币类型数据的常量表现形式。取值范围为-922337203685477.5808～922337203685477.5807。

3. 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。一般格式为：

Const 常量名 = 表达式[，常量名= 表达式]……

其中，“常量名”是一个名字，同变量的命名规则，可以加类型说明符，符号常量习惯上用大写字母进行定义。“表达式”由字符常量、算术运算符（指数运算符“^”除外）、逻辑运算符组成，可以使用字符串，但是不能使用字符串连接符、变量及用户定义的函数或内部函数。例如，使用符号常量 PI 来代替圆周率常数 3.1415926535897，可以按照如下方式定义：

```
Const PI = 3.1415926535897
```

这样，在程序中凡是用到圆周率的地方，都可以用 PI 来代替。符号常量习惯上用大写字母来书写。在使用符号常量的时候，需要注意：

在声明符号常量时，可以在常量名后面加上类型说明符，例如：

```
Const Ten&=10
Const Twenty#=20
```

前者声明为长整型常量，需要 4 个字节；后者声明为双精度常量，需要 8 个字节。如果不使用类型说明符，则根据表达式的求值结果确定常量类型。字符串表达式总是产生字符串常数；对于数值表达式，则按最简单（即占字节数最少）的类型来表示这个常数。例如，当表达式的值为整数，则该常数被作为整型常数处理。

当在程序中引用符号常量时，通常省略类型说明符。例如，可以通过名字 Ten 和 Twenty 引用上面声明的符号常量 Ten&和 Twenty#。省略类型说明符后，常量的类型取决于 Const 语句中表达式的类型。

类型说明符不是符号常量的一部分，定义符号常量后，在定义变量时要慎重。例如，声明了 Const Num=45，则 Num#、Num%、Num@、Num&和 Num！不能再被用于变量名或者常量名。

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，即变量的“可见性”。定义了一个变量后，为了能正确地使用变量的值，应当明确可以在程序的什么地方访问该变量。

前面讲过，Visual Basic 应用程序由三种模块组成，即窗体模块（Form）、标准模块（Module）和类模块（Class）。本书不介绍类模块，因此将着重介绍窗体模块和标准模块。窗体模块包括事件过程（Event Procedure）、通用过程（General Procedure）和声明部分（Declaration），而标准模块由通用过程和声明部分组成，它们之间的关系，如图 3-2 所示。根据定义位置和所使用的变量定义语句的不同，Visual Basic 中的变量可以分为三类，即局部（Local）变量、模块（Module）变量和全局（Public）变量，其中模块变量包括窗体模块变量和标准模块变量。各种变量位于不同的层次，以下将逐一介绍。

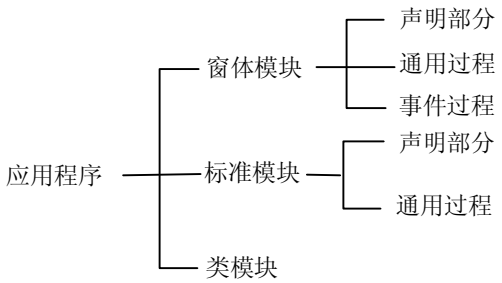


图 3-2　Visual Basic 应用程序的构成

1. 局部变量

在一个过程（事件或者通用过程）内部定义的变量就叫做局部变量，其作用域是它所在的过程。局部变量通常用来存放中间结果或者用作临时变量。某一过程的执行只对该过程的变量产生作用，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。如果需要，可以通过“过程名.变量名”的形式分别引用不同过程中相同名字的变量。

以下将通过一个实例来熟悉局部变量的作用范围。在这个程序里，有一个窗体和两个命令按钮。两个命令按钮的 Click 事件过程中都定义了两个同名的局部变量 I 和 J，我们将验证两个过程中的同名局部变量 I 和 J 是互相独立的。

（1）运行 VB，在窗体上添加两个 CommandButton 控件。

（2）打开代码编辑器，添加 Form1 的 Paint 过程代码，以及 Command1、Command2 控件的 Click 事件过程和代码，如下所示：

```
'窗体的绘画过程
Private Sub Form_Paint()
    Print "      Command1 的 I 和 J 值                    Command2 的 I 和 J 值"
End Sub
"命令按钮 1 的 Click 事件过程，通过点击事件对 I 和 J 变量进行累加
Private Sub Command1_Click()
    Static I As Integer
    Static J As Integer
    Print Tab(5); "I="; I, "J="; J
    I = I + 1
    J = J + 2
End Sub
'命令按钮 2 的 Click 事件过程，通过点击事件对 I 和 J 变量进行累加
Private Sub Command2_Click()
    Static I As Integer
    Static J As Integer
    Print Tab(34); "I="; I, "J="; J
    I = I + 3
    J = J + 4
End Sub
```

分析以上代码，Command1_Click 事件过程中，I 和 J 的值每运行一次，分别增加 1 和 2；而在 Command2_Click 事件过程中，I 和 J 的值每运行一次，分别增加 3 和 4。两个过程的同名局部变量 I 和 J 是分别独立的。按下 F5 键运行程序，轮流单击 Command1 和 Command2 按钮，可以验证以上结果，如图 3-3 所示。

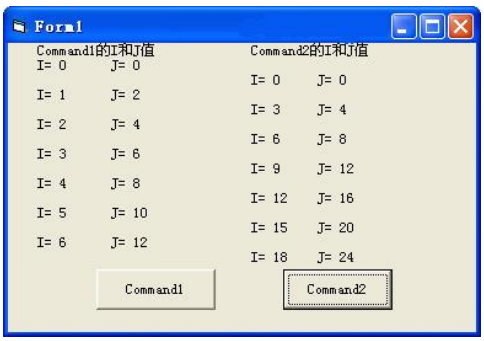


图 3-3　不同过程定义的同名局部变量是相互独立的

2. 模块变量

模块变量包括窗体变量和标准模块变量。

窗体变量可以用于该窗体内的所有过程。一个窗体包含有若干个过程（事件过程或通用过程），这些过程连同窗体一起存入窗体文件（.frm）中。当同一窗体内的不同过程使用相同的变量时，必须定义窗体变量。

在使用窗体变量前，必须先进行声明。窗体变量是不能隐式声明的。其方法是：

在程序代码窗口的“对象”框中选择“通用”，并在“过程”框中选择“声明”，然后就可以在程序代码窗口中声明窗体变量。

标准模块是只含有程序代码的应用程序文件，扩展名为.bas。建立一个标准模块，步骤如下：

（1）执行“工程”菜单中的“添加模块”命令。

（2）在“添加模块”对话框中选择“新建”选项卡。

（3）单击“模块”图标，然后单击“打开”按钮，即可打开标准模块代码窗口，在该窗口中输入代码。

标准模块中模块变量的声明和使用与窗体模块中的窗体变量类似。

在默认情况下，模块变量对该模块中的所有过程都是可见的，但对其他模块中的代码不可见。模块变量在模块的声明部分用 Private 或者 Dim 声明。例如，

```
Private intMoudle As Integer
```

或者

```
Dim intMoudle As Integer
```

在声明模块级变量时，Private 和 Dim 没有什么区别，但是 Private 相对来说好些，因为可以将它与声明全局变量的 Public 区分开来，使得代码具有更强的可读性。

3. 全局变量

全局变量可以被程序中任何一个模块和窗体引用，但它们必须在专门的标准模块文件中用 Public（公用）语句来定义，而不能在窗体中定义。

标准模块由全局变量声明、模块层声明及通用过程等几部分组成。其中全局变量声明放在标准模块的首部，标准模块中的全局变量声明总是在启动时执行。

全局变量声明的一般格式是：

Public 变量名 As 数据类型

如图 3-4 所定义的全局变量

```
Public I As Integer
```

标准模块在编辑完代码后，可以用“文件”菜单中的“保存文件”命令独立存盘，扩展名为.bas。

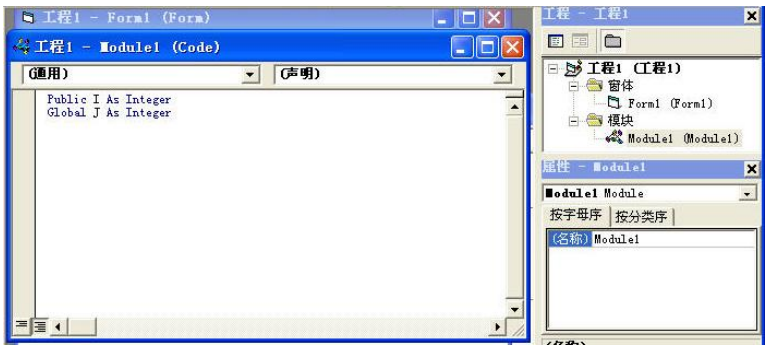


图 3-4　全局变量的定义

3.2.6　变体变量类型

与其他基本数据类型一样，用户也可以定义变体类型（Variant）的变量。

1. Variant 变量的定义

Variant 变量可以用普通数据类型变量的格式定义，也可以采用默认方式定义。例如：

```
Dim Change1 As Variant
Dim Change1
```

以上两种方式都可以把 Change1 定义为 Variant 变量。如果一个变量未经定义而直接使用，则该变量为 Variant 变量。

在 Variant 变量中可以存放任何类型的数据，包括数值、文本字符串、日期和时间。向 Variant 变量赋值时不必进行任何转换，Visual Basic 自动执行必要的转换。例如：

```
'存入有效字符串
Change1 = "100"
'Change1 变为数值 50
Change1 = Change1 - 50
'Change1 变为字符串 123A50
Change1 = "123A" + Change1
```

从以上代码可以看出，随着赋值的不同，变体变量的类型也在不停变化，这就是“变体数据类型”的含义。

Variant 变量用起来很方便，因为在对它赋值的时候，不需要考虑类型转换的问题。但是它的使用也存在以下一些问题：

如果对 Variant 变量进行算术运算，则需要保证变量中存放的是某种形式的数值，包括整数、浮点数、定点数或可以解释为数值的字符串。如果 Variant 变量中的内容是 ABC，则不能对其进行算术运算，因为 ABC 不是有效的数值。

运算符“+”既可以用于数值相加，也可以用于字符串连接，当在两个 Variant 变量之间使用“+”运算时，其结果将取决于两个变量的内容。为了避免这种情况的出现，在进行字符串连接时，最好使用“&”。

2. Variant 变量中的数值

在 Variant 变量中存放数值时，Visual Basic 以紧凑的方式存储。如果是较小的整数，则以 Integer 类型存储，而较大的或带有小数部分的数值则用 Long 类型或 Double 类型存储。

如果需要用指定的类型来存储 Variant 变量的值，则必须用类型转换函数。如果 Variant 变量中存放的不是数值或可以解释为数值的内容（如日期/时间或含有数字的字符串），则对其进行算术运算或函数运算时会发生错误。因此，应当在运算前对 Variant 变量中的值进行判断，这可以通过 IsNumeric 函数来实现。例如：

```
If IsNumeric(x) Then x = x + 1
```

上述语句的含义是，如果 Variant 变量 x 是一个数值，则执行 x=x+1。

3. Variant 变量中的空值

Variant 变量在被赋值前为空值（内部表示为 Empty 或者 0），它不同于数值 0，不同于空的字符串（" "），也不同于 NULL，通过 IsEmpty 函数可以判断一个变量自声明以来是否被赋值过：

```
If IsEmpty(Y) then Y=0
```

当 Variant 变量为空值时，可以用在表达式中，Visual Basic 将根据具体情况来解释为数值 0 或者空字符串。如果将一个空值 Variant 变量赋值给一个非空值 Variant 变量，后者将变为空值。

3.3 常用内部函数

Visual Basic 提供了大量的函数与语句。在这些函数中，有些是通用的，有些则与某种操作相关，大体上可以分为 5 类：数学函数、转换函数、字符串函数、日期和时间函数、随机数函数。以下将简要介绍这些函数的功能。

3.3.1　数学函数

数学函数可以用来进行一些基本的数学运算，如求三角函数、绝对值、平方根、对数、指数以及对数值进行取整处理或者符号处理等。表 3-2 列出了 Visual Basic 中常用的数学函数。

表 3-2　常用的数学函数

函数	功能
Sin(x)	返回自变量 x 的正弦值，x 为弧度值
Cos(x)	返回自变量 x 的余弦值，x 为弧度值
Tan(x)	返回自变量 x 的正切值，x 为弧度值
Atn(x)	返回自变量 x 的反正切值，单位为弧度
Abs(x)	返回自变量 x 的绝对值
Sgn(x)	返回自变量 x 的符号，即当 x 为负数时，返回-1；当 x 为 0 时，返回 0；当 x 为正数时，返回 1
Sqr(x)	返回自变量 x 的平方根，x 必须大于或等于 0
Exp(x)	返回以 e 为底，以 x 为指数的值，即求 e 的 x 次方
Fix(x)	返回参数的整数部分（向上取整）
Log(x)	返回参数的绝对数值
Int(x)	返回参数的整数部分（向下取整）
Round(x)	四舍五入取整

说明：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，可以用以下公式转换为弧度：1 度=π/180=3.14159/180（弧度）。

例 3.2　求下列函数的值。

Abs(-3.5)　　结果为：3.5
sqr(9)　　结果为：3
sqr(-4)　　出错，参数不能为负数
sin60°　　Sin(3.14159/180*60)
Fix(-9.6)　　结果为：-9
Int(-9.6)　　结果为：-10
Round(3.5)　　结果为：4

3.3.2　随机数函数

随机数函数 Rnd(x)主要用来产生一个随机数，其中参数 x 是一个实型数，可以省略。Rnd(x)函数产生一个 0～1 之间（包括 0 但不包括 1）的单精度随机数。每一次要产生的随机数受参数 x 的影响。具体影响情况如下：

当 x<0 时，每次产生的随机数相同。

当 x=0 时，所产生的随机数与上次产生的随机数相同。

当 x>0 或者省略时，产生下一个随机数。

如果需要产生一个随机整数，可以通过把随机数乘以一个整数求得。例如，用语句"Int(Rnd*整数)+1"可以产生 1～“整数”范围内的随机数。

在 Visual Basic 中，与 Rnd(x)函数配套使用的还有一个 Randomize(x)函数。Randomize(x)

函数可以消除 Rnd(x)函数使用时重复出现同一序列随机数的现象。参数 x 是一个整型数，它是随机数发生器的“种子数”，可以省略。如果省略，则 Visual Basic 取 Timer 整数（返回从午夜开始到现在经过的秒数）的时间值作为新随机数的种子数。由于内部时钟在不停地变化，所以每次执行时随机数种子数也不相同，从而可以产生不同的随机数序列。如果给出种子数（x 不省略），则产生与 x 对应的一个特定序列的随机数。

产生一定范围内的随机数，公式为：

Int(Rnd*范围+基数)

例如：产生［30～50］之间的随机数

Int(Rnd*21+30)

3.3.3 转换函数

转换函数主要用于类型或者形式的转换，如将十进制转换成十六进制，将字符转换成对应的 ASCII 码等。表 3-3 列出了 Visual Basic 中常用的转换函数。

表 3-3 常用的转换函数

函数	功能
Lcase(x)	将大写字母转换成小写字母，已符合要求的字母保持不变
Ucase(x)	将小写字母转换成大写字母，已符合要求的字母保持不变
Hex$(x)	把一个十进制数转换为十六进制数
Oct$(x)	把一个十进制数转换为八进制数
Asc(x$)	返回字符串 x$中第一个字符的 ASCII 字符
Chr$(x)	把 x 的值转换为相应的 ASCII 字符
Str$(x)	把 x 的值转换为一个字符串
Cint(x)	把 x 的小数部分四舍五入，转换为整数
Ccur(x)	把 x 的值转换为货币类型值，小数部分最多保留四位，且自动四舍五入
CDbl(x)	把 x 值转换为双精度数
CLng(x)	把 x 的小数部分四舍五入转换为长整型数
CSng(x)	把 x 值转换为单精度数
Cvar(x)	把 x 值转换为变体类型值
Val(x)	把字符串转换成数值

例 3.3 求下列函数的值。

```
Asc("A")            结果：65
Chr(97)             结果：a
Asc(Chr(122))       结果：122
Asc("Abcd123")      结果：65
Asc("asdf")         结果：97
str(256)            "_256"
str(-256.90000)     "-256.9 "
val("1.2sa10")      1.2
val("abc123")       0
val("-1.2e3eg")     -1200
val("-1.2ee3eg")    -1.2
```

3.3.4　字符串函数

字符串函数主要用于对字符串进行操作和处理。例如，取得字符串长度、删除字符串中的空格以及截取字符串等。表 3-4 列出了 Visual Basic 中常用的字符串函数。

表 3-4　常用的字符串函数

函数	功能
LTrim$(字符串)	去掉字符串左边的空格
RTrim$(字符串)	去掉字符串右边的空格
Left$(字符串,n)	取字符串左边的 n 个字符
Right$(字符串,n)	取字符串右边的 n 个字符
Mid$(字符串,p,n)	从位置 p 开始取字符串的 n 个字符
Len(字符串)	测试字符串的长度
String$(n,字符串)	返回由 n 个字符组成的字符串
Space$(n)	返回 n 个空格
InStr(字符串 1,字符串 2)	在字符串 1 中查找字符串 2

例 3.4　求下列函数的值

```
Left("ABCDE",2)              结果为："AB"
Right("ABCDE",2)             结果为："DE"
Mid("ABCDE",2,3)             结果为："BCD"
Len("ABCDE")                 结果为：5
String(3, "ABC")             结果为：AAA
Instr("ABCDEFG", "CDE")      结果是：3
```

3.3.5　日期和时间函数

日期和时间函数主要用于对日期和时间进行处理。表 3-5 列出了 Visual Basic 中常用的日期和时间处理函数。

表 3-5　日期和时间处理函数

函数	功能
Day(Now)	返回当前的日期
WeekDay(Now)	返回当前的星期
Month(Now)	返回当前的月份
Year(Now)	返回当前的年份
Hour(Now)	返回小时（0～23）
Minute(Now)	返回分（0～59）
Second(Now)	返回秒（0～59）

3.4　运算符与表达式

运算（即操作）是对数据的加工。最基本的运算形式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，被运算的数据称为运算量或操作数。由运算符和操作数组成的表达式描述了对哪些数据，以何种顺序进行什么样的操作。操作数可以是常量也可以是变量，还可以是函数。例如：I+1、J+Cos(x)、X=I+J、PI*r*r 等均为表达式，此外单个变量或常量也可以看成是表达式。

Visual Basic 提供了丰富的运算符，可以构成多种表达式。

3.4.1 算术运算符

算术运算符是最常用的运算符，用来执行简单的算术运算。Visual Basic 提供了 8 种算术运算符，见表 3-6。

表 3-6 算术运算符

算术运算	运算符	表达式例子	优先级
指数运算	^	X^Y	1
取负运算	–	–X	2
乘法运算	*	X*Y	3
浮点除法运算	/	X/Y	3
整数除法运算	\	X\Y	4
取模运算	Mod	X Mod Y	5
加法运算	+	X+Y	6
减法运算	–	X-Y	6

在 8 种运算符中，除取负运算符是单目运算符外，其他均为双目运算符（即需要两个操作数）。加、减、乘、除运算符与数学中的含义基本相同，下面介绍其他几种运算符的操作。

1. 指数运算

指数运算用来计算乘方和方根，其运算符为^，2^3 表示 2 的 3 次方，而 2^(1/3)是计算 2 的 3 次方根。当指数是一个表达式时，必须加上括号，如 x 的 y+z 次方，必须写作 x^(y+z)，而不能写成 x^y+z，后者的含义是 x 的 y 次方值与 z 相加。例：5^2 的结果为 25。

2. 浮点数除法与整数除法

浮点数除法运算符“/”执行标准除法操作，其结果为浮点数，例如，表达式 3/2 的结果是 1.5，与数学中的除法一样。整数除法运算符“\”执行整除运算，结果为整型值。因此，表达式 3/2 的结果是 1，而不是 1.5。整数除法的操作数一般是整型值，当操作数带有小数时，首先被四舍五入为整型数或长整型数，然后进行整除运算，运算结果被截断为整型数或长整型数，不进行四舍五入处理。例：10/3 的结果为：3.33333；10\3 的结果为：3。

3. 取模运算

取模运算符 Mod 又称为求余运算符，其结果为第一个操作数整除第二个操作数所得的余数。例如，如果用 7 除 4，余数为 3，则 7 Mod 4 的结果为 3。再如表达式 13.3 Mod 2.99，首先通过四舍五入把 13.3 和 2.99 分别变为 13 和 3，故上式的结果为 1。

4. 运算符的优先级

Visual Basic 规定了运算符的优先级和结合性，在表达式求值时，先按运算符的优先级高低次序执行。在表 3.6 中列出的 8 种运算符中，其优先级从上往下依次降低，其中乘和浮点除是同级运算符，加和减是同级运算符。当一个表达式中含有上述多种运算符时，必须严格按照

上述顺序求值，如先乘除后加减，在表达式 a-b*c 中，左侧为减号，右侧是乘号，而乘号优先于减号，因此，该表达式相当于 a-(b*c)。此外，如果表达式中含有括号，则先计算括号中的表达式；有多层括号时，先计算内层括号内表达式的值。

3.4.2　关系运算符和逻辑运算符

关系运算符也称为比较运算符，用来对两个表达式的大小进行比较，比较的结果是一个逻辑值，即真（True）或假（False），用关系运算符将关系表达式或逻辑量连接起来就是关系表达式。Visual Basic 提供了 6 种关系运算符，见表 3-7。

表 3-7　关系运算符

运算符	测试关系	表达式例子
=	等于	X=Y
<>	不等于	X<>Y
<	小于	X<Y
>	大于	X>Y
<=	小于或等于	X<=Y
>=	大于或等于	X>=Y
Like	比较样式	Visual Basic 6.0 新增的比较符
Is	比较对象变量	Visual Basic 6.0 新增的比较符

用关系运算符连接的两个算术表达式所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布尔值，即 True 或 False。Visual Basic 把任何非 0 值都认为是“真”，但一般以“-1”表示真，以“0”表示假。关系运算符既可以进行数值的比较，也可以进行字符串的比较。

关于优先次序，前两种关系运算符(=，<>)的优先级别相同，后 4 种关系运算符(<，>，<=，>=)的优先级别也相同。但前两种关系运算符的优先级别低于后四种关系运算符。

关系运算符的优先级低于算术运算符。

关系运算符的优先级高于赋值运算符(=)。

字符串比较，则按字符的 ASCII 码值从左到右一一比较，直到出现不同的字符为止。例：

"ABCDE" > "ABRA"　　结果为 False

3.4.3　逻辑运算符

逻辑运算也称为布尔运算，其作用是将操作数进行逻辑运算，结果是逻辑值 True（真）或 False（假）。用逻辑运算符将关系表达式或逻辑量连接起来的式子称为逻辑表达式，也称布尔表达式。Visual Basic 提供的逻辑运算符有 5 种，见表 3-8。

表 3-8　逻辑运算符

运算符	说明
Not	非，由真变假或由假变真
And	与，两个表达式同时为真时值为真，否则为假
Or	或，两个表达式都为假时值为假，否则为真
Xor	异或，两个表达式同时为真或同时为假，值为假，否则为真
Eqv	等价，两个表达式同时为真或同时为假，值为真，否则为假

3.4.4 字符串运算符

字符串运算符有两个：“&”和“+”，它们都是将两个字符串拼接起来。注意：变量与&之间要加一个空格。

例 3.5

```
"12" + "456"      结果 "123456"
"12" & "456"      结果 "123456"
```

区别：“+”两边的操作数必须是字符串，只要有一个不是字符串，则进行加法运算。当进行加法运算时，如无法转换为数值型，则出错。“&”不管两个操作数是什么类型都将进行连接运算。

例 3.6

```
"abcdef" + 12345          出错
"abcdef" & 12345          结果为 "abcdef12345"
"123" + 456               结果为  579
"123" &  456              结果为 "123456"
```

3.4.5 表达式

在 Visual Basic 中，与运算符相对应，表达式有算术表达式、关系表达式和逻辑表达式三种。

算术表达式由算术运算符、数值型常量、变量、函数和圆括号组成，其运算结果为数值。在算术表达式中，如果操作数具有不同的数据精度，那么按照 Visual Basic 的规定，运算结果的数据类型采用精度高的数据类型。还可由字符串常量、字符串变量、字符串函数和连接运算符组成算术表达式中的特例——字符串表达式。

关系表达式是用关系运算符将两个表达式连接起来，并对两个表达式的值进行比较的式子，比较的结果是一个布尔值（True 或 False）。

逻辑表达式是用逻辑运算符连接若干个关系表达式或布尔值构成的式子，运算结果是一个布尔值（True 或 False）。

3.4.6 表达式的执行顺序

一个表达式可能含有多种运算，计算机按照一定的顺序对表达式进行求值。一般运算顺序如下：

（1）先进行函数运算。

（2）然后进行算术运算。算术运算的顺序为：

幂（^）→取负（-）→乘、浮点除（*、/）→整除（\）→取模（Mod）→加、减（+、-）→连接（&）。

（3）进行关系运算（=、>、<、<>、<=、>=）。

（4）进行逻辑运算，顺序为：Not→And→Or→XOR。

在进行运算的时候，需要注意以下几点：

- 当乘法和除法同时出现在表达式中时，按照它们从左到右出现的顺序进行计算。用括号可以改变表达式的优先顺序，强制某些低级的运算优先执行。括号内的运算总是优先于括号外的运算。

- 字符串连接运算符（&）不是算术运算符，它的优先顺序位于所有算术运算符之后，而在所有关系运算符之前。
- Like 的优先顺序与所有关系运算符都相同，实际上是模式匹配运算符。Is 运算符是对象引用的关系运算符。它并不将对象或者对象的值进行比较，而只是确定两个对象引用是否参照了相同的对象。
- 上述操作顺序有一个例外，就是当幂和负数相邻时，负号优先。
- 乘号（*）不能省略，也不能用“·”代替。
- 在一般情况下，不允许两个运算符相连，应当用括号隔开。
- 括号可以改变运算顺序。在表达式中只能使用圆括号，不能使用方括号或大括号。
- 幂运算符表示自乘，如 A^B 表示 A 的 B 次方，即 B 个 A 连乘。当 A 和 B 不是单个常量或者变量时，用括号括起来，例如(A+B)^(C+D)。

3.5　常用语句

3.5.1　赋值语句

赋值语句是程序设计中最基本、最常用的语句。用赋值语句可以把指定的值赋给某个变量或者带有属性的对象。赋值语句的一般使用格式有以下三种：

1．给变量赋值

该过程是将右边表达式的值赋给左边的变量，形式如下：

变量=表达式

例如：

```
Dim A As Integer
Dim B As Integer
A = Val("10")   '将字符串"10"转化为数值
B = 3.14159   'B 为整型变量，系统进行强制转换时，自动将浮点数进行四舍五入，结果 B 的值为 3
```

2．为对象的属性赋值

在 Visual Basic 应用程序的设计中，可以在程序中用赋值语句为对象的属性设置属性值。形式如下：

对象名. 属性名=属性值

例如，可以为命令按钮 Command1 的 Caption 属性设置一个新值：

```
Command1.Caption = "显示"
```

也可以把数值变量 A 转换为字符串赋给带有 Text 属性的对象：

```
Text1.Text=Str$(A)
```

3．为用户自定义类型声明的变量的各元素赋值

为用户自定义类型声明的变量的各元素赋值，其形式如下：

变量名. 元素名=表达式

例如，本章节中所定义的自定义类型 StudentInfo，先进行变量声明"Dim student as StudentInfo"，定义了 StudentInfo 类型的自定义变量后，就可以通过以下赋值语句给 student 变量中的 Name（姓名）成员赋值了：

```
student.Name = "李四"
```

3.5.2 注释语句

为了便于对程序的阅读和理解，通常编程人员会在程序的适当位置加上必要的注释。Visual Basic 中的注释语句以 Rem 或'开头，之后紧跟注释内容。其一般格式为：

Rem 注释内容
'注释内容

例如：

```
'This is a rem
Rem 这是一个注释内容
```

注释语句是非执行语句，用来对程序的有关语句进行注释。虽然在程序清单中注释内容会被完整的列出，但它不被解释和编译。任何字符（含中文字符）都可以放在注释行中作为注释内容。注释语句一般放在过程或模块的开头作为标题用，可以放在一些执行语句的后面作为备注。放在执行语句后面时，注释语句必须是最后一个语句。此外还需要注意的是，注释语句不能放在续行符的后面。

3.5.3 暂停语句

暂停语句是用来暂停程序的执行，也就是 Stop 语句。暂停语句的作用类似于执行“运行”菜单中的“中断”命令。当执行暂停语句时，将自动打开立即窗口。暂停语句的使用格式如下：

Stop

在 Visual Basic 的解释系统中，暂停语句保持文件打开，并且不退出 Visual Basic 开发环境。因此，可以在调试程序时使用 Stop 语句设置断点，以便对程序进行检查和调试。但是，如果在可执行文件（扩展名为.exe）中含有 Stop 语句时，所有文件都将关闭。因此，当一个应用程序通过编译并且能够正常运行、不需要再进入中断模式时，最好删去程序源代码中的所有 Stop 语句，然后再编译程序，并生成新的可执行文件。

3.5.4 结束语句

结束语句通常用来结束一个程序的执行，即 End 语句。结束语句的使用格式如下：

End

例如：

```
Private Sub Command1_Click()
   End
End Sub
```

Command1 控件的 Click 过程用来结束程序，即单击 Command1 控件时，程序将结束。

结束语句除了用来结束程序外，还可以用于其他一些方面。例如，用于结束一个 Sub 过程（End Sub）、结束一个 Function 过程（End Function）、结束一个 If 语句块（End If）、结束自定义类型的定义（End Type）和结束情况选择语句（End Select）等。

习题三

一、选择题

1．执行语句 Dim x, y As Integer 后，（　　）。（2009.3）

A．x 和 y 均被定义为整型变量

B．x 和 y 均被定义为变体类型变量

C．x 被定义为整型变量，y 被定义为变体类型变量

D．x 被定义为变体类型变量，y 被定义为整型变量

2．以下关系表达式中，其值为 True 的是（　　）。（2009.3）

A．"XYZ">"XYz"　　B．"VisualBasic"<>"visualbasic"

C．"the"="there"　　D．"Integer"<"Int"

3．执行以下程序段

```
a$ = "Visual Basic Programming"
b$ = "C++"
c$ = UCase(Left(a, 7) & b & Right(a, 12))
```

后，变量 C 的值为（　　）。（2009.3）

A．Visual BASIC Programming

B．Visual C++ Programming

C．VISUAL C++ PROGRAMMING

D．VISUAL BASIC PROGRAMMING

4．若变量 a 未事先定义而直接使用（例如：a = 0），则变量 a 的类型是（　　）。（2008.9）

A．Integer　　B．String　　C．Boolean　　D．Variant

5．表达式 2*3^2+4*2/2+3^2 的值是（　　）。（2008.09）

A．30　　B．31　　C．49　　D．48

6．为把圆周率的近似值 3.14159 存放在变量 pi 中，应该把变量 pi 定义为（　　）。（2008.9）

A．Dim pi As Integer　　B．Dim pi(7) As Integer

C．Dim pi As Single　　D．Dim pi As Long

7．在 Visual Basic 中，表达式 3*2\5 mod 3 的值是（　　）。（2008.4）

A．1　　B．0　　C．0　　D．出现错误提示

8．以下选项中，不合法的 Visual Basic 的变量名是（　　）。（2008.4）

A．a5b　　B．_xyz　　C．a_b　　D．andif

9．下列可以正确定义 2 个整型变量和 1 个字符串变量的语句是（　　）。（2007.4）

A．Dim n, m As Integer, s As String

B．Dim a%, b$, c As String

C．Dim a As Integer, b, c As String

D．Dim x%, y As Integer, z As String

10．下列表达式中不能判断 x 是否为偶数的是（　　）。（2007.4）

A．x/2=Int(x/2)　　B．x Mod 2 = 0

C．Fix(x/2) = x/2　　D．x\2 = 0

11．设 a=2，b=3，c=4，下列表达式的值是（　　）。（2006.9）

```
Not a <= c Or 4 * c = b ^ 2 And b <> a + c
```

A．-1　　B．1　　C．True　　D．False

12．有下列用户定义类型：

```
Type student
   number As String
```

```
    name As String
    age As Integer
End Type
```

则下列正确引用该类型成员的代码是（　　）。（2006.9）

A．student.name = "李明"

B．Dim s As student
　　s.name = "李明"

C．Dim s As Type student
　　s.name = "李明"

D．Dim s As Type
　　s.name = "李明"

二、填空题

1．描述“X 是小于 100 的非负整数”的 Visual Basic 表达式是_____。（2006.9）

2．下列语句的输出结果是_____。（2006.4）

```
Print Format(Int(12345.6789 * 100 + 0.5) / 100, "00,000.00");
```

3．Print DateDiff("m", #2002/09/24#, #2002/09/25#)输出结果为_____。