第 2 章　VB 语言基础

VB 程序除了界面之外，还应该包含数据描述和数据操作两个方面的内容。在编写程序时，首先需要考虑如何描述数据，并为它们在内存安排存储空间，然后考虑采用恰当的控制结构和表达式，对数据进行操作。

本章介绍 VB 语言的数据类型，讲解常量、变量、表达式和语句等程序的基本元素，最后介绍窗体，为编写 VB 程序打下基础。

2.1　数据类型

2.1.1　基本数据类型

计算机中的数据是现实世界中信息的具体表现形式，具有一定的数据类型，数据类型确定了数据的取值范围和能够进行的操作。在计算机的存储器中，不同类型的数据所占存储空间的长度也有所不同。数据不仅是程序处理的对象，也是计算机运算产生的结果。程序员在编程解决某一问题时，必须要为数据设计合适的数据类型，这样才能合理地存储和处理数据。

VB 语言的数据类型十分丰富，在数值计算和文本处理方面功能很强。图 2-1 列出了 VB 提供的基本数据类型，其中字节型、整型和长整型用于描述整数，而单精度型、双精度型和货币型用于描述实数。VB 还允许程序员以基本数据类型为基础，自定义新的数据类型。

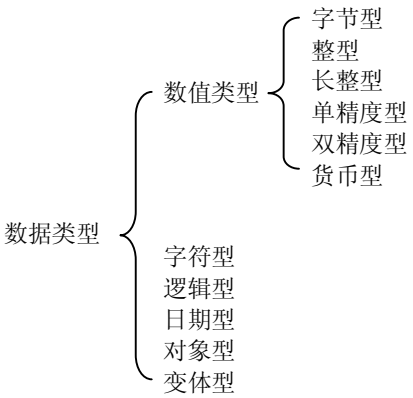

图 2-1　VB 语言的数据类型

2.1.2　标识符

在 VB 程序中经常会出现许多符号，这些符号分别代表不同的含义。VB 语言常见的符号主要有关键字和标识符。

1. 关键字

关键字又称保留字，是 VB 语言预先规定的具有固定含义的一些单词，例如 Integer、If

和 While 等。程序员只能按预先规定的含义来使用它们，不能擅自改变其含义。

2. 标识符

标识符是程序员给程序中的实体所取的名字，这些实体可以是变量、常量、数组、函数和控件对象等。VB 语言标识符的命名规则是：以字母开始，由字母、下划线和数字组成。例如 sum、n_i2 和 A1 是合法的标识符，而 di$s 和 2day 则是非法的标识符。

说明：标识符不能与关键字同名。在标识符中并不区分字母的大小写，例如 VB 把 a1 和 A1 看作是同一个标识符。

2.2 常量与变量

2.2.1 常量

不同类型的数据在程序中既可以以常量的形式出现，也可以存放在相应的变量中。常量是指在程序执行期间其值不发生变化的量，变量的字面含义是指在程序执行期间其值可以变化的量，实际上对应了内存的一段存储空间。常量有不同的数据类型，它可以分为直接常量和符号常量。直接常量是指可以从字面直接识别的常量，符号常量则是指用标识符描述的常量。

1. 整型常量

整型常量包括字节型、整型和长整型，有十进制、八进制和十六进制三种形式。具体说明如下：

（1）十进制整数。例如：123、-270、0。

（2）八进制整数，以&或者&O 作为前缀。例如：&123 和&O123 都表示八进制整数$(123)_8$，所对应的十进制整数为 $1\times8^2+2\times8^1+3\times8^0=83$。

（3）十六进制整数，以&H 作为前缀。例如：&H123 表示十六进制整数$(123)_{16}$，所对应的十进制整数为 $1\times16^2+2\times16^1+3\times16^0=291$。

如果在一个整型数据的尾部加上&，则表示长整型常量，例如 456&。

2. 实型常量

实型常量包括单精度型、双精度型和货币型，有定点和指数两种形式。具体说明如下：

（1）定点形式，由数字和小数点组成。例如：3.2556、0.289、-458.、.899。

（2）指数形式，由尾数、指数符号（E）和指数组成。要求尾数不能省略，指数是整数。例如：32.6E+2 或 32.6E2 都表示 32.6×10^2。

如果在一个实型数据的尾部加上#，或者用指数符号（D）代替指数符号（E），则表示双精度型常量。例如：3.2556#、32.6D2、32.6E2#。

如果在一个实型数据的尾部加上@，则表示货币型常量，例如 210.8@。货币型常量用于表示金额，其精度是整数部分最多保留 15 位，小数部分最多保留 4 位。

3. 字符型常量

字符型常量又称为字符串，是由一对双引号括起来的字符序列，例如"CHINA"、"Mp3"和"集结号"等。字符串的长度是指字符串中字符的个数，""是空串，表示不包含任何字符，其长度为 0。

需要指出的是，计算机只能存储二进制数据，无法直接表示字符型数据，因此通常采取

编码的方式来处理。常用的是 ASCII 编码，从 0 开始，共有 256 个，可以表示英文字母、标点符号等字符。VB 采用的是 Unicode 编码，用两个字节表示一个字符，每一个字符对应一个 Unicode 码，汉字也有自己的 Unicode 码。为了与 ASCII 编码兼容，从 0 开始的 256 个 Unicode 码所表示的字符与相应的 ASCII 码所表示的字符完全相同。例如换行符的 Unicode 码是 10，字符 0 的 Unicode 码是 48，大写字母 A 的 Unicode 码是 65，小写字母 a 的 Unicode 码是 97。

思考：0 和"0"是同一个常量吗？"A"和"a"是两个相同的字符串吗？

4. 逻辑型常量

逻辑型常量只有 True 和 False 两个值，分别表示“真”和“假”。

思考：True 和"True"是同一个常量吗？

5. 日期型常量

日期型常量由一对“#”括起来，表示日期和时间。它有多种形式，例如#4/29/08#和#2008-4-29#均表示 2008 年 4 月 29 日，#15:20:30#和#8/8/08 8:8:0 PM#也都是合法的日期型常量。为避免引起不必要的混乱，建议读者尽量采用如下的标准格式：

```
#月/日/年 时:分:秒 AM|PM#
```

说明：AM|PM 表示两个选项任选其一。

思考：#08-4-29#表示哪一天？

6. 符号常量

如果在程序中多次出现某个常量，则可以定义符号常量以取代该数据。这样做不仅增加了程序的可读性，而且也便于维护。定义符号常量的一般格式如下：

```
Const 符号常量[As 类型]=表达式
```

说明：[As 类型]表示[ ]中的内容为可选项。

例如：

```
Const PI As Single=3.14159
```

定义了一个单精度型符号常量 PI，在程序中出现标识符 PI 即表示常量 3.14159。

思考：如果需要 PI 代表 3.1415926，应如何修改程序？

实际上 VB 语言也提供了很多符号常量，它们均以 vb 开头，程序员可以在程序中直接使用。例如 vbCr 是格式控制常量，表示回车符，vbLf 表示换行符，而 vbCrLf 则表示回车/换行符；vbRed 是颜色常量，表示红色；vbMaximized 是窗口状态常量，表示窗口最大化。

2.2.2　变量

变量实际代表了内存中某一段存储空间，其中可以存放数据即变量的值，存储空间的大小则由变量的数据类型来决定。变量有名字，程序员在程序中可以通过变量名访问变量所对应的内存空间。如图 2-2 所示，a 是变量名，3 是变量 a 的值。变量类似于现实生活中的抽屉、书包等容器，人们能够在其中随意放置物品（即数据），而每次存放的具体物品可以变化，变量也由此得名。

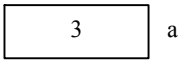

图 2-2　变量示意图

VB 各种类型变量的基本情况如表 2-1 所示。从表中可以发现，变量的取值范围是有限的，而且其所占内存的字节数越多，相应的取值范围就越大。

表 2-1　VB 基本类型的变量

类型	关键字	类型符	所占字节数	取值范围
字节型	Byte		1	0～255
整型	Integer	%	2	-32768～32767
长整型	Long	&	4	-2147483648～2147483647
单精度型	Single	!	4	-3.4×10^{38} ～$+3.4\times10^{38}$
双精度型	Double	#	8	-1.7×10^{308} ～$+1.7\times10^{308}$
货币型	Currency	@	8	-922337203685477.5808～922337203685477.5807
逻辑型	Boolean		2	True 或 False
字符型	String	$	字符串的长度	
日期型	Date		8	100 年 1 月 1 日～9999 年 12 月 31 日
对象型	Object		4	
变体型	Variant			

在 VB 程序中，所有的变量在使用之前一般要先定义。变量定义主要是指出变量的名称，确定变量的类型。变量定义语句的格式如下：

```
Dim 变量 1 As 类型[,变量 2 As 类型,…]
```

例如：

```
Dim a As Integer, b As Single, c As String
```

一共定义了 3 个变量，分别是整型变量 a、单精度型变量 b 和字符串变量 c。如果定义变量时在其尾部附上类型符，则可以省略类型说明部分。上一条语句也可以写为以下的等价形式：

```
Dim a%, b!, c$
```

思考：40000 能否存放在整型变量 a 中？

在同一个作用域中，变量不允许被重复定义。当某个变量被定义之后，VB 会为其分配相应长度的内存空间，并进行初始化。数值型变量的初值是 0，字符串变量的初值是空串，而逻辑型变量的初值是 False。

字符串变量一般能够存放不固定长度的字符串，也可以在程序中定义定长的字符串变量。例如：

```
Dim s As String * 20
```

定义了一个长度固定为 20 的字符串变量 s。如果存放在定长字符串变量中的字符数小于给定长度，系统会自动用空格在字符串的后面予以填补；如果大于给定长度，系统则会自动截去超出部分的字符。

如果变量未经定义而直接使用，或者在定义时没有进行类型说明，则系统默认该变量为变体型（Variant）。变体型变量在使用时并不安全，建议在程序中尽量不要采用。

思考：如果要定义两个 Double 型变量 d1 和 d2，使用语句 Dim d1,d2 As Double 可以吗？

2.3　运算符与表达式

运算符用于对数据进行运算，被运算的数据称为操作数。表达式描述对哪些数据以什么顺序施以什么样的操作，它由运算符和操作数组成。操作数既可以是常量，也可以是变量，还可以是函数调用。

VB 语言的运算符按功能来分，常用的有算术运算符、关系运算符、逻辑运算符和赋值运算符等。运算符也可以根据运算所需操作数的个数进行分类，只需一个操作数的运算符称为单目运算符，需要两个操作数的运算符称为双目运算符。学好运算符要注意以下几点：

（1）运算符的功能。

（2）运算符的优先级。

（3）运算符所需的操作数个数和类型。

本节重点介绍算术表达式，在第 4 章将介绍关系表达式和逻辑表达式。

2.3.1　算术表达式

1. 算术运算符

VB 语言的算术运算符用于实施加、减、乘、除等常见的数值计算，其操作数通常是数值型的数据，如表 2-2 所示。除了取负运算符为单目运算符之外，其他都是双目运算符。

表 2-2　算术运算符

运算符	优先级	功能
^	1	指数（幂运算）
-	2	取负
*	3	乘
/	3	除
\	4	整除
Mod	5	取余
+	6	加
-	6	减

表 2-2 按优先级由高到低的顺序，依次列出了 8 个算术运算符。指数运算符（^）的优先级最高，而加（+）和减（-）的优先级最低。其中乘（*）和除（/）是同级运算符，加（+）和减（-）也是同级运算符。

说明：

（1）整除（\）运算是取整数相除的商，取余（Mod）运算是取整数相除的余数，这两种运算的操作数都要求是整型数据。例如 1\2 的值是 0，1 Mod 2 的值是 1。如果操作数是实数，则自动按四舍五入的原则转换成整数，再进行运算。例如 7.4\3.8 的值是 1，7.4 Mod 3.8 的值是 3。

（2）除（/）与整除（\）不同，它是针对实数的除法运算。例如 1/2 的值是 0.5，5.4/1.2

的值是 4.5。

（3）指数（^）运算的幂次既可以是整数，也可以是实数。例如 2^3 的值是 8，8^(1/3)相当于对 8 开立方，它的值是 2。

2. 算术表达式

用算术运算符和括号将操作数连接起来，构成符合 VB 语言规则的式子，称为算术表达式。应从左边开始计算一个表达式的值，如果遇到括号，就先计算括号中的内容；如果出现不同类型的运算符，则按照当前优先级的高低顺序依次计算。例如表达式 4*6 Mod 9+4\3，先计算 4*6，值为 24；再计算 24 Mod 9，值为 6；然后计算 4\3，值为 1；最后计算 6+1，整个表达式的值为 7。

思考：表达式 4*6 Mod (9+4\3)的值是什么？

如果参加算术运算的操作数具有不同的数据类型，为保证数据运算的精度，VB 规定运算结果的数据类型以高类型为准。所谓高类型，是指其所占内存的字节数较多。例如 Integer 型数据和 Double 型数据进行运算，则运算结果的数据类型为 Double 型。

2.3.2 字符串表达式

连接运算符（&）用来连接两个字符串，它的优先级低于算术运算符，例如"Visual"&" Basic"构成了一个字符串表达式，其值是"Visual Basic"。加（+）也可以用来连接字符串，例如字符串表达式"Visual"+" Basic"的值同样是"Visual Basic"。

两种运算符虽然都能实现字符串的连接，但是有着各自不同的特点，如表 2-3 所示。

表 2-3 字符串连接运算符的比较

左操作数	右操作数	&	+
"123"	"456"	"123456"	"123456"
"123"	456	"123456"	579
123	456	"123456"	579
123	"456abc"	"123456abc"	类型不匹配，出错

说明：

（1）&是专用的字符串运算符，无论其操作数是何种类型，系统都会将它转换为字符串，然后强制进行连接。

（2）+运算符对操作数的类型要求较为严格，只有两个操作数均为字符串，才进行连接操作。如果其中一个操作数是数值，另一个操作数是字符串，则又分为两种情况：如果字符串中全部为数字字符，则进行算术求和操作；如果字符串中含有其他字符，则系统就会报错。

（3）&既是 Long 型数据的类型符，又是八进制整数的前缀。因此建议使用&运算符时，用空格将两个操作数与&分开，以免出现不必要的错误。

2.3.3 日期表达式

日期型数据可以进行加减运算，构成日期表达式。有以下两种情况：

（1）两个日期型数据相减，结果是一个数值，表示两个日期之间相差的天数。例如

#5/3/2008#-#4/29/2008#的值是 4，而#5/3/2008#-#5/8/2008#的值是-5。

（2）一个日期型数据与一个数值相加或相减，结果是一个日期型数据，表示向后或向前推算日期。例如#5/3/2008#+5 的值是#5/8/2008#，而#5/3/2008#-4 的值是#4/29/2008#。

2.4　语句

正如高楼大厦是由一砖一瓦堆砌而成的，程序代码则是由一条一条的语句组成的。语句是构成 VB 程序的最小单位，程序中的语句经过编译之后，生成了若干条机器指令。根据这些指令，计算机系统就能够完成运算操作，或者实现对操作流程的控制。

2.4.1　书写规则

与写文章一样，编写程序也应该遵守一定的规范。这样做不仅符合 VB 语法的要求，而且还增强了程序的可读性。

1. 注释

适当的注释有助于理解语句和程序的功能。注释不是语句，它不会被 VB 编译和执行。注释有以下两种格式：

（1）使用单引号（'）引导，一般出现在一条语句的后面。例如：

```
Dim sum As Long              '定义一个长整型变量 sum
```

（2）使用 Rem 引导，必须单独一行。例如：

```
Rem 定义一个长整型变量 sum
Dim sum As Long
```

2. 续行

如果一条语句过长，为便于阅读，可以用续行符（ _）将这条语句分成多行书写。例如：

```
s = "工作单位：" & "湖北省十堰市"_
& "湖北汽车工业学院"
```

注意：续行符的写法是空格紧跟下划线，它只能出现在一行的末尾。

3. 语句分隔

通常情况下一行只写一条语句，也可以用冒号（:）把几条语句分隔，然后写在同一行。例如：

```
t = a: a = b: b = t              '3 条语句写在同一行
```

2.4.2　赋值语句

赋值语句是 VB 程序中经常使用的基本语句，它的一般形式如下：

```
变量|对象.属性=表达式
```

说明：

（1）=是赋值运算符，它需要两个操作数，优先级最低。

（2）赋值运算符的右操作数通常是算术表达式、字符串表达式和函数调用表达式，左操作数是变量或者对象的属性。

（3）执行赋值语句时，首先计算赋值运算符右边的表达式，然后把值赋给左边的变量或者对象的属性。

（4）两个操作数的数据类型应尽量保持一致。如果类型不一致，系统会将右操作数的类型强制转换为左操作数的类型。这样做不仅有可能降低数据的精度，而且也有可能出现错误。例如把字符串"123abc"赋给一个整型变量，程序运行时系统会报错。

赋值语句的作用主要有以下两个：

（1）保存数据运算的结果。对数据进行计算之后，应通过赋值运算把结果及时保存在变量中，否则这样的操作会没有实际意义。例如计算球体体积的语句如下：

```
Dim r As Single, v As Single
r = 2                          '设置球的半径值为 2
v = 4 / 3 * 3.14 * r ^ 3       '计算球的体积，结果存放在变量 v 中
```

思考：在语句 v = 4/3*3.14*r ^3 中，4/3 写成 4\3 可以吗？

（2）在程序中修改对象的属性值。在界面设计阶段，利用属性窗口设置控件对象的属性值，称为静态设置。在程序中利用赋值语句设置控件对象的属性值，称为动态设置。例如把文本框对象 Text1 的背景色属性设置为红色，相应的赋值语句如下：

```
Text1.BackColor = vbRed   '改变文本框的背景色
```

说明：

每一个界面中的控件在程序里都有一个唯一的对象名。Text1.BackColor 的含义是通过成员运算符（.）访问对象 Text1 的 BackColor 成员。

赋值运算符的右操作数可以是函数调用表达式，它由函数名和参数列表组成。函数调用表达式的作用是通过调用某个函数，完成特定的功能。其一般形式如下：

```
函数名(参数列表)
```

VB 语言提供了大量的内部函数，它们能够完成一些预先设定好的功能，如计算数学函数值、字符串处理以及类型转换等。例如：

```
Dim a As Integer
a = Val("123abc")        '把字符串转换为数值后赋给整型变量 a
```

说明：Val 函数的功能是把字符串转换为数值，并自动过滤数字之后的非数字字符。经过函数调用之后再赋值，变量 a 的值是 123，从而确保了赋值的安全。

例如：计算 $y=\sqrt{|5\sin x-3\cos x|}+e^{2x}$，x=2。

分析：首先定义变量 x 和 y，把 2 赋给 x；然后调用 Sin 函数和 Cos 函数分别求正弦值和余弦值，调用 Abs 函数求绝对值，调用 Sqr 函数开平方，调用 Exp 函数求 e 的幂次；最后把函数调用表达式的值赋给 y。对应的 VB 语句段如下：

```
Dim x As Single, y As Single
x = 2
y = Sqr(Abs(5 * Sin(x) - 3 * Cos(x))) + Exp(2 * x)
```

说明：注意数学公式与 VB 表达式的不同之处，2x 应写为 2*x。

调用内部函数时应正确书写函数名称，注意参数的个数、类型以及实际意义，了解函数返回值的类型。例如求 60 度角的正弦值，应该调用内部函数 Sin，它只需要一个参数。注意到该参数接收的是弧度值，因此写成 Sin(3.14*60/180)。

2.4.3 流程控制语句

流程控制语句并不参与对数据的操作，而是控制程序执行的流程。它可以分为两类：

一类是流程结构语句，如 If 语句、For 语句等，形成某种控制结构；另一类是流程转向语句，例如使用 Exit 语句可以直接跳出循环结构或过程。在第 4 章和第 5 章将重点介绍流程控制语句。

End 语句

End 语句的功能是立即结束程序的执行，它的一般形式如下：

```
End
```

设计 VB 程序时，通常在窗体中画出一个用于退出的命令按钮，然后在该按钮的单击事件过程中安排一条 End 语句，为程序的执行设置一个终点。

2.5　窗体

开发 VB 程序的第一步就是设计程序界面，窗体则是设计界面的基本平台，所有的控件都是添加在窗体中的。窗体（Form）是 VB 程序的重要对象，也是所有控件的容器，如图 2-3 所示。程序的每一个窗体都是 VB 工程中的一个模块，并单独保存在一个窗体文件（.frm）中。在程序运行时，每一个窗体对应于一个具有 Windows 风格的窗口。

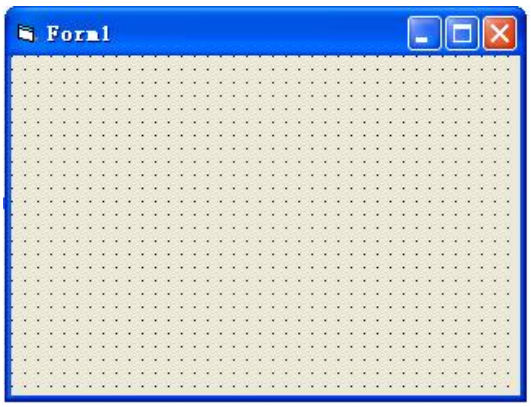

图 2-3　窗体

2.5.1　属性

窗体的属性描述了窗体的外观、位置等特性。表 2-4 列出了窗体的一些常用属性。

表 2-4　窗体的常用属性

属性	作用
Name	设置窗体的对象名
Caption	设置窗体的标题
AutoRedraw	确定是否自动重画被遮住的窗体内容
BorderStyle	设置窗体边框的类型
BackColor	设置窗体的背景颜色

续表

属性	作用
ForeColor	设置窗体的前景颜色
Font	设置窗体中显示的文字的字体
Height	设置窗体的高度
Width	设置窗体的宽度
Top	设置窗体距屏幕顶端的距离
Left	设置窗体距屏幕左端的距离
Moveable	确定程序运行时窗体能否移动
Visible	确定程序运行时窗体是否可见
WindowState	设置窗体在启动时的状态

说明：

（1）Name 是所有控件都具有的属性，其属性值就是控件对象在程序中的对象名。程序第一个窗体的默认对象名是 Form1，第 n 个窗体的默认对象名是 Formn，依此类推。

（2）BorderStyle 的属性值有 6 个，默认值是 2，如表 2-5 所示。

表 2-5　BorderStyle 属性值

常量	值	含义
None	0	窗体无边框
Fixed Single	1	窗体为单线边框，可以移动但不能改变尺寸
Sizable	2	窗体为双线边框，可以移动而且能改变尺寸
Fixed Dialog	3	窗体为固定对话框，不能改变尺寸
Fixed ToolWindow	4	窗体为工具栏风格，不能改变尺寸
Sizable ToolWindow	5	窗体为工具栏风格，可以改变尺寸

（3）窗体的高度、宽度以及距离等属性值的单位是 Twip，1 英寸=1440Twip。

（4）WindowState 的属性值有 3 个，默认值是 0，如表 2-6 所示。

表 2-6　WindowState 属性值

常量	值	含义
Normal	0	正常状态
Minimized	1	窗体最小化
Maximized	2	窗体最大化

2.5.2　事件

窗体作为对象，能够响应许多事件，表 2-7 列出了窗体的一些常用事件。

表 2-7　窗体的常用事件

事件	来源
Click	鼠标单击窗体
DblClick	鼠标双击窗体
Load	窗体装入工作区
Unload	卸载窗体
Activate	窗体成为活动状态
DeActivate	窗体成为不活动状态
Resize	调整窗体的尺寸

说明：

（1）装入窗体时会自动触发 Load 事件，因此可以在窗体的 Load 事件过程中对控件对象和变量进行初始化。当 VB 程序启动时，即可自动执行相应的初始化工作。

（2）Activate 事件和 DeActivate 事件往往发生在拥有多个窗体的 VB 程序中。例如某个程序有 A 和 B 两个窗体，当前 A 处于活动状态，B 处于不活动状态。如果单击 B 窗体，则 B 窗体成为活动状态，触发 Activate 事件；而 A 窗体成为不活动状态，触发 DeActivate 事件。

2.5.3　方法

方法是对象自身所具有的行为，也是为用户提供的功能。方法的调用形式如下：

[对象.]方法 [参数列表]

窗体的常用方法有 Print、Cls 和 Show 等，如表 2-8 所示。

表 2-8　窗体的常用方法

方法	功能
Print	在窗体中输出文本
Cls	清除窗体中显示的文本和图形
Show	显示窗体
Hide	隐藏窗体
Move	移动窗体，并可以改变其尺寸

说明：

（1）Print 方法不仅用于窗体，而且也可以用于图片框和打印机等其他对象。将在第 3 章详细介绍 Print 的用法。

（2）装入窗体并不表示一定会自动显示，需要调用 Show 方法显示窗体。如果窗体尚未装入内存，则调用 Show 方法时会自动加载。Show 方法有一个可选参数用于指定窗体模式，0 是默认值，表示非模态，系统不限制其他窗体的操作；1 表示模态，系统只允许当前活动窗体的操作，不允许切换到其他窗体，除非当前窗体被隐藏或卸载。例如以模态方式显示窗体 Form2，可以写为：

```
Form2.Show 1
```

（3）Move 方法的调用形式如下：

```
[对象.]Move left[,top[,width[,height]]]
```

left 是必选参数，表示对象移动后的左边距，其余 3 个参数是可选参数，分别表示对象的顶边距、宽度和高度。例如将窗体向屏幕的左下方移动并做适当缩小，可以写为：

```
Move Left-10,Top+10,Width-50,Height-50
```

2.6 小结

在程序中数据描述是通过数据类型体现的，VB 的基本数据类型主要有整型、实型和字符型等。各种数据类型都有常量和变量，变量对应了内存的一段存储空间，存储数据是通过变量实现的。各种类型的变量占据的内存字节数是不同的，它们能够表示的数据范围也是不同的。

数据处理是通过运算符和表达式完成的，本章主要学习了算术表达式和字符串表达式。算术表达式完成数据的加、减、乘、除等算术运算，注意算术运算符的优先级，以及整除与实数除法的区别。

语句和窗体是构成 VB 程序的基石。VB 语句主要有赋值语句和流程控制语句，对数据的操作是由前者完成的，后者负责控制程序执行的流程。赋值语句既可以用来修改变量的值，也可以用来设置对象的属性值。窗体是程序设计时的平台，也是程序运行时的窗口。本章学习了窗体常用的属性、事件和方法，熟悉了 VB 对象的一些特点，为进一步学习其他的控件对象奠定了基础。

习　题

1．下列哪些是合法的变量？

s1　integer　m_day　har?　sum　2n

2．VB 提供了哪些基本数据类型？

3．什么是类型符？如果未指定变量的类型，其默认类型是什么？

4．变量定义之后，VB 如何对其进行初始化？

5．计算下列表达式的值。

（1）x+y Mod 3*(x+y)\6，其中 x=4.2，y=5。

（2）(a+b)/5+a^2，其中 a=3，b=4。

（3）"VB" & 6

（4）"12"+34 & 5

（5）#6/7/2008#+3-#5/30/2008#

6．写出下列数学表达式对应的 VB 表达式。

（1）ax^2+bx+c

（2）$\sqrt{s(s-a)(s-b)(s-c)}$

（3）$\cos^3(a-b)$

（4）sin2a+ln|b-c|