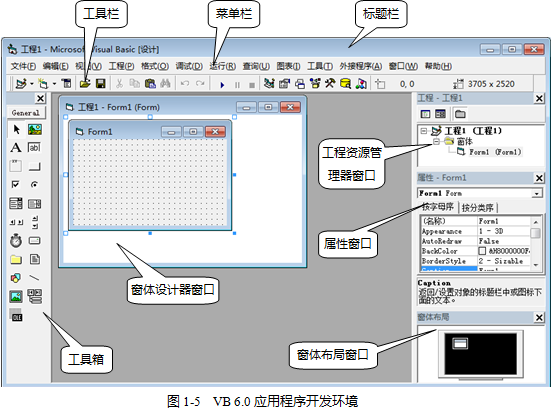
**Vb考试范围以及对应知识点**

|  |  |  |  |
| --- | --- | --- | --- |
| Visual Basic  程序设计 | 基本操作题 | 30 | 15% |
| 简单应用题 | 20 | 10% |
| 综合应用题 | 30 | 15% |

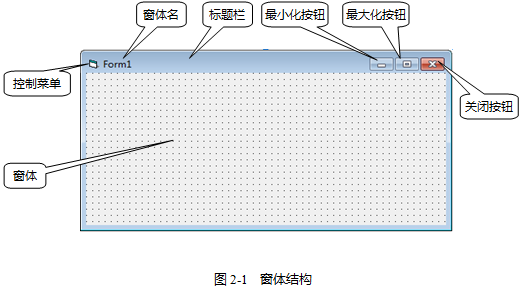
1. 集成开发环境（IDE）及其主要窗口的使用；



1. 应用程序的建立、编辑、调试、运行和保存；
2. 窗体的常用属性、方法和事件；

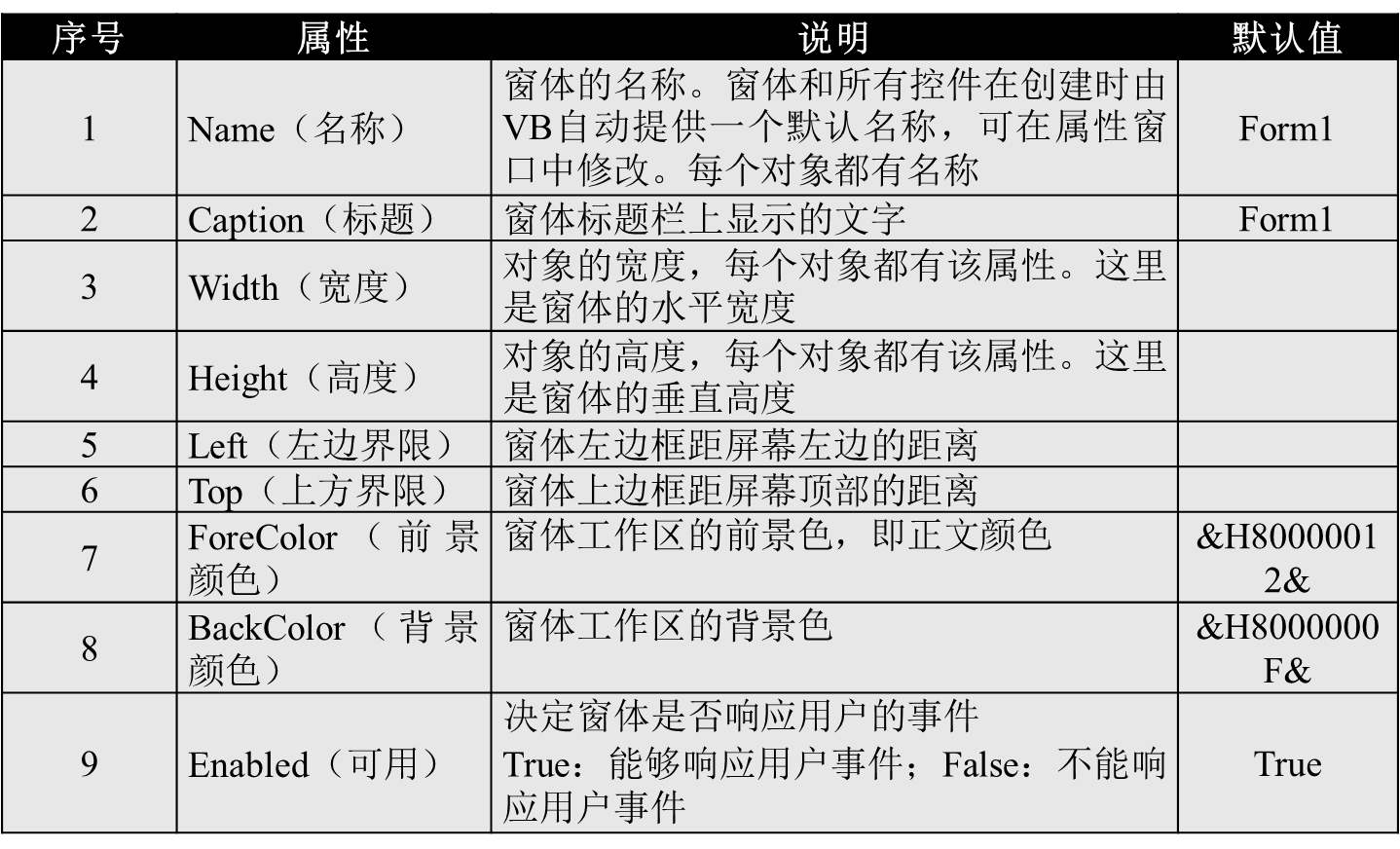
3.1定义

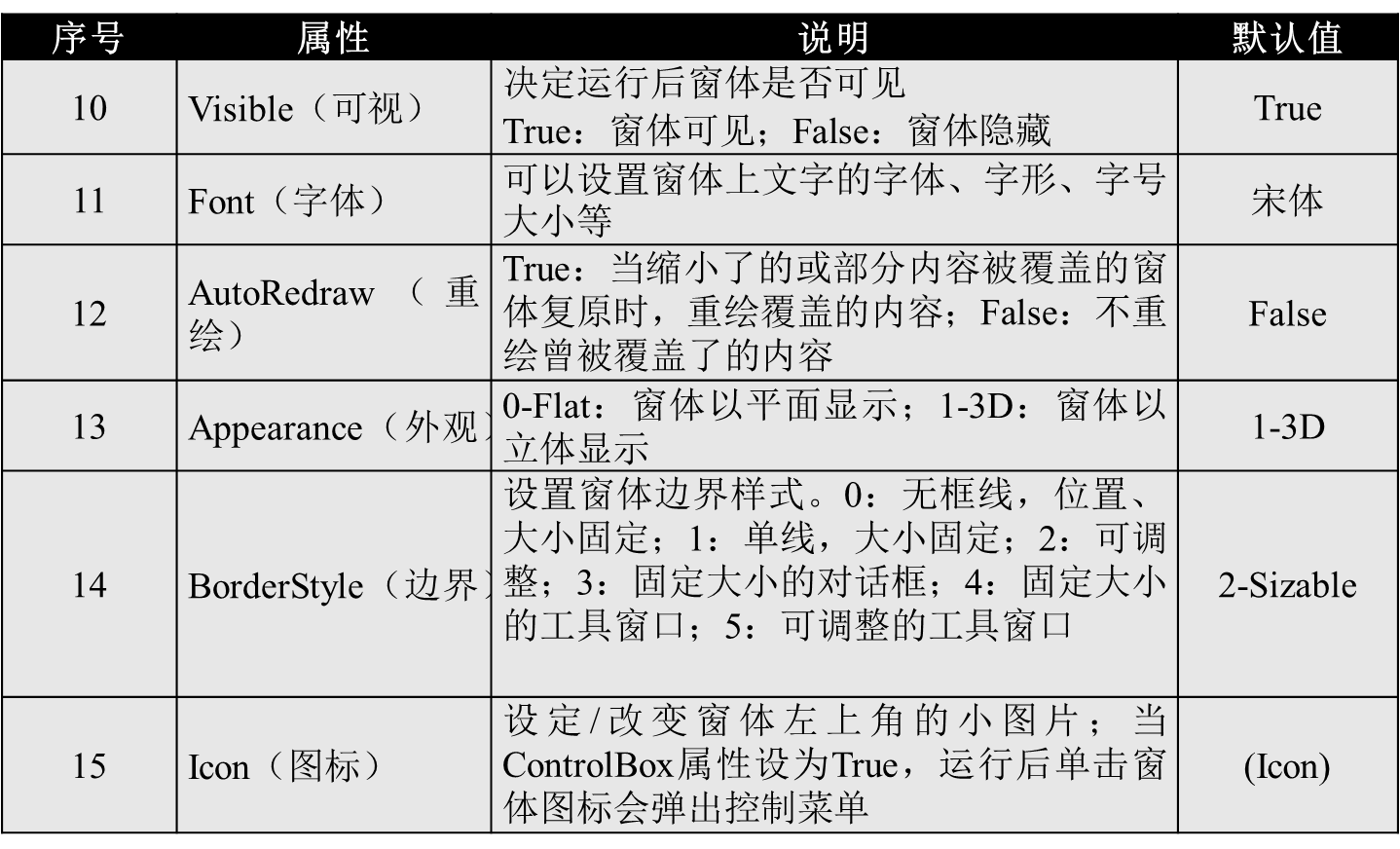
窗体结构与Windows下的窗口十分类似。在程序的设计阶段，这些用户界面称为窗体，在程序运行后称为窗口。窗口可以任意缩放、移动，可最大化也可以最小化。



3.2窗体的主要属性

窗体属性决定了窗体的外观





3.3常用方法

（1）Show方法

格式：对象名.Show

功能：显示被遮住的窗体，或将窗体载入内存后再显示。

（2）Hide方法

格式：[对象名.]Hide

功能：使窗体从屏幕上暂时隐藏，但并没有从内存中清除。

（3）Print方法

格式：[对象名.]Print [表达式表]

功能：该方法用来在窗体上输出文本和数据。

（4）Move方法

格式：[对象名.]Move X[，Y[，Width[，Height]]]

功能：窗体调用该方法可以进行移动，并可在移动中动态改变窗体的大小。

说明：参数X和Y表示移动到目标位置的坐标；Width和Height表示移动到目标位置后窗体的宽度和高度，通过这两个参数实现窗体大小的调整。若省略Width和Height参数，则移动过程中窗体大小不变。

例如，要将Form1移动到屏幕的（100，100）处，并使其大小变为高2000、宽3000，可使用如下语句：

Form1.Move 100, 100, 3000, 2000

（5）Cls方法

格式：[对象名.]Cls

功能：用于清除窗体上的文本或图形。

说明：如果省略对象名，则清除当前窗体中所显示的内容

3.4常用事件

（1）Initialize事件：仅当窗体第一次创建时（用对象的方法）触发该事件。编程时一般将窗体或其他对象的属性设置的初始化代码放在该事件过程中。

（2）Load事件：当窗体装入到内存时就会触发Load事件。编程时，一般把设置控件属性默认值和窗体级变量的初始化代码放到Load事件过程中。

（3）Activate、Deactivate事件：当窗体变为活动窗口时触发Activate事件，而在另一个窗体变为活动窗口前触发Deactivate事件。

（4）UnLoad事件：当从内存中清除一个窗体时触发该事件。如果重新装入该窗体，则窗体中所有的控件都要重新初始化。

（5）Click事件：单击鼠标左键时发生的事件。程序运行时，单击窗口内的空白处将调用窗体的Form\_Click事件过程，否则调用控件的Click事件过程。

（6）DblClick事件：双击鼠标左键时发生的事件。

（7）Paint事件：为了确保程序运行时不至于因某些原因使窗体内容丢失，通常用Paint事件过程来重画窗体内容。程序运行时，如果出现以下情况会自动触发Paint事件：

① 窗体被最小化成图标，然后又恢复为正常显示状态。

② 全部或者部分窗体内容被遮住。

③ 窗体的大小发生改变。

（8）Resize事件：运行时如果改变窗体的大小，则会自动触发该事件。

1. 基本数据类型和运算符；

4.1基本数据类型：



4.2运算符

字符串运算符

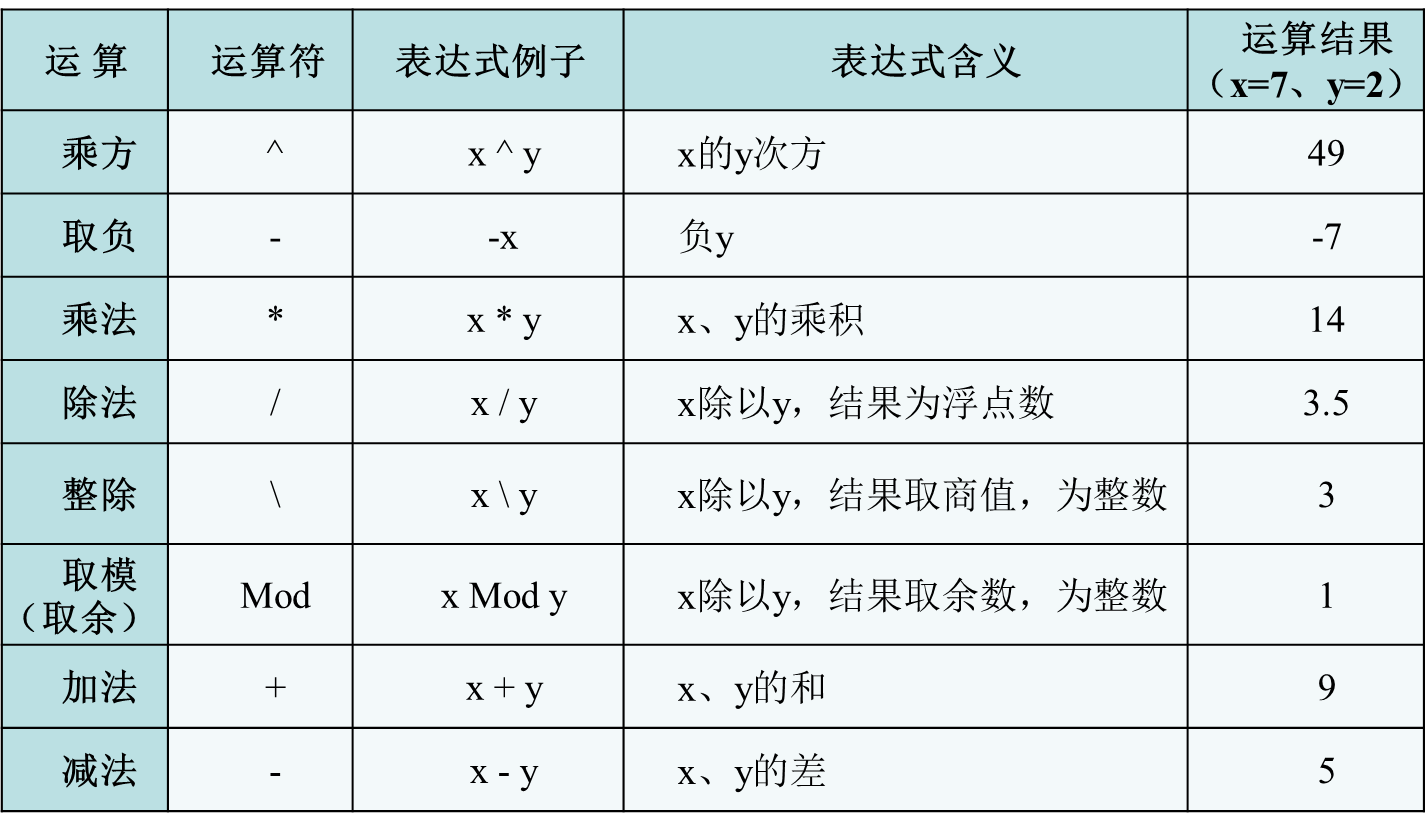
Visual Basic提供了两个字符串运算符：“&”和“+”。

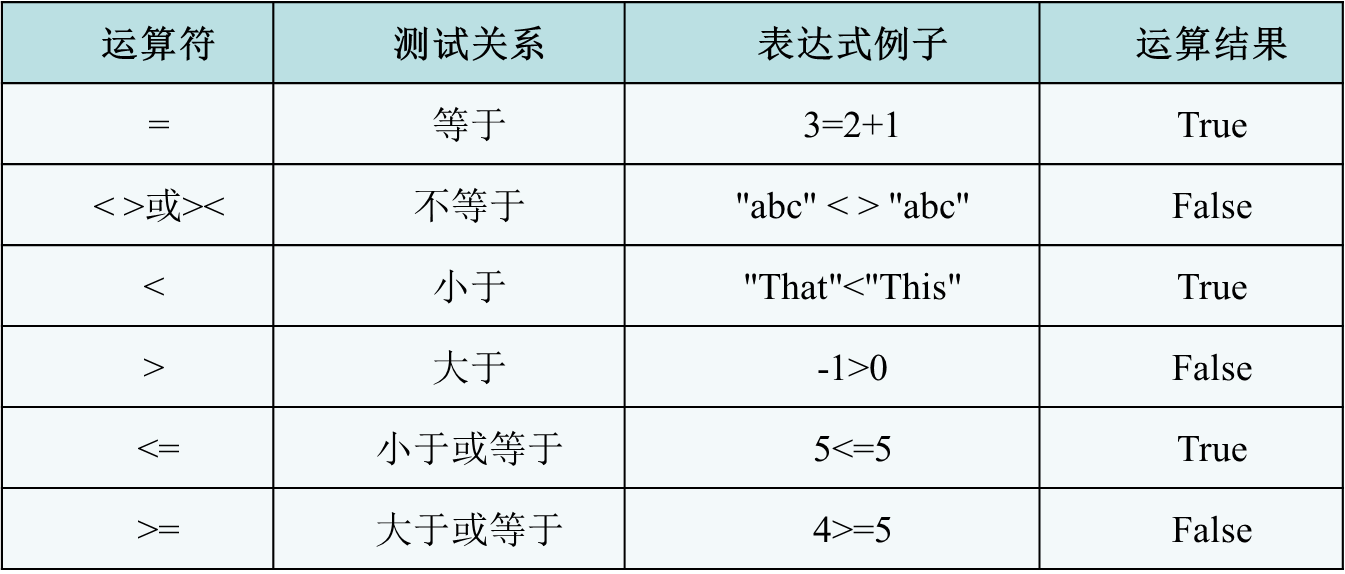
它们用于将两个字符串首尾连接起来形成一个新的字符串。

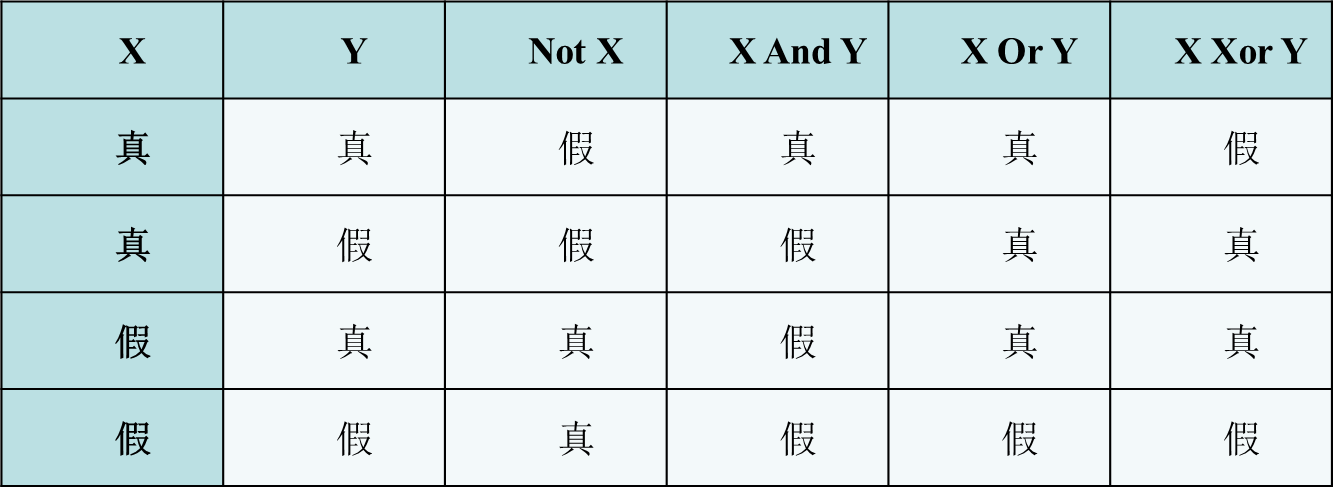
（1）“+”：两个操作数都是字符串时，做字符串连接运算；都是数值或一个是数值而另一个是数字字符串时，做加法运算；其它类型的操作数会出错。

（2）“&”：如果两边的操作数如果不是字符串而是其他数据类型，进行连接时系统先将操作数转换为字符串，然后再连接。

（3）因为符号“&”同时还是长整型的类型符，所以在使用“&”时要格外注意，“&”在用作运算符时，操作数与运算符“&”之间应加一个空格，否则会出错。







1. 常量、变量、表达式的定义和使用；

5.1常量

VB中的常量分为3种：文字常量、符号常量和系统常量。

5.1.1文字常量

（1）字符串型常量：必须使用英文的双引号将实际的字符括起来，双引号是字符串常量的定界符，表示字符串的开始和结束。例如："计算机程序设计基础"，"SQL Server 2000"。

（2）数值型常量：分两类。

① 字节型、整型、长整型常量：有三种形式，即十进制、十六进制、八进制。

十进制整型数：由若干个十进制数字（0~9）组成，可以带有正负号。如，123、-58等。

十六进制整型数：由若干个十六进制数字（0~9及a~f或A~F）组成，前面冠以前缀&H（或&h）。例如：&H36、&H2F8。

八进制整型数：由若干个八进制数字（0~7）组成，前面冠以前缀&O（大写字母O而不是数字0）。例如：&O726等。

可以在整型常量后面加类型符“%”或“&”来指明该常量是整型常量还是长整型常量；如不加类型符，VB系统会根据数值大小自动识别，将选择需要内存容量最小的表示方法。

② 浮点型常量：分为单精度浮点型常量和双精度浮点型常量。

日常记法：包括正负号、0~9、小数点。如果整数部分或小数部分为0，则可以省略这一部分，但要保留小数点。例如：3.1415926、-12.9、35. 、-.69 。

指数记法：用mEn来表示m×10n ，其中m是一个整型常量或浮点型常量，n必须是整型常量，m和n均不能省略。例如，1.23E4表示1.23×104 。“E”可以写成小写“e”，如果双精度常量，则需用“D”或“d”来代替“E”。

可以在浮点型常量后面加类型符“!”或“#”来指明该常量是单精度浮点型常量还是双精度浮点型常量；如不加类型符，VB系统会根据数值大小自动识别，将选择需要内存容量最小的表示方法。

（3）布尔型常量：也称逻辑型常量，只有两个值，即True和False。注意，它们没有定界符。“True”和“False”不是布尔型常量，而是字符串型常量。

（4）日期型常量：使用“#”作为定界符。只要用两个“#”将可以被认作日期和时间的字符串括起来，都可以作为日期常量。例如：#1949-10-1#，#2007-7-1 9:20:37 AM# 。

5.1.2字符常量

为了便于程序阅读和修改，对于程序中经常使用的常数值，通常采取用户自定义符号的形式。

格式：Const 符号常量名 [As 类型]= 表达式

说明：

（1）“符号常量名”需符合标识符的命名规则。

（2）“As类型”用来指定常量的数据类型，如果省略，则数据类型由“表达式”决定。

（3）“表达式”是由文字常量、小括号以及各种运算符组成。

（4）一行中可以定义多个符号常量，但各常量之间要用逗号隔开。

（5）如果符号常量只在过程或某个窗体模块中使用，则在定义时可以加上关键字Private（可省略）；如果要在多个模块中使用，则必须在标准模块中定义，并且要加上关键字Public。

（6）常量一旦声明，在其后的程序代码中只能引用，而不能改变常量值。

例如：

Const MAX As Integer =100，MIN=MAX-99

Private Const TODAY As Date = #2007-7-1#

Const PI#=3.1415926

5.1.3系统常量

除了用户通过声明创建的符号常量外，VB系统还提供了大量的已经定义好的符号常量，一般以“vb”为前缀，称为系统常量。

在“对象浏览器”中的Visual Basic（VB）、Visual Basic for Applications（VBA）等对象库中列举了VB的常量。

如：vbCrLf（回车换行）、vbRed（红色）、vbGreen（绿色）等。

5.2变量

5.2.1定义

变量实际上代表一些临时的内存单元，这些内存单元中可以存放数据，其内容随着程序的运行而变化。程序中可以通过变量名来引用内存单元中的变量值。

使用变量前，一般必须先声明变量名及其数据类型。在VB中，变量声明方式分为显式声明和隐式声明。

5.2.2显式声明

使用变量前用声明语句声明变量。

格式：Dim 变量名 [As 数据类型]

说明：

（1）关键字Dim还可以是Static、Private、Public或Global，他们的区别是声明的变量的作用范围不同，这一点将在第5章详细介绍。

（2）变量名需符合标识符的命名规则。

（3）变量名的尾部可以加上类型符，用来标识不同的数据类型。用类型符定义的变量，在使用时可以省略类型符。例如，用Dim name$ 定义了一个字符串变量name$，则引用这个变量时既可以写成name$，也可以写成name。

（4）数据类型决定了该变量所占内存空间的大小，若未指定数据类型且变量名末尾也没有类型说明符，则默认为变体型。

（5）在一个定义语句里可以同时定义多个变量。例如，下列语句定义的三个变量中，a是变体型，b是变长字符串型，c是长整型。

Dim a，b As String，c As Long

（6）用Dim可以定义变长字符串变量，也可以定义定长字符串变量。定义定长字符串变量的格式为： Dim 变量名 As String \* 正整数

例如：Dim stuID As String \* 4

这里，把变量stuID定义为长度是4的定长字符串。如果实际赋值给该变量的字符串长度小于4个字符，则不足的部分用空格补充；反之，如果超出4个字符，则超出的部分被忽略。

（7）所定义的变量根据不同的数据类型有不同的默认初值。数值型变量默认初值为0，字符串型变量默认初值为"" （空串），布尔型变量默认初值为False 。

5.2.3隐式声明

VB允许使用未经声明语句声明的变量，这种方式称为隐式声明。隐式声明的变量默认为变体型。

尽管默认声明很方便，但有可能带来麻烦，使程序出现无法预料的结果，而且较难查出错误。因此，为了安全起见，最好能显式地声明程序中使用的所有变量。

VB提供了强制用户对变量进行显式声明的措施。其操作过程如下：单击“工具”菜单中的“选项”命令，在打开的“选项”对话框中，选择“编辑器”选项卡，选中“要求变量声明”的复选框，这样就可以在所有新建模块的通用声明段自动插入Option Explicit语句。

5.3算术表达式

算术表达式是用算术运算符和小括号将数值型常量、变量、函数、对象属性等连接成的一个有意义的运算式子。

（1）书写VB表达式时，应注意与数学中表达式写法的区别。VB表达式不能省略乘法运算符。例如，数学表达式b2-4ac，写成VB表达式应为b^2-4\*a\*c或b\*b-4\*a\*c。

（2）VB表达式中所有的括号一律使用小括号，并且括号左右必须配对。例如，数学表达式2[x/(a+b)-c]，写成VB表达式应为2\*(x/(a+b)-c)。

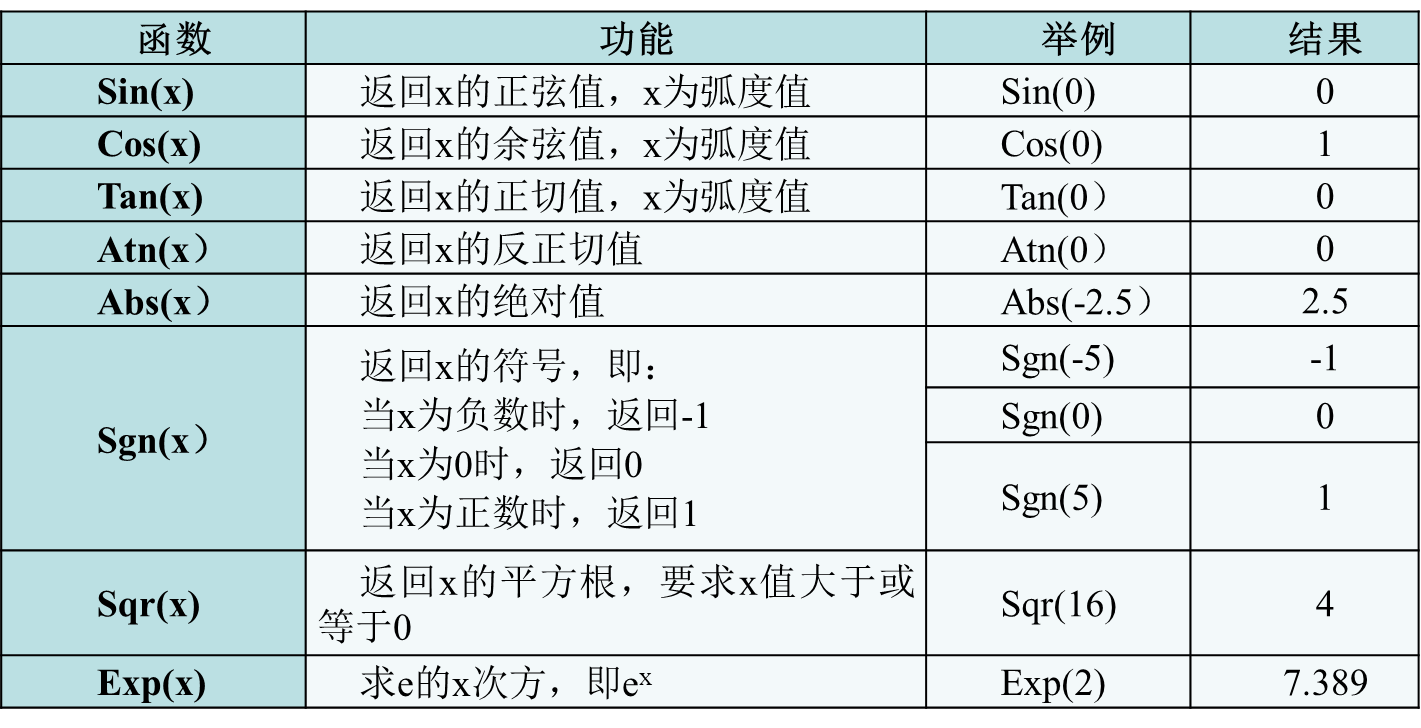
（3）一个表达式中各运算符的运算次序由优先级决定，优先级高的先运算，低的后运算，优先级相同的按从左到右的次序运算。算术运算符的优先级从高到低为：乘方→取负→（乘法、除法）→整除→取模→（加法、减法） 。小括号可以改变运算优先级，即小括号的优先级最高。如果表达式中含有小括号，则先计算小括号内表达式的值；有多层小括号时，先计算内层小括号。

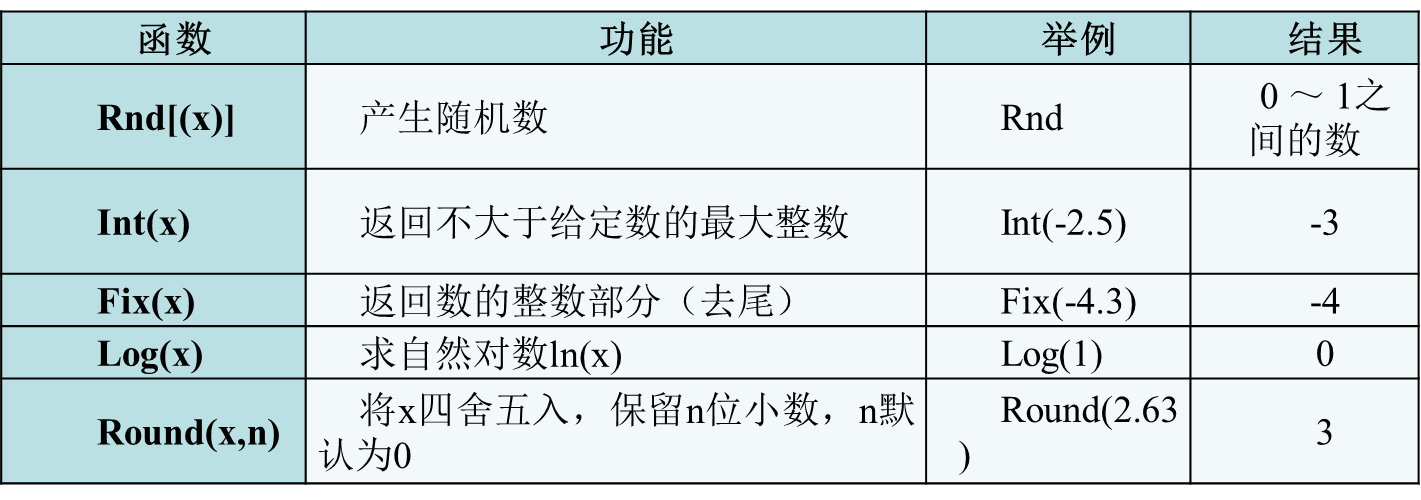
5.4字符串表达式

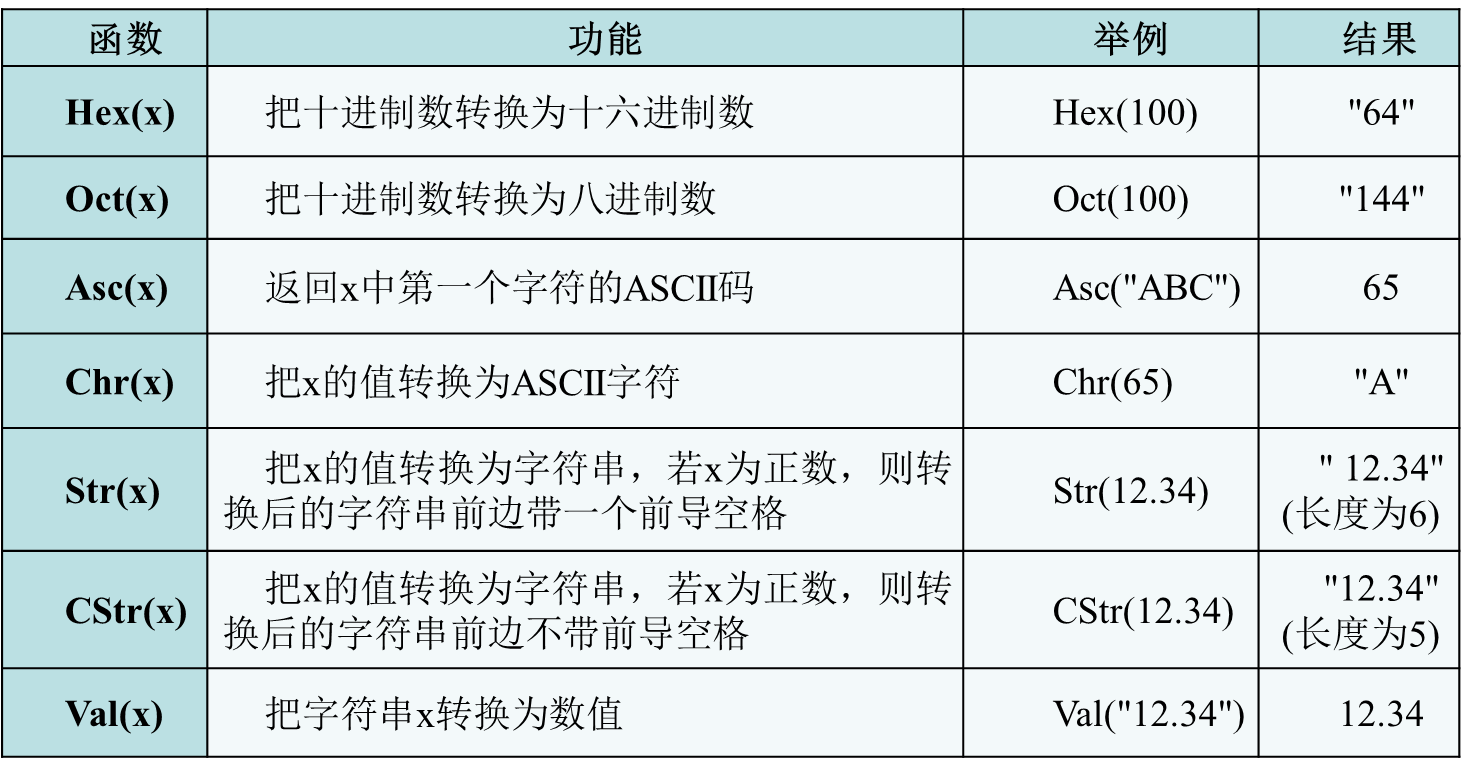
字符串表达式是用字符串运算符和小括号将一些常量、变量、函数等连接成的一个有意义的运算式子。字符串表达式的结果是一个字符串。

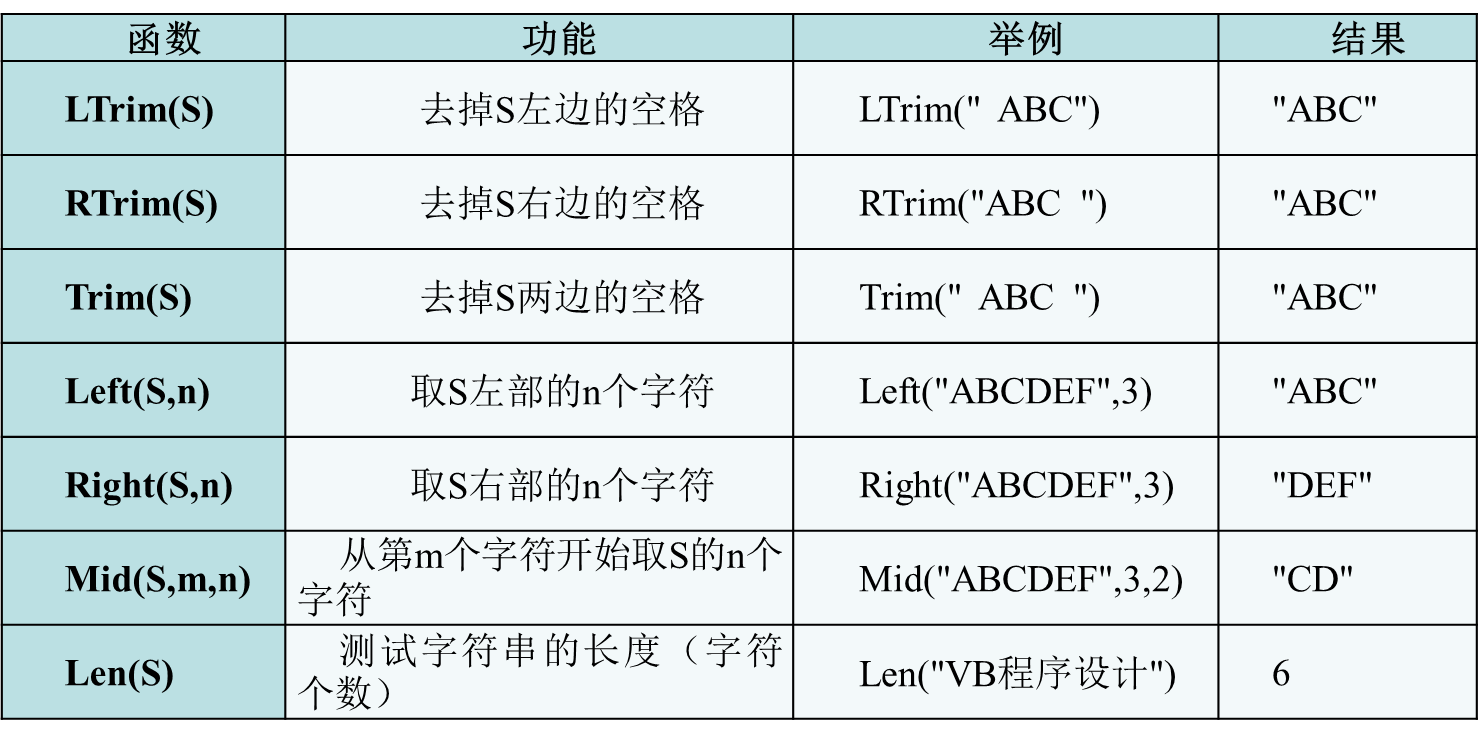
例如，表达式："This "+ "is "& "a " & "book." 结果是 "This is a book."

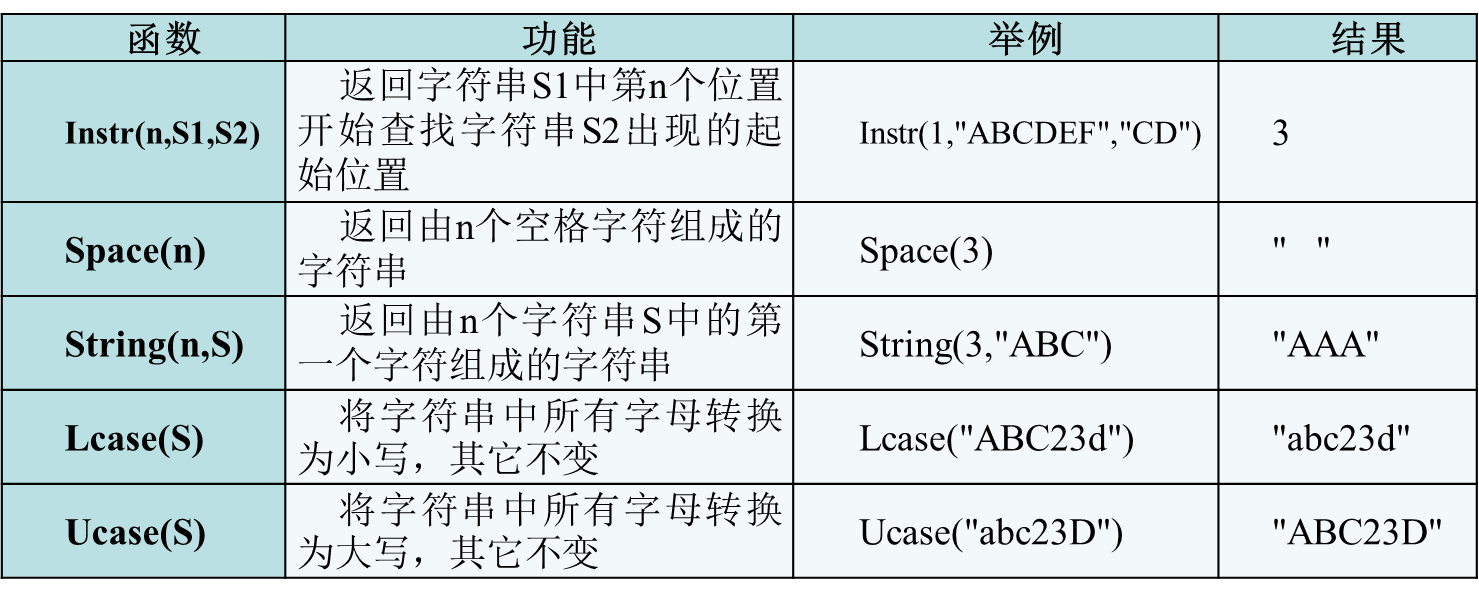
1. 常用内部函数的使用；











格式输出函数

格式：Format(<表达式>,<格式字符串>)

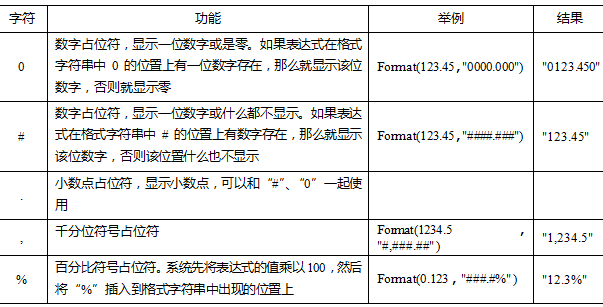
功能：用于控制输出数据的格式。

说明：

（1）函数名Format后面可以加上“$”符号，功能不变。该函数返回值类型为字符串型。

（2）<表达式>是指要格式化的数值、日期或字符串型表达式。

（3）<格式字符串>指定表达式的值的输出格式，格式字符串要加双引号。格式字符有3类：数值型格式、日期型格式和字符型格式。







1. 赋值语句、结束语句的使用；

7.1赋值语句

[Let]目标操作符=源操作符（变量，表达式，常量，带有属性的对象）

1. 赋值语句兼有计算与赋值的双重功能
2. “=”与数学上的等号意义不同
3. 目标操作符，源操作符的数据类型一致
4. 关键字Let 可省略

7.2注释语句

目的提高程序的可读性

Rem 注释内容

'注释内容

7.3暂停语句Stop

VB遇到Stop语句，这条语句就暂停程序执行转换到中断模式

4结束语句

End Sub 结束Sub过程

End Function 结束Function过程

End If 结束If语句块

End Type 结束记录类型

End Select 结束情况语句

1. Print语句的使用；

8.1Print方法

格式： ［对象名.］Print ［<表达式表>］［,|；］

功能：在指定的对象中输出表达式的值。

说明：

（1）对象名可以是窗体名、图片框名、打印机或立即窗口（Debug） 。若省略对象名，则表示在当前窗体上输出。

（2）<表达式表>可以是一个或多个表达式。各表达式之间如果使用分号分隔符，则以紧凑格式输出。如果是数值表达式，则先计算表达式的值，然后输出，输出数值的前面有一个符号位（正数显示空格，负数显示负号“-”），数值后面有一个空格；如果是字符串表达式，则原样输出，输出的字符串前后都没有空格；若省略表达式，则输出一个空行。

（3）若各表达式之间以逗号作分隔符，则按标准格式输出数据，每个表达式占14个字符位置。

（4）在一般情况下，每执行一次Print方法都会自动换行，即后一个Print语句的执行结果总是显示在前一个Print语句的下一行。为了仍在同一行上显示，可以在Print语句的末尾加上逗号或者分号。其中，分号表示紧凑格式，逗号表示标准格式。

8.2与Print方法有关的函数

（1）Tab函数

格式：Print Tab(n);表达式

功能：与Print方法配合使用，从第n列开始输出表达式的值，n为整数。

说明：Tab(n)与表达式之间必须用分号分隔。

（2）Spc函数。

格式：Print Spc(n);表达式

功能：与Print方法结合使用，从当前位置跳过n个空格后再输出表达式的值，n为整数。

说明：

① Spc函数与Tab函数区别：Tab函数是从第1列开始计数，而Spc函数是从前一个输出项结束位置开始计数，表示两个输出项之间的间隔。

② Space函数与Spc函数区别：Space函数返回值是字符串类型，Space函数和输出项之间可以用分号分隔，也可使用字符串连接运算符进行连接；而Spc函数和输出项之间只能用分号分隔。

1. InputBox输入函数的使用；

格式：InputBox（提示信息［,对话框标题］［,默认值］）

功能：屏幕显示一个输入框，等待用户输入信息后，将输入信息作为字符串返回。

说明：

（1）<提示信息>：字符串表达式。在对话框内显示提示信息，提示用户输入的数据的范围、作用等。如果要显示多行信息，则可在各行行末用回车符Chr(13)、换行符Chr(10)、回车换行符的组合Chr(13)&Chr(10)或系统常量vbCrLf来换行。

（2）<对话框标题>：字符串表达式，可选项。运行时该参数显示在对话框的标题栏中。如果省略，则在标题栏中显示当前的工程名。

（3）<默认值>：字符串表达式，可选项。显示在文本框中，在没有其他输入时作为缺省值。如果省略，则文本框为空。

例如：

MyStr = InputBox("请输入学生年龄" & Chr(13) & "范围：17～23岁", "输入年龄对话框", "20")

10.消息对话框MsgBox函数的使用；

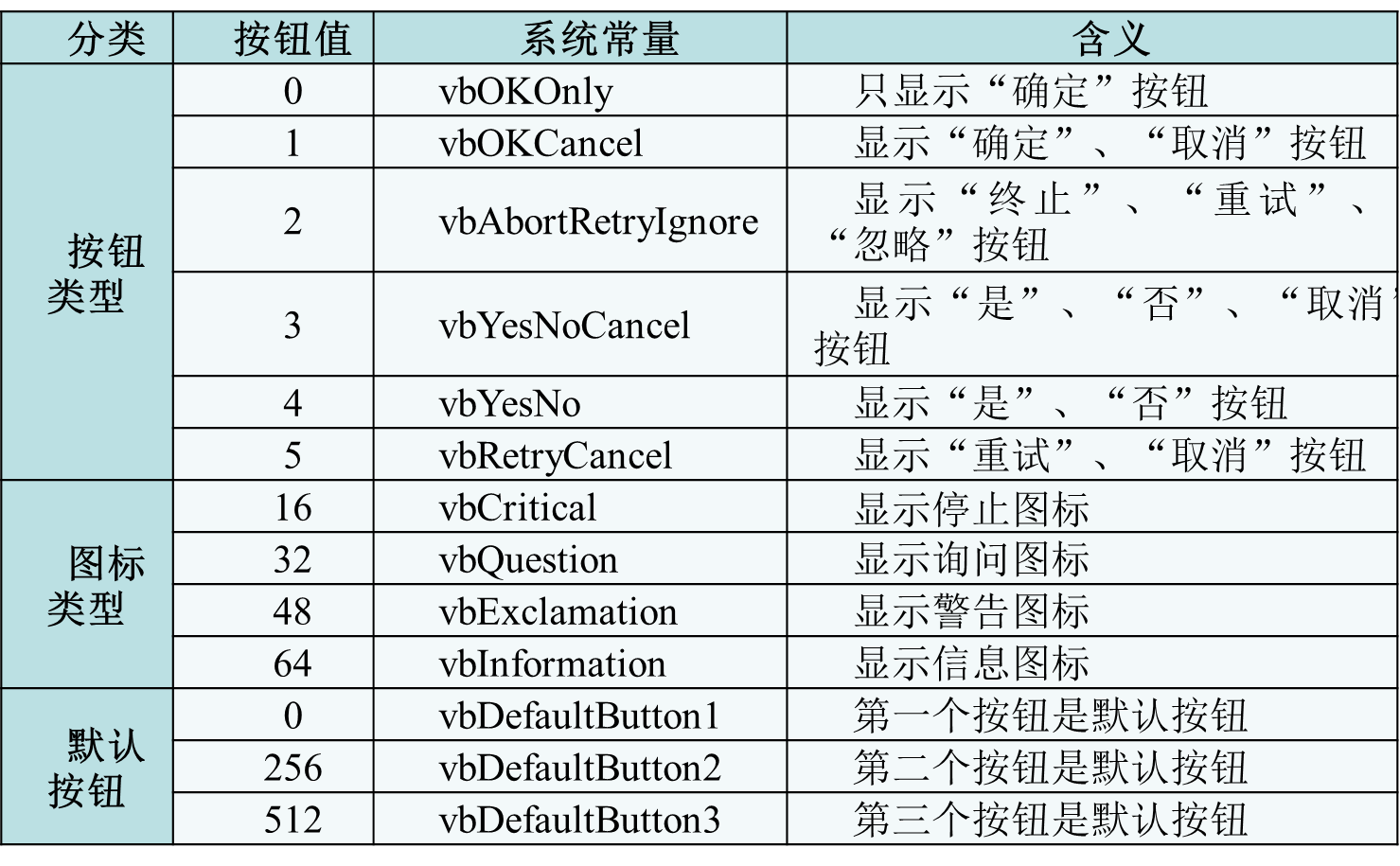
MsgBox（<提示信息>［，<按钮类型>］［，<对话框标题>］）

功能：屏幕显示一个消息框，等待用户单击按钮，返回一个整数告诉用户单击了哪个按钮。

说明：

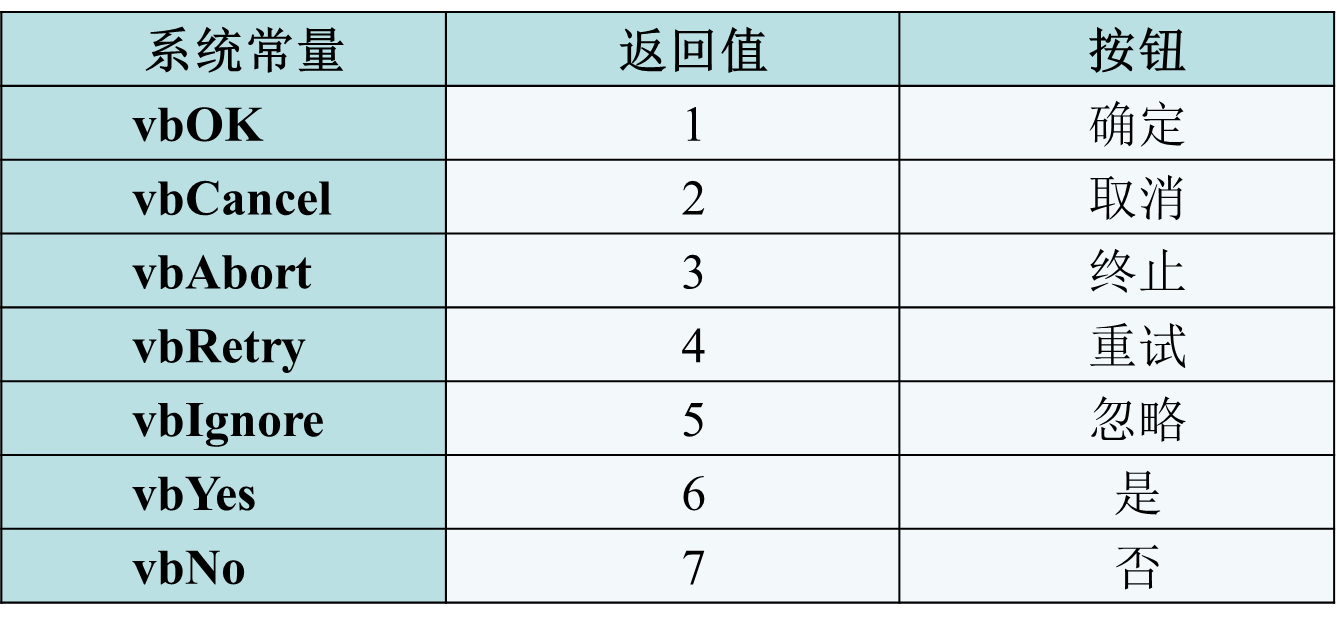
（1）<提示信息>：字符串表达式，用于指定显示在对话框中的信息。如果要显示多行信息，则可在各行行末用回车符Chr(13)、换行符Chr(10)、回车换行符的组合Chr(13)&Chr(10)或系统常量vbCrLf来换行。

（2）<按钮类型>：数值型数据，是可选项，用来指定对话框中出现的按钮和图标的种类及默认按钮。可以用按钮值，也可以用系统常量。“按钮类型”的设置值及含义如下表所示。



（3）<对话框标题>：字符串表达式，是可选项，它显示在对话框的标题栏中，如果省略，则在标题栏中显示工程名。

（4）MsgBox函数的返回值是由用户在消息框中选择的按钮决定的。每个按钮都有对应的返回值



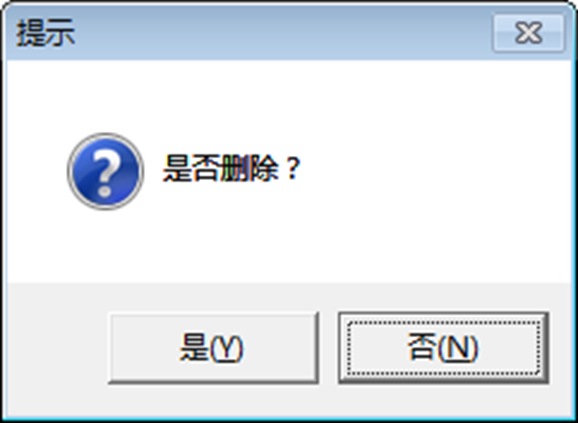
例如：

Private Sub Command1\_Click()

a = MsgBox("是否删除？", 4 + 32 + 256, "提示")

End Sub

效果如下图所示，如果用户选择“是”按钮，则a获得的值为6，否则获得7。



1. 选择结构程序设计；
2. 逻辑表达式的正确书写；
3. If语句和Select Case语句的使用；

14.1 If语句

（1）单行结构If语句

格式： If <条件> Then <语句组1> [Else <语句组2>]

功能：

如果条件表达式为真（True），则执行语句组1，否则执行语句组2，然后去执行该If语句的下一条语句。

[Else 语句组2]可以省略，此时，如果条件表达式为真（True），则执行语句组1，否则直接去执行该If语句的下一条语句。

（2）块结构双分支If语句

格式：

If <条件> Then

<语句组1>

[Else

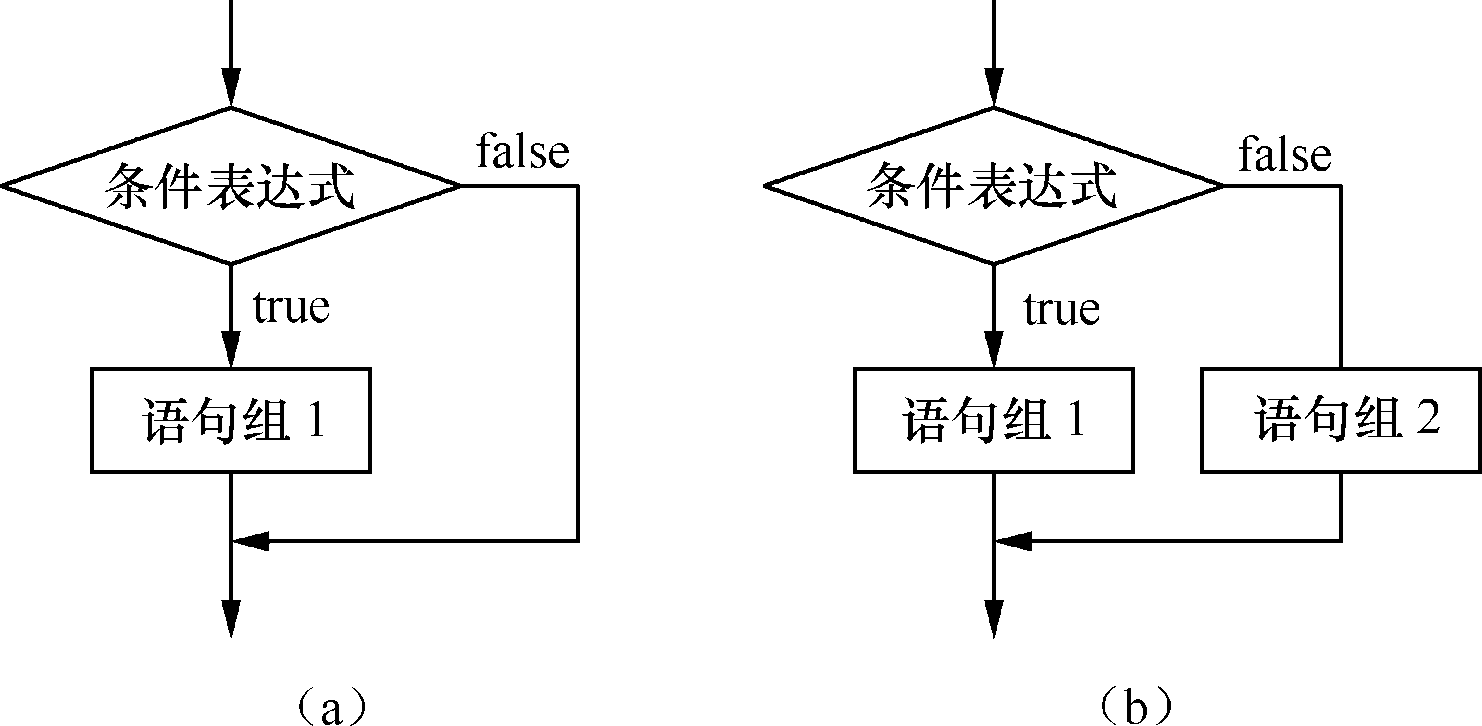
<语句组2>]

End If

功能：

如果条件表达式为真（True），则执行语句组1，否则执行语句组2，然后去执行该If语句的下一条语句（即End If下方的语句）。

Else子句可以省略，此时，如果条件表达式为真（True），则执行语句组1，否则直接去执行该If语句的下一条语句（即End If下方的语句）。



（3）块结构多分支If语句

格式：

If <条件1> Then

<语句组1>

ElseIf <条件2> Then

<语句组2>

……

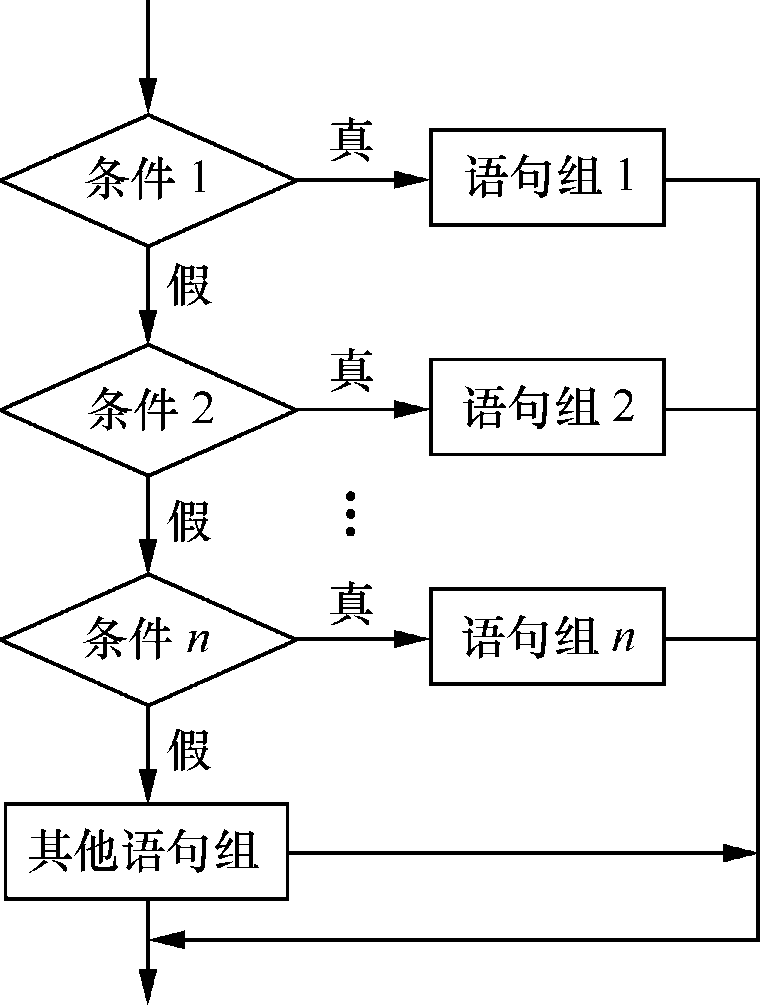
ElseIf <条件n> Then

<语句组n>

[Else

<其他语句组>]

End If



14.2 If语句的嵌套

如果一个If的语句组中又包含另一个If语句,称为If语句的嵌套。

格式：

If <条件1> Then

If <条件2> Then

<语句组1>

Else

<语句组2>

End If

Else

<语句组3>

End If

14.3 IIf函数

格式：IIf（<条件>,<表达式1>,<表达式2>）

功能：函数先判断<条件>的值，若为True，则函数返回<表达式1>的值；若为False，则函数返回<表达式2>的值。

说明：

（1）函数中的三个参数均不能省略。

（2）可以将IIf函数看作是单行If语句结构。例如，用IIf函数求变量x和y中的最大值存到max中：max = IIf(x > y, x, y)

等价于：If x > y Then max = x Else max = y 。

（3）IIf函数也可以嵌套。例如，求x、y、z三个数中的最大值。

max = IIf(z>IIf(x > y, x, y),z, IIf(x > y, x, y))

14.4 Select Case语句

格式：

Select Case 表达式

Case 表达式列表1

语句组1

Case 表达式列表2

语句组2

……

Case 表达式列表n

语句组n

Case Else

语句组n+1

End Select

功能：

若表达式的值与某个表达式列表的值相匹配，则执行该表达式列表后的相应语句组。

Select Case语句可以更方便地实现多分支程序的设计，且结构清晰，容易阅读，效率更高。

说明：

（1）表达式：可以是一个数值表达式或字符串表达式，通常是一个变量。

（2）Case表达式列表：是Case子句，如果表达式与某个Case子句的表达式列表相匹配，则执行该Case子句中的语句组。Case子句中的“表达式列表”可以有3种表示形式：

① 一个或多个常量，多个常量之间用“,”分开；

② 使用To关键字，用以指定一个数值范围，要求值小的数在To之前，如1 To 10；

③ Is关键字与关系运算符配合使用，用以指定一个数值范围，如Is >10。

在每个 Case 子句的“表达式列表”中，以上3种形式可以任意组合使用。如：

Case 3，5，7 To 9,Is>10

（3）Case Else：当表达式的值与前面所有的Case子句的表达式列表都不匹配时，执行语句组n+1。Case Else子句可以省略。

（4）End Select：为多分支结构语句的结束标志。

（5）Select Case语句也可以实现嵌套，每个语句组中又可以出现其它If语句或Select Case语句。当然，块结构If语句中也可以出现Select Case语句。

1. 循环结构程序设计；
2. For、Do和While循环语句的使用；

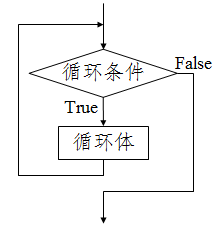
16.1 Do和While循环语句

格式一：前测试当型循环

Do While <条件>

[循环体]

Loop

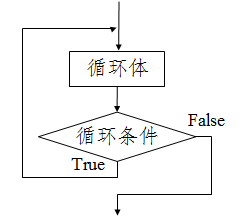


格式二：后测试当型循环

Do

[循环体]

Loop While <条件>



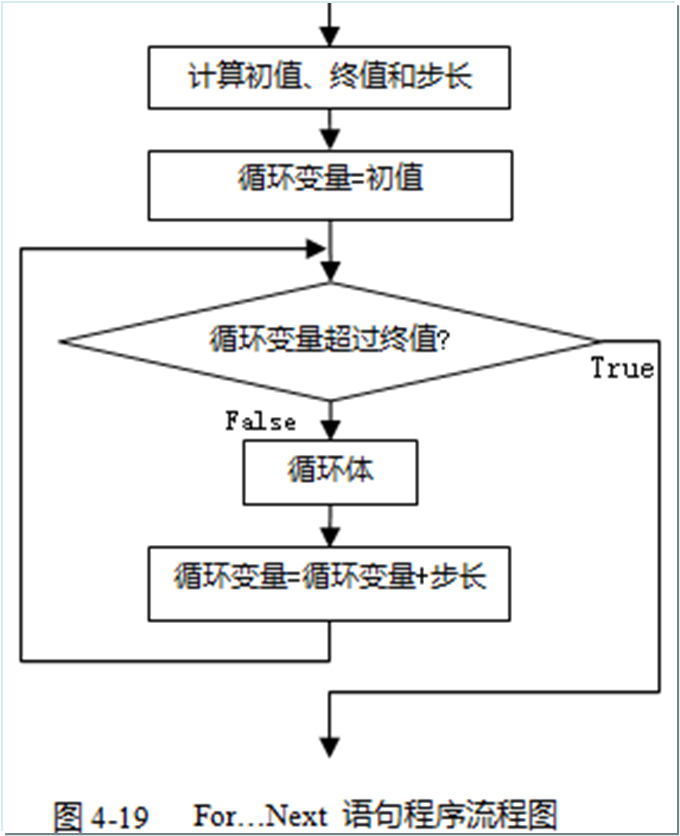
16.2 For…Next 语句

格式：

For <循环变量> = <初值> To <终值> ［Step <步长>］

［循环体］

Next <循环变量>



For…Next循环执行过程：

首先计算初值、终值和步长，将初值赋给循环变量。

若循环变量的值大于终值（步长为正数）或小于终值（步长为负数），则退出循环；

否则继续循环，执行循环体。

然后执行Next语句，将循环变量的值加上一个步长值，再返回去判断条件（与终值做比较），如果满足条件继续循环，否则结束For循环，转去执行Next语句之后的语句

说明：

（1）当步长为1时，Step 1可以省略。

（2）循环体中可以包含Exit For语句，该语句一般会出现在If语句中，用于表示当某种条件成立时，强行退出循环，转去执行Next语句之后的语句。

（3）For循环也称为计数循环，适合于循环次数已知的场合。

（4）循环次数计算公式：Int（（终值−初值）/步长）+1。

1. 数组的定义、初始化及基本应用；

17.1一维数组

17.1.1一维数组的定义

格式：Dim 数组名([<下界> to] <上界>)[As <数据类型>]

说明：

（1）数组名需要遵循标识符的命名规则。

（2）下界和上界：必须是常量。下界是数组元素的最小下标，上界是最大下标。若省略下界，则默认为0。下列语句可以改变下界的默认值：

Option Base 0|1

如果希望下界的默认值为1，则只需在相应模块的通用声明段输入如下语句：

Option Base 1

（3）As 数据类型：说明数组元素是什么类型。默认是Variat型。

例如：下列三组数组定义语句是等价的。均定义了数组a，共10个元素：a(1)、a(2)、...a(10)。

① Dim a(1 To 10) As Integer

② Dim a%(1 To 10)

③ Option Base 1 '该语句应放置在模块的通用声明部分

Dim a(10) As Integer

但下列语句定义的却是11个元素：a(0)、a(1)、a(2)、...a(10)。

Option Base 0 '该语句应放置在模块的通用声明部分

Dim a(10) As Integer

（4）数组必须先定义后使用。

17.1.2一维数组元素的引用

（1）在数组定义完之后，可以使用下标法引用数组中的每一个元素，格式：数组名(下标)。

（2）一般来说，程序中凡是简单变量出现的地方，都可以用相同类型的数组元素代替。也可以用循环语句对同一数组的多个元素进行成批操作。

（3）引用数组元素时,数组名、类型和维数必须与定义时一致。

（4）在同一个过程中，数组和简单变量不能同名,否则会出错。

17.1.3一维数组元素的赋值

（1）用InputBox函数为数组元素赋值。例如，

Dim a(1 To 10) As Integer, i As Integer

For i = 1 To 10

a(i) = Val(InputBox("请输入第" & i & "个数", "数组输入"))

Next i

（2）将数组元素赋值为有规律的数列。例如，

Dim a(1 To 10) As Integer, i As Integer

For i = 1 To 10

a(i) = 2 \* i - 1

Next i

17.1.4一维数组元素的输出

例如：在窗体上输出数组元素的值。

Dim a%(1 To 10)

For i = 1 To 10

Form1.Print a(i);

Next i

例如：在文本框中输出数组元素的值。

Dim a%(1 To 10)

For i = 1 To 10

Text1.Text = Text1.Text & a(i) & Space(1)

Next i

17.2二维数组

17.2.1二维数组的定义

格式：

Dim 数组名（[<下界>] to <上界>,[<下界> to] <上界>） [As <数据类型>]

说明：其中的参数与一维数组完全相同。

例如：Dim a( 2 , 3 ) As Single

二维数组在内存存放顺序是“先行后列”。数组a存放顺序： a(0,0)→a(0,1)→a(0,2)→a(0,3)→

a(1,0)→a(1,1)→a(1,2)→a(1,3)→

a(2,0)→a (2,1)→a(2,2)→a(2,3)

17.2.2二维数组的引用

格式：数组名(下标1,下标2)

例如：a(1,2)=10

a(i+2 ,j)=a(2,3)\*2

在程序中常常通过二重循环来操作使用二维数组元素。

17.2.3二维数组的基本操作

例如，给二维数组b输入数据的程序段如下：

Const N=4,M=5

Dim b(1 to N,1 to M) As Integer, i% , j%

 For i=1 to N

For j=1 to M

b(i,j)=Val(InputBox("请输入整数："))

Next j

Next i

1. 标签框、文本框、命令按钮、复选框、单选框、列表框、计时器等常用组件的应用；
2. 常见的标签（Label）控件属性

|  |  |
| --- | --- |
| 属性 | 说明 |
| (Name) | 标签控件的名称 |
| Alignment | 设置Caption属性文本的对齐方式, 取值为： 0 左对齐 1 右对齐 2 中间对齐 |
| Appearance | 是否用立体效果绘制,取值为： 0 平面 1 3D(立体) |
| AutoSize | 控件对象的大小是否随标题内容的大小自动调整,取值为： True 是 False 否 |
| BorderStyle | 设置边界样式,取值为： 0 None(无边界线) 1 FixedSingle(固定单线框) |
| Caption | 标签控件的标题 |
| Enabled | 用于设定是或对事件产生响应,取值为： True 可用 False 不可用,在执行程序时,该对象用灰色显示,并且不响应任何事件 |
| Index | 在对象数组中的编号 |
| TabIndex | 获得或设置此对象在父窗体的编号(父窗体中对象响应Tab键的顺序) |
| Visible | 设置此对象的可见性, 取值为： True 该对象可见 False 该对象不可见 |
| WordWrap | 获得或设置一个值,决定控件是否扩大以显示标题文字。取值为： True False |

* 1. 文本框控件(TextBox)的常用属性

|  |  |
| --- | --- |
| 属性 | 说明 |
| (Name) | 文本框控件的名称 |
| Alignment | 设置Caption属性文本的对齐方式, 取值为： 0 左对齐 1 右对齐 2 中间对齐 |
| Appearance | 是否用立体效果绘制,取值为： 0 平面 1 3D(立体) |
| Enabled | 用于设定是否对事件产生响应,取值为： True 可用 False 不可用,在执行程序时,该对象用灰色显示,并且不响应任何事件 |
| Index | 在对象数组中的编号 |
| Locked | 设置文本框的内容是否可以编辑。取值为： True 锁住文本框的Text属性内容,只能显示,不能通过键盘作任何更改,成为只读文本。此时在文本框中可以使用"复制"命令,不能使用"剪切"和"粘贴"命令。但是通过程序代码仍可以改变文本框的内容。 False 能通过键盘修改文本框的Text属性内容 |
| MaxLength | 获得或设置Text属性中所能输入的最大字符输。如果输入的字符数超过MaxLength设定的数目时,系统将不接受超出部分,并且发出警告声。 |
| MultiLine | 设置文本框对象是否可以输入多行文字。取值为： True 当文本超过控件边界时,自动换行。 False 需要注意的是：若该属性为False时,文本框控件对象的Alignment属性无效。 |
| PasswordChar | 该属性将文本显示的内容全部改为所设置的内容。他常用于设置密码,如PasswordChar设定为"\*",那么无论用户输入什么字符,都只显示"\*"。 |
| ScrollBoars | 设置边框滚动条模式： 0 无滚动条 1 水平滚动条 2 垂直滚动条 3 水平和垂直滚动条 |
| SelLength | 返回或设置选定文本的长度(字符数)。 该属性没有列在属性窗口中,但在程序中可以使用这些属性。 |
| SelStart | 返回或设置选定文本的起始位置,如果没有文本被选中,则指出插入点的位置。 该属性没有列在属性窗口中,但在程序中可以使用这些属性。 |
| SelText | 返回或设置选定文本, 如果没有字符串被选中,则为空字符串。 该属性没有列在属性窗口中,但在程序中可以使用这些属性。 |
| Text | 显示的文本内容 |
| Visible | 设置此对象的可见性, 取值为： True 该对象可见 False 该对象不可见 |

* 1. 框架控件(Frame)的常用属性

|  |  |
| --- | --- |
| 属性 | 说明 |
| (Name) | 框架控件的名称 |
| Appearance | 是否用立体效果绘制,取值为： 0 平面 1 3D(立体) |
| BackColor | 背景颜色,可从弹出的调色板选择。 |
| BorderStyle | 设置边界模式,取值为： 0 None(无边界线) 1 FixedSingle(固定单线框) |
| Caption | 框架控件的标题 |
| Enabled | 框架控件是否可用,取值为： True 可用 False 不可用,在执行程序时,该框架控件的标题暗淡虚显, 连同框架上的其他控件一起被禁用 |
| Font | 字型,可从弹出的对话框选择字体,大小和风格 |
| Index | 在对象数组中的编号 |
| Visible | 设置此对象的可见性, 取值为： True 该对象可见 False 该对象不可见 |

1. 命令按钮控件(CommandButton)的常用属性

|  |  |
| --- | --- |
| 属性 | 说明 |
| (Name) | 命令按钮的名称 |
| Appearance | 是否用立体效果绘制,取值为： 0 平面 1 3D(立体) |
| Cancel | 用于设定默认的取消按钮(指出命令按钮是否为窗体的取消按钮)。取值为： True 不管窗体上的哪个控件有焦点,按下"Esc"键后，就相当单击该默认按钮 False 否 |
| Caption | 命令按钮的标题。该属性最多包含255个字符。如果其内容超过了命令按钮的宽度则会转到下一行, 如果其内容超过255个字符, 则标题的超出部分被截掉。该属性也可以为命令按钮创建快捷方式，其方法是在做为的快捷字母前加一个连字符(&) |
| Default | 设置该命令按钮是否为窗体默认的按钮。取值为： True 用户按下Enter键,就相当单击该默认按钮 False 否 |
| Enabled | 用于设定是或对事件产生响应,取值为： True 可用 False 不可用,在执行程序时,该对象用灰色显示,并且不响应任何事件 |
| Index | 在对象数组中的编号 |
| Visible | 设置此对象的可见性, 取值为： True 该对象可见 False 该对象不可见 |

1. 复选框控件(CheckBox)的常用属性

|  |  |
| --- | --- |
| 属性 | 说明 |
| (Name) | 复选框控件的名称 |
| Alignment | 设置标题文本的对齐方式, 取值为： 0 左对齐 1 右对齐 |
| Appearance | 是否用立体效果绘制,取值为： 0 平面 1 3D(立体) |
| Caption | 复选框控件的标题,此标题也支持快捷键 |
| Enabled | 用于设定是或对事件产生响应,取值为： True 可用 False 不可用,在执行程序时,该对象用灰色显示,并且不响应任何事件 |
| Index | 在对象数组中的编号 |
| Value | 设置复选框处在什么状态。复选框控件有三种状态： 0 Unchecked 未选定 1 Checked 选定 2 Graved 灰色 |
| Visible | 设置此对象的可见性, 取值为： True 该对象可见 False 该对象不可见 |

1. 单选钮控件(OptionButton)的常用属性

|  |  |
| --- | --- |
| 属性 | 说明 |
| (Name) | 单选钮控件的名称 |
| Alignment | 设置标题文本的对齐方式, 取值为： 0 左对齐 1 右对齐 |
| Appearance | 是否用立体效果绘制，取值为： 0 平面 1 3D(立体) |
| Caption | 单选钮控件的标题，此标题也支持快捷键 |
| Enabled | 用于设定是或对事件产生响应，取值为： True 可用 False 不可用,在执行程序时，该对象用灰色显示,并且不响应任何事件 |
| Index | 在对象数组中的编号 |
| Value | 获得或设置单选钮处在什么状态。取值为： True选中 False 未选中 用户可以在属性窗口中设置该属性，也可以在程序中用代码来设置。该属性常用于判定是否选定选项，并作出相应的响应。 |
| Visible | 设置此对象的可见性, 取值为： True 该对象可见 False 该对象不可见 |

1. 组合框控件(ComboBox)常用属性

|  |  |
| --- | --- |
| 属性 | 说明 |
| (Name) | 组合框控件的名称 |
| Appearance | 是否用立体效果绘制,取值为： 0 平面 1 3D(立体) |
| Enabled | 用于设定是或对事件产生响应,取值为： True 可用 False 不可用,在执行程序时,该对象用灰色显示,并且不响应任何事件 |
| Font | 字型,可从弹出的对话框选择字体,大小和风格 |
| ForeColor | 前景颜色,可从弹出的调色板选择。 |
| Height | 组合框控件的高度 |
| HelpContextID | 指定一个对象的缺省帮助文件上下文标识符 |
| Index | 在对象数组中的编号 |
| IntegralHeight | 是否自动调整自身大小(获得或设置一个值,指出控件是否显示部分项目)。取值为： True False |
| ItemData | 数据列表(获得或设置组合框控件中每一个项的指定额管理号) |
| List | 项目列表(获得或设置控件的列表部分中包含的项) |
| Text | 选中的项目内容 |
| Visible | 设置此对象的可见性, 取值为： True 该对象可见 False 该对象不可见 |

1. 列表框控件(ListBox)的常用属性

|  |  |
| --- | --- |
| 属性 | 说明 |
| (Name) | 列表框控件的名称 |
| Appearance | 是否用立体效果绘制,取值为： 0 平面 1 3D(立体) |
| Cloumns | 制定列表框中列的数目。列表框中的项目可以单列垂直显示,也可以水平单行或多列列表显示。取值为： 0 垂直单列列表 1 水平单行列表 >1 水平多列列表 |
| Enabled | 用于设定是或对事件产生响应,取值为： True 可用 False 不可用,在执行程序时,该对象用灰色显示,并且不响应任何事件 |
| Index | 在对象数组中的编号 |
| ItemData | 数据列表(获得或设置列表框控件中每一个项的指定额管理号) |
| List | 项目列表(获得或设置控件的列表部分中包含的项) |
| ListCount | 返回列表框中项目的数目。该属性是一个只读属性,不能在属性窗口中设置, 只能在程序运行时访问它 |
| ListIndex | 该属性是一个只读属性,不能在属性窗口中设置,一般在程序运行中设置或返回控件中当前选中项目的索引。 |
| MultiSelect | 是否允许多项选择。取值为： 1 不允许多项选择 2 简单多项选择,即用鼠标单击或用空格键光标键操作 3 扩充多项选择,即用Shift键和Ctrl键配合操作 |
| Text | 该属性总是对应用户在程序运行时所选定的列表项目。Text是一个只读属性,不能在属性窗口中设置,也不能在程序中设置。它只用于获取当前选定的项目值 |
| Visible | 设置此对象的可见性, 取值为： True 该对象可见 False 该对象不可见 |

1. 水平滚动条控件(HScrollBar)和垂直滚动条控件(VScrollBar)的常用属性

|  |  |
| --- | --- |
| 属性 | 说明 |
| (Name) | 控件对象的名称 |
| Enabled | 用于设定是或对事件产生响应,取值为： True 可用 False 不可用,在执行程序时,该对象用灰色显示,并且不响应任何事件 |
| Index | 在对象数组中的编号 |
| LargeChange | 用于设置单击滚动条中间区域时,每单击一次滚动条移动的数量。< |
| Max | 定义Value属性值的最大值 |
| Min | 定义Value属性值的最小值 |
| SmallChange | 用于设置单击滚动条两端箭头时,每单击一次滚动条移动的数量。 |
| Value | 该属性值对应于滚动框在滚动条中的相应位置,是一个整数。 |
| Visible | 设置此对象的可见性, 取值为： True 该对象可见 False 该对象不可见 |

1. 定时器控件的常用属性

|  |  |
| --- | --- |
| 属性 | 说明 |
| (Name) | 定时器控件的名称 |
| Enabled | 用于设定是或对事件产生响应,取值为： True 可用 False 不可用,在执行程序时,该对象用灰色显示,并且不响应任何事件 |
| Index | 在对象数组中的编号 |
| Interval | 获得或设置两次调用Timer控件的Timer事件间隔的毫秒数。 |

1. 图片框和图像框的区别

图片框

|  |  |
| --- | --- |
| 属性 | 说明 |
| (Name) | 图片框控件的名称 |
| AutoSize | 用于调整控件的大小是否自动调整, 取值为： True 控件自动调整大小以适应加载的图形 False 不调整,当图形原始大小比尺寸大时,则超出部分自动被裁掉。 |
| Enabled | 用于设定是或对事件产生响应,取值为： True 可用 False 不可用,在执行程序时,该对象用灰色显示,并且不响应任何事件 |
| Picture | 该属性用于设置控件要显示的图形, 该属性不管是在属性窗口还是在运行时用程序设置,均要求有完整的路径名和文件名。 |

Picture1.Picture = LoadPicture(App.Path & "\1.jpg") '加载图像，App.Path 表示该

图像框

|  |  |
| --- | --- |
| 属性 | 说明 |
| (Name) | 图像控件的名称 |
| Picture | 该属性用于设置控件要显示的图形, 该属性不管是在属性窗口还是在运行时用程序设置,均要求有完整的路径名和文件名。 |
| Stretch | 该属性用于设置由Picture属性设定的图片是否随着控件的大小自动调整其大小,图形的伸展可能导致图象质量的降低。 |

Image1.Picture = LoadPicture(App.Path & "\1.jpg") '加载图像，App.Path 表示该工程所在路径

1. 常用算法的实现：累加、累乘、求平均、求最大/最小值、条件计数、选择排序、冒泡排序、直接排序、顺序査找等。

19.1数组输入、输出

Dim a(1 To 10) As Integer

Private Sub Command1\_Click()

Dim i As Integer

For i = 1 To 10

a(i) = InputBox("输入整数")

Print a(i);

Next i

Print

End Sub

Private Sub Command2\_Click()

For i = 1 To 10

Print a(i);

Next i

End Sub

19.2数组初始化、累加、累乘、求平均值

Private Sub Command1\_Click()

Dim i As Integer

Dim s As Integer '如果是累乘，建议定义为Long类型

Dim a()

a = Array(5, 4, 3, 2, 1, 10, 9, 8, 7, 6)

s = 0 '如果是累乘，应初始化为1

For i = 0 To 9

s = s + a(i) '如果是累乘，应使用乘法

Next i

Print s '如果是求平均值，此处应写上s / 10

End Sub

19.3求最大、最小值

Private Sub Command1\_Click()

Dim i As Integer

Dim s As Integer

Dim max As Integer

Dim min As Integer

Dim a()

a = Array(5, 4, 3, 2, 1, 10, 9, 8, 7, 6)

max = a(0)

min = a(0) '把第一个数当成最大、最小值

For i = 1 To 9 '对剩下的数依次比较

If a(i) > max Then max = a(i) '更新最大值

If a(i) < min Then min = a(i) '更新最小值

Next i

Print max; min

End Sub

19.4条件计数

Private Sub Command1\_Click()

Dim i As Integer

Dim s As Integer

Dim t As Integer

Dim a()

a = Array(5, 4, 3, 2, 1, 10, 9, 8, 7, 6)

t = 0 '计数器初始为0

For i = 0 To 9

'统计数组当中能被3整除的数的个数

If a(i) Mod 3 = 0 Then t = t + 1

Next i

Print t

End Sub

19.5顺序查找

Private Sub Command1\_Click()

Dim i As Integer

Dim s As Integer

Dim t As Integer

Dim key As Integer

Dim a()

a = Array(5, 4, 3, 2, 1, 10, 9, 8, 7, 6)

key = InputBox("输入待查找的数")

t = -1 't用来保存找到的位置，-1表示找不到

For i = 0 To 9

'在数组中顺序查找

If a(i) = key Then

t = i '若找到，则记住其位置，并退出循环

Exit For

End If

Next i

If t >= 0 Then '找到，则输出位置

Print t

Else

Print "找不到"

End If

End Sub

19.6冒泡排序

For i = 0 To n - 2

For j = 0 To n-2-i

If a(j) > a(j+1) Then

t=a(j)： a(j)=a(j+1)： a(j+1)=t

End if

Next j

Next i

19.7 选择排序

For i = 0 To n-2

For j = i + 1 To n-1

If a(i) > a(j) Then

t = a(i): a(i) = a(j): a(j) = t

End If

Next j

Next i