第 2 章　Visual Basic 语言基础

本章主要介绍 Visual Basic 6.0 的编程基础知识，包括常量、变量的基本概念；数据类型及定义；运算符与表达式的使用；程序结构；自定义过程、函数的基本语法和使用。本章还详细探讨了变量作用范围，介绍了 VB 应用程序的启动模式等，为后续学习面向对象的程序设计，打下坚实的基础。

2.1　常量、变量与数据类型

2.1.1　Visual Basic 的数据类型

数据是分类型的，数据的类型表明数据可以参与的运算以及所需存储空间的大小与范围。VB6 的数据类型包括标准数据类型和用户自定义类型。其中，标准数据类型包括字节型（Byte）、逻辑型（Boolean）、整型（Integer）、长整型（Long）、货币型（Currency）、无符号整型（Decimal）、单精度浮点型（Single）、双精度浮点型（Double）、日期型（Date）、字符型（String）、对象型（Object）、变体型（Variant）等。

在 VB6 中，数据类型及所占存储空间大小和范围列表如表 2-1 所示。

表 2-1　数值类型

类型名称	关键字	字节数	取值范围及用途
字节型	Byte	1	0～255，用于存储二进制数据，如图像、声音等
整型	Integer	2	-32768～32767，用于表示没有小数点的数据
长整型	Long	4	-2147483648～2147483647，当使用整型不足以表示数据时，使用长整型
单精度浮点型	Single	4	负数时：-3.402823E38～-1.401298E-45； 正数时：1.401298E-45～3.402823E38
双精度浮点型	Double	8	负数时：-1.79769313486232E308～-4.94065645841247E-324； 正数时：4.94065645841247E-324～1.79769313486232E308
货币型	Currency	8	-922337203685477.5808～922337203685477.5807，用于精确计算，整数部分有 15 位数字，小数部分有 4 位数字
无符号整型	Decimal	14	无小数点时：±79228162514264337593543950335，小数点右边有 28 位数时：±7.9228162514264337593543950335；最小的非零值：±0.0000000000000000000000000001

续表

类型名称	关键字	字节数	取值范围及用途
字符型（变长）	String	10 字节加字符串长度	0 到大约 20 亿
字符型（定长）	String	字符串长度	1 到大约 65,400
日期型	Date	8	100 年 1 月 1 日 到 9999 年 12 月 31 日
逻辑型	Boolean	2	True 或 False
对象型	Object	4	任何 Object 引用
变体型（数字）	Variant	16	任何数字值，最大可达 Double 的范围
变体型（字符）	Variant	22 个字节加字符串长度	与变长 String 有相同的范围
用户自定义	利用 Type	所有元素所需数目	每个元素的范围与它本身的数据类型的范围相同

初学者大可不必马上记住每种数据类型的存储空间大小和范围。2.1.3 节将结合变量的学习对数据类型进行分类，以帮助读者掌握常用数据类型的定义和使用。

2.1.2 常量

常量即常数。在 VB6 中，常量分为系统常量和自定义常量。

1. 常量

在整个程序执行过程中，常量的值保持不变。使用常量可增加代码的可读性和可维护性。

常量是有意义的名字，取代重复出现而难以记住的数字、不变的数值或字符串。在程序运行过程中，常量的值不能像变量那样被修改，也不能被赋以新值。

VB 有许多内部常量，用户也可以建立自定义常量。

（1）内部常量：由系统定义的、可直接使用的常量，可用来在代码中的任何地方代替实际值。表 2-2 列出了 VB 的内部常量。

表 2-2 VB 内部常量

常数	相当于	描述
vbCrLf	Chr(13) + Chr(10)	回车符与换行符结合
vbCr	Chr(13)	回车符
vbLf	Chr(10)	换行符
vbNewLine	Chr(13) + Chr(10)	平台指定的新行字符；适用于当前平台
vbNullChar	Chr(0)	值为 0 的字符
vbNullString	值为 0 的字符串	用来调用外部过程；与长度为 0 的字符串（""）不同
vbObjectError	-2147221504	用户定义的错误号应当大于该值，例如： Err.Raise Number = vbObjectError + 1000
vbTab	Chr(9)	Tab 键
vbBack	Chr(8)	退格符
vbFormFeed	Chr(12)	在 Microsoft Windows 中没有作用
vbVerticalTab	Chr(11)	在 Microsoft Windows 中没有作用

此外，还有表示颜色的常量，如 vbRed 表示红色（vb+颜色的英文单词）；表示键盘代码的常量，如 vbKeyA 代表 A 键（vbKey+键名）等。需要时，可以通过 MSDN 查看“Visual Basic 常数”的相关内容。

例如：VbRed：颜色常量，代表红色。

VbCrLf：代表回车换行。

常量名采用大小写混合的格式，其前缀表示定义常量的对象库名。来自 VB 和 VBA（Visual Basic for Applications）对象库的常量以“vb”开头，例如，vbBSNone 表示窗体无边框；来自数据访问对象库的常量以“db”开头，等等。

（2）用户自定义常量：用 Const 语句声明的常量。

声明格式：[Public|Private] Const 常量名[As Type]＝表达式

其中：常量名是有效的符号名（规则与变量名一样），表达式由数值常数或字符串常数以及运算符组成。

以下是正确的常量声明：

```
Const Pi = 3.14159265358979   '声明了常量 Pi，其值为 3.14159265358979
Public Const MaxPlanets As Integer = 9   '声明了公用整型常量 MaxPlanets
Const ReleaseDate = #1/1/95#
Const CodeName = "Enigma"
Public Const Pi = 3.14, MaxPlanets = 9,WorldPop = 6E+09
Const Pi2 = Pi * 2
```

等号右边的表达式除数字或字符串外，也可以是运算结果为数字或字符串的表达式（表达式中不能包含函数调用）。

2. 用户自定义常量的范围

和变量声明一样，用 Const 语句声明的常量也有作用范围。规则如下：

（1）在过程内部声明的常量，仅存在于过程中。

（2）在模块声明段中声明的常量，对模块中所有过程都有效，但对模块外任何代码都无效。

（3）在标准模块声明段中声明的常量，在整个应用程序中有效，可在 Const 前面放置 Public 关键字，但在窗体模块或类模块中不能声明 Public 常量。

2.1.3　变量

应用程序在运行期间，可以用变量临时存储数据。变量是指程序运行期间其值可以变化的量，实质是程序运行过程中保存临时数据的内存单元，内存单元的名字即变量名。

变量由名字和数据类型组成。变量的名字用于在程序中标识变量和使用变量的值，通过变量名可以引用变量中存储的数据。数据类型则确定变量中能保存哪种类型的数据。

1. 变量的命名规则

（1）变量必须以字母开头，由字母、汉字、数字或下划线组成。

（2）变量的长度不得超过 255 个字符。

（3）变量名不能使用 VB 的关键字。

（4）在同一个范围（即可以引用变量的域，如一个过程、一个窗体等）内，变量名必须是唯一的。

（5）为了增加程序的可读性，可在变量名前加一个类型前缀，以表示该变量的数据类型。

例 2-1 已知某电视机的销售价格为 1500 元/台，编程统计某商店销售电视机的总数量及总销售额。

【分析】

销售前并不知道销售电视机的数量。设计两个变量 IntTVnumber 和 CurTVPrice 保存数量和销售额，根据变量设计一个表达式，不必事先知道实际的输入是多少。每次运行程序时，输入销售电视机的数量后，通过以下代码计算销售额：

```
CurTVPrice = 1500 * IntTVnumber
```

从变量名中可见，CurTVPrice 的数据类型是货币类型，IntTVnumber 的数据类型是整数类型。

2. 变量的声明

在 C 语言中，在使用变量之前必须声明变量；而在 VB 中，变量可以不需要声明就可以直接使用，且变量名不区分大小写。

例如，下面的语句都是正确的：

X=10

Y=20

Z=x+y

但是，声明变量可以节省编程时间，减少因键入操作引起的错误。例如，将 userName 写成了 ueseName，如果没有事先定义 userName 变量，VB 不会自动提示输入错误。如果要强制在 VB 中必须先声明变量才可以使用，可以在代码窗口的首行添加以下一行代码：

```
Option Explicit
```

这行代码表示所有的变量必须先声明才可以使用，否则提示错误，如图 2-1 所示。

图 2-1 编译错误提示

声明变量的语法格式：Dim 变量名 [As 类型]

说明：

（1）Dim 是变量定义的关键字（还可能是变量作用范围修饰关键字 Public 或 Private，“作用范围修饰关键字”将在变量作用范围讨论）。

（2）[As 类型名]是可选项。As 可以解读为“为”，如果省略该语句，变量默认为 Variant（变体）类型。Variant 数据类型在不同场合可代表不同的数据类型。

（3）变量名：必须满足变量命名规则。

通过 Dim 语句进行变量的声明，可以事先将变量通知程序。

以下是变量类型定义语句的几个例子：

```
Dim strName As String        '定义字符串类型变量
```

```
Dim intNum As Integer        '定义整型变量
Dim dblMoney As Double       '定义实型变量
Dim dBirthday As Date        '定义日期类型变量
Dim bolYesNo As Boolean      '定义布尔类型变量
```

3. 变量的声明方式

（1）隐式声明。使用变量前，可以不声明该变量。这种方法虽然方便，但如果拼错了变量名，将会导致一个难以查找的错误。

（2）显式声明。显式声明变量后，系统一旦遇到一个未经声明的变量名时，将发出错误警告。

显式声明变量的方法：在类模块、窗体模块或标准模块的声明段中加入 Option Explicit 语句；或在菜单栏上选择“工具→选项”选项，在“选项”对话框的“编辑器”选项卡中选择“要求变量声明”复选框，可在任何新模块中自动插入 Option Explicit 语句（不会在已经建立的模块中自动插入）。若需要向已有模块添加 Option Explicit，只能用手工方法。

注意：Option Explicit 语句的作用范围仅限于语句所在模块，对每个需要强制显式声明变量的窗体模块、标准模块及类模块，必须将 Option Explicit 语句放在这些模块的声明段中。

4. 变量的数据类型

（1）数值类型变量：包含整数和小数（实数）。

①整数类型。

关键字：Integer（整型）和 Long（长整型）。

当变量保存的数在整数范围-32768～32767 之间时，可以定义为 Integer 类型，超过该范围的整数，则定义为 Long 类型。

例如：Dim a As Integer, b As Long

```
a=100 ：b=40000
a=b    '该行在运行时，将产生“溢出”的运行错误
```

说明：

- 可以在一行中定义多个变量，类型说明之间用逗号分隔，后面的变量定义必须去掉 Dim 关键字。其中，“=”为赋值符，含义是将数值保存到变量；“：”为并行符，可以将多行语句通过“：”号写在一行中，使代码更加紧凑。“'”后面为代码注释内容，程序不会执行该符号后面的内容。
- 如果写为以下形式：

  Dim a, b As Long

则 a 为 Variant 类型（变体型）。根据默认的规定，如果在声明中没有说明数据类型，则变量的数据类型为 Variant。

提示：定义变量时，最好同时声明其类型，程序运行可以避免类型转换所需的时间开销，使得程序的运行效率更高。

- 如果变量 b 不是赋予整数类型的数值（如 b="abc"），将产生“类型不匹配”的运行错误；如果将小数 89.9 赋值给变量 b，则变量 b 的值将是四舍五入后的整数。
- 所有 Integer 类型数据都可以保存到 Long 类型变量中，反之不成立。例如，上面代码中的第三行 a=b。如果不能确定保存的整数将会有多大，应将变量 a 定义为 Long 类型。

②浮点数类型。

关键字：Single（单精度）和 Double（双精度）。

如上所述，若不能确定保存的小数有多大时，应定义为 Double 类型来保存。Double 类型比 Single 类型可以分配更大的内存空间。

例如，分别定义 x、y 两个变量如下：

```
Dim x As Single
Dim y As Double
```

（2）字符类型。

关键字：String。

字符型变量的值包含一连串字符。字符串可以包括字母、数字、空白和标点符号。

字符串有两种：变长与定长的字符串。变长字符串在定义时不指定长度，定长字符串在定义时通过“*”号指定长度。

例如：Dim 变量名 as String [* 长度]

例 2-2 在当前窗体中添加一个 Button1 按钮，如图 2-2 所示；在按钮的 Click 事件中添加以下代码：

```
Private Sub Command1_Click()
   Dim s1 As String, s2 As String * 10
   s1 = "hello!"
   s2 = "1234567890abc"
   MsgBox （s1 & vbCrLf & s2）
End Sub
```

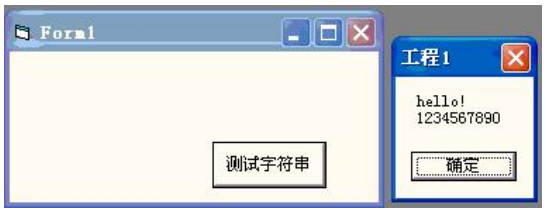

图 2-2 使用字符串的例子

图 2-3 测试长度

【分析】

本例在按钮事件中定义变长字符串 s1 和定长字符串 s2，当对定长字符串赋值超过其定义的长度时，超过的长度将被截取；如果数据不足定义的长度，则以空格填充。变长字符串长度是实际保存的字符串数据长度。可以使用 VB6 中的 Len 函数进行验证，将上面代码中的最后一行改为测试长度的代码：

```
MsgBox (Len(s1) & vbCrLf & Len(s2))
```

运行结果如图 2-3 所示。

- MsgBox 是 VB6 内部函数，作用是弹出信息对话框，信息内容为括号中的参数。本书将经常使用 MsgBox 函数来测试显示某个变量值。
- &是连接符，可以将不同类型的数据，连接成字符串，这里连接 s1 和换行符以及 s2 的值。

（3）日期类型。

关键字：Date。

日期类型变量可以保存的日期数据范围从 100 年 1 月 1 日到 9999 年 12 月 31 日，时间可

以从 0:00:00 到 23:59:59。任何可辨认的文本日期都可以赋值给 Date 变量。

日期类型常量必须使用定界符“#”括起来，当字符串中包含正确的日期格式时，也可以赋值给日期类型变量。例如：

```
Dim d As Date
d = #9/9/2009 12:30:30 PM#
```

或：

```
d="2009-10-09 12:30:30"
```

例 2-3　在标题栏中显示当前的日期和时间，如图 2-4 所示。

在当前窗体的 Load 事件中添加如下代码：

```
Private Sub Form_Load()
   Me.Caption = "当前时间:" & Now()
End Sub
```

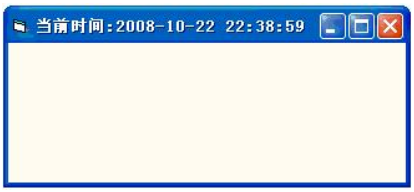

图 2-4　显示当前日期时间

【分析】

以上代码中，Now()为 VB6 中同时取得日期和时间的函数。Date 不但是日期类型的关键字，也是取得当前日期的函数；Time 为取得当前时间的函数。

VB6 还提供以下处理日期类型数据的函数：

- Date()：取得当前日期，Time()取得当前时间。
- Year(日期类型数据)：取得当前日期中的年份。相似的，Month 和 Day 分别取得月和日；而 Hour、Minute 和 Second 则分别取得日期时间类型中的时、分和秒。

例 2-4　在窗体上显示当前日期、时间和星期。

在窗体上添加三个标签 Label 控件，名称默认，界面布局如图 2-5 所示。

在窗体的 Load 事件中，编写如下代码：

```
1   Private Sub Form_Load()
2   Dim d As Date
3   d = Date()
4   Label1.Caption = "今天的日期是:" & _
5   Year(d) & "年" & Month(d) & "月" & Day(d) & "日"
6   Label2.Caption = "当前时间是 " & Time()
7   Label3.Caption = "今天是星期 " & Weekday(d, vbMonday)
8   End Sub
```

【代码说明】

行 3：取日期函数，赋值给变量 d。

行 4：设置 Label 控件的 Caption 属性，属性值为“=”号后面的字符串。用函数 Year、Month 和 Day 分别求得变量 d 中的年、月和日，并连接成中文日期格式。

提示：符号“_”（下划线）是 VB6 中的续行符，前面必须带空格。当程序行太长时，可

以用续行符将本该属于一行的语句分成多行书写，便于阅读代码。

行 6：用函数 Time 取得时间。

行 7：函数 WeekDay 求星期几，其中，参数 d 是日期类型数据，第二个参数为常数。默认星期日为一周的第一天。为了符号中国的习惯，将星期一 vbMonday 设置为一周的第一天。

【运行结果】

如图 2-6 所示。

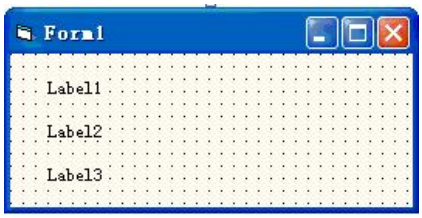

图 2-5　界面布局

图 2-6　运行结果

（4）逻辑类型。

关键字：Boolean。

逻辑类型又称布尔类型，通常用来表示逻辑判断的结果。该数据类型只有两个取值：True 和 False。

例如：Dim bol As Boolean

　　bol=True

（5）变体类型。

关键字：Variant。

如果定义变量时没有指明变量的数据类型，则变量为 Variant 类型；或当未使用 Option Explicit 语句要求强制变量定义时，直接使用的变量也为 Variant 类型。Variant 数据类型没有类型声明字符。

Variant 是一种特殊的数据类型，除定长String数据及用户定义类型外，可以包含任何种类的数据。

例 2-5　对可变类型变量赋值，验证其最终数据类型。

在当前窗体的 Load 事件中，添加如下代码：

```
1   Private Sub Form_Load()
2      Dim x
3      x = 100
4      x = "abc"
5      x = Now
6      MsgBox ("x 的值是:" & x & vbCrLf & "x 的类型是:" & TypeName(x))
7   End Sub
```

【代码说明】

行 2：在定义变量 x 时，并没有指明其类型，因此默认为 Variant 类型。

行 3：将整数 100 赋于变量 x，则此时 x 为整数类型 Integer，值为 100。

行 4：将字符串 abc 赋于变量 x，则此时 x 为字符串类型 String，值为 abc。

行 5：将日期类型数据赋于变量 x，则此时 x 为日期类型 Date，值为当前日期值。

行 6：当程序执行到该行时，将弹出对话框，如图 2-7 所示。运行结果说明了对于可变类

型变量，最终赋予它什么类型数据，则其为该类型变量。本例最终赋予它为日期类型数据，因此它最后将是日期类型变量。TypeName 是求指定变量的数据类型，返回值为类型关键字。

【运行结果】

如图 2-7 所示。

图 2-7　运行结果

提示：Variant 变量虽然使用起来很方便，但不便于阅读程序，而且容易产生混乱，建议编程时最好不使用这类变量。

（6）自定义类型：Type。

如果基本类型不能满足使用要求，可以使用自定义类型。自定义类型相当于 C 语言中的结构体类型，即将各种基本数据类型组合成需要的类型。

语法格式：[Private/Public] Type 类型名

　　变量名 As 基本数据类型

　　……

End Type

注意：如果在窗体的代码窗口中使用自定义类型定义，其定义语句 Type 必须放在声明部分，不能放置在过程或函数内，而且必须使用作用范围关键字 Private（在模块文件中可以省略关键字 Private，默认为 Public 类型），同时，自定义类型内的变量定义不能再使用 Dim 关键字进行基本类型变量的定义，否则会出现编译错误。

自定义类型可以包含任何的基本类型，包括后面讲到的数组以及其他的自定义类型（即嵌套使用自定义类型）。

例 2-6　假设需要定义一个保存各个学生三门课程的成绩，可以自定义名称为 student 的类型，代码如下：

```
Private Type student
    Name As String '姓名
    Sex As Boolean '性别
    ClassName As Integer '班级名称
    Score1 As Single '分数 1
    Score2 As Single '分数 2
    Score3 As Single '分数 3
End Type
```

定义 student 类型后，在程序中可以像使用普通数据类型那样使用 student 类型。

用 student 类型声明变量的语句：Dim stu As student。

给变量 stu 赋值时，必须为该变量的每个成员赋值。自定义类型的使用方式与对象属性的使用方式相同，如图 2-8 所示。

例如，以下代码对 stu 变量进行初始化：

```
stu.intNumber=5
```

```
stu.strName="张三"
stu.strSex ="男"
stu.intAge =18
stu.blnFlag=False
```

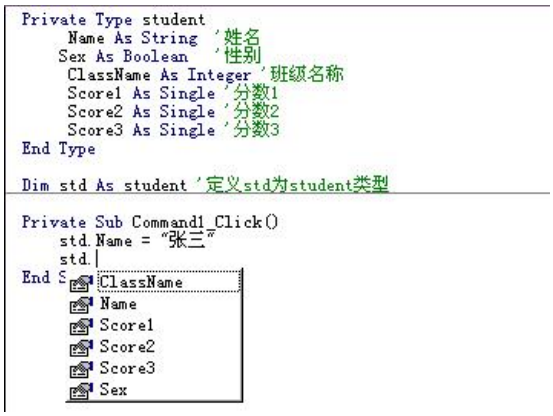

图 2-8　自定义类型的使用

定义自定义类型后，在程序中只要写出变量名和后面的小数点，即可显示该自定义类型的成员变量。

为了简化输入，可用 With 语句对某个对象执行一系列的语句，不必重复指出对象的名称。

With 语句格式：

With 对象名
[语句系列]
End With

功能：在一个单一对象或一个用户定义类型上执行一系列的语句。

若用 With 语句对 stu 变量进行初始化，代码如下：

```
With stu
  .intNumber = 5
  .strName = "张三"
  .strSex = "男"
  .intAge = 18
  .blnFlag = False
End With
```

注意：Type 语句只能在模块级使用，其默认作用域是 Public。如果需要在类模块或对象模块中使用，必须在 Type 关键字前加上关键字 Private。使用类型名和变量名时不要混淆。

在例 2-6 中，student 为类型名，stu 为变量名。With 与 End With 必须配对使用。程序一旦进入 With 语句，对象就不能改变。因此，不能用一个 With 语句设置多个不同的对象。

2.2　运算符与表达式

运算是对数据的加工。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，被运算的数据称为运算量或操作数。

2.2.1　算术运算符和算术表达式

算术表达式又称数值表达式，由算术运算符、数值型常量、变量、函数和圆括号组成，其运算结果一般为数值。

算术运算符是常用的运算符，用于执行简单的算术运算。VB6 常用的算术运算符如表 2-3 所示。

表 2-3　算术运算符

运算符	说明	示例
+（加）	返回两个操作数之和	5 + 4
-（减）	返回两个操作数之差	5 - 4
*（乘）	返回两个操作数之积	5 * 4
/（除）	返回两个操作数之商	5 / 4
\（整除）	将两个数相除并返回以整数形式表示的结果	5\4
Mod（取余）	将两个数相除并只返回余数	10 Mod 5
^（幂）	求以某个数为底、以另一个数为指数的幂	3^3

例 2-7　算术运算符应用范例。

（1）Mod 运算

```
Dim testResult As Double
testResult = 10 Mod 5       '结果为：0
testResult = 10 Mod 3       '结果为：1
testResult = 12 Mod 4.3     '结果为：0，将小数四舍五入成整数
```

注意：如果有一个数是小数，则将小数四舍五入成整数再进行运算。

（2）除运算

```
Dim resultValue As Double
resultValue = 10 / 4        '结果为：2.5
resultValue = 10 / 3        '结果为：3.333333…
```

注意：即使两个操作数都是整数常数，结果始终为浮点类型（Double）。

（3）整除运算

```
Dim resultValue As Integer
resultValue = 11 \ 4        '结果为：2
resultValue = 67 \ -3       '结果为：-22
```

注意：结果是一个整数，表示两个操作数的整数商，余数被丢弃。

（4）幂运算

```
Dim exp1, exp2, exp3, exp4, exp5, exp6 As Double
exp1 = 2 ^ 2                '结果为 4（2 的平方）
exp3 = (-5) ^ 3             '结果为 -125（-5 的立方）
```

2.2.2　字符串运算符与字符串表达式

字符串表达式由连接运算符、字符串常量、字符串变量和字符串函数组成。

VB6 提供两个字符串运算符“+”和“&”，用于连接两个或更多的字符串。

说明：

+：对字符串类型实现连接；对数字类型实现相加运算。

&：可以将所有数据类型的数据连接成字符串。

例 2-8 字符串表达式范例。

```
Dim sampleStr As String
sampleStr = "Hello" & " World"      'sampleStr 的结果为："Hello World"
sampleStr = "Hello" + " World"      'sampleStr 的结果为："Hello World"
sampleStr = "12" + " 34"            'sampleStr 的结果为："1234"
```

注意： “+”运算符尽可能执行算术加法运算，只有当两个表达式均为字符串时，才执行连接操作。

2.2.3 关系运算符和关系表达式

关系运算符比较两个表达式的值，以判断它们之间的大小关系，常用在判断语句中，用来决定程序执行的流程。

常用关系运算符见表 2-4。表 2-4 还列出关系表达式运算结果为 True 或 False 的条件。

表 2-4 关系运算符

运算符	满足条件，表达式值为 True	满足条件，表达式值为 False
<（小于）	表达式 1 <表达式 2	表达式 1 >=表达式 2
<=（小于等于）	表达式 1 <=表达式 2	表达式 1 >表达式 2
>（大于）	表达式 1 >表达式 2	表达式 1 <=表达式 2
>=（大于等于）	表达式 1 >=表达式 2	表达式 1 <表达式 2
=（等于）	表达式 1 =表达式 2	表达式 1 <>表达式 2
<>（不等于）	表达式 1 <>表达式 2	表达式 1 =表达式 2

注意：运算符“=”也用作赋值运算符。

关系运算符两边可以是数值、字符串、日期、逻辑类型的数据。比较运算后得到的运算结果是逻辑类型，即值为 True 或 False。

例 2-9 关系运算符的使用。

```
Dim testResult As Boolean
testResult = 45 < 35
testResult = 45 = 45
testResult = 4 <> 3
testResult = "5" > "4444"
```

例 2-9 中，第一次关系运算的结果为 False，其余关系运算的结果均为 True。

2.2.4 赋值运算符

基本赋值运算符是“=”，用于给变量或属性赋值。

格式：[Let] 变量名＝表达式

　　[Let] 对象.属性 ＝表达式

功能：将“＝”右边表达式的值赋给左边的变量（或对象的属性）。

说明：

（1）“＝”称为赋值号，不是等号。

（2）Let 关键字表示赋值，一般我们在使用时都省略不写。

（3）表达式中可以包含变量、常量、属性值。

（4）赋值号两边的数据类型通常应保持一致。

赋值语句举例：

```
Dim i As Integer
  i = 5                          'i 为整型变量，将整数 5 赋给 i
  i = i + 1                      '在 i=5 的基础上加 1，其结果为 6，再将结果 6 赋给 i
  ch = "abcd"                    '将“abcd”赋给变量 ch
  Text1.Text = ""                '清除文本框的内容
  Form1.Caption = "启动"         '指定窗体 Form1 的标题为“启动”
```

2.2.5　逻辑运算符

逻辑运算符一般用于判断某个条件是否成立，然后根据该条件决定程序执行哪一个分支。逻辑运算符可以对多个逻辑表达式或关系表达式进行逻辑运算，运算结果是逻辑类型。

VB6 提供的常用逻辑运算符见表 2-5。

表 2-5　逻辑运算符

运算符	运算	运算规则	例子	结果
And	与	当两表达式结果均为真时，结果才为真	T And T	T
Or	或	当两表达式结果均为假时，结果才为假	T Or T	F
Not	非	当表达式为假时，结果为真	Not F	T
Xor	异或	两表达式结果相同时，结果为假	T Xor T	F

例 2-10　逻辑运算符的使用。

```
Dim a As Integer = 10,b As Integer = 8,c As Integer = 6
  Dim val As Boolean
  val = (a > b) And (b > c)      '结果为 True
  val = (b > a) And (b > c)      '结果为 False
  val = a > b Xor b              '结果为 True
  val = a > b Xor b > c          '结果为 True
```

2.2.6　运算优先级

当一个表达式中包含多个运算符时，存在运算符的优先次序问题。一般先进行算术运算，其次进行关系运算，最后进行逻辑运算。在同一级运算符中，按运算符出现的顺序从左向右进行计算。

各运算符的优先次序由高到低排列的顺序为：圆括号（由里向外逐层展开）、函数、算术运算符、关系运算符（其优先级相同）、逻辑运算符。

当表达式包含不止一种运算符时，按照以下规则进行计算：

（1）算术运算符和连接运算符的优先级高于比较运算符、逻辑运算符和位运算符。算术

运算符和连接运算符之间的优先级顺序如下：

①求幂（^）。

②正、负标识符。

③乘法和浮点除法（*、/）。

④整数除法（\）。

⑤取余（Mod）。

⑥加法和减法（+、 - ），字符串连接（+）。

⑦字符串连接（&）。

（2）比较运算符之间具有相同的优先级，它们的优先级均高于逻辑运算符，低于算术运算符和连接运算符。

（3）逻辑运算符的优先级均低于算术运算符、连接运算符和比较运算符。逻辑运算符之间的优先级顺序如下：

①非（Not）。

②与（And）。

③或（Or）。

④异或（Xor）。

（4）具有相同优先顺序的运算符按照它们在表达式中出现的顺序从左至右进行计算。

注意：

- 这里的“=”运算符是相等比较运算符，不是赋值运算符。
- 字符串连接运算符“&”不是算术运算符，但在优先级方面与算术运算符属于一组。

例 2-11 以下运算结果等效。

```
Dim a, b, c, d, f, h As Double
a = 8.0
b = 3.0
c = 4.0
d = 2.0
h = 1.0

f = a - b + c / d * h              '结果为 7
f = (a - b) + ((c / d) * h)        '结果为 7
```

2.3 数组与字符串

所谓数组，是保存在一片连续内存单元中的一组数。数组中，每一个元素的名称相同，索引（下标）连续，且具有相同的数据类型。数组单元通过数组的索引（下标）引用，VB6 数组的下标默认从 0 开始。例如，a(0)表示数组 a 的第 1 个元素，a(1)表示数组 a 的第 2 个元素，依此类推。

图 2-9 所示是一个保存 5 个整数的数组 Arr 在内存单元中的示意图。

一般变量在内存中存放的位置没有规律，而且不同类型的变量占用的内存空间也不相同。数组具有相同数据类型、相同变量名、连续的下标。用户可以根据下标区分不同的元素。由于这些变量的变量名相同、下标不同，用户使用循环来访问其中的某些变量或所有变量比较方便。

图 2-9　数组 Arr 在内存单元中的示意

例如，求一个班 40 个学生某门课程的平均分，为了保存这些成绩，可以定义 40 个变量。这样虽然可以解决问题，但显得很繁琐，而用数组可以很好地解决这个问题。

2.3.1　一维数组的声明

定义一维数组的语法：Dim 数组名(上限) [As 数据类型]

例如：Dim arr(4) As Integer

说明：该语句建立一个有 5 个元素的数组 arr，数组的下限为 0，上限为 4。

数组默认的下限为 0，也可以在定义时指定下限，例如：

Dim stdScore(1 To 10) As Single

以上语句建立有 10 个元素的数组 stdScore，数组下限为 1，上限为 10。数组 stdScore 定义后，若对数组元素 stdScore(0)赋值，将出现“下标越界”的运行错误。

如果在某代码模块的所有数组定义时改变默认的下标（定义下标为 1），可以在代码模块的声明部分添加以下开关语句：Option Base 1，则该模块中所有定义的数组下限默认为 1 而不是 0。

VB6 中提供两个函数，以求出数组的上限和下限：

UBound(数组名)：求指定数组的上限。

LBound(数组名)：求指定数组的下限。

例 2-12　求数组上限和下限。

```
Option Base 1
Private Sub Form_Load()
  Dim arr(4) As Integer
  MsgBox (LBound(arr) & UBound(arr)) '结果为 14，共 4 个元素
  '如果去掉第一句 Option Base 1，则结果为 04，共有 5 个元素
End Sub
```

2.3.2　一维数组的使用

定义数组后，可以将数据逐一保存到数组单元中，也可以通过循环将所有元素取出来。

例 2-13　以下代码将数据逐一保存到数组单元中，然后显示第一个数组的第一个元素和第二个数组的第一个元素。

```
1   Private Sub Form_Load()
2     Dim arr(4) As Integer, brr(4) As String
3      Dim tmpArr(1)
4
5      For i = LBound(arr) To UBound(arr)
6       arr(i) = i + 1
7       brr(i) = "A" & i
```

```
8       Next
9
10      tmpArr(0) = arr
11      tmpArr(1) = brr
12
13      MsgBox tmpArr(0)(0) & vbCrLf & tmpArr(1)(0)
14  End Sub
```

【代码说明】

行 2：分别定义一个包含 5 个元素的整型数组 arr 和一个字符串数组 brr。

行 3：定义一个变体类型数组。变体类型数组可以用来保存其他任何类型数组。

行 5～8 ：用循环语句对 arr 和 brr 数组赋值。

行 10：将 arr 数组保存到变体数组 tmpArr 的第一个单元。

行 11：将 brr 数组保存到变体数组 tmpArr 的第二个单元。

行 13：显示变体数组中第一个数组的第一个元素和第二个数组的第一个元素。

图 2-10　运行结果

【运行结果】

如图 2-10 所示。

2.3.3　二维数组与多维数组

二维数组相当于一个二维表格。例如，以下语句声明了一个 3×4，共 12 个元素的二维数组：

```
Dim arr(2,3) as string
```

多维数组的定义类似，维数之间使用逗号分隔。例如，以下语句定义了一个三维数组：

```
Dim mulArr (3, 1 To 10, 1 To 15)
```

数组的大小为 4 × 10 × 15，元素总数为三个维数的乘积 600。

二维数组的操作通常使用循环语句，如以下代码所示：

```
Private Sub Form_Load()
   Dim i As Integer, j As Integer
   Dim arr(0 To 2, 0 To 3) As Double
   For i = 0 To 2
      For j = 0 To 3
         arr(i, j) = i * 3 + j
      Next j
   Next i
End Sub
```

注意：增加数组的维数时，数组所占的存储空间会大幅度增加。因此，要慎用多维数组。使用 Variant 类型数组时更要格外小心，因为它们需要更大的存储空间。

2.3.4　动态数组

在某些情况下，数组的长度不能事先确定，希望能够在运行时改变数组的大小。动态数组可以在任何时候改变大小。因此，在系统中可短时间内使用一个大数组，当不使用该数组时，可将内存空间释放。动态数组具有灵活、方便的特点，可以有效地管理内存。

1. 动态数组的声明

先用 Dim 语句声明不定长度的数组，例如：

```
Dim DyArray ()  '将数组声明为一个空维数组，即可将数组声明为动态数组。
```

2. 用 ReDim 语句动态声明某一长度的数组

ReDim 语句可重新为数组分配内存空间（即用 ReDim 语句分配实际的元素个数），例如：

```
ReDim DyArray(4 to 12)  'ReDim 语句给数组 DyArray 分配一个有 9 个元素的矩阵
```

说明：

（1）可用变量设置动态数组的边界，例如：

```
Dim n As Integer
Dim A() As String
 …
n=2*n
ReDim A(n)
```

（2）ReDim 语句是可执行语句。应用程序运行时，执行 ReDim 语句将删除数组中原有的数据。

（3）每个 ReDim 语句只能改变数组的长度及上下界，但不能改变数组的维数。

（4）Erase 语句用于删除动态数组并释放动态数组占用的内存空间，例如：Erase A。

（5）每次执行 ReDim 语句时，当前存储在数组中的值都会全部丢失。若希望改变数组大小又不丢失数组中的数据，可用带 Preserve 关键字的 ReDim 语句为数组重新分配空间。

可用 UBound 函数引用上界，使数组扩大，增加一个元素，而现有元素的值不会丢失，例如：

```
ReDim Preserve A（UBound（A）＋1）
```

使用 Preserve 关键字时，只能改变多维数组中最后一维的上界；如果改变了其他维或最后一维的下界，运行时将会出错。

例 2-14　求任意多个从键盘输入的整数之和。

【建立工程】

新建工程，按默认名称保存工程为“工程 1”，窗体名称保存为 Form1。

在窗体上添加两个按钮，Button1 和 Button2，标题属性 Caption 分别改为“录入数据”和“查看结果”，如图 2-11 所示。

图 2-11　界面布局

【程序代码】

```
1   Dim arrNum() As Integer
2   Dim TmpNum As Integer
3
4   Private Sub Form_Load()
5       ReDim arrNum(0)
6   End Sub
7
8   Private Sub Command1_Click()
9       TmpNum = Val(InputBox("请输入一个整数:", "输入一个整数", 0))
10
11     ReDim Preserve arrNum(UBound(arrNum) + 1)
12     arrNum(UBound(arrNum)) = TmpNum
13  End Sub
```

```
14
15  Private Sub Command2_Click()
16      Dim sum As Integer
17      For i = 1 To UBound(arrNum)
18          sum = sum + arrNum(i)
19      Next
20
21      MsgBox "你输入了[" & UBound(arrNum) & "]个数,计算结果为:" & sum
22  End Sub
```

【代码说明】

行 1：定义变体数组（注意：可变数组不能指定上限）。

行 2：临时保存录入的整数变量。

行 4～6：初始化数组上限和下限为 0。

行 9：通过 InputBox 函数弹出对话框，接受用户输入数据。函数中各参数对应对话框的位置，参见图 2-12，该对话框返回字符串类型数据，因此必须通过 Val()函数转换为数值类型保存到整型变量 TmpNum 中；单击“取消”按钮时，该对话框将返回空字符串，Val()函数可以将一切非数字字符串的内容转换为数值 0。

行 11：在数组原来的长度基础上，增加一个元素。如果是首次执行该语句，那么第一个输入的值将保存到数组中下标为 1 的元素中。计算时要注意，下标为 0 的元素其实没有保存数据，参见行 17 的分析。

行 12：将整数保存到当前数组最后一个元素中。

行 17～19：将数组从第 2 个元素（下标为 1 的元素）到最后一个元素的整数累加起来，保存到 sum 变量中，最后显示计算结果，如图 2-13 所示。

图 2-12　InputBox 对话框

图 2-13　计算结果

例 2-15　随机生成 10 个 1～100 范围之间的整数，并进行从大到小排序。

【分析】

（1）利用随机函数产生 10 个两位整数，存入一维数组 arr 中。

（2）排序的基本思想：先从要排序的数中选择最大的数并放在第一个位置，然后从剩下的数中选择最大的数放在第二个位置，如此类推，最后在剩下的两个数中选择大的数放在倒数第二个位置，剩下的一个数放在最后位置，排序完成。

（3）产生指定范围的随机数。

为了生成某个范围内的随机整数，可使用以下公式：

Int((upperbound - lowerbound + 1) * Rnd + lowerbound)

这里，upperbound 是随机数范围的上限，而 lowerbound 则是随机数范围的下限。

【程序代码】

```
1   Private Sub Form_Load()
```

```
2     Dim I, j, temp As Integer, a(1 To 10) As Integer
3     Dim S As String
4
5     S = "产生的 10 个随机整数如下：" & vbCrLf
6
7     For I = 1 To 10
8       a(I) = Int((100 - 1 + 1) * Rnd + 1)  '产生 1～100 之间的整数
9       S = S & a(I) & " "
10    Next
11
12     For I = 1 To 9          '进行排序
13      For j = I + 1 To 10
14        If a(I) < a(j) Then
15          temp = a(I)
16          a(I) = a(j)
17          a(j) = temp
18        End If
19      Next j
20    Next I
21
22    S = S & vbCrLf & vbCrLf & "排序后输出 10 个整数如下：" & vbCrLf
23
24    For I = 1 To 10
25      S = S & a(I) & " "
26    Next
27    MsgBox S
28  End Sub
```

【代码说明】

本例使用了循环结构。如果对行 12～20 的排序算法难以理解，请学习循环语句后再回到该例来理解。

行 5：保存用于对比排序前后的数组元素结果的字符串变量。

行 7～10：产生随机数，并保存到数组中去。本例每次运行的结果都是一样的，为了产生完全不同的随机数，可以在使用 Rnd 函数前调用一次初始化随机数生成器 Randomize 函数。如在行 6 中加入该函数语句，则每次运行程序产生的随机数都将不一样。

行 12～20：排序算法。

【运行结果】

如图 2-14 所示。

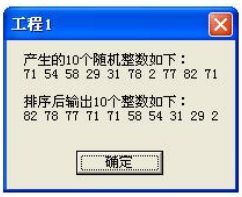

图 2-14　运行结果

2.3.5　控件数组

所谓控件数组，是一组类型相同的控件，使用相同的名称，通过数组的方式来使用。控件数组可以使代码的编写更加简洁。

在窗体 Form1 上放置一个命令按钮控件 Command1，复制该控件并粘贴到当前窗体，如图 2-15 所示。此时，将弹出如图 2-16 所示的提示框，如果选择“是”按钮，窗体上将出现两个名称（Name）为 Command1 的按钮控件。

VB6 通过控件数组的下标来区分数组中不同控件的操作。例如，以下代码可以实现：单击两个按钮时，区分是对哪个按钮的操作。

图 2-15　复制 Command1 控件

图 2-16　提示是否建立控件数组

```
Private Sub Command1_Click(Index As Integer)
    If Index = 0 Then
        MsgBox "你点击了第 1 个按钮"
    ElseIf Index = 1 Then
        MsgBox "你点击了第 2 个按钮"
    End If
End Sub
```

事件参数中的 Index 表明单击控件数组的下标。其他类型控件数组的判断方法类似。

VB6 中，除了在程序设计时产生控件数组外，还可以在程序运行时动态建立控件数组。实现方法如下：

（1）在窗体上建立控件并设置该控件的相应属性，其中，设置该控件的 Index 属性值为 0，表示该控件为数组，建立控件数组的第一个元素。

（2）编写代码时，通过 Load 方法添加所需的数组元素个数，元素在窗体上的位置通过对 Top 和 Left 属性赋值确定。

例 2-16　动态创建和卸载控件的示例。

在窗体上添加两个按钮控件，分别设置属性如图 2-17 所示。单击“动态建立按钮”按钮时，自动创建标题为 2～10 的按钮，并且 5 个按钮为一行分行显示，运行结果如图 2-18 所示。再次单击该按钮时，卸载刚建立的动态按钮。

【建立工程】

新建工程，按默认名称保存工程为“工程 1”，窗体名称保存为 Form1。

按图 2-17 进行界面布局。

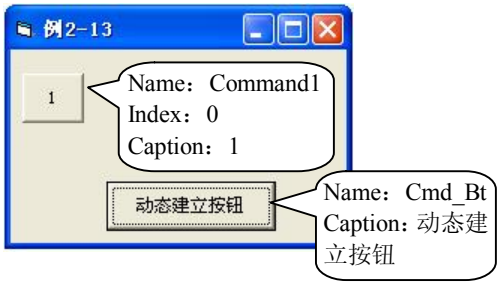

图 2-17　界面布局

【程序代码】

```
1   Private Sub Cmd_Bt_Click()
2       Dim CurTop As Integer, CurLef As Integer
3
4       If Command1.Count = 1 Then
5           Cmd_Bt.Caption = "卸载动态按钮"
6
```

```
7           CurTop = Command1(0).top
8           CurLef = Command1(0).Left
9
10          For i = 1 To 9
11                 Load Command1(i)
12                 Command1(i).Caption = i + 1
13                 Command1(i).Visible = True
14                 CurLef = CurLef + Command1(i - 1).Width
15
16                  If i Mod 5 = 0 Then
17                     CurTop = Command1(0).top + Command1(0).Height
18                     CurLef = Command1(0).Left
19                 End If
20
21                 Command1(i).Left = CurLef
22                 Command1(i).top = CurTop
23          Next
24      Else
25          Cmd_Bt.Caption = "动态建立按钮"
26          For i = 1 To 9
27             Unload Command1(i)
28          Next
29      End If
30  End Sub
```

【代码说明】

行 4：如果当前命令按钮数为 1，则执行创建操作。动态创建的命令按钮，以已经存在的数组命令按钮为基础创建。创建动态命令按钮前，必须将存在的命令按钮的 Index 属性设置为整值（如 0），使其成为数组按钮中的第 1 个按钮。

行 5：改变命令按钮的标题 Caption。

行 7，8：初始化新创建的动态命令按钮在窗体上的位置。

行 11：创建第 i 个动态命令按钮。

行 12：设置新命令按钮的标题。

行 13：使动态创建的命令按钮在窗体上可见，默认是不可见的。

行 14：动态创建的命令按钮的 Left 位置，由于分行后将再次初始化，这里先保存到变量中。

行 16～19：5 个命令按钮建立后，分行显示第 6 个及其后的命令按钮。

行 21，22：动态命令按钮在窗体上的位置。

行 27：卸载已经建立的第 i 个动态命令按钮。

【运行结果】

如图 2-18 所示。

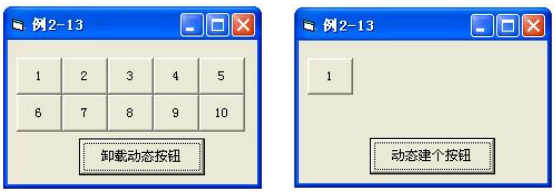

图 2-18　运行结果

2.3.6　字符串

字符串类型的数据在程序设计中经常使用，本节主要介绍实现字符串处理的一些常用函数。

1. 大小写转换函数

UCase(s)：返回将字符串 s 转换为大写后的值，s 本身的值不变。

LCase(s)：返回将字符串 s 转换为小写后的值，s 本身的值不变。

例 2-17 显示字符串变量 s1 大小写转换前后的结果。

【程序代码】

```
Private Sub Form_Load()
    Dim s1 As String, s2 As String
    s1 = "abcDeF"
    s2 = UCase(s1) & vbCrLf & LCase(s1) & vbCrLf & s1
    MsgBox s2
End Sub
```

【运行结果】

如图 2-19 所示。

图 2-19 运行结果

2. 拆分字符串函数 Split

返回按指定的分隔符，将字符串拆分成一个数组。

语法：Split(s,分隔符)

注意：以下的例子需要在窗体上添加 Command1 命令按钮，并在命令按钮的 Click 事件中添加代码。

例 2-18 按逗号分隔字符串 s。

【程序代码】

```
Private Sub Command1_Click()
    Dim s As String, arr
    s = "A,B,C,D"
    arr = Split(s, ",")
    For i = 0 To UBound(arr)
        MsgBox arr(i)
    Next
End Sub
```

【运行结果】

依次弹出内容为 A、B、C、D 的对话框。

注意：保存 Split 返回值的变量 arr 不能定义为数组，只能为变体类型变量，即使已经知道了返回值是个数组数据。

3. 组合字符串函数 Join

组合字符串正好与拆分字符串作用相反，是将字符串数组元素，按指定的分隔符连接成字符串。

例 2-19 将数组 arr 元素连接成字符串。

【程序代码】

```
Private Sub Command1_Click()
    Dim s As String
    Dim arr(3) As String
    arr(0) = "A"
    arr(1) = "B"
    arr(2) = "C"
    arr(3) = "D"
   s = Join(arr, ",")
   MsgBox s
End Sub
```

【运行结果】

图 2-20　运行结果

如图 2-20 所示。

4. 替换字符串函数 Replace

将字符串中的字符或部分字符串替换为给定的新字符串的值。

语法：Replace(原字符串,被替换的字串,替换为新字符串)

例 2-20　将 s1 中的“中国”替换为“中华人民共和国”。

【程序代码】

```
Private Sub Command1_Click()
    Dim s1 As String, s2 As String
    s1 = "我爱中国"
    s2 = Replace(s1, "中国", "中华人民共和国")
    MsgBox s2 & vbCrLf & s1
End Sub
```

【运行结果】

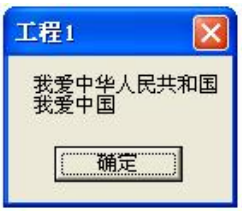

图 2-21　运行结果

如图 2-21 所示。

5. 取子符串函数 Mid

语法：Mid(s,start,length)

从 s 的指定位置 start 开始取长度为 length 的子串，如果省略了参数 length，则取 s 字符串全部。其中，start 从 1 开始计算。

例 2-21　Mid 函数使用的各种情况。

【程序代码】

```
Private Sub Command1_Click()
    Dim s1 As String, s2 As String, _
    s3 As String, s4 As String

    s1 = "ab 中华人民共和国,Ok"
    s2 = Mid(s1, 1, 1)
    s3 = Mid(s1, 3, 7)
    s4 = Mid(s1, 3)
    MsgBox s2 & vbCrLf & s3 & vbCrLf & s4
End Sub
```

【运行结果】

图 2-22　运行结果

如图 2-22 所示。

还可以使用 Left 和 Right 函数取字符串。

例 2-22　取子串。

【程序代码】

```
Private Sub Command1_Click()
    Dim s1 As String, s2 As String, _
    s3 As String, s4 As String
    s1 = "ab 中华人民共和国,Ok"
    s2 = Left(s1, 4)
    s3 = Left(s1, 100)
    s4 = Right(s1, 4)
MsgBox s1 & vbCrLf & s2 & vbCrLf & s3 & _
vbCrLf & s4
End Sub
```

【运行结果】

图 2-23　运行结果

如图 2-23 所示。

2.4 程序结构

VB 是面向对象的程序设计语言，尽管采用了事件驱动的机制，但在设计过程的程序代码时，仍需要对过程的流程进行控制，即采用结构化程序设计的方法完成过程代码设计。

结构化程序设计方法把程序的结构分为顺序、选择和循环三种基本结构。程序设计的基本原则是尽量避免语句之间的跳转，自顶向下、逐步求精、模块化设计等。目的是在团队开发大型软件时，实现高效率、高可靠性。

2.4.1 顺序结构

顺序结构是程序设计中最简单、最常用的基本结构。该结构中，各语句的执行按照语句的书写次序逐条顺序执行，是任何程序的基本结构。

例 2-23 一个最简单的顺序结构程序。

【程序代码】

```
Private Sub Command1_Click()
   Dim x As Integer
   Dim y As Integer
   x = 10
   y = 20
   MsgBox "x+y=" & x + y
End Sub
```

【代码说明】

单击命令按钮 Command1 时，将逐条执行 Click 事件内的语句。

2.4.2 分支结构

通常，计算机按语句在程序中书写的顺序执行语句。但在许多情况下，语句执行的顺序依赖于输入数据或中间运算结果。在这种情况下，必须根据某个变量或表达式的值作出判定，以决定执行哪些语句和跳过哪些语句不执行。VB 应用程序中，可以通过分支结构实现这种功能。分支结构又称条件结构。

VB 提供三种分支语句，即：

（1）If…Then 语句

（2）If…Then…Else 语句

（3）Select Case 语句

1．If…Then 语句

格式 1：If 条件 Then 语句

格式 2：If 条件 Then

　　语句块

　End If

功能：如果条件成立（为真），则执行 Then 后面的语句或语句块。否则，执行 If 语句的下一条语句。流程图如图 2-24 所示。

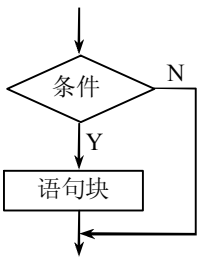

图 2-24 If…Then 语句流程图

说明：

（1）条件一般为关系表达式、逻辑表达式。

（2）格式 1 中的语句只能是一个语句或写在一行上用“：”分隔的多个语句。

（3）语句块指一行或多行语句。

例 2-24　比较 x、y 中的数，将大数放在 x 中，小数放在 y 中。

【程序代码】

用格式 1：

```
If x<y Then w=x:x=y:y=w    '引入中间变量 w，使 x、y 中的值进行交换
```

用格式 2：

```
If x<y Then
   w=x
   x=y
   y=w
End If
```

例 2-25　已知 x、y、z 三个数，要求按从小到大的顺序进行排列。

【程序代码】

```
Private Sub Command1_Click()
  X = 8: y = 9: z = 3                  '设定 x、y、z 的初值
  If x > y Then w = x: x = y: y = w '将 x、y 中的小数存放于 x 中
  If x > z Then w = x: x = z: z = w 'x 中存放 x、z 中的小数,此时 x 为三个数中的最小数
  If y > z Then w = y: y = z: z = w 'y 中总是存放 y、z 中的小数
  MsgBox x & "," & y & "," & z
End Sub
```

程序运行时，单击 Command1 按钮，将弹出按从小到大的顺序显示 x、y、z 值的对话框。

2. If…Then…Else 语句

格式 1：If 条件 Then 语句 1 Else 语句 2

格式 2：If 条件 Then
　　语句块 1
　Else
　　语句块 2
　End If

格式 3：If 条件 1 Then
　　语句块 1
　ElseIf 条件 2 Then
　　语句块 2
　Else
　　语句块 3
　End If
End If

功能：如果条件成立（为真），则执行 Then 后面的语句 1（或语句块 1）；否则，执行 Else 后面的语句 2（或语句块 2）。流程图如图 2-25 所示。

说明：

（1）格式 3 构成了条件语句的嵌套，若条件 1 不成立，则判断条件 2 是否成立。

（2）ElseIf 不能写成 Else If。在 Else 语句中又出现 If 语句，称为 If 语句的嵌套。使用 If 的嵌套结构时，若 If 语句不写在一行上，则必须与 End If 配对。多个 If 嵌套时，End If 与其最接近的 If 配对。

例 2-26 在火车站托运行李时，需要根据行李的重量按不同标准收费。若重量不超过 50kg，按每公斤 0.3 元收费；若重量超过 50kg，其中 50kg 按每公斤 0.3 元收费，超出 50kg 部分按每公斤 0.6 元收费。要求设计程序，根据输入的托运行李重量计算并输出托运费。

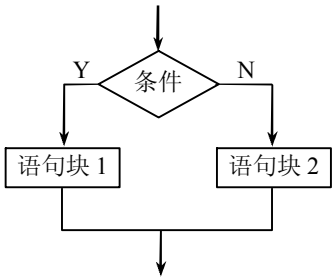

图 2-25 If…Then…Else 语句流程图

【分析】

设托运行李重量为 weight，托运费为 pay，计算公式如下：

$$\begin{cases} pay=weight*0.3 & \text{当 } weight \leqslant 50 \\ pay=50*0.3+(weight-50)*0.6 & \text{当 } weight>50 \end{cases}$$

程序运行时，单击窗体，利用 inputBox 函数接收输入的行李重量；计算后，在窗体上显示行李托运费。

【程序代码】

```
Private Sub Command1_Click()
    Dim weight, pay
    weight = InputBox("请输入行李的重量")
    If weight < 50 Then
      pay = weight * 0.3
    Else
      pay = 50 * 0.3 + (weight - 50) * 0.6
    End If

    MsgBox "行李托运费为：" & pay & "元"
End Sub
```

【运行结果】

如图 2-26 所示。

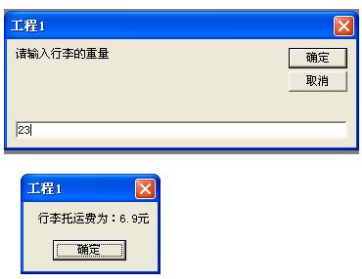

图 2-26 例 2-26 程序运行界面

例 2-27　以下程序段将学生成绩的百分制转换成 A、B、C、D、E 等级制。

【程序代码】

```
Private Sub Command1_Click()
   score = Val(InputBox("请输入分数值："))
   If score > 90 Then
        grade = "A"
   ElseIf score >= 80 Then
        grade = "B"
   ElseIf score >= 70 Then
        grade = "C"
   ElseIf score >= 60 Then
        grade = "D"
   Else
        grade = "E"
   End If

   MsgBox "分数等级为：" & grade
End Sub
```

3. Select Case 语句

Select Case 语句也可以实现选择，有时比 If 语句还简单、直观。

格式：Select Case 表达式

```
Case 表达式列表 1
   语句块 1
Case 表达式列表 2
   语句块 2
  …
[Case Else
   语句块 n+1]
End Select
```

功能：根据测试表达式的值，选择执行不同的语句。若表达式的结果与 Case 子句中表达式列表 i 的值相等，则执行语句块 i。

说明：表达式列表与表达式的类型必须一致。

表达式列表可以是下列形式之一：

（1）一个具体的值，如 5。

（2）用逗号分隔的一组数据，如 1,3,4,6。

（3）用 To 表示的一个区间，如 3 To 9。

（4）用 Is 代表测试值，后带关系运算符和比较值，如 Is>7。

例 2-28　将例 2-27 中用 If 嵌套结构实现的功能改用 Select Case 语句实现。

【程序代码】

```
Select Case  score
   Case is>=90
       Grade="A"
   Case 80 to 89
```

```
        Grade="B"
    Case 70 to 79
        Grade="C"
    Case 60 to 69
        Grade="D"
    Case Else
        Grade="E"
Case Select
```

2.4.3 循环结构

程序设计中，常常会遇到一些计算不复杂但需要反复多次计算的问题，如显示九九乘法表等。这种问题可用循环结构来实现。

循环结构程序设计实际上是一种特殊结构的选择结构，程序根据循环条件的判断结果决定是否执行循环体语句。循环体语句是循环结构中的处理语句块，用来执行重复的任务。

循环结构有多种形式，可以根据需要选择。无论采用何种类型的循环结构，循环体的执行次数必须视其循环类型与条件而定，且必须确保循环的重复执行能在适当的时候得以终止（即非死循环）。此外，循环结构还能嵌套使用。

VB 的循环语句有以下四种形式。

（1）For ... Next

（2）For Each ... Next

（3）Do…Loop

（4）While…Wend

1. For ... Next 结构

格式：For 循环变量 [As 类型]=初始值 To 终止值 [Step 步长]

　　语句块

　　[Exit For]

Next

功能：For 循环语句是一种定次循环语句，即在设计程序时已经可以确定需要循环的次数。执行 For 语句时，首先判断循环变量当前值是否大于终值，若不是，则进入循环执行循环体语句。执行到 Next 语句时，将循环变量的值加上步长值，返回到 For 语句（循环开始处）；否则，退出循环执行 Next 语句之后的语句。

说明：

- 循环变量：整型变量，用于记录循环次数的数值型变量。
- 类型：说明循环变量的类型。
- 初始值：循环变量的起点值。
- 终止值：决定何时退出循环。
- 步长值：数值类型。步长值决定每执行一次循环，循环变量的增量。每次循环后，循环变量的增量（默认）增 1，可以是正数或负数。
- Exit For：控制转移到 For 循环外，即退出循环。
- 初始值、终止值和步长变量在循环前必须赋值，且为整数。

例 2-29　从 1 累加到 100，显示累加结果。

【程序代码】

```
Private Sub Command1_Click()
    Dim sum As Integer
    sum = 0
    For i = 1 To 100
       sum = sum + i
    Next
    MsgBox ("计算结果为:" & sum)
End Sub
```

图 2-27　运行结果

【运行结果】

如图 2-27 所示。

例 2-30　显示九九乘法表。

For 循环语句可以嵌套使用。本例中，被乘数从 1 变化到 9，乘数也从 1 变化到 9，采用两层循环嵌套。

【程序代码】

```
Private Sub Command1_Click()
    Dim i As Integer, j As Integer
    Dim s As String

    For i = 1 To 9
      For j = 1 To i
         s = s & i & "*" & j & "=" & i * j & " "
      Next
      s = s & vbCrLf
    Next
    MsgBox (s)
End Sub
```

【运行结果】

如图 2-28 所示。

图 2-28　九九乘法表

2. For Each... Next 结构

For Each...Next 循环与 For...Next 循环类似。该语句对数组或对象集合中的每一个元素重复一组语句，而不是重复语句一定的次数。在不知道一个集合有多少元素的场合，For Each...Next 循环非常有用。

格式：For Each element In 对象或集合、数组

语句块
[Exit For]
Next

功能：可以列出数组中的每一个元素，或一个集合中的每一个元素。

其中：

- 对集合，element 只能是 Variant 类型变量，或一般的 Object 类型变量，或“对象浏览器”中列出的对象。
- 对数组，element 只能是 Variant 类型变量。

例 2-31 用 For Each 结构显示数组中每一个元素。

【程序代码】

```
Private Sub Command1_Click()
    Dim arr(9) As Integer
    Dim s As String
    '为了测试，先对数组赋值
    For i = 0 To UBound(arr)
        arr(i) = i + 1
    Next
    '使用 For Each 结构列举出数组的每一个元素
    For Each a In arr
        s = s & a & " "
    Next

    MsgBox "数组元素如下:" & s
End Sub
```

【运行结果】

图 2-29 运行结果

如图 2-29 所示。

例 2-32 用 For Each 结构显示窗体上所有控件对象的信息。

【建立工程】

新建工程，按默认名称保存工程为“工程 1”，窗体名称保存为 Form1。

界面布局如图 2-30 所示。

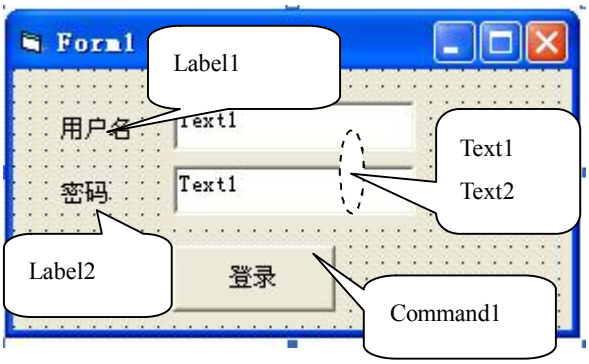

图 2-30 界面布局

【程序代码】

```
Private Sub Command1_Click()
    For Each c In Me.Controls
```

```
        s = s & "控件名称是:" & c.Name & _
                " 控件类型是:" & TypeName(c) & vbCrLf
    Next
    MsgBox "窗体包含以下控件:" & s
End Sub
```

【代码说明】

TypeName 函数求取变量的类型名。

【运行结果】

如图 2-31 所示。

图 2-31　运行结果

3．While 结构

While 循环结构有几种形式，如表 2-6 所示。

表 2-6　While 循环结构的形式

While 结构 1	While 结构 2	While 结构 3
While <条件> 语句块 Wend	Do While<条件> 语句块 [Exit Do] Loop	Do 语句块 [Exit Do] Loop While <条件>

表中 While 结构 1 和 While 结构 2 实现相同的功能，只是写法不同，这种结构称为“当型”循环结构。即当满足条件时，执行循环体（语句块）。

While 结构 3 是“直到型”循环结构，无论是否满足条件，至少执行循环体（语句块）一次，直到不满足条件才退出循环。

注意：While…Wend 没有 Exit While 的语句，Do...Loop 语句提供了一种结构化与适应性更强的方法来执行循环。

例 2-33　求从 1 累加到 100 的过程中，结果超过 1000 的数是多少。

【程序代码】

```
Private Sub Command1_Click()
    Dim i As Integer
    Dim sum As Integer

    Do While 1 '形式上是无限循环，在循环体中必须有退出的判断
      sum = sum + i
      If sum > 1000 Then Exit Do '当满足条件时,退出循环
      i = i + 1
   Loop
   MsgBox ("当结果超过时,该数为:" & i)
End Sub
```

2.5 变量作用范围

每个变量都有自己的作用域，即变量的作用范围。如果在变量的作用域之外使用该变量，将产生编译错误。VB 允许在声明变量时指定其有效范围。一般来说，声明变量时，尽量选择较小的作用域，保持较小的作用域有助于节省内存并减少代码指向非法变量的可能性。

在一个过程内部声明的变量，只有该过程内的代码才能访问或改变其值。若要使变量的值在同一模块的所有过程中都有效，甚至在整个应用程序的所有过程都有效，必须在声明变量时指定其作用范围。

1. 过程级变量

只有在声明该变量的过程中才能被使用，又称局部变量。所有过程或函数中定义的变量都是局部变量。

用关键字“Dim”声明过程级变量。这种变量只有在执行该过程时才存在，过程结束，该变量的值即不存在，变量占用的内存被释放；下一次再执行该过程时，其中的局部变量重新初始化。

由于在一个过程中声明的变量只有在本过程中才可以使用，在其他过程中无法使用，因而在不同的过程中可以使用相同的变量名。

例 2-34 在窗体上添加一个 Command1 命令按钮，并添加以下代码，单击命令按钮时，显示的结果是什么？

【程序代码】

```
Private Sub Command1_Click()
   MsgBox i
End Sub

Private Sub Form_Load()
   Dim i As Integer
   i = 100
End Sub
```

【分析】

程序启动时，首先触发窗体的 Load 事件，执行 Load 事件中的代码，即定义变量 i 并使其值为 100。当窗体显示出来时，该过程已经结束，则过程变量 i 已经不存在。

单击 Command1 命令按钮时，执行其 Click 事件中的代码。该代码中的变量 i 不再是 Load 事件过程中的变量 i，而是新定义的变体类型的变量 i。因此，单击命令按钮时，将显示空对话框，然后该变量从内存中消失。

2. 模块级变量

在模块的声明段中，使用 Private 或 Dim 关键字声明的变量，其作用范围是该模块中的所有过程。Private 表示私有的，相对应的关键字是公有的 Public；用 Private 声明的变量只能在本模块中使用，其他模块的代码不可用；用 Public 声明的变量不但本模块可以使用，其他模块也可以使用。

窗体代码文件中，在所有过程或函数之外（代码声明部分）定义的变量，即为模块级变量。若在任何过程中改变了该变量，该变量最终的值是最后一次改变的值。

模块级变量定义中，Private 和 Dim 之间没什么区别。另外，在过程（如事件过程和后面讲到的自定义过程）和自定义函数中，不能用 Private 和 Public 关键字声明变量。

例 2-35　在窗体上添加三个命令按钮，名称分别为 Command1、Command2 和 Command3。分析以下代码的执行结果：

```
Dim a  As Integer

'对 a 赋值
Private Sub Command1_Click()
   a = 100
End Sub

'对 a 赋值
Private Sub Command2_Click()
   a = 200
End Sub

'查看 a 的值
Private Sub Command3_Click()
   MsgBox a
End Sub
```

【分析】

程序运行时，如果单击 Command1 后再单击 Command2，则单击 Command3 时，变量 a 的值将是 200；然后再单击 Command1，变量 a 的值是 100。即无论按什么顺序单击 Command1 或 Command2，再单击 Command3 时显示的变量 a 的值都是最后一次赋的值。

例 2-36　先在窗体 Form1 定义用户名变量 UserName，然后在窗体 Form2 中显示出来。

【分析】

添加两个窗体，名称分别为 Form1 和 Form2。为了在窗体模块 Form2 中使用窗体模块 Form1 中定义的变量，该变量必须使用关键字 Public 定义。

在现有工程中添加窗体的方法：在菜单栏上选择“工程→添加窗体”选项，弹出“添加窗体”对话框，在“新建”选项卡中选择“窗体”，单击“打开”按钮。

【程序代码】

（1）窗体 Form1 的代码

```
Public UserName As String
Private Sub Form_Load()
   UserName = "张三"
End Sub

Private Sub Command1_Click()
   Form2.Show
End Sub
```

（2）窗体 Form2 的代码

```
Private Sub Form_Load()
   Me.Caption = "欢迎你," & Form1.UserName
End Sub
```

工程中的代码如图 2-33 所示。

【代码说明】

如果要使用一个窗体模块中的公有变量，需要通过如下格式引用：

窗体名.变量名

【运行结果】

如图 2-32 所示。

图 2-32　运行结果

从上面的例子中可以看出，一个窗体使用另一个窗体定义的变量时，需要使用“窗体名.变量名”格式。实际上，如果一个变量（后面还会使用到公有的自定义过程和函数）需要在多个模块中使用到，一般将该变量定义在全局的标准模块文件中。

标准模块文件建立的方法：在菜单栏上选择“工程→添加模块”选项，弹出“添加模块”对话框，在“新建”选项卡中选择“模块”，单击“打开”按钮。

模块文件默认文件名为 Module1，内容为空白。

例 2-37　改写例 2-36，将变量 UserName 定义为标准模块文件中的变量。

【分析】

在例 2-36 的基础上，新添加标准模块文件 Module1，将 Form1 中定义的代码 Public UserName As String 移动到 Module1 中，如图 2-33 所示。

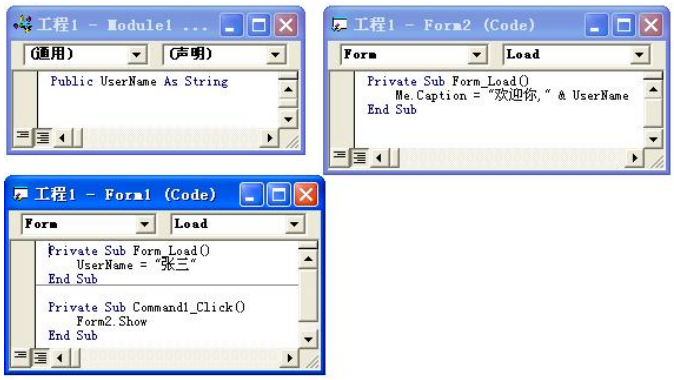

图 2-33　工程中的代码

【运行结果】

和例 2-36 一样。

可见，在 Module1 文件中定义的公有变量，其他模块都可以直接使用，不需要在变量名前加“模块名”。在 Module1 中用 Private 或 Dim 定义的变量，只能在本模块中使用，其他模块是不能使用的。

3. 静态变量

用 Static 关键字声明的局部变量，变量值在程序运行中一直有效。将局部变量定义为静态时，可保留变量的值。若希望过程中所有的局部变量为静态变量，可在过程的起始处加上 Static 关键字。

例如：Static Sub RunningTotal (num)

对于 Dim 声明的局部变量以及声明局部变量的过程，仅当过程在执行时，这些变量才存在。通常，执行一个过程时，所有局部变量将重新初始化。若需要过程结束后仍然保持过程中某些局部变量的值，应将这些变量声明为静态变量。

[练习] 试分析下面两段代码中，每次单击命令按钮 Command1 时，变量 num 的值。

（1）代码 1

```
Private Sub Command1_Click()
   Static num As Integer
   num = num + 1
   MsgBox num
End Sub
```

（2）代码 2

```
Private Sub Command1_Click()
   Dim num As Integer
   num = num + 1
   MsgBox num
End Sub
```

注意：静态变量在被声明的方法或类的范围内值保持不变。一个静态变量可以保持其值，直到该变量的值被重新设置，或者应用程序结束。

2.6　函数与过程

函数是完成某一功能的程序代码段，又称程序模块。该段程序代码以一定格式实现并通过函数名来调用。编写函数的目的是便于维护，避免代码重复编写，提高程序设计效率。

一般来说，函数是有返回值的，函数的返回值即函数执行的结果。实际上，过程也是函数，只是没有返回值的函数；所谓方法，是存在于某个对象中的函数或过程。

函数分为系统函数和自定义函数。系统函数是 VB 提供的内部函数，可以直接通过函数名调用，如 MsgBox()函数、Len()函数等。自定义函数是用户自己设计的函数。

2.6.1　自定义函数

格式：[作用范围关键字] Function 函数名（参数列表）As 返回值类型

　　代码段

　　[Exit Function]

　　函数名=返回结果

End Function

说明：

- 关键字 Function…End Function：函数的定义结构。

- 作用范围关键字：使用 Public 或 Private 关键字，指定该函数是在本模块使用还是可以被其他模块使用，含义与变量作用范围中说明的相同，不指明作用范围时，默认是使用 Public 关键字。但不能使用 Dim 关键字。
- 参数：需要传递到函数中处理的数据，多个参数以逗号分隔；调用函数时，传递给函数的参数必须和函数定义时的个数、位置和类型一致。
- 函数返回值：通过对函数名赋值方式返回调用函数的结果，返回值必须与函数定义时的数据类型一致，函数只能通过函数名返回一个值，不能返回多个值。

例 2-38 定义一个名为 Add 的函数，实现任意两个整数相加，并返回相加的结果。

【程序代码】

```
'函数的定义
Private Function Add(ByVal x As Integer, ByVal y As Integer) As Integer
   Dim Sum As Integer
   Sum = x + y
   Add = Sum
End Function

'函数的使用
Private Sub Command1_Click()
    Dim Sum As Integer
    Sum = Add(1, 2)
    MsgBox ("计算结果是:" & Sum)
 End Sub
```

2.6.2 自定义过程

过程分为事件过程和自定义过程。事件过程是在对象上触发某个事件时，系统自动调用的过程。程序设计中接触最多的是命令按钮的 Click 事件过程，本章大部分例子都通过命令按钮的事件过程来说明。

过程名可以自定义。VB 中对象的事件过程按“对象名_事件名”格式自动命名，表示在什么对象上发生什么事件时，调用该过程。

格式：Sub 过程名(参数列表)

　　代码段

　　[Exit Sub]

End Sub

说明：过程不能有返回值。

例 2-39 编写一个 MyMsg 过程，模拟 MsgBox 函数的功能。

【程序代码】

```
'自定义过程

Sub MyMsg(ByVal msg As String)
  MsgBox (msg)
End Sub

'过程的使用
```

```
Private Sub Command1_Click()
    MyMsg ("这是 MyMsg 过程")
End Sub
```

2.6.3　参数传递

函数参数（包括过程参数）的传递方式有两种：值传递和引用传递。

一般，函数的参数称为形式参数，实际传递给函数参数的数据称为实际参数。实际参数可以是变量或常量。

所谓值传递（ByVal），是在调用函数时，把实际参数的值传递给函数对应的形式参数；所谓引用传递（ByRef），是在调用函数时，形式参数对应实际参数的内存地址。因此，引用传递实际参数必须是变量。

两种传递方式的最大区别：引用传递可以改变实际参数的值，值传递不影响实际参数的值。在例 2-40 中可以看出两者之间的差别。

例 2-40　值传递与引用传递的区别。

【程序代码】

```
'形式参数为值传递，使用关键字 ByVal 进行说明
Sub Testval(ByVal x As Integer)
    x = x + 10
End Sub
'形式参数为值传递，使用关键字 ByRef 进行说明
Sub TestRef(ByRef x As Integer)
    x = x + 10
End Sub
'在窗体上添加按钮 Button1
Private Sub command1_click()
   Dim m As Integer
   m = 10
   Testval m
   MsgBox ("传值调用的结果是:" & m) '结果为 10,m 的值没有改变

   m = 10
   TestRef m
   MsgBox ("传地址调用的结果是:" & m) '结果为 20,m 的值在 TestRef 中被改变了
End Sub
```

【运行结果】

如图 2-34 所示。

图 2-34　运行结果

说明：过程调用时，参数不能使用括号，否则将可能无法得到正确结果；如果要使用括号，必须使用 Call 关键字，如：Call TestRef(m)。

2.6.4 使用可选的参数

在过程的参数列表中列入 Optional 关键字，可以指定过程的参数为可选，如果指定了可选参数，则参数表中该参数后面的其他参数也必须是可选的，并且要用 Optional 关键字声明。可选参数也可以指定默认值。

以下代码中，自定义过程 MyMsg 具有可选参数并指定了默认值，如果调用该过程时带参数，则显示参数的值；如果不带参数，则显示默认值。

```
Sub MyMsg(Optional ByVal msg As String = "abc")
MsgBox msg
End Sub
Private Sub Command1_Click()
MyMsg
MsgBox "OK"
End Sub
```

2.7 程序启动方式

2.7.1 选择启动窗体

默认情况下，程序启动时以第一个创建的窗体作为启动窗体。建立工程时，默认的第一窗体是 Form1。如果选择工程中的其他窗体作为启动窗体，可以在菜单中选择“工程→工程 x 属性”（工程 x 表示工程名）选项进行设置，或按图 2-35 所示操作。

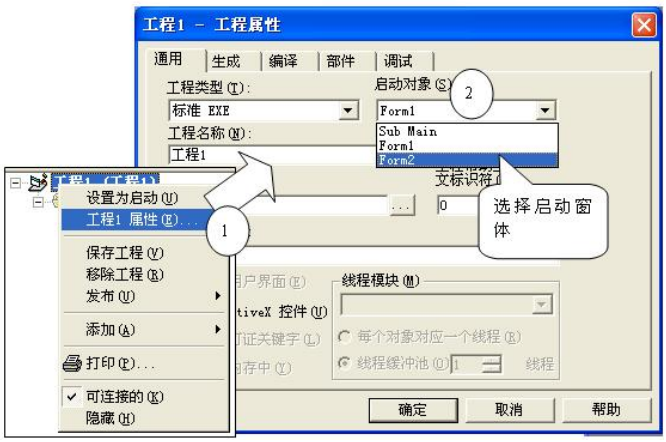

图 2-35 选择启动窗体

2.7.2 在标准模块中启动窗体

程序不一定从一个窗体中开始运行，可能窗体出现前还要做其他初始化操作，例如，检测环境配置，如果不符合程序运行要求，直接退出程序；或在主窗体前显示 Logo 窗体，同时在后台进行一些耗时的操作。VB6 提供了另外一种启动程序的方法，即从标准模块的 Main 过程中开始运行程序。Main 过程很像 C 语言中的 main 函数，可以决定首先执行哪些操作，也

可以决定其他窗体的启动顺序。

如果项目中没有标准模块，应添加标准模块，并在标准模块中手工添加一个 Main 过程。

Module1 中的全部代码如下：

```
Sub Main()
    '首先启动 Form1 窗体
    Form1.Show vbModal

    'Form1 关闭,再启动 Form2 窗体
    Form2.Show vbModal
End Sub
```

其中，Show 方法后面有 vbModal 参数，该窗体为模式窗体，表示该窗体关闭后才执行本行的下一行语句；如果不带 vbModal 参数，该窗体为非模式窗体，表示打开窗体 Form1 后马上执行下一条语句，即同时启动两个窗体。这也是模式与非模式窗体的区别。

项目标准模块中只能有一个 Main 过程。要使程序启动时首先执行 Main 过程，必须对项目进行设置。在 VB6 中，可以通过选择“启动对象”为“Sub Main”进行设置，如图 2-35 所示。

2.8　典型例题

例 2-41　以下符号中，哪个不能作为 VB 中的变量名？

A．ABCDE　　B．P123456　　C．89XYZDEF　　D．xyz

【分析】

根据 VB 变量名的命名规则，4 个变量中只有 89XYZDEF 不是以字母开头的，不能作为变量名；其他 3 个变量名都满足 VB 中变量名的命名规则，是合法的变量名。

例 2-42　以下数据中，哪些是变量？哪些是常量？是什么类型的常量？

name	"name"	false	ff	"11/15/99"
cj	"129"	n	#11/15/1999#	123.45

【分析】

变量有 name、ff、cj、n。常量有"name"、false、"11/15/99"、"129"、#11/15/1999#、123.14。其中："name"、"11/15/99"、"129"为字符型常量；false 为布尔型常量；#11/15/1999#为日期型常量；123.14 为数值型常量。

例 2-43　把下列数学表达式改写为等价的 VB 算术表达式。

A．$\dfrac{1+\dfrac{y}{x}}{1-\dfrac{y}{x}}$　　B．$x^2+\dfrac{3xy}{2-y}$

C．$\sqrt{|ab-c^3|}$　　D．$\sqrt{s(s-a)(s-b)(s-c)}$

【分析】

VB 算术表达式与数学表达式在写法上有区别：VB 中每个符号占 1 个格，所有符号都必须一个一个并排写在同一横线上，不能在右上角或右下角写方次或下标；原来在数学表达式中省略的内容必须重新写上；所有括号都用小括号（），括号必须配对；要把数学表达式中的某些符号，改成 VB 中可以表示的符号。

A．(1+y/x)/(1-y/x)

B．x^2+3*x*y/(2-y)

C．(Abs(a*b-c^3))^0.5 或(Abs(a*b-c^3)) ^ (1/2)或 Sqr(Abs(a*b-c^3))

D．(s*(s-a)*(s-b)*(s-c))^0.5 或 (s*(s-a)*(s-b)*(s-c)) ^ (1/2) 或 Sqr (s*(s-a)*(s-b)*(s-c))

例 2-44 根据给出的条件，写出逻辑表达式。

（1）闰年的条件是：年号（year）能被 4 整除，但不能被 100 整除；或者能被 400 整除。

（2）征兵的条件是：男性（sex）年龄（age）在 18~20 岁之间，身高（size）不低于 1.65 米；或者女性（sex）年龄（age）在 16~18 之间，身高（size）不低于 1.60 米。

【分析】

（1）被某个数整除，可以用数值运算符 Mod 或 Int 函数来实现，即：

(year Mod 4=0 And year Mod 100 <>0) Or (year Mod 400 = year/400)

或：

(Int (year/4) And Int (year/100) <> year/100) Or (Int (year/400) = year/400)

（2）设性别 sex 值为 True 代表男性，sex 值为 False 代表女性，则对应的逻辑表达式为：

(sex And age >=18 And age <= 20 And size >= 1.65) Or (Not sex And age>=16 And age <= 18 And size>=1.60)

例 2-45 函数 Int(Rnd(0)*10)的值是在哪个范围内的整数？

【分析】

函数 Rnd (0)是 0～1 之间的数，因此，Int (Rnd (0)*10)的值是在 0～10 之间的整数。

例 2-46 以下 VB 语句中，正确的注释语句是（ ）。

A．Dim x(10) As Integer Rem 这是 VB 程序中的语句

B．'这是 VB 程序中的语句

```
Private Sub Command1_Click()
…
```

C．x=1:y+2: _Rem 这是 VB 程序中的语句:z=3

D．If Shift=6 And Button=2 Then

```
    Print "AAAA"        Rem 这是 VB 程序中的语句
End If
```

【分析】

注释出现在一个语句行后面时，只能用撇号作为注释符，不能用 Rem；在复合语句行中，注释必须是最后一个语句，不能放在复合语句行中；注释语句不能放在续行符的后面。本题正确的答案应选 B。

一、填空题

1．若声明数组时不明确指定其数据类型，默认是_____类型。

2．用语句 dim A(5,4)定义的数组，若设置 option base 1，则有_____个元素。

3．用 Redim 命令多次为动态数组重新分配内存时，若在 Redim 后出现了关键字_____，则不能改变数组的维数。

4．若使用函数 Array 为变体数组各元素赋值，该数组必须是_____数组。

5．通常_____循环用于循环次数确定的循环结构。

6．While…Wend 循环中，条件_____时退出循环。

7．在某 For…Next 循环中，若其步长为负数，则当循环变量初值_____，循环体语句一次也不执行。

8．若在函数中不明确指定参数的传递方式，默认的传递方式是_____，其关键字是_____。

9．在 VB 中，若要将整个数组作为实参传递给过程，应将_____放入实参列表中。

二、简答题

1．VB 提供了哪些标准数据类型？声明类型时，其类型关键字分别是什么？所占用的字节数分别是多少？

2．“+”与“&”在进行字符串连接运算时有何异同？

3．过程级静态变量与窗体级变量在程序运行时有何异同？

4．以下表达式的运算结果是多少？

9/3+5*7-8\5-9 mod 5

5>3 And "aeoplane"<"airplane"

5．求下列表达式的值，并写出其类型。

（1）((8+(7*9-13)/5)/9)^2

（2）"我们爱中国"&"的山山水水，"&"何时能畅游一番？"

（3）2*3+6<= (7+2)/3

（4）#12/31/2001# - #12/20/2001#

（5）2<3 And 7>8

6．写出下列函数的值。

（1）Int(-3.14159)

（2）Sqr(Sqr(16))

（3）Fix (3.1415926)

（4）Int (Abs (99-100)/2)

（5）Sgn(7*3+2)

（6）Left ("Hello",2)

（7）Val ("16 Year")

（8）Str(-459.65)

（9）Len("Hello")

7．找出下列变量名中哪些是错误的。

（1）n　　（2）3w　　（3）Abs　　（4）x-y

（5）x%y　　（6）e f　　（7）出生日期　　（8）grade_1

8．把下列数学表达式写成 VB 表达式。

（1）$v_0t-\frac{1}{2}at^2$

（2）$\frac{\sin\alpha\cos\beta}{\alpha\beta}$

（3）ax^2-bx-c

（4）0<x≤5

9．计算以下表达式的值。

（1）8/4*5/2.5*(3.25+6.75)

（2）3*7\2

（3）26\3 mod 0.4*int(2.5)

（4）Ture and 8-3>=6

（5）#8/5/1999# -10

10．构造条件表达式时，一般需要用到什么运算符？

11．静态数组和动态数组有什么区别？

12．用什么方法可以建立控件数组？

13．如何实现既增加动态数组的元素个数，又保存该数组中原有的值？

14．Select Case 语句中，表达式列表可以有哪几种写法？

15．在固定次数的 For…Next 循环中，如果循环体中存在改变循环变量数值的语句，可能出现什么情况？

16．在 For…Next 循环中，若终值是以变量的形式表示时，是否可以在循环体内通过其他语句修改循环终值以达到改变循环次数的目的？为什么？

17．参数传递的方式有哪两种？有哪些区别？

18．在过程调用的参数传递中，有哪些方式，它们在使用中有什么区别？

19．过程的形参和实参在什么情况下类型必须一致，什么时候不必完全一致？

三、程序设计题

1．输入 20 个数，求出它们的最大值、最小值和平均值。

2．输入一组数，以 9999 结束，统计其中正、负和零的个数。

3．编程实现：输入三个整数，按照从大到小的顺序进行输出，程序代码放在 Command1_Click 事件过程中。

4．编程实现：给出一个百分制成绩，要求输出成绩等级“A”、“B”、“C”、“D”、“E”。90 分以上为“A”，80～89 分为“B”，70～79 分为“C”，60～69 分为“D”，60 分以下为“E”。

5．编程实现：输入一个整数，在窗体中显示所有小于等于该整数的质数。

6．编写一个函数 isprime，用于判断其参数 num 是否是素数，并输出相应信息。