

VB2010 应用程序的开发步骤中一个重要的环节就是编写程序代码,这是继创建用户界面之后一项关键的任务。应用程序中的核心部分就是程序代码,程序的大部分功能都是通过代码来实现的。要想在 VB2010 开发环境下编写正确的程序,必须掌握 VB2010 语言基础。本章主要介绍构成 VB2010 应用程序的基本元素,包括数据类型、常量、变量、运算符、表达式等内容,重点是常量、变量、表达式的应用。

3.1 常量、变量、表达式应用实例

【实例 3.1】 计算圆的周长和面积。

实例功能:

(1) 已知圆的半径 r , 根据数学公式 $c = 2\pi r$ 和 $s = \pi r^2$, 计算圆的周长 c , 圆的面积 s 。

(2) 半径 r 的值从文本框输入的数据获得, 计算结果 c 和 s 格式化为保留两位小数的数值, 并将其分别显示在对应的文本框内, 其界面如图 3-1 所示。

实例分析:

在程序设计中, 存储和计算的数据是有类型区分的, 不同类型的数据其表现出来的值是不一样的。代码在编写过程中也是要遵循一些约定的。



图 3-1 圆的计算结果(单精度型)

3.2 程序编码规则

每一种程序设计语言都有自己的语法格式和编码规则, 编写程序时必须遵循这些格式和规则, 不然会出现编译错误。这里介绍 VB2010 中编码格式和书写规则, 语法格式在后续章节中介绍。

3.2.1 程序编码

1. 标识符

标识符是在用户编程时为程序中出现的各种元素所起的名称,这些元素包括变量、常量、对象、方法、函数、过程等。在 VB2010 中规定,标识符必须遵循如下命名规则:

(1) 名称的首字符必须为字母字符或下划线,后面可跟字母、下划线、数字等字符,可以使用汉字、希腊字符(不常用)。

(2) 不能使用 VB2010 中的关键字。

(3) 不区分字母的大小写,以声明时为准。

(4) 同一范围不允许出现相同的名称。

例如,下面给出的标识符是错误的:

2ab	'不能以数字开头
My File	'不能有空格
x/y	'不允许使用/符号
If	'不能使用关键字

2. 关键字

关键字是程序设计语言中有特定含义的字符串。在 VB2010 中,保留关键字不能作为变量或过程等编程元素的名称。保留关键字有很多,例如 If、For、Loop、Exit、End、Me、Dim、As 等。

3.2.2 编码约定

用户编程时为了提高程序的可读性和可维护性,代码的编写往往要遵循一些约定。

1. 代码不区分大小写

VB2010 代码不区分英文字母的大小写,但是,为了方便用户阅读和修改程序,常做如下约定:

(1) 关键字的首字母自动被转换为大写。

(2) 如果关键字由多个英文单词组成,系统会将每个单词的首字母转换成大写。

(3) 用户自定义的标识符,系统会以第一次声明为准,以后的输入会自动转为首次定义的格式。

2. 标识符约定

(1) 使用易读的标识符名称,尽量不使用缩写,可读性比简洁性更重要。

(2) 如果标识符由多个单词组成,不要在各单词之间使用分隔符,如下划线(_)或连字符(-)等,而应使用大小写来指示每个单词的开头。

3. 布局约定

(1) 通过整齐排列功能,用默认设置(智能缩进、4 字符缩进、将制表符保存为空格)

对代码的格式进行设置。通过选择“工具”→“选项”命令,在“选项”对话框中的“文本编辑器”中设置,如图 3-2 所示。

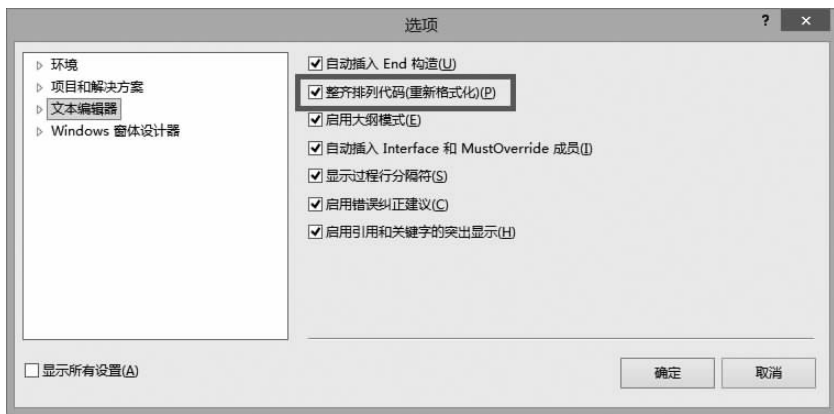


图 3-2 文本编辑器格式

(2) 同一行可以书写多条语句,用冒号“:”分隔。为了程序可读性,每行尽量使用一个语句或一个声明。

(3) 单条语句太长也可以分若干行书写,可在行后加续行符(依次为一个空格、一个下划线“_”和一个回车符)分隔;还可以使用 VB2010 新增的隐式行继续功能,在下列情况下直接按 Enter 键在下一行继续书写语句,而无须使用下划线字符_。

- ① 在逗号之后。
- ② 在左括号(之后或右括号)之前。
- ③ 在左大括号{之后或右大括号}之前。
- ④ 在串联运算符 & 之后。
- ⑤ 在赋值运算符=、&.=、:=、+=、-=、*=、/=、\=、^=、<<=、>>=之后。
- ⑥ 在表达式内的二元运算符+、-、/、*、Mod、<>、<、>、<=、>=、^、>>、<<、And、AndAlso、Or、OrElse、Like、Xor 之后。
- ⑦ 在 Is 和 IsNot 运算符之后。

例如:

```
Label1.Text=
    "欢迎使用 Visual Basic 2010!"
```

4. 注释约定

注释是为了方便阅读而为代码添加的简短的解释性说明,通常用“'”符号或 REM 语句来实现。计算机会忽略注释的内容不去执行,因此,调试程序时也常用注释功能。

注释可以和语句同行并跟随其后,也可以另占一整行。如果注释需要多行,必须在每行的前面使用注释符号。

例如:

```
'标签显示文本信息
```

使用注释需要注意以下几点：

- (1) 在过程的开头加入一段说明过程功能特征(过程的作用)的简短注释是一个很好的编程做法。
- (2) 每个重要的变量声明前都应有注释,用以描述被声明变量的用途。
- (3) 变量、控件和过程的命名应当足够清楚,使得只在遇到复杂的实现详细情况时才使用注释。
- (4) 注释不能与续行符同行。
- (5) 通过选择一行或多行代码,然后在“标准”工具栏上选择“注释”按钮和“取消注释”按钮,可以添加或移除某段代码的注释符。

3.3 数据类型

在程序设计过程中,描述不同种类的数据要用不同的类型加以区分,这就是数据类型。每种数据类型都有一个类型名称,其主要作用是对所处理的数据规定存放的形式、取值的范围和能进行的运算精度。

MSDN 中这样描述数据类型：

编程元素的“数据类型”指编程元素可以持有什么类型的数据及如何存储该数据。数据类型应用于可以存储在计算机内存中或参与表达式计算的所有值。每个变量、文本、常数、枚举、属性、过程参数、过程变量和过程返回值都具有数据类型。

VB2010 中的数据类型分为两大类：基本数据类型和复合数据类型。其详细分类情况如图 3-3 所示。

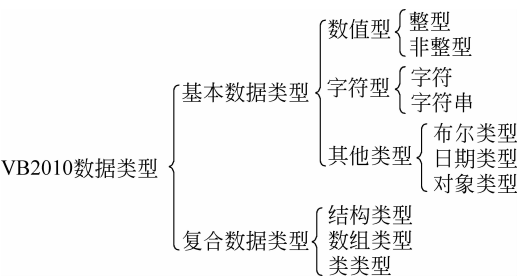


图 3-3 VB2010 的数据类型

3.3.1 基本数据类型

基本数据类型包括数值型数据类型、字符数据类型和其他数据类型。表 3-1 列出了 VB2010 中常用的基本数据类型的大小及其取值范围。

表 3-1 基本数据类型

数据类型	类型字符	大小(字节)	取值范围
Byte	无	1	0~255(无符号)
SByte	无	1	-128~127(有符号)
Short	无	2	-32 768~32 767(有符号)
Integer	%	4	-2 147 483 648~2 147 483 647(有符号)
Long	&	8	-9 223 372 036 854 775 808~9 223 372 036 854 775 807 ($9.2\cdots E+18$)(有符号)
UShort	无	2	0~65 535(无符号)
UInteger	无	4	0~4 294 967 295(无符号)
ULong	无	8	0~18 446 744 073 709 551 615 ($1.8\cdots E+19$)(无符号)
Single	!	4	负值,-3.402 823 5E+38~-1.401 298E-45 正值,1.401 298E-45~3.402 823 5E+38
Double	#	8	负值,-1.797 693 134 862 315 70E+308~-4.940 656 458 412 465 44E-324 正值,4.940 656 458 412 465 44E-324~1.797 693 134 862 315 70E+308
Decimal	@	16	0~+/-79 228 162 514 264 337 593 543 950 335(+/- $7.9\cdots E+28$),不包含小数点; 0~+/-7.922 816 251 426 433 759 354 395 033 5,包含小数点右边 28 位; 最小非零数为+/-0.000 000 000 000 000 000 000 000 000 000 000 1(+/-1E-28)
Char	无	2	0~65 535(无符号)
String	\$		0 到大约 20 亿个 Unicode 字符
Boolean	无	2	True 或 False
Date	无	8	0001 年 1 月 1 日午夜 0:00:00 到 9999 年 12 月 31 日晚上 11:59:59
Object	无	4 或 8	任何类型都可以存储在 Object 类型的变量中

在代码中出现的文本,计算机将依据其文本形式来自动确认其数据类型。其中默认的数据类型如表 3-2 所示。

表 3-2 默认数据类型

文 本 形 式	默认数据类型	示 例
无小数数值	Integer	2147483647
无小数数值(超出 Integer)	Long	2147483648
带小数数值	Double	1, 23
双引号括起来	String	"123"
数字符号括起来	Date	# 5/1/2014 #

除此之外,VB2010 还提供了一组强制文本类型字符,用来强制文本采用某种特定数

据类型,而不是按其默认的数据类型。将强制文本类型字符追加到文本末尾来实现。
表 3-3 列出了常用的几种强制文本类型字符。

表 3-3 强制文本类型字符

文 本 形 式	数 据 类 型	示 例
S	Short	I=123S
I	Integer	I=123I
L	Long	I=123L
D	Decimal	I=123D
F	Single	I=123F
R	Double	I=123R
US	UShort	I=123US
UI	UInteger	I=123UI
UL	ULong	I=123UL
C	Char	M="A"C

另外,Boolean、Byte、Date、Object、SByte 或 String 数据类型或任何复合数据类型(如数组或结构)都没有文本类型字符。

下面分别介绍这些数据类型。

1. 数值型数据类型

VB2010 中提供了几种数值型数据类型,用来处理各种形式的数字信息,分为整型和非整型。数值型数据类型的默认值均为 0。

1) 整型数值类型

整型数值类型就是指表示整数(没有小数部分)的数据类型。根据符号有无可以分为有符号类型和无符号类型。有符号整型数据类型包括 SByte(短字节型)、Short(短整型)、Integer(整型)及 Long(长整型);无符号整型数据类型包括 Byte(字节型)、UShort(无符号短整型)、UInteger(无符号整型)及 ULong(无符号长整型)。无符号数不能表示负数。

用整型进行算术运算比用其他数据类型速度快。VB2010 中 Integer 和 UInteger 类型的算术运算速度最快。

各种整型数值类型由于所占的存储空间大小不同,所能存储的数据范围也不一样。需要存储的整数很大,超出 Integer 范围时采用 Long 类型,它的存储范围最大。如果存储小整数可以使用 Short 和 SByte 类型,但是它们的运算速度都没有 Integer 快。

如果永远不需要存储负数,可以使用无符号类型 Byte、UShort、UInteger 和 ULong。这些数据类型所能存储的正整数是相应的有符号类型(SByte、Short、Integer 和 Long)的两倍。

如果将整型变量设置为其类型范围以外的数字将会出错。如果将其设置为小数精

度,则数字将向上或向下舍入为最接近的整数值。如果数字同样接近两个整数值,则值将舍入为最接近的偶数整数。例如下面的代码:

```
Dim i As Integer
'i 的取值范围应该是-2 147 483 648~+2 147 483 647
i=2147483648      '语句出错,因为 i 超出了取值范围
i=3.9             'i 的值为 4
i=2.5             'i 的值为 2
i=3.5             'i 的值为 4
```

实例 3.1 中半径 r 应该使用整型数据类型来表示。

2) 非整型数值类型

非整型数据类型是表示同时带有整数部分和小数部分的数字的类型。

非整型数值数据类型有 Decimal(小数型)、Single(单精度)和 Double(双精度),它们都是有符号类型。如果某个变量可以包含小数,则将其声明为这些类型之一。

Single 类型和 Double 类型是浮点类型。Single 类型可以精确到 7 位十进制数,精度不高,而 Double 类型能精确到 15 位十进制数,所以当运算要求精度较高时,可以采用 Double 类型。

Decimal 不是浮点数据类型。Decimal 数字具有一个二进制整数值和一个用于指定值的哪一部分是小数部分的整数比例因子。

Decimal 类型中可以存储非常精确的数字,在小数点后可以保留 28 位小数。当小数位为 0 时,它支持很大的正数或负数,具体范围如表 3-1 所示。

浮点数字的范围比 Decimal 数字大,但可能会导致四舍五入错误。浮点类型支持的有效位比 Decimal 支持的少,但它可以表示更大的值。

Double 是小数数据类型中效率最高的,因为目前各平台的处理器都是以双精度来进行浮点运算。但是,使用 Double 的运算不如使用整型(如 Integer)的运算速度快。

在为 Decimal 变量或常量赋值时,最好将其强制文本类型字符加到数值后面,以免数值太大超出 Long 数据类型。例如:

```
Dim i As Decimal
i=9223372036854775808      '溢出,超出了 Long 数据类型
i=9223372036854775808D    '不溢出,强制为 Decimal 数据类型
```

实例 3.1 中的周长和面积应该用单精度类型来表示。

2. 字符数据类型

字符数据类型用来处理 Unicode 字符,包括 Char(字符型)和 String(字符串型)两种类型。

Char 数据类型是单个双字节(16 位)Unicode 字符。如果一个变量总是仅存储一个字符,则将其声明为 Char。Char 的默认值是码位为 0 的字符。

String 数据类型是 0 或更多个双字节(16 位)Unicode 字符的序列。如果一个变量可以包含任意个数的字符,则将其声明为 String。String 的默认值为 Nothing(null 引用)。

在 VB2010 中,String 型数据是用双引号""括起来的若干字符,如果不包含任何字

符,则该字符串为空字符串,表示为""。注意 Nothing 与空字符串的不同。

3. 其他数据类型

除了数值数据类型和字符数据类型外,VB2010 还提供了几种不是针对数字或字符的数据类型。它们处理特殊的数据,如是/否值、日期/时间值、对象地址等。

1) 布尔型

Boolean 数据类型是一个无符号值,该值只能为 True 或 False。当一个变量仅包含真/假、是/否、开/关这样的两种状态时,就可以将其声明为 Boolean。其默认值为 False。

2) 日期类型

Date 数据类型是一个存储日期和时间信息的 64 位值。每个增量表示从公历第 1 年的 1 月 1 号(12:00AM)开始经过的 100 毫微秒时间。如果变量可以包含日期值、时间值或两者,则将其声明为 Date。

Date 的默认值为 0001 年 1 月 1 日的 0:00:00(午夜)。

在 VB2010 中必须将 Date 文本括在数字符号“##”内。必须以 m/d/yyyy 格式指定日期值,例如 #5/31/2014#。

3) 对象型

Object 数据类型是一个指向应用程序或其他某个应用程序中的对象实例的 32 位地址。Object 变量可以引用应用程序识别的任何对象或任何数据类型的数据。如果一个变量存储指向某个类实例的指针,而在编译时不知道该类,或者如果该指针可以指向各种数据类型的数据,则将其声明为 Object。

Object 的默认值为 Nothing(null 引用)。

如果在代码中没有声明变量的数据类型,则编译器默认变量为 Object 数据类型,可以将任何数据类型的变量、常数或表达式赋给 Object 变量。Object 数据类型为引用类型。但是,当 Object 变量引用值类型的数据时,VB2010 将此变量视为一个值类型。

无论它引用什么数据类型,Object 变量都不包含数据值本身,而是指向该值的一个指针。它总是在计算机内存中使用 4 个字节,但这不包括表示变量值的数据的存储。由于使用指针定位数据的代码的缘故,访问持有值类型的 Object 变量比访问显式声明类型的变量速度稍慢。

3.3.2 复合数据类型

VB2010 除了提供基本数据类型外,还提供复合数据类型,即将不同类型的数据项组合起来以创建新的类型。声明为复合数据类型的变量可以保存多个值。

复合数据类型主要包括结构、数组和类。可以从基本类型和其他复合类型生成复合数据类型。例如,可以定义结构元素的数组或者具有数组成员的结构。

这三种复合数据类型的定义方法分别是:

- (1) 使用 Structure 语句定义结构。
- (2) 使用 Dim 语句(Visual Basic)定义数组。
- (3) 使用 Class 语句(Visual Basic)定义类。

具体的使用方法见后续章节的详细介绍。

3.3.3 数据类型的实现

在 VB2010 中,数据类型是基于其类别实现的。根据特定类型的变量存储的是自己的数据还是指向数据的指针,可以对数据类型进行分类。如果存储的是自己的数据,则是值类型;如果保存指向内存中其他位置的数据的指针,则是引用类型。

1. 值类型

如果数据类型在它自己的内存分配中存储数据,则该数据类型就是“值类型”。值类型包括:

- (1) 所有数值数据类型。
- (2) Boolean、Char 和 Date。
- (3) 所有结构,即使其成员是引用类型。
- (4) 枚举。

2. 引用类型

“引用类型”包含指向存储数据的其他内存位置的指针。引用类型包括:

- (1) String。
- (2) 所有数组,即使其元素是值类型。
- (3) 类类型,如 Form。
- (4) 委托。

3.3.4 数据类型转换

VB2010 中将值从一种数据类型改变为另一种数据类型的过程就叫数据类型转换。

1. 扩大转换和收缩转换

转换到取值范围更大或精度更高的数据类型被称为扩大转换;而转换到取值范围更小或精度更低的数据类型被称为收缩转换。

扩大转换在运行时始终是成功的,且从不会导致数据丢失。无论类型检查开关 (Option Strict 语句)是设置为 On 还是 Off,扩大转换始终隐式执行。

例如:

```
Dim i As Integer
Dim j As Single
i=1234
j=i '从整型到单精度的扩大转换
```

收缩转换在运行时并不总会成功,可能会失败或导致数据丢失。如果目标数据类型无法接收正转换的值,将会出错。例如:

```
Dim i As Integer
Dim j As Short
i=32768
```

2. 隐式转换和显式转换

按照代码中是否使用特殊语法,可以分为隐式转换和显式转换。隐式转换是自动完成的,不需要任何特殊语法。例如上面的扩大转换例子就是隐式转换。

显式转换需要使用类型转换函数,利用这些函数强制将表达式转换为目标数据类型。类型转换函数详见表 2-13 所示。

当代码中使用 Option Strict On 语句时,不允许收缩转换的隐式转换。例如:

```
Dim i As Integer
Dim j As Short
i=327
j=i 'Option Strict On 不允许从 Integer 到 Short 的隐式转换
```

这时,必须将 j=i 改为 j=CShort(i)。

另外还有一个函数 CType,其格式如下:

CType(表达式,类型名)

其中,第一个参数是将要转换类型的表达式,第二个参数是目标数据类型。

3. 转换过程中值的更改

从值类型进行的转换在转换目标中存储源值的副本。但是,此副本不是源值的完全相同的映像。目标数据类型以不同的方式存储值,甚至正表示的值也可能更改,具体取决于正执行的转换类型。

“收缩转换”更改源值的目标副本,同时会有潜在的信息丢失。例如,小数值在转换为整型时被四舍五入,而数值类型在转换为 Boolean 后将缩减为 True 或 False。

提示:

将数字数据类型值转换为 Boolean 时,0 转换为 False,非 0 值转换为 True。将 Boolean 值转换为数字类型时,False 转换为 0,True 转换为 -1。

“扩大转换”保留源值,但可以更改值的表示形式。从整型转换为 Decimal,或从 Char 转换为 String 时会发生扩大转换。

原始源值不因转换而更改。

从“引用类型”进行的转换只复制指向值的指针,而值本身既不复制也不以任何方式更改。唯一可以更改的是存储指针的变量的数据类型。

3.4 常量与枚举

3.4.1 常量

常量也称为常数,是指在程序运行过程中其值始终保持不变的量。在 VB2010 中有三种常量:直接常量、用户自定义的符号常量和系统预定义的常量。

1. 直接常量

直接常量是指在程序代码中直接给出的数据,其值直接反应其类型。每种不同的数据类型有不同的常量表示形式,参照表 3-2。例如:

字符常量: "Hello World", "北京", "2"。

数值常量: 278, -91, 18.33。

逻辑常量: True, False。

日期常量: #04/10/2009#, #Jan 1, 2009#。

在 Visual Basic 中还允许使用八进制数和十六进制数,八进制数以 &O 开头,十六进制数以 &H 开头,例如 &O153、&H3F。

2. 符号常量

符号常量是以标识符代表一个具体常量值形式出现的常量。如果有一个值(如圆周率)在应用程序中始终不变,则可以定义为一个符号常量,并用该符号常量取代该值。名称比值更容易记忆,也便于维护。可以只定义该常量一次,然后在代码中的多处使用它。如果需要修改其值,只需修改声明语句即可。符号常量要先声明后使用,声明语句格式为:

[访问修饰符] Const 常量名 [As 类型]=表达式

功能: 将表达式的值赋给指定的符号常量。

其中:

- Const: 声明常量的关键字。
- 常量名: 是用户为常量命的名,遵循标识符命名规则。
- As 类型: 可选项,用来说明常量的数据类型。如果省略,数据类型由表达式的值决定。
- 表达式: 可以是数值、字符串及由它们和运算符组成的简单表达式。
- 访问修饰符: 可选项。常用的访问修饰符如表 3-4 所示。

表 3-4 访问修饰符

访问修饰符	别 名	作 用
Public	公有成员	可供同一程序集的任何代码访问,或被其他程序集引用
Private	私有成员	在同一类中的代码可以访问该类型或成员
Protected	保护成员	在同一类或派生类中的代码可以访问该类型或成员
Friend	内部成员	可供同一程序集的任何代码访问,其他程序集不可以访问
Protected Friend	内部保护成员	可供同一程序集的任何代码或其他程序集中任何派生类访问

在任何过程之外的模块级别声明的常量是“成员常量”,该常量是声明它的类、结构或模块的成员,可以使用访问修饰符来调整它们的访问级别。在过程级别声明的常数是局部常数,该常数局限于声明它的过程或块,不能对这些常数使用任何访问修饰符。

例如：

```
Public Class Form1
    Const MIN=10                                '定义模块级符号常量 MIN
    Public Const MAX As Integer=100             '定义公有符号常量 MAX
    Private Sub Button1_Click (ByVal sender As System.Object, ByVal e As System.
        EventArgs) Handles Button1.Click
        Const MyWord="Hello"                    '定义常量 MyWord,字符串
        Const MyInteger As Integer= 32          '定义常量 MyInteger,整型数
        Const PI=3.1415926                      '定义圆周率为符号常量 PI
    End Sub
End Class
```

提示：

一旦声明了常量,就不能改变它的值。而且常量有作用范围,如果在过程内部声明常量,它只在该过程有效;如果在通用代码段中声明,它在整个窗体模块的所有过程内都有效。

在一行中使用逗号作为分隔符可以同时声明多个常量。例如：

```
Const DaysInWeek As Short=7,DaysInYear As Short=365
```

3. 系统预定义常量

VB2010 提供许多系统预定义的常量以方便编程。这些常量一般以 vb 字母开头,后面跟有意义的符号。例如,常用的输出和显示常量如表 3-5 所示。

表 3-5 常用的输出和显示常量

预定义常量	说 明	预定义常量	说 明
vbCrLf	回车/换行组合符	vbTab	Tab 字符
vbNullChar	null 字符	vbBack	退格字符

3.4.2 枚举

枚举提供一种使用成组的相关常数及将常数值与名称相关联的方便途径。例如,可以为一组与一周中的七天相对应的整数常数声明一个枚举,然后在代码中使用这七天的名称而不是它们的整数值。枚举有用户创建和预定义两种。

1. 创建枚举

使用 Enum 语句创建枚举。其语句的声明格式如下：

```
[访问修饰符] Enum 枚举名 [As 类型]
    成员列表
End Enum
```

功能：用户创建一个枚举。

其中：

- [访问修饰符]：可选项，访问级别，表明枚举的有效范围。
- 枚举名：用户为枚举起的名，遵循标识符命名规则。
- As 类型：可选项，用来说明枚举的数据类型，必须是整数类型之一，默认为 Integer 类型。
- 成员列表：声明的枚举的成员常量的列表。单行显示多个成员。
- End Enum：终止 Enum 块。

Enum 语句必须在类或模块的声明部分使用。

每个成员均按下列语法格式声明：

```
member name [=initializer]
```

member name 是成员的名称，initializer 是设定此成员的表达式（可选）。如果没有为成员指定 initializer，VB2010 会将其初始化为 0（如果它是成员列表中的第一个），或初始化为比该成员的前一个值大 1 的值。

枚举成员的引用格式如下：

枚举名.成员名

例如：

```
Enum filemenu As Short
    newfile=1
    openfile=2
    savefile=4
    closefile=5
End Enum
```

该语句声明了一个 filemenu 枚举，其数据类型为 Short，共 4 个成员，每个成员都有一个设定好的值，其数据类型也是 Short。成员引用如 filemenu.newfile 代表的值为常量 1。

将上例修改成如下：

```
Enum filemenu
    newfile
    openfile
    savefile
    closefile
End Enum
```

该枚举 DaysInWeek 和其成员的数据类型均为 Integer，其中 newfile 的值为 0，openfile 的值为 1，savefile 的值为 2，closefile 的值为 3。

2. 预定义枚举

VB2010 提供了很多预定义的枚举，如表 3-6 所示。

表 3-6 常用预定义的枚举

预定义枚举	说 明
DateFormat	指示在调用 FormatDateTime 函数时如何显示日期
FirstDayOfWeek	指示在调用与日期相关的函数时使用的每周的第一天
FirstWeekOfYear	指示在调用与日期相关的函数时使用的每年的第一周
MsgBoxResult	指示在 MsgBox 函数返回的消息框上所按的按钮
MsgBoxStyle	指示在调用 MsgBox 函数时要显示的按钮
OpenAccess	指示调用文件访问函数时如何打开文件
OpenMode	指示调用文件访问函数时如何打开文件
OpenShare	指示调用文件访问函数时如何打开文件
VariantType	指示由 VarType 函数返回的变量对象的类型
VbStrConv	指示在调用 StrConv 函数时要执行哪种类型的转换

3.5 变 量

变量是指在程序运行过程中其值可以不断改变的量。例如在执行计算时,需要计算若干个值,并对它们进行比较,这时必须保留这些值。在 VB2010 中使用变量来存储值,以供程序计算需求。

变量由三个要素组成:变量名、变量的数据类型和变量的值。

(1) 变量名就是按照标识符命名规则给所要存储的值取的一个名字,以便编程中的应用。

(2) 变量的数据类型表示该变量可以存储什么样的数据,即存储的值的 datatype。

(3) 变量的值就是存储在变量中的数据信息。

3.5.1 变量的声明

VB2010 规定,变量必须先声明才可以使用。变量的声明实际上是指定了变量名和数据类型。变量声明成功后,系统会根据数据类型为变量分配相应的存储空间,并在该存储空间中存放变量的值。

变量的声明语句是 Dim 语句,其格式如下:

```
Dim 变量名 [As 类型] [=初始值]
```

其中:

- 变量名: 用户定义的标识符,遵循命名规则。
- As 类型: 可选项,声明变量的数据类型。若省略,系统默认变量为 Object 类型。

- 初始值：可选项，给变量赋的初值。

变量声明时需要注意以下几点：

(1) 使用 Dim 语句可以声明变量为任何数据类型，对于引用类型可使用 New 关键字来创建由数据类型制定的类或结构的新实例。例如：

Dim mydata As Integer	'声明整型变量 mydata
Dim myname As String	'声明字符串类型变量 myname
Dim flag As Boolean	'声明布尔类型变量 flag
Dim mydialog As System.Windows.Forms.Form	'声明窗体类变量 mydialog
Dim mylabel As New System.Windows.Forms.Label	'创建标签类的一个实例 mylabel

(2) 声明的变量都有其作用范围，详见 3.5.3 节。

(3) 变量的声明有隐式声明和显式声明，可通过两种方法来设定。

① 选项设置

选择“工具”→“选项”命令，在弹出的“选项”对话框中选择“项目和解决方案”下的“VB 默认值”选项，如图 3-4 所示。如果 Option Explicit 为 On(默认值)，则为显示声明，编译器将要求必须声明使用的每个变量；如果 Option Explicit 为 Off，关闭显示声明(即为隐式声明)，则可以不声明就使用变量。

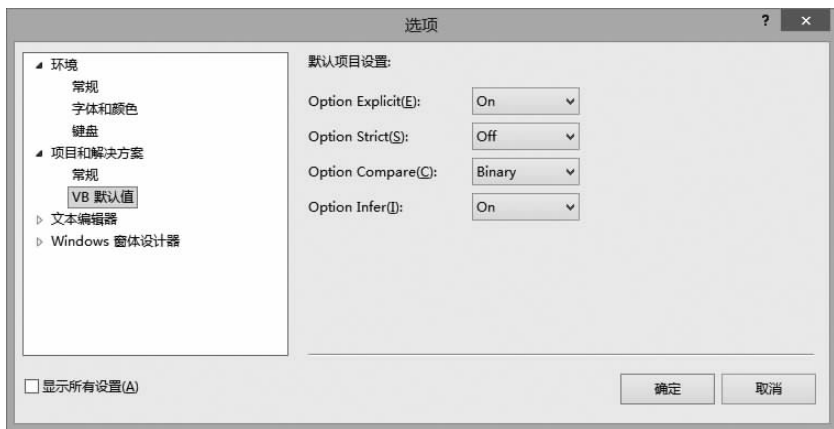


图 3-4 变量声明的选项设置

② Option Explicit 语句

语句格式如下：

Option Explicit On|Off

如果未指定 On 或 Off，则默认值为 On。当 Option Explicit On 或 Option Explicit 出现在代码中时，必须显式声明所有变量。如果尝试使用未声明的变量名，编译时将发生错误。Option Explicit Off 语句允许变量的隐式声明。

(4) 声明变量的同时可以给其指定初始值。例如：

Dim mydata As Integer=10	'声明整型变量 mydata，同时赋给初始值 10
--------------------------	---------------------------

```
Dim myname As String="Zhang San" '声明字符串变量 myname,同时赋给初始值"Zhang San"  
Dim flag As Boolean=True '声明布尔类型变量 flag,同时赋给初始值 True
```

如果不指定初始值,则 VB2010 会把它初始化为相应数据类型的默认值;如果指定了初始值,但没有指定数据类型,则根据初始值来判断数据类型。详见数据类型介绍。

(5) 在一条声明语句中可以声明多个变量,用逗号分隔。例如:

```
Dim i,j,k As Integer '声明三个整型变量 i,j,k  
Dim str1,str2 As String '声明两个字符串型变量 str1,str2  
Dim a As Integer,b As Single,c As Double  
'声明三个不同类型变量,a 是整型,b 是单精度,c 是双精度
```

(6) 可以使用标识符类型字符指定变量的数据类型,而不使用“As 类型”。VB2010 规定了可用的数据类型的标识符类型字符,详见表 3-1。例如:

```
Dim mydata% '声明整型变量 mydata  
Dim myname$ '声明字符串变量 myname
```

3.5.2 变量的使用

变量声明后就可以在程序中进行使用,包括变量的赋值和变量的引用。

1. 变量的赋值

变量的赋值就是将变量名放在赋值语句的左侧,将赋值语句右侧生成的值存储在变量中。

赋值语句一般格式如下:

变量名=表达式

功能: 将表达式的值计算出来,然后赋值给左边的变量。

表达式的类型应与变量的类型保持一致,如同时为数值型或同时为字符型。当类型为数值型而两者的精度不同时,系统强制将表达式的值转换为变量的精度。例如:

```
Dim i As Integer '将 i 定义为整型  
i=4.6 'i 的值为 5,按照四舍五入的原则将 4.6 转换成整型
```

提示:

赋值号与关系运算符“等号”都是用“=”表示,但两者意义不同,系统会根据所处的位置自动判断是何种意义的符号。

2. 变量的引用

变量的引用就是将变量名用在表达式中或放在赋值语句等号的后面,用它参与运算或直接给其他变量赋值,这就是变量的引用。例如:

```
Dim i,j,k As Integer '声明三个整型变量 i,j,k  
k=3 '给 k 赋初值,k=3
```

i=k

'引用 k 的值,将其赋给 i,i=3

j=(i+5)^2

'引用 i 的值参与运算,将运算结果赋给 j,j=64

实例 3.1 计算圆的周长和面积应用程序设计。

实例分析:

程序要用到三个变量,半径 r 为整型变量,周长 c 和面积 s 是单精度变量。Button1_Click 事件过程中完成计算功能。

设计步骤:

(1) 按照图 3-1 创建应用程序界面。

(2) 设计程序代码:

```
Public Class Form1
```

```
Private Sub Button1_Click (ByVal sender As System.Object, ByVal e As System.  
EventArgs) Handles Button1.Click
```

```
    '定义符号常量
```

```
    Const PI=3.1415926
```

```
    '定义半径、周长和面积为整型变量
```

```
    Dim r As Integer
```

```
    Dim c,s As Single
```

```
    r=Val(TextBox1.Text)           '得到半径的值
```

```
    c=2 * PI * r                   '计算周长的值
```

```
    s=PI * r * r                   '计算面积的值
```

```
    '将周长和面积格式化后输出
```

```
    TextBox2.Text=Format(c,"0.00")
```

```
    TextBox3.Text=Format(s,"0.00")
```

```
End Sub
```

```
End Sub
```

(3) 在文本框中输入一个半径的值 5,单击“计算”按钮,程序运行结果如图 3-1 所示。

试一试:

如果修改程序代码中声明变量的语句,即将代码 Dim c,s As Single 改为 Dim c,s As Integer,再次运行程序,输入半径 5,单击“计算”按钮,其运行结果如图 3-5 所示。

图 3-5 中显示的运行结果与图 3-1 中的结果有很大差异,造成这种错误的原因是存放结果的周长变量 c 和面积变量 s 的数据类型发生了改变。当两个变量均声明为整型变量时,它们只能存放整型数值,所以精确度降低了。由此可看出,不同数据类型表达的数值含义及其计算结果不一样。



图 3-5 圆的计算结果(整型)

3.5.3 变量的生存期和变量的范围

1. 变量的生存期

变量的生存期是指变量可以存储值的时间周期。变量是唯一具有生存期的元素。在生存期内变量的值可以改变,但变量总是存储某些值。

1) 局部变量和成员变量的生存期

“局部变量”是指在过程内部声明的变量。“成员变量”是 VB2010 类型的成员,它在模块级别上的类、结构或模块内部(但不能在该类、结构或模块内部的任何过程中)声明。“局部变量”(在过程内部声明)仅在声明变量的过程的运行阶段存在。“成员变量”(在模块级,并且在任何过程外部声明)的生存期一般与声明该变量的元素(如过程或类)的生存期相同。

当控制进入声明局部变量的过程时,局部变量的生存期便开始了。过程一开始运行,每个局部变量即被初始化为其数据类型的默认值。当过程终止时,不保留其局部变量的值,VB2010 将回收它们占用的内存。下次调用该过程时,将再度创建并重新初始化它的所有局部变量。

2) 静态变量的生存期

如果用 Static 关键字声明局部变量,该变量的生存期要比其过程的执行时间长。将该变量称为静态变量。一般过程终止后,此过程中的局部变量将立即消失。静态变量可继续存在,并保留其最新值。代码下次调用该过程时,变量不会重新初始化,仍然保存已赋给它的最新值。静态变量在定义它的类或模块的生存期内继续存在。

其语句格式如下:

```
Static 变量名 [As 类型]
```

【实例 3.2】 设计一个应用程序,显示单击按钮的次数。

实例功能:

单击按钮统计次数,将结果显示在标签上。

实例分析:

(1) 触发的应该是按钮的 Click 事件。

(2) 统计次数实际上是计数的过程,根据变量的含义,可以使用累加操作,即 $i=i+1$ 。

(3) 由于 i 的值需要保留,不能将其定义为局部变量,而是使用静态变量。

设计步骤:

(1) 创建应用程序界面,窗体上添加一个按钮和一个标签,设置对象的基本属性,如图 3-6 所示。

(2) 代码实现:



图 3-6 实例 3.2 运行结果

```
Private Sub Button1_Click (ByVal sender As System.Object, ByVal e As System.  
EventArgs) Handles Button1.Click
```

```
Static i As Integer
```

```

i=i+1
Label1.Text="单击了按钮" & i & "次"
End Sub

```

(3) 运行程序,连续单击 5 次按钮,其结果如图 3-6 所示。

2. 变量的范围

变量的范围是指不必限定变量名便可以引用变量的所有代码的集合。变量的范围由声明变量的位置决定。位于给定区域中的代码不必限定在该区域中定义的变量的名称便可以使用它。变量的范围级别如表 3-7 所示。

表 3-7 变量的范围级别

级 别	说 明
块范围	只适用于在其中声明变量的代码块
过程范围	适用于在其中声明变量的过程中的所有代码
模块范围	适用于在其中声明变量的模块、类或结构中的所有代码
命名空间范围	适用于在其中声明变量的命名空间中的所有代码

1) 块范围

块是初始声明语句与终止声明语句之间的一组语句,例如 Do 和 Loop、For 和 Next、If 和 End If 等。在某个块声明的变量只能在该块中使用。例如:

```

If x<100 Then
    Dim y As Integer
    y=x^3
End If

```

'y 变量的范围是 If 和 End If 之间的块

2) 过程范围

在过程内声明的变量在该过程外是不可用的,该级别的变量也称为“局部变量”。在一个过程内用 Dim 或 Static 声明变量,只能在本过程中使用。

实例 3.1 中所有变量都是过程范围,实例 3.2 中的 i 变量也是过程范围,但由于使用了 Static 关键字,其生存期延长,当过程结束时仍保留最新值,因此起到了计数的作用。

试一试:

如果将 Static 关键字换成 Dim,结果会怎样? 分析原因。

3) 模块范围

通过将声明语句放在模块、类或结构中的任一过程或块的外部来声明该级别的变量。当在模块级声明时,由所选的访问级别确定范围。包含模块、类或结构的命名空间也影响范围。

将变量声明语句放在模块、类或结构的内部,所有过程的外部,从模块、类或结构内部的任何位置都可以引用该变量,但是不能从外部引用。

已声明变量的“访问级别”是指能够对其进行访问的程度,即什么代码对其具有读取或写入权限。指定访问级别的关键字称为“访问修饰符”,最常用的有 Public 和 Private。

声明语句中的 Public 关键字指定可从以下位置访问元素：同一项目中任意位置的代码、引用该项目的其他项目及由该项目生成的任何程序集。仅可以在模块、接口或命名空间级别使用 Public。将包含 Public 关键字的变量声明语句放置在模块、类或结构的内部，所有过程的外部，从包含该模块、类或结构的命名空间中的任何位置都可以引用该变量。

声明语句中的 Private 关键字指定仅可以从同一模块、类或结构内访问元素。仅可以在模块级别使用 Private。这意味着可以在模块、类或结构内部声明 private 元素，但不能在源文件级别或命名空间级别、接口内部或者过程内声明它。在模块级别，不带任何访问级别关键字的 Dim 语句与 Private 声明等效。

改变实例 3.2 的代码，i 变量声明使用 Dim 语句，将其声明放在窗体模块，即事件过程外，代码如下：

```
Public Class Form1
    Dim i As Integer '声明 i 变量是模块范围
    Private Sub Button1_Click (ByVal sender As System.Object, ByVal e As System.
        EventArgs) Handles Button1.Click
        i=i+1
        Label1.Text="单击了按钮" & i & "次"
    End Sub
End Class
```

运行程序，连续单击 5 次按钮，程序运行结果和图 3-6 一样。

此时，i 变量的作用范围扩展到了窗体模块，只要在窗体内部均可以引用，当 Button1_Click 过程结束时，i 的值仍保留，所以达到计数效果。

4) 命名空间范围

如果使用 Friend 或 Public 关键字声明模块级变量，则该变量变为可用于在其中声明该变量的整个命名空间内的所有过程。

例如，将上述程序中声明 i 变量的代码改为 Public i As Integer，那么 i 变量的作用范围是命名空间范围，可以在整个命名空间内引用该变量。

3.6 运算符与表达式

运算符就是可以实现某种运算的符号，这些运算包括算术运算、连接运算、关系运算、逻辑运算等。VB2010 中的值元素包括变量、常量、属性、函数返回值等。运算符通过执行计算、比较或其他运算来处理值元素。与运算符组合在一起的值元素称为该运算符的操作数。运算符与值元素组合在一起就形成了表达式，它将产生一个新值。

3.6.1 运算符

VB2010 提供了以下几种运算符：

1. 算术运算符

算术运算符用来进行算术运算。VB2010 提供了 8 种算术运算符，如表 3-8 所示列出