# 第5章 一"块"程序——过程和函数

过程和函数就是在程序中可以被重复使用的代码。当应用程序比较复杂时,开发人员可能需要在程序中多处使用相同的代码。在 Visual Basic 应用程序开发中,重要任务之一就是创建可重复使用的代码。通过学习本章,读者可以熟练掌握过程和函数的定义及使用方法。

## 5.1 过 程

过程就是在应用程序可以被多次重复使用的代码段。使用过程具有以下优点:复杂任 务分解成多个简单代码段;易于读写,具有较强的可读性和可维护性。前面章节介绍的事 件过程就是 Visual Basic 定义过程的一种。本节介绍的过程是用户创建的自定义过程。

## 5.1.1 什么是 Sub 过程

Sub 过程是可以被重复执行的无返回值的代码段。其标准语法如下:

```
[Private | Public] [Static] Sub Name [(Arglist)]
  [statements]
  [Exit Sub]
  [statements]
End Sub
```

参数说明:

- □ Private|Public 为可选项。表明模块作用域。具体内容请参见 5.1.5 小节。
- □ Name 为必需项。表明 Sub 过程名称。在其他程序中,如果需要调用此过程,则必须使用该名称。
- □ Arglist 为可选项。表明在 Sub 过程中传递的参数名称。如果为多个参数,可以用 逗号分开。参数默认声明为 Variant 数据类型。
- □ End Sub 为必需项。表明该 Sub 过程结束。

## 5.1.2 创建 Sub 过程

添加 Sub 过程有两种方法如下所示。

方法一:

(1)选择"工程"|"添加模块"命令,打开"添加模块"对话框,如图 5.1 所示。在 "新建"选项卡中选择"模块"图标,单击"打开"按钮,弹出"模块编辑"对话框。 (2)选择"工具"|"添加过程"命令,弹出"添加过程"对话框,如图 5.2 所示。
(3)在"名称"文本框内输入模块名称,如 showmsg。在"类型"选项区域中选择"子程序"单选按钮,在"范围"选项框中选择"公有的"单选按钮。单击"确定"按钮。
(4)在模块编辑窗体可见自动添加 showmsg 子过程,如图 5.3 所示。

<b>合</b> 添加模块	×
新建 现存	
模块	
	1 <u>1</u> #W
	帮助(H)
□ 不再显示这个对话框(0)	

图 5.1 "添加模块"对话框

添加过程 <u>名称 (P)</u> showmsg		■×■
×型 ● 子程序 (S) ○ 函 <u>数</u> (C)	〇 属性 (2) 〇 事件 (2)	取消
<ul> <li>范围</li> <li>○ 公有的 (B)</li> <li>○ 所有本地変量为積</li> </ul>	<del>6 私有的 (2)</del> 静态变重 (a)	

图 5.2 "添加过程"对话框

💭 工程1 - Form1 (	Code)	-	
(通用)	•	showmsg	•
Public Suk	show	msg()	-
End Sub			
≡≣ •			- -

图 5.3 模块编辑窗口

方法二:

(1)选择"工程"|"添加模块"命令,弹出代码编辑窗口。

(2) 手动输入子过程名称, 按下 Enter 键后显示如下代码。

```
Sub showmsg()
End Sub
```

(3) 在 Sub 和 End Sub 之间手动输入代码。

【示例 5-1】创建显示信息 showmsg 过程。

(1) 新建一个工程,并新建一个窗体。

(2)选择"工程"|"添加模块"命令,打开"添加模块"对话框。选择"新建"|"模块"图标,单击"打开"按钮,弹出"模块编辑"对话框。

(3)选择"工具"|"添加过程"命令。在"名称"文本框内输入模块名称,如 showmsg。 在"类型"选项框中选择"子程序"单选按钮,单击"确定"按钮。

(4) 在模块编辑窗体可见自动添加 showmsg 子过程。输入以下代码:

```
Public Sub showmsg(smsg As String)
MsgBox smsg
End Sub
```

• 51 •

□○说明:该代码运行弹出信息对话框,显示变量 smsg 中文字信息。

至此,一个子过程创建结束。过程必须被调用才可以执行过程中的代码。

#### 5.1.3 调用 Sub 过程

调用 Sub 过程主要使用 Call 语句或过程名+参数形式调用。

□ 用 Call 语句调用子过程,其标准语法如下:

Call SubName [Arglist]

参数说明: Call 关键字为必需项; SubName 为子过程名称; Arglist 参数列表为可选项。 □ 过程名+参数形式调用,其标准语法如下:

SubName [Arglist]

参数说明: SubName 为子过程名称; Arglist 参数列表为可选项。

【示例 5-2】调用显示信息 showmsg 子过程。

(1)新建一个工程,并新建一个窗体。

(2) 双击工具箱中的 CommandButton 图标,自动在窗体添加一个 CommandButton 控件,设置 caption 属性为"显示过程"。双击"显示过程"按钮,弹出代码编辑窗口。在左侧对象下拉列表框中选择 Command1 选项,在右侧过程下拉列表框中选择 Click 事件。

(3) 输入以下代码:

```
Call showmsg("这是第一个子过程")
```

```
或
```

Showmsg "这是第一个子过程"

## 5.1.4 使用 Sub Main

Sub Main 是一个特殊的子过程,可以控制应用程序的开始与结束。应用程序启动时必须设置启动窗体或 Sub Main 过程。启动窗体作为第一个显示窗体,控制其他应用程序窗体的显示。Sub Main 中通过控制代码流程,选择顺序启动窗体。一般情况下,Windows 应用程序都有一个 Splash 窗体,显示程序名称、版本号和公司名称等相关信息。

1. 设置启动窗体

在默认情况下,应用程序中的第一个窗体被指定为启动窗体。当应用程序开始运行时, 此窗体就被显示出来。如果想在应用程序启动时显示别的窗体,那么就得改变启动窗体。 要改变启动窗体,按照以下步骤执行。

(1)选择"工程"|"工程属性"命令,在弹出的"工程属性"对话框中选择"通用"

• 52 •

选项卡。

(2) 在"启动对象"列表框中,选取要作为新启动窗体的窗体,如 Form1 窗体。

(3) 单击"确定"按钮,关闭"工程属性"对话框。

#### 2. 无启动窗体时启动

无启动窗体时,必须使用 Sub Main 过程。欲将 Sub Main 过程设为启动对象,可选择 "工程"|"工程属性"命令。在弹出的"工程属性"对话框中选择"通用"选项卡,从"启 动对象"列表框中选定 Sub Main 选项。输入代码如下:

01	Sub Main()	
02	Dim intStatus As Integer	
03	'调用一个函数过程来检验用户状态	
04	intStatus = GetUserStatus	
05	'根据状态显示某个启动窗体	
06	If intStatus = 1 Then	
07	frmMain.Show	'显示主窗体
8 0	Else	
09	frmPassword.Show	'显示登录窗体
10	End If	
11	End Sub	

在 Visual Basic 应用程序开发中,如果程序非常复杂,最好调用 Sub Main 过程设定为 启动过程。

## 5.1.5 设定过程作用范围

Sub 过程创建时需要设定过程作用域。作用域就是过程的作用范围。一般情况下,包括以下作用域。

#### 1. 公共过程

在定义过程时使用 Public 关键字,该过程在整个程序中可以被调用。使用 Public 关键 字可以允许一个窗体或模块中的过程被其他窗体或模块调用。声明公共过程必须要为其设 定唯一名称。

#### 2. 局部过程

在定义过程时使用 Private 关键字,该过程在定义的窗体或模块中可以被调用。使用 Private 关键字不允许一个窗体或模块中的过程被其他窗体或模块调用。声明局部过程在当 前作用域必须要为其设定唯一名称。该作用域优点是节省了系统资源,缺点是无法被其他 程序调用。

#### 3. 静态变量过程

应用程序开发时,变量在过程中创建并使用,当过程终止时自动清空该变量。个别情况希望保留变量值以备将来调用过程时使用,可以使用 Static 关键字将其声明为静态变量。静态变量在运行时自动保存变量数值,如打印机报表页码变量就是这样一种静态变量。

• 53 •

●注意:作用域的使用可以节省资源、提高效率。尽量使用局部过程或函数,减少使用公共过程和静态变量过程,Visual Basic 可以更加有效地管理内存。

## 5.2 块与块之间的关系——传递参数

Visual Basic 提供了非常灵活的传递参数方式,允许使用可选参数和可变参数,使用传值引用和传地址引用方法。传递的 Arglist 参数的标准语法如下:

[Optional] [ByVal | ByRef] [ParamArray] varName[()] [As type] [= defaultvalue]

参数说明:

- Optional 为可选项,表明传递的参数不是必需的。如果使用了该关键字,则 Arglist 中的后续参数都必须是可选的,且使用 Optional 关键字声明。如果使用了 ParamArray,则任何参数都不能使用 Optional 声明。
- □ ByVal 为可选项,表明传值引用参数。
- □ ByRef 为可选项,表明传地址引用参数。
- □ ParamArray 为可选项,可选的。只用于 Arglist 的最后一个参数,指明最后这个参数是一个 Variant 类型的 Optional 数组。使用 ParamArray 关键字可以提供任意数目的参数。ParamArray 关键字不能与 ByVal、ByRef 或 Optional 一起使用。
- □ Defaultvalue 为可选项,表明预先赋予参数默认值。

## 5.2.1 可选参数和可变参数

可选参数表明指定参数是否可选。可变参数表明接受任意数目的参数。

1. 可选参数

过程中可能接受指定参数范围内的一个或几个参数。如果为可选参数,可以使用 Optional 关键字来声明,并在过程中使用 Ismissing 函数来判断是否传递可选参数。

【示例 5-3】创建 Multi 过程。本示例将定义 3 个参数,最后一个为可选参数,计算 3 个数的乘积并调用 msgbox 函数显示计算结果。

(1)新建一个工程,并新建一个窗体。

(2)选择"工程"|"添加模块"命令,打开"添加模块"对话框。在"新建"选项卡 中选择"模块"图标,单击"确定"按钮,弹出"模块编辑"对话框。

(3)选择"工具"|"添加过程"命令,弹出"添加过程"对话框。

(4) 在"名称"文本框内输入模块名称,如 multi。在"类型"选项框中选择"子程序" 单选按钮,单击"确定"按钮。

(5) 在模块编辑窗体可以看见自动添加 multi 子程序。在 Sub 和 End Sub 之间手动输入以下代码。

01 Public Sub multi(a As Integer, b As Integer, Optional c)

• 54 •

02	Dim s As Integer	'声明变量
03	s = a * b	'赋予变量
04	'使用 ismissing 语句判断参数是否包含内容。	
05	If Not IsMissing(c) Then	
06	s = s * c	
07	End If	
08	MsgBox "数字乘积为: " & s	'显示信息
09 E	nd Sub	

○说明:本例过程定义3个参数a、b、c,其中c为可选参数。当调用 multi 过程时,首先 计算a和b乘积并赋予变量s。使用 Ismissing 函数来判断参数c是否存在。如果 参数存在,继续计算s与c的乘积并赋予s。最后调用 msgbox 函数显示最后3个 参数的乘积结果。

(6) 双击工具箱中的 CommandButton 图标,自动在窗体中添加一个 CommandButton 控件,设置 caption 属性为"可选参数乘积"。

(7) 双击"可选参数乘积"按钮,弹出代码编辑窗口。输入代码如下:

Call	multi(12,	3)		'调用子程序
Call	multi(12,	З,	2)	

#### 2. 可变参数

在函数中如果指定参数个数,则应与函数说明的个数相等,并对应被传递。Visual Basic 中可能传递可变参数个数。可以使用 ParamArray 来表明函数将接收任意数目的可变参数。

【示例 5-4】创建接收不定参数计算乘积的 multi\_more 过程。该示例将改写 5-3 示例中 代码计算不定参数的乘积。

(1) 选项"工具"|"添加过程"命令,弹出"添加过程"对话框。

(2) 在"名称"文本框内输入模块名称,如 multi\_more。在"类型"选项框中选择"子程序"单选按钮,单击"确定"按钮。

(3) 在"模块编辑"对话框中可以看见自动添加 multi\_more 子过程。在 Sub 和 End Sub 之间手动输入以下代码。



(4) 双击工具箱中的 CommandButton 图标,自动在窗体中添加一个 CommandButton 控件,设置 caption 属性为"可变参数乘积"。

(5) 双击"可变参数乘积"按钮,弹出代码编辑窗口。输入代码如下:

```
Call multi_more(1, 2, 4, 5, 6)
```

## 5.2.2 传值引用和传地址引用

传值引用表明传递变量副本。传地址引用表明读取变量内存地址访问变量数值。

#### 1. 传值引用

Byval 关键字表明为传值引用。该关键字只传递变量的副本。如果过程更改此变量数 值,则该改动只影响副本,不影响变量本身。标准语法如下:

```
Sub prog(Byval a as integer)
'添加代码
End Sub
```

2. 传地址引用

Byref 关键字表明为传地址引用。该关键字传递变量内存地址去访问实际变量内容。 通过过程可以永远改变传递变量值。传地址引用为 Visual Basic 默认的传递方式。其标准 语法如下:

```
Sub prog(Byref a as integer)
'添加代码
End Sub
```

【示例 5-5】传值和传地址引用过程,传递数值并计算两个数值之和。

(1) 选择"工具" | "添加过程"命令,弹出"添加过程"对话框。

(2) 在"名称"文本框内输入模块名称,如 Sub\_ref。在"类型"选项框中选择"子程 序"单选按钮,单击"确定"按钮。

(3) 在"模块编辑"对话框中可见自动添加的 Sub\_ref 子过程。在 Sub 和 End Sub 之间手动输入以下代码。

```
Public Sub Sub_ref(ByRef a As Integer) 

a = a + 10
End Sub
```

(4) 再添加一个过程, 名称为 Sub val。输入代码如下:

(5) 双击工具箱中的 CommandButton 图标,自动在窗体中添加一个 CommandButton 控件,设置 caption 属性为"传值和传地址"。双击"传值和传地址"按钮,弹出代码编辑 窗口。输入代码如下:

01	Private Sub	Command1_Click()	'按钮单击事件
02	Dim a As	s Integer	'声明变量

• 56 •

03	Dim b As Integer	
04	a = 10	'赋予变量数值
05	b = 10	
06	Call Sub_ref(a)	'调用子程序
07	Print "传地址引用结果为: " & a	「显示结果 <b>□</b> □ <b>□</b> ×
08	Call Sub_val(b)	(5) FORTH (1) (5) (5) (5) (5) (5) (5) (5) (5) (5) (5
09	Print "传值引用结果为: " & b	後值引用结果为: 10 传值和传地址
10 E	Ind Sub	

## 5.3 函 数

函数为编辑代码时可以被程序调用带有返回值的代码段。本节将介绍函数定义和使用 方法。

## 5.3.1 定义函数

Function 过程又被称为函数过程,与 Visual Basic 内置函数完全相同。函数与子过程区别为,函数带有返回值,而子过程没有返回值。定义函数的标准语法如下:

```
[Public | Private | Static] Function Name [(Arglist)] [As type]
  [statements]
  [Name = Expression]
  [statements]
  [Name = Expression]
End Function
```

参数说明: Name 为必需项,作为调用函数名称。Arglist 为可选项,作为参数列表。 参数列表格式如下:

[ByVal | ByRef] varName [As type]

其中, ByVal 和 ByRef 表明引用方式, ByVal 为传值引用, ByRef 为传地址引用。区 别在于传值引用返回值不变, 传地址引用可以更改返回值。默认是 ByRef 传地址引用。

- □ Statements 为可选项,可以为函数、过程和代码。
- □ Name=Expression 为可选项,为函数返回值。
- □ Exit Function 为可选项,可以从当前 Function 过程中退出。程序从调用该 Function 过程的语句之后的语句执行。在 Function 过程的任何位置都可以使用 Exit Function 语句。
- □ End Function 为必需项,表明函数过程结束。

添加函数和添加过程非常相似,注意在"类型"选项框中选择"函数"单选按钮。

【示例 5-6】创建返回两个参数之和的 calc()函数。该函数可以计算 a 和 b 的总和,并

• 57 •

作为函数返回值返回程序。

(1)选择"工程"|"添加模块"命令,打开"添加模块"对话框。新建一个模块。

(2)选择"工具"|"添加过程"命令,打开"添加过程"对话框。在"名称"文本框 内输入模块名称 calc。注意,在"类型"选项框中选择"函数"单选按钮,范围选择"公 有的"单选按钮。"模块编辑"对话框自动添加以下代码。

Public Function calc() End Function

(3) 手动添加两个参数, 定义一个返回值。代码修改如下:

```
01 Public Function calc (ByRef a As Integer, ByRef b As Integer) As Integer
```

```
02 'calc 为返回值
```

```
03 calc = a + b
```

```
04 End Function
```

## 5.3.2 设定函数作用范围

函数创建以后,要设定使用范围以及是否可以被其他程序调用。函数包括以下作用域。

#### 1. 公共函数

如果希望函数在整个程序中可以被调用,需要在定义函数时使用 Public 关键字。使用 Public 关键字可以允许一个窗体或模块中的函数被其他窗体或模块调用。声明公共函数必 须要为其设定唯一名称。

#### 2. 局部函数

如果希望函数在定义的窗体或模块中可以被调用,需要在定义函数时使用 Private 关键 字。使用 Private 关键字不允许一个窗体或模块中的函数被其他窗体或模块调用。声明局部 函数在当前作用域必须要为其设定唯一名称。局部函数优点是减少占用系统资源,缺点是 无法被其他作用域调用。

#### 3. 静态变量函数

应用程序运行时,变量在函数中创建并使用,当程序终止时自动清空该变量。个别情况希望保留变量值以备将来调用函数时使用,可以使用 Static 关键字将其声明为静态变量。静态变量在运行时自动保存变量数值,如打印机报表页码变量就是这样一种静态变量。

#### 5.3.3 调用函数

函数调用和过程调用基本相似,唯一不同之处在于函数可以有返回值。如果无返回值, 可以用 Call 语句调用函数。其标准语法如下:

#### Call funName [Arglist]

参数说明: Call 关键字为必需项。FunName 为函数名称。Arglist 参数列表为可选项。 如果有返回值,使用函数名+参数形式调用。其标准语法如下:

• 58 •

```
Return=funName [Arglist]
```

其中, Return 为必需项。funName 为函数名称。Arglist 参数列表为可选项。

【示例 5-7】调用 Calc()函数,返回值为两个参数之和。该函数可以计算 a 和 b 的总和, 并作为函数返回值返回程序。

(1) 双击工具箱中的 CommandButton 图标,自动在窗体添加一个 CommandButton 控件,设置 caption 属性为"求和函数"。双击"求和函数"按钮,弹出代码编辑窗口。在左侧对象下拉列表框中选择 Command1 选项,右侧过程下拉列表框中选择 Click 事件。输入以下代码:

0	1 Private Sub Command1_Click()	'按钮单击事件
0	2 Dim smsg As String	'声明变量
0	3 smsg = "a和b总和为: "	
0	4 '调用函数计算	
0	5 Dim n As Integer	Form1
0	6 n= calc(12, 56)	a和b总和为: 68
0	7 Print smsg & n	·显示信息
0	8 End Sub	

(2) 该函数采用 n=calc(12,56)的方式返回数值,并在窗体上显示出来最后结果为68。

## 5.3.4 传递参数和返回值

函数传递参数的方法与过程完全相同,在此不再赘述。函数与过程最重要的区别是函数有返回值。函数的数据类型决定其返回值的数据类型。可以在 Arglist 之后使用 As 关键 字来指定函数返回值的数据类型。在 Arglist 中,可以使用 As 子句来指定任何传给该过程的参数的数据类型。不但可以指定为任何标准数据类型,还可以在 Arglist 中指定 As Any 来禁止类型检查,从而允许将任意数据类型传递给该过程。

【示例 5-8】创建计算矩形面积 rect 函数。该函数可以计算矩形面积,并作为函数返回 值返回程序。

(1)选择"工程"|"添加模块"命令,打开"添加模块"对话框。新建一个模块。

(2)选择"工具"|"添加过程"命令,在"名称"文本框内输入模块名称 rect。注意, 在"类型"选项框中选择"函数"单选按钮,范围选择"公有的"单选按钮。"模块编辑" 对话框自动添加以下代码。

```
Public Function rect()
End Function
```

(3) 手动添加两个参数, 定义一个返回值。代码修改如下:

```
01 Public Function rect(ByRef a As Integer, ByRef b As Integer) As Integer
02 rect = a * b
```

```
03 End Function
```

(4) 双击工具箱中的 CommandButton 图标,自动在窗体添加一个 CommandButton 控件,设置 caption 属性为"矩形面积",双击"矩形面积"按钮,弹出代码编辑窗口,左侧 对象选择"Command1",右侧过程选择"click"过程。输入以下代码:

01	<pre>Private Sub Command1_Click()</pre>	'按钮单击事件
02	Dim n As Integer	'声明变量
03	Dim a As Integer	
04	Dim b As Integer	
05	a = 12	'赋值变量
06	b = 22	
07	'调用 rect 函数计算	
8 0	n = rect(a, b)	5. Form1 (1) 100 (1) (1) 100 (1) 100 (1) 100 (1) 100 (1) 100 (1) 100 (1) (1) 100 (1) 100 (1) (1) 100 (1) (1) 100 (1) (1) (1)
09	Print "矩形面积为:" & n	短形面积
10	End Sub	

□ ↓ Gineric Construction C

5.4 小 结

本章主要介绍了过程和函数的定义,以及如何根据应用程序的需要调用过程和函数。 本章重点及难点包括过程和函数如何传递参数、函数如何返回值。其中,函数传递参数和 返回值结构复杂、难于掌握。第6章将介绍数组。

## 5.5 习 题

创建过程、函数实现计算阶乘数之和、计算多个数值中最大值。

(1) 创建 Countall 过程计算阶乘数之和。

计算1:+5:+8:+10! 总和,并在窗体上显示计算结果。

新建一个工程,并新建一个窗体,添加一个 CommandButton 控件,命名为"计算阶乘", 打开代码编辑窗口,输入代码如下:

```
01 Private Sub Command1_Click()
02 Call Countall(1, 5, 8, 10) '调用计算过程
03 End Sub
```

【参考过程实现】

01	<pre>Private Sub Countall(ParamArray nums())</pre>	
02	Dim total As Long, x As Variant	
03	Dim n As Integer, s As Long	
04	total = 0	'计算总和
05	For Each x In nums	'枚举循环
06	n = 1	
07	s = 1	'! 计算总和
8 0	For n = 1 To x	'For 循环计算
09	s = s * n	
10	Next	
11	total = total + s	'计算总和赋值
12	Next x	
13	Print "总和为" & total	'在窗体上显示
14	End Sub	

• 60 •

程序运行时,单击"计算阶乘"按钮在窗体上显示计算结果。运行窗体如图 5.4 所示。 (2) 创建和调用 max 函数计算最大值。

计算3个数的最大值,并在窗体上显示计算结果。

添加一个 CommandButton 控件, 命名为"求最大值"。打开代码编辑窗口, 输入以下 核心代码。

01 Private Sub Command2\_Click() 02 Print "三个数最大值为: " & max(4, 25, 10) '显示函数返回值 03 End Sub

#### 【参考函数实现】

```
01 Private Function max(ByRef a As Integer, ByRef b As Integer,
02
                                                 ByRef c As Integer)
03
      If a > b Then
                                                     '显示比较值
04
                                                     '比较大值赋值 max
        max = a
05
      Else
06
        max = b
07
      End If
                                                     '最大值 max 与 c 比较
80
      If max < c Then
                                                     '结果为 true 赋值 max
09
        max = c
10
      End If
11 End Function
```

程序运行时,单击"求最大值"按钮在窗体上显示计算结果,如图 5.4 所示。



图 5.4 程序运行结果