

# 第 1 章 Excel VBA 入门

本章主要讲解 Excel VBA 的基础入门知识,主要包括使用宏和 VBE 环境。宏是由 VBA 语句组成的命令集,实现 Excel 操作。VBE 是用户创建和编辑代码的主要功能,在本章中,将详细介绍 VBE 的创建操作方法。

## 1.1 创建和管理宏

用户对 VBA 知识还不熟悉的时候,宏是最好的入门方法。用户可以在不了解 VBA 语法和对象的前提下,通过录制宏了解宏完成的基本功能。在本小节中,将详细讲解如何创建和管理宏。

### 案例 1 录制宏

#### 1. 功能说明

本实例的主要功能是演示如何在 Excel 2007 中录制宏,在用户对 Excel VBA 某些方面的知识不太了解的时候,通过录制宏来了解 Excel 如何“记录”操作是最好的方法。

#### 2. 语法说明

本案例涉及到的语法是“录制”技术,这是 Excel 本身就提供的功能。用户只需启动“录制”功能,然后进行操作。在操作结束后,设置“录制”结束,就完成整个录制过程。

#### 3. 案例说明

在案例将需要录制一个简单的宏,这个宏实现的功能是设置单元格中数字的属性(位数、颜色等)。设置这些属性的过程并不复杂,用户可以直接查看录制后的代码。

#### 4. 编写代码

(1) 输入数值,然后启动录制宏功能。在单元格 B2 中输入数值“123”,然后单击“开发工具”|“代码”|“录制宏”按钮,如图 1.1 所示。

(2) 设置宏。在打开的“录制新宏”对话框中,在“宏名”选框中输入“设置格式”,在“说明”选框中输入“该宏的功能是设置数字格式”,并设置宏的快捷键是“Ctrl+t”,如图 1.2 所示。

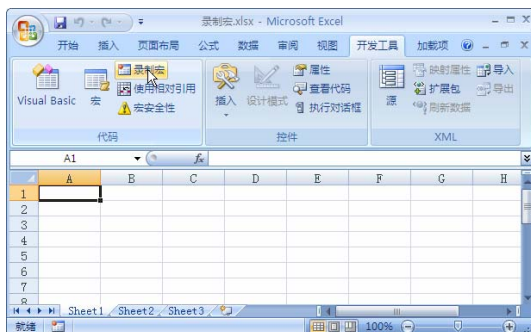


图 1.1 启动录制宏的功能

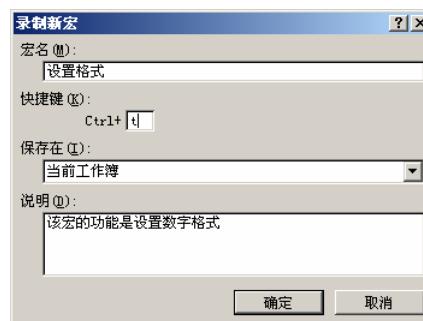


图 1.2 设置宏的属性

(3) 设置数字的格式。选中数字单元格，然后选择“开始”选项卡，然后依次数值数字的位数、格式和颜色等，如图 1.3 所示。

(4) 结束录制宏。切换到“开发工具”选项卡，然后单击“停止录制”按钮，如图 1.4 所示。

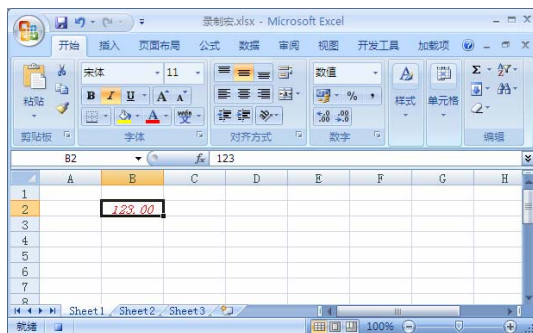


图 1.3 设置数字属性

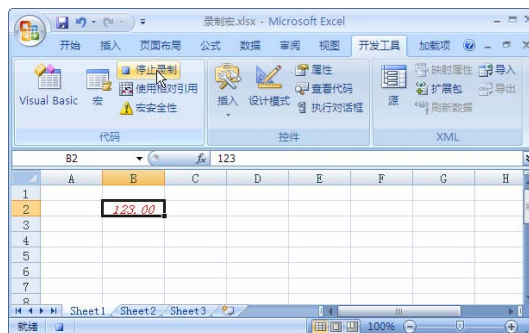


图 1.4 结束录制宏

## 5. 运行结果

查看录制宏对应的 VBA 代码。调出 VBE，查看宏对应的代码，如图 1.5 所示。



图 1.5 宏对应的程序代码

## 6. 程序分析

在使用 Excel 宏的时候，尽量将宏的有用信息都填写完整，这样当用户使用宏的时候，可以通过信息来了解宏的内容。

## 案例 2 编辑宏

### 1. 功能说明

在 Excel VBA 中，尽管用户可以使用 Excel 进行自动“录制”宏，但是，Excel 自动记录的 VBA 代码具有“自动性”。常常需要用户对其代码进行编辑，在本小节中，将详细演示如何编辑宏。

### 2. 语法说明

本小节将讲解如何进入编辑宏的环境，然后对 VBA 宏代码进行修改。在本小节中，将重新编写了宏，希望用户从最后的代码对比中，查看出编辑宏的方法。

### 3. 案例说明

本例的主要功能是编写和案例 1 自动宏功能一样的宏代码，也就是，编写设置数字格式的宏代码。

### 4. 编写代码

(1) 选择编辑宏。选择“开发工具”|“代码”|“宏”选项，打开“宏”对话框，然后选择对应的宏，单击“编辑”按钮，如图 1.6 所示。

(2) 查看对应的宏。当用户单击“编辑”按钮，进入 VBE 环境，编写新的宏代码，如图 1.7 所示。

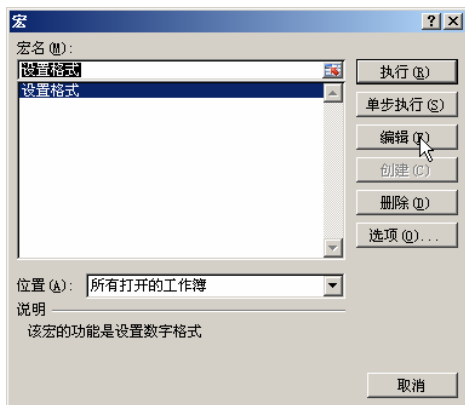


图 1.6 选择编辑宏

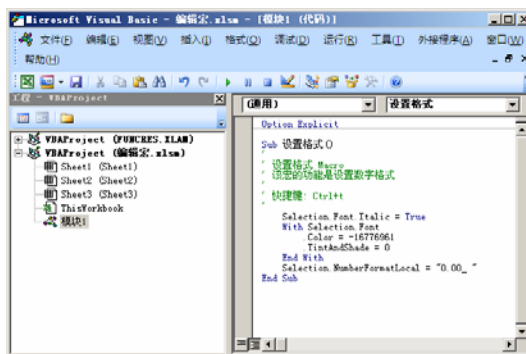


图 1.7 查看相应的宏

## 5. 运行结果

查看新的代码，其具体代码如下：

```
Sub 设置格式()  
'  
' 设置格式 Macro  
' 该宏的功能是设置数字格式  
'  
' 快捷键: Ctrl+t  
'  
  
    Selection.Font.Italic = True  
    With Selection.Font  
        .Color = -16776961  
        .TintAndShade = 0  
    End With  
    Selection.NumberFormatLocal = "0.00_ "  
End Sub
```

## 6. 程序分析

本小节所涉及的程序代码知识和后面章节的内容类似，关于其具体的语法知识，请用户参考后面章节的内容。

# 案例 3 运行宏

## 1. 功能说明

在 Excel VBA 中，用户编写宏的主要功能是运行宏。利用宏的代码，多次完成类似的操作和功能。

## 2. 语法说明

运行宏在技术上等于程序的运行和调试。如果录制完成的宏，则属于程序的运行，因为不会出现语法错误。

## 3. 案例说明

在本例中，将介绍如何在 Excel 中运行宏。为了让读者对宏的功能有直观的认识，本小节将运行前面案例的宏。

## 4. 编写代码

(1) 添加原始的数据。前面案例中宏的功能是设置数字格式，因此，首先需要输入原始数据，如图 1.8 所示。

(2) 选择运行宏。选择单元格 A2，然后选择“开发工具”|“代码”|“宏”选项，如图 1.9 所示。

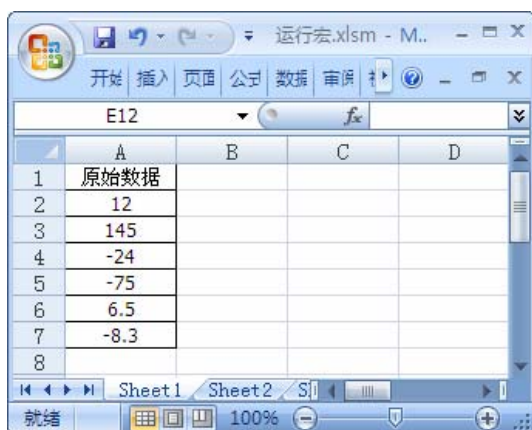


图 1.8 原始数据

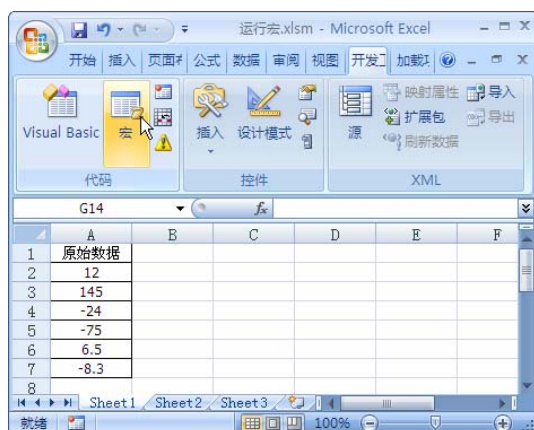


图 1.9 选择运行宏

(3) 执行宏功能。在前面步骤中打开的“宏”对话框中，单击“执行”按钮，选择执行对应的宏功能，如图 1.10 所示。

(4) 查看宏运行的结果。当单击“执行”按钮后，查看运行的结果，如图 1.11 所示。

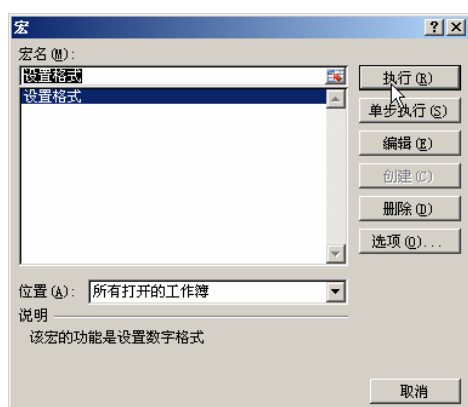


图 1.10 执行宏

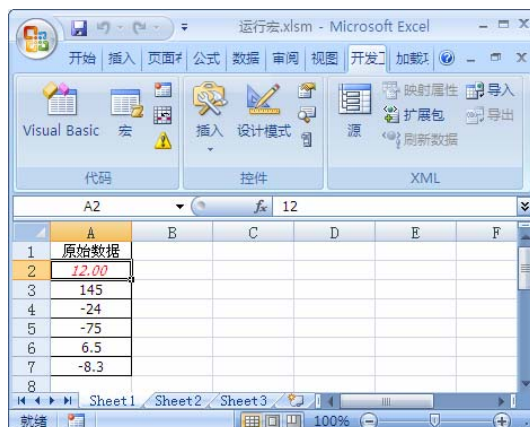


图 1.11 查看宏执行的结果

## 5. 运行结果

依次选择其他单元格，然后分别运行宏，查看得到的结果，如图 1.12 所示。



图 1.12 执行的结果

## 6. 程序分析

由于用户录制和编写的宏是设置数字格式的，因此，执行这些宏会有直观的设置结果。但是，不是每个宏都会产生直观的结果，有些宏只是完成中间处理功能。

## 1.2 使用 VBE

在本小节中，用户将了解 VBA 的编辑环境 VBE。熟悉 VBE 各部分和组件的功能和设置，将会给读者编写代码带来很大的便利。用户可以根据自己编码的习惯显示不同的窗口，和设置 VBE 的编码属性。在本小节中，将结合具体例子讲解如何使用 VBE。

### 案例 4 添加模块

#### 1. 功能说明

模块是 VBA 编程的起点，几乎所有的代码都保存在某个模块中。本小节将讲解如何添加模块。

#### 2. 语法说明

在 Excel VBA 中，通常有三种方法向工程中添加模块：

- ☐ 录制宏。当用户录制宏时，系统将自动插入 VBA 模块，并将宏代码放在该模块中。
- ☐ 在 VBE 中，选择工具栏中的“插入”|“模块”选项。
- ☐ 在 VBE 中，右击“工程”子窗口，弹出快捷菜单，选择“插入”|“模块”选项。

#### 3. 案例说明

在本例中，首先演示如何在 VBE 中添加新的模块，然后设置模块的属性。

#### 4. 编写代码

(1) 添加新的模块。右击“工程”子窗口，弹出快捷菜单，选择“插入”|“模块”选项，如图 1.13 所示。添加后的新模块如图 1.14 所示。

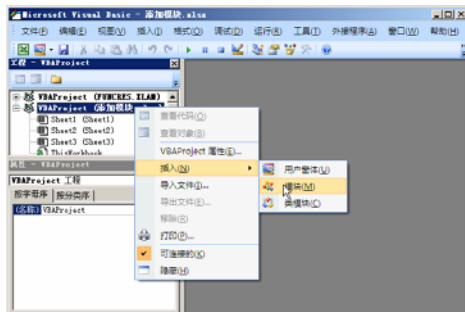


图 1.13 进入 VBE 环境

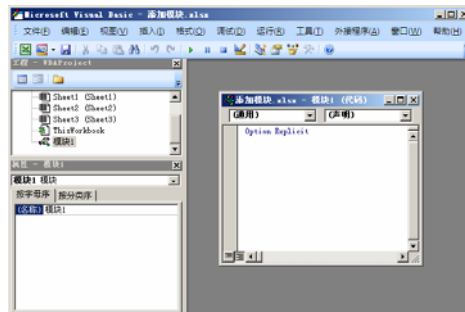


图 1.14 添加的新模块

(2) 修改模块的名称。在“属性”列表中，在“名称”选项中输入“MyModule”，修改模块的名称，结果如图 1.15 所示。

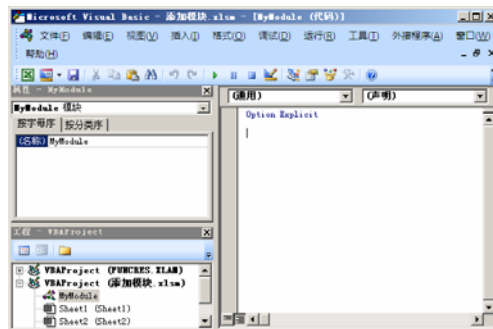


图 1.15 修改模块的名称

#### 5. 运行结果

从前面的结果中，用户可以查看到，添加的模块如图 1.16 所示。



图 1.16 添加的模块

## 6. 程序分析

当工程中的模块较少时，可以使用默认的名称。但是，当工程比较复杂，模块比较多时，建议用户在添加模块的时候，自行设置符合工程属性的名称。

## 案例 5 导出模块

### 1. 功能说明

当用户在实际开发时，特别是大型程序时，会遇到功能雷同的模块。例如，在多个项目中，都需要使用“设置格式”功能。用户可以避免每次都编写同样的模块，而只需编写模块，然后将其导出保存。当需要使用这种功能时，导入这个模块。

### 2. 关键技术

在 Excel VBA 中，根据输出类型的不同，用户可导出三种模块：模块、类模块和窗体。根据不同类型的模块，可以导出的文件类型也不同。具体文件类型如下：

- ❑ BAS: 导出模块文件的扩展名；
- ❑ FRM 或 FRX: 导出用户窗体的扩展名；
- ❑ CLS: 导出类模块的扩展名。

### 3. 案例说明

在本例中，将演示如何将编写的模块导入并保存。

### 4. 编写代码

(1) 复制前面小节中的“录制宏”文件，并将其名称修改为“导出模块”。然后按快捷键“ALT+F11”，进入 VBE，如图 1.19 所示。

(2) 选择导出模块。选中前面章节添加的模块，右击鼠标，在弹出的快捷菜单中选择“导出文件”选项，如图 1.20 所示。

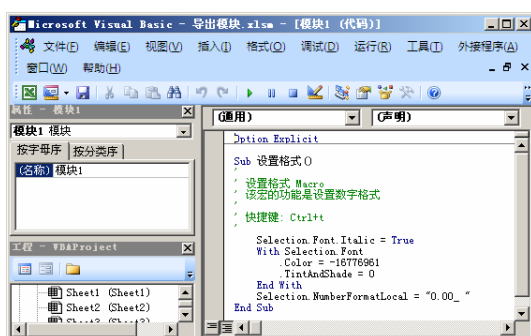


图 1.19 进入“导出模块”的 VBE

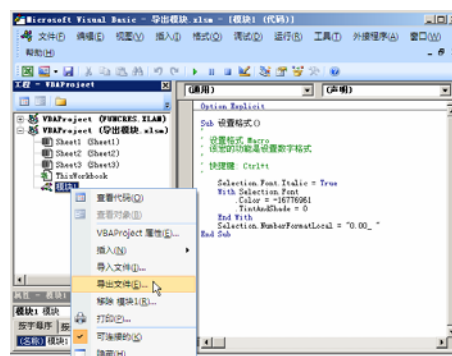


图 1.20 导出模块



(3) 选择文件的保存路径。选择对应的选项后，Excel 会弹出“导出文件”对话框，在其中选择导出模块的保存路径，如图 1.21 所示。

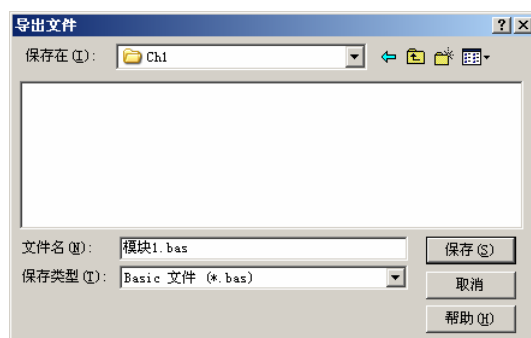


图 1.21 选择保存路径

## 5. 运行结果

查看导出的模块。在模块保存的路径中，用记事本打开模块文件，如图 1.22 所示。



图 1.22 查看导出的模块

## 6. 程序分析

当用户导出模块时，可以自定义模块的名称，用户可以根据定义的模块名称，来查看或者分析模块功能。

# 案例 6 导入模块

## 1. 功能说明

前面案例已经讲解到，导出模块的主要功能是为了在其他情况下，能否使用该模块的功能。其中，使用模块功能的主要方法就是导入模块。

## 2. 语法说明

导入模块是导出模块的反向操作。当用户导入模块时，VBE 根据扩展名来判断导入的模块类型，具体是模块、类模块或窗体。

## 3. 案例说明

在本案例中，将演示如何向 VBA 文档中导入之前保存的模块文件。

## 4. 编写代码

(1) 新建 Excel 文件，然后进入 VBE 环境，选择工具栏中的“文件”|“导入文件”命令，如图 1.23 所示。

(2) 选择需要导入的文件。在前面步骤中打开的“导入文件”对话框中，选择前面步骤中导出的文件模块，然后单击“打开”按钮，如图 1.24 所示。

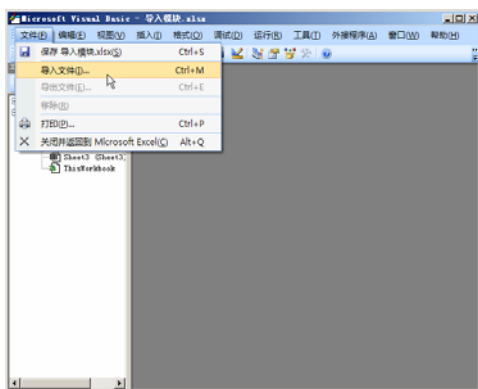


图 1.23 选择导入文件

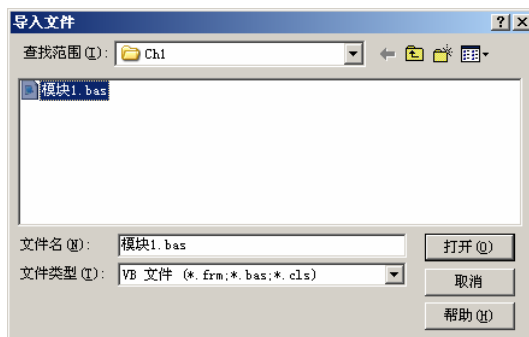


图 1.24 导入文件

## 5. 运行结果

当用户单击“打开”按钮后，Excel VBA 中导入相应的模块，如图 1.25 所示。

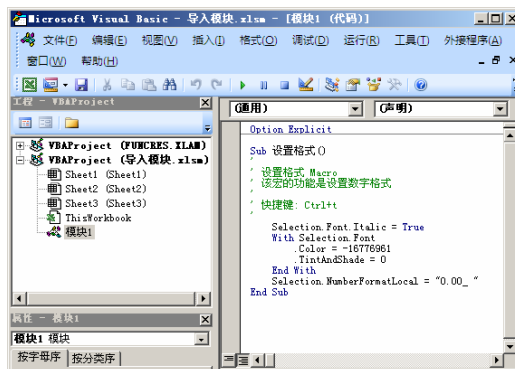


图 1.25 导入的模块

## 6. 程序分析

当用户向 Excel VBA 文件中导入其他类型的模块文件，操作方法和本案例类似，这里就不重复介绍。

## 案例 7 显示属性列表

## 1. 功能说明

当用户在进行 VBA 编程时，经常需要用到对象的属性或者方法。Excel VBA 对同一个对象提供了多个属性。如何用户自行输入属性的代码，会经常犯错误。在 VBE 中，当用户在窗口中输入对象名称和句点时，窗口中会显示属性列表，用户可以在列表中选择。

## 2. 语法说明

在 VBE 代码窗口中，当用户输入对象名和一个句点后，如果没有弹出“属性/方法”列表框，可使用以下方法显示该列表框。

- ❑ 按下“Ctrl+J”组合键；
- ❑ 单击“编辑”工具栏中的“属性 / 方法列表”按钮；
- ❑ 右击代码窗口中的对象代码，在弹出的快捷菜单中选择“属性 / 方法列表”命令。

### 3. 案例说明

本例将演示如何在 VBE 的代码窗口中显示属性 / 方法列表。

#### 4. 编写代码

(1) 新建 Excel 文件，然后进入 VBE 环境，插入新的模块和过程，并在其中输入代码 “ThisWorkbook”，然后输入一个句点，系统会弹出属性和方法的列表，如图 1.26 所示。

(2) 完成代码语句。在属性列表中选择“ActiveSheet”选项，完成代码语句，如下：

ThisWorkbook.ActiveSheet.Cells(2, 3) = "Excel VBA"

如图 1.27 所示。

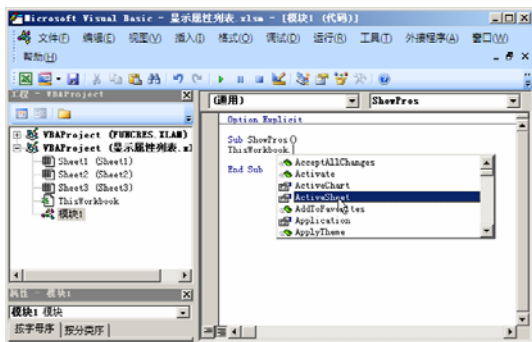


图 1.26 显示列表

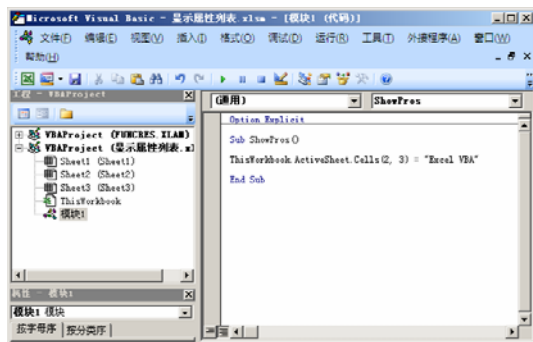


图 1.27 完成代码语句

## 5. 运行结果

当用户完成代码后，按快捷键 F5，测试程序，得到的结果如图 1.28 所示。

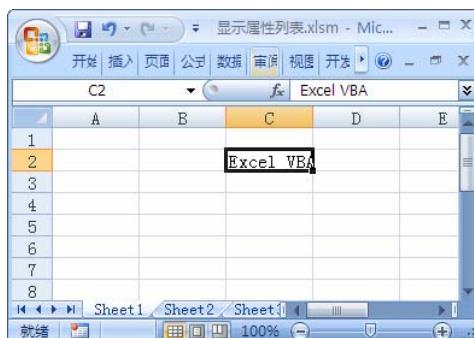




图 1.28 程序调试结果

## 6. 程序分析

在“属性/方法”列表中，图标表示内容属于方法，图标表示内容属于属性。通过上面的图标，用户可以很简单的判断对应的内容。