Excel VBA
应用程序专业设计
实用指南

黄睿　马然　编著

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

Excel VBA应用程序专业设计实用指南

黄 睿 马 然 编著

電子工業出版社
Publishing House of Electronics Industry
北京 · BEIJING

内 容 简 介

Excel在数据挖掘和企业信息处理中占有重要的开发地位。本书采用更为前瞻的视角向读者介绍了当今Excel开发的先进知识，从系统架构师的角度为读者分析了Excel开发的环境和应当如何才能够开发出专业的Excel VBA应用程序。本书将Excel的开发分为8个部分向读者进行了介绍，从应用程序的结构和设计方法以及错误排除，到面向对象的程序设计和面向接口的程序设计等。通过本书的学习，读者能够从更高层次认识Excel VBA的开发。

本书面向的读者对象是有一定基础的Excel开发人员。

图书在版编目（CIP）数据

Excel VBA应用程序专业设计实用指南/黄睿，马然编著.—北京：电子工业出版社，2006.4
ISBN 7-121-02304-0

Ⅰ.E… Ⅱ.①黄…②马… Ⅲ.电子表格系统，Excel—程序设计 Ⅳ.TP391.13

中国版本图书馆CIP数据核字（2006）第012370号

责任编辑：朱 巍
特约编辑：陈 虹
印 刷：北京天竺颖华印刷厂
出版发行：电子工业出版社
北京市海淀区万寿路173信箱 邮编：100036
北京市海淀区翠微东里甲2号 邮编：100036
经 销：各地新华书店
开 本：787×1092 1/16 印张：15.625 字数：380千字
印 次：2006年4月第1次印刷
定 价：23.00元

凡购买电子工业出版社的图书，如有缺损问题，请向购买书店调换，若书店售缺，请与本社发行部联系。联系电话：010-68279077。质量投诉请发邮件至zlts@phei.com.cn，盗版侵权举报请发邮件至dbqq@phei.com.cn。

前　言

在VBA图书市场上，绝大部分图书的重点都是介绍基本的Excel VBA知识，很少有书籍以通观全局的视角来具体把握整个Excel VBA的开发。本书就是这样一本书，我们希望能够帮助读者了解当今Excel VBA开发的前沿知识，并且能够站在一个更高的角度上看待Excel VBA的开发，帮助用户在需求分析、功能设计、应用程序的结构设计和用户界面设计几个不同的方面来看待Excel VBA的开发。这样读者就能够从一个应用程序架构设计师的角度来重新审视Excel VBA的开发。

本书分成8个部分来介绍Excel VBA开发中的一些高级议题。

在第1章中分别介绍了基本的Excel应用程序的运行和调试知识，同时也介绍了Excel对象模型中一些比较重要的类和基本操作。首先通过宏的介绍，拉开了对Excel VBA应用程序整体开发的大幕。通过简单的宏，可以了解Excel应用程序的运行机制和机理——任何复杂的Excel应用程序都可以看做是复杂的宏的演变。本章还分别介绍了Application对象、Workbook对象、Worksheet对象以及VBA中基本的开发调试知识。

在第2章中首先分析了Excel在实际工程中应用的价值，让读者能够全面地衡量在工程中使用Excel作为开发工具是否恰当，并且给出了其他几大类的产品作为比较。可以看到，Excel在整个开发产品的过程中占有比较独特的地位，但是总的说Excel适合于开发单机版、访问量与开发维护量都不是很大、对数据有分析建模功能的应用程序。当然，使用Excel作为开发工具还有一个巨大的优势——Excel的用户群巨大，可以使应用程序的用户大大缩短学习周期。

在第3章中重点关注了应该从哪些方面着手编写成熟的代码，从而在编写Excel VBA代码的时候能够养成良好的习惯并打下坚实的基础。我们分别关注了命名规范的使用、应用程序的组织结构以及代码注释等问题，最后还了解了如何实现版本控制的方法。总之在本章中，给读者提供的信息是在后面的Excel VBA开发中需要在编程时一些要注意的问题的综合。作为开发的整体而言，本章中的内容不会影响任何功能开发，但是会给编程开发习惯做一个很好的约束，使我们能够开发出整洁高效的代码。

在第4章中介绍了当使用工作表为用户界面设计基础的时候需要注意的一些问题，其中包括如何从整体上设计用户界面，以及一些设计用户界面上的小技巧。同时强调了设计用户界面并不仅仅是设计外观，更重要的是要把用户界面当做是三层应用程序结构中的第一层，要重视应用程序和用户的互动与沟通。

在第5章中介绍了Excel加载项的概念和种类，同时也介绍了如何创建相应的加载项。加载项的使用扩展了使用Excel的能力，同时它为创建不同类型的应用程序提供了更多的选择，特别是在实现优秀结构的应用程序的时候，加载项的引入可以称为实现良好结构的基石。

在第6章中关注了在Excel VBA开发过程中非常重要的类模块的开发，通过使用模块，我们在思考问题的时候就可以以对象为中心来分析需求，这对于VBA开发而言是非常重要的

财富。在本章中根据类模块的特点，分别介绍了构成类模块的要件，创建类模块的过程，以及如何在类模块中使用事件来进一步增强开发的层次。

在第7章中主要了解面向接口的编程方式，它并不是面向对象编程的对立面，而是面向对象编程的良好补充。通过使用面向接口的编程方式，可以使实现部分和功能定义进行分离，特别是在设计接口相同而实现差别很大的类时使用，可以使整个程序的结构趋向合理化，而且更加容易扩展，并容易使用。

在第8章中具体了解了如何使用CommandBar对象的相关知识，先后了解了CommandBar对象模型以及一些重要的对象，然后通过一些具体的实例让读者了解其中重要的方法和属性。

在第9章中介绍了如何使用用户窗体作为应用程序的界面。用户窗体向用户提供了一种标准的与系统进行沟通的手段，充分利用用户窗体可以增强应用程序的亲和度。本章分别介绍了用户窗体中的一些基本类和控件，并通过具体的代码介绍了它们的使用方法。

在第10章中向读者介绍了在Excel VBA中如何使用 Windows API调用的一些相关知识，分别介绍了在VBA中使用Windows API概念以及一些在编程中经常用到的Windows API使用操作。

在第11章中了解了在Visual Basic编辑器中调试程序的相关知识。我们首先归纳了错误的类型，然后具体介绍了在Visual Basic编辑器中调试程序的一些基本知识。在下一章中将了解编程中另一个重要的概念——错误处理。错误处理和程序调试是一个既有联系又有区别的编程中的两个方面。

在第12章中介绍了错误处理的概念以及一些常用的方法，最后还介绍了在较为复杂的应用程序中应当使用集中的错误处理方式。错误处理在整个应用程序的开发过程中占有很重要的地位，任何严肃的应用程序都应该认真地对待错误处理功能。

在第13章中介绍了企业信息系统这种较为特殊的Excel应用程序，这种应用程序需要在很多方面都要覆盖Excel的原有功能，所以使用这种应用程序必须能够灵活驾驭Excel的方方面面。在本书中因为受到篇幅限制，无法更进一步地介绍企业信息系统的各种表现形式，但是它们的共性是非常明显的，即都需要存储和恢复Excel设置，并且设定独特的用户环境，在此基础上设计用户自定义的用户界面。在这种应用程序中会大量用到在前面章节中所提到的各种设计技巧，如类模块的设计、面向接口的设计、命令条设计以及用户窗体的设计等。

在第14章中了解了如何使用Excel VBA访问并更改数据库中的数据。首先对关系型数据库进行了概要的介绍，之后介绍了数据库应用的核心——SQL语句，在后面主要介绍了ADO对象模型以及对象模型中的RecordSet、Connection和Command对象的用途。

本书由黄睿、马然编著，马忠颖、张慧丽、王军、王冬梅、龚蕊、张慧霞、董立强、孙妍和王鹏等同志在本书的编写过程中给予了很大的帮助，在此感谢他们的辛勤付出。同时要感谢北京美迪亚电子信息有限公司和电子工业出版社的广大老师们的辛勤工作，没有他们就不会有本书的出版。由于编者水平有限，错误在所难免，希望读者能够批评指正。

目　录

第一部分　Excel应用程序结构

第二部分　程序设计最佳实践

第三部分　扩展应用程序结构

第四部分　高级用户界面设计

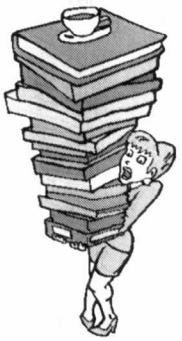
第五部分　Windows API调用

第六部分 Excel VBA应用程序的调试与错误处理

第七部分 企业信息系统

第八部分 Excel与外部数据

第一部分　Excel应用程序结构



第1章

Excel应用程序介绍

开发Excel应用程序就好像是要建一座房子，所不同的是，建造房子所使用的建筑材料都是由Excel提供的，在这一章中我们将会熟悉这些建筑材料，了解它们的特性以及在整个工程实践中它们都能发挥什么样的作用。本书的读者对象一般认为应当具备初步的Excel使用知识和基本的VBA编程基础。之所以写一章主要是为了让大家更系统地了解Excel的构件，并给大家提供一些独特的思路，使我们在后面的编程实践中能够更加灵活地应用这些要素。同时这一章的另一个作用就是希望能作为一个基本的参考，如果读者在后面的实践中有任何疑惑，也许都可以在这一章中找到基本问题的答案。

建议比较熟练的Excel开发者略读这一章，甚至直接跳到下一章中，对于其他读者，我们建议你能够完成这一章的阅读后再开始阅读下一章。

本书的宗旨就是希望能够给读者提供一个从高处俯视Excel开发的一个平台，然后借助这个平台，能够从整体上把握整个开发的方方面面，把所有参与到开发中的构件都作为棋子来考虑，运筹帷幄。由于不希望本书只是作为提供小技巧和小参考的大全书籍，所以在这一章里面不会详尽说明Excel应用程序对象中每一个类的功能，主要给读者提供一个概念，在整个编程环境中每一个棋子将会在整个的开发中能够起的作用。

我们将从下面几个方面来介绍Excel应用程序基本结构：

- Excel基本组成部分
- VBA编程初步
- Excel开发环境
- VBA基本元素

1.1　Excel应用程序与VBA概述

Excel应用程序的基本形态是Excel的宏语言通过特定语法（VBA）来控制Excel内置对象，来完成一些自动化过程，而这些过程通常都用来完成一些重复性的工作。Excel中的宏

可以称为Excel应用程序编程的起点，它可以展示基本的Excel VBA程序工作的步骤。

虽然Excel VBA编程的能量远远大于宏能够达到的地方，但是这是一个很好的起点。

下面就通过使用Excel记录并运行宏，来看看宏和Excel VBA的运作模式。

1.1.1 记录宏

要开始记录宏，过程非常简单，请打开“工具|宏|录制新宏”，如图1-1所示。

然后，出现图1-2所示的窗口。

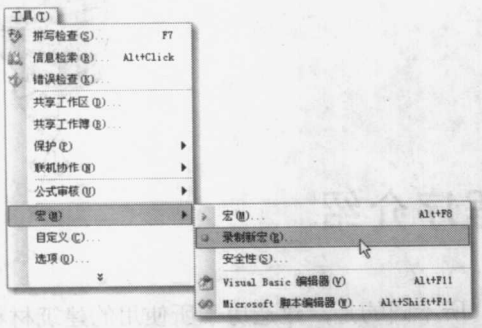


图1-1 打开“录制新宏”

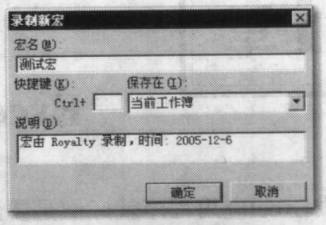


图1-2 “录制新宏”窗口

请在上面的窗口中输入宏的名称、说明文字以及保存位置，然后单击“确定”。在Excel工作簿上面将会出现一个浮动工具条，上面有一个停止按钮，如图1-3所示。

图1-3 停止按钮

此时就可以开始录制动作了，在这里将选中的单元格范围的颜色设置为黄色，设置完成之后，请单击上面的停止按钮。这样就成功地创建了一个宏，之后需要测试一下这个宏是否能够达到想要的效果。

下面来看一看Excel为我们创建的宏的VBA代码：依次打开“工具|宏|宏”，选中想要查看代码的宏，单击“编辑”。打开Visual Basic编辑器，如图1-4所示。

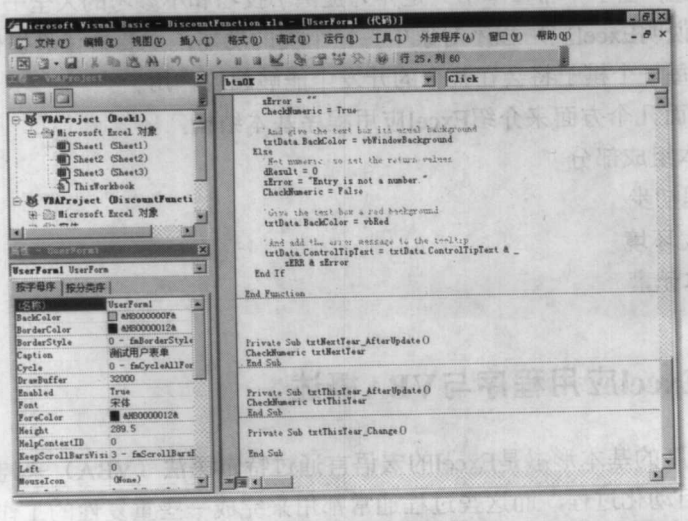


图1-4 Visual Basic编辑器窗口

这个窗口是由多个部分所组成的，具体每一个部分将会在本章的后面进行详细介绍。可以从这部分代码中很清晰地看到，它的作用就是将选中的范围的底色调整成为黄色（ColorIndex = 6）。

1.1.2 运行宏

运行宏的过程也是相当简单的，只需单击“工具|宏|宏”，然后选中需要运行的宏之后，单击“运行”就可以了。也可以通过在位于该对话框下面的“位置”下拉框中选择要运行的宏所在的工作簿，来运行其他工作簿中的宏。

也可以让宏在打开工作簿的时候自动运行，比如说要在某个特定的工作簿打开的时候设置工作簿的打印页面，只需将这个宏命名为Auto_Open即可，因为这个宏名对于Excel是一个特殊的名称。

还有其他三种打开宏的方式，分别是：

- 使用快捷键打开宏
- 在菜单中打开宏
- 在工具栏中打开宏

下面就简单介绍一下如何在这三种环境中打开宏。

使用快捷键打开宏

1. 依次打开“工具|宏|宏”，选中希望设置快捷键的宏。
2. 单击“选项”。
3. 在“快捷键”框中键入一个字母。如果键入的是小写字母，那么快捷键就是Ctrl+键入的字母，如果键入的是大写字母，那么快捷键就是Ctrl+Shift+键入的字母。

快捷键只能使用字母，不能使用数字，也不能使用其他字符。

4. 单击“确定”。
5. 单击“取消”。

在工具栏中打开宏

从“宏”对话框中运行宏的方式是内置运行宏的方式。如果要在多个工作表中应用宏，使用快捷键虽然快捷，但是有时并不直观，所以这种时候可以使用工具栏。将宏指定到工具栏上，然后就可以直接从工具栏上运行宏。当要使用的宏多起来的时候，将一两个宏指定到普通工具栏上可能难以找到要使用的宏，但是如果专门指定一个新的工具栏专门用做放置宏的按钮，效果则会好得多。

要将宏指定到一个新建的工具栏上：

1. 单击“工具”➤“自定义”，然后在“自定义”对话框中打开“工具栏”选项卡。如图1-5所示。
2. 在对话框中输入宏的名称，然后单击“确定”，新的工具栏就建好了。如图1-6所示。

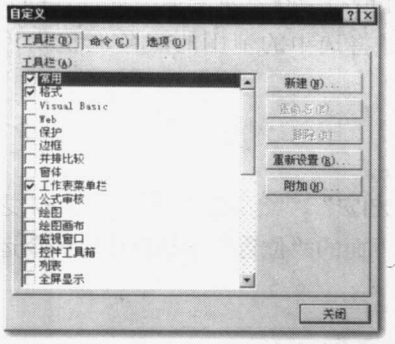


图1-5　自定义工具栏对话框

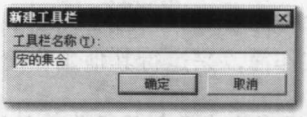


图1-6　输入新工具栏的名称

3. 在自定义对话框中单击“命令”选项卡。
4. 在“类别”列表中选择“宏”。
5. 将“自定义按钮”拖动到新的工具栏上。
6. 在刚刚建立的工具栏上，单击新添加的按钮，打开图1-7所示的对话框。

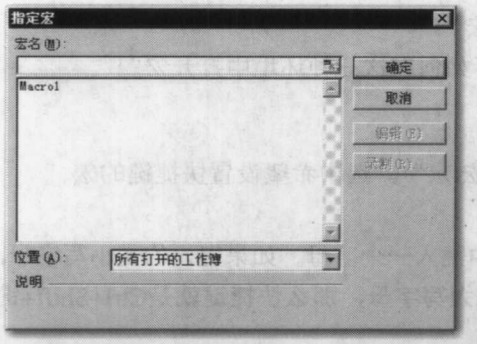


图1-7　指定置于工具栏上的宏

7. 选中宏，然后单击“确定”就可以了。

从菜单中运行宏

在菜单中添加宏命令和在工具栏中添加宏命令过程类似，可以根据需要来选择。在菜单中添加宏命令的步骤如下：

1. 单击“工具”➤“自定义”，在“自定义”对话框中打开“命令”选项卡。
2. 在“类别”列表中选择“新菜单”，将新菜单拖到主菜单栏的最右侧。
3. 在“自定义”对话框中的“类别”列表中，选择“宏”。
4. 将“自定义按钮”拖到新菜单的空白项上。
5. 单击刚刚添加的按钮，打开图1-7所示的对话框。
6. 单击“确定”即可。

调试宏

宏的调试也就是对VBA代码的调试对于一个严肃的VBA开发者而言是非常重要的。任

何代码几乎都不可能没有错误，那么要想修正错误，首先就要找到错误，这就是调试开始发挥作用的时候。VBE提供了一些帮助调试VBA代码的工具，可以发现在微软的工具中，包括Visual Studio中，调试的工具都是相似的。如果之前使用过调试工具，那么会发现它们有很多共同之处。不管是在Visual Studio或者是Java世界中的调试工具，调试带来的重要手段就是能够在代码中步进，以及在代码中查看任意变量的当前值的能力。

开始调试过程也非常简单，只要在VBE中打开需要调试的宏。然后在“调试”工具栏上选择“逐语句”运行，按下F8键步进，如图1-8所示。

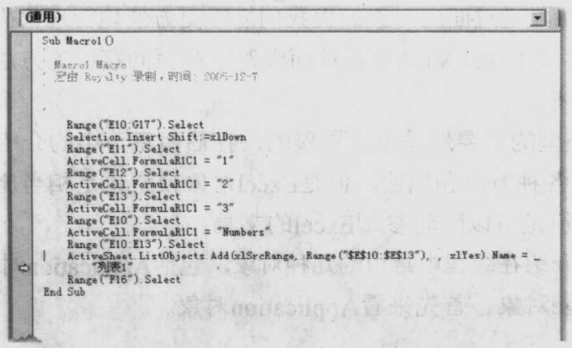


图1-8 步进调试宏

每次按下F8键都会使黄色高亮显示向下执行一句，可以很清晰地看到整个宏的执行过程。

如表1-1所示，可以看到其他调试过程中常用的快捷键。

表1-1 VBE中的快捷键

快捷键	作用
F5	运行宏代码
F8	逐语句执行过程
Shift+F8	在当前执行点所在位置的过程中，执行其余的程序行
Ctrl+Shift+F8	停止运行宏
Ctrl+F8	使得宏可以直接运行到光标所在处
Shift+F9	监视选中变量的值
F9	在当前光标处添加一个断点
Ctrl+Shift+F9	清除所有断点
Ctrl+F9	将选中语句作为下一条运行的语句

完成对宏的基本操作和调试的过程提供了很多关于Excel VBA代码运行调试的基本概念，Excel VBA的开发事实上也就是不同表现形式的宏的开发。熟悉上面的内容有助于我们进一步理解更高级的Excel VBA操作。

1.2 Excel对象模型

要知道什么是对象模型，首先应该了解什么是对象。简言之，对象就是一种将代码按照实体来进行代码封装的编程方式，虽然说VBA并不能算是完全面向对象的语言（不具备面向对象语言的全部要素），而是一种面向过程的语言，但是面向对象的思想在VBA中无处不在。如在Excel VBA中对任何Excel的控制都是通过操作Excel对象来实现的。面向对象编程是一种重要的思想方法，通过面向对象编程我们能够更方便地管理代码以及编写出具备良好结构的代码。关于如何在Excel VBA中设计面向对象和面向接口的方法与实践，我们将在本书的第11章中有所论述。

了解Excel对象模型的重要性是毋庸置疑的，在后面章节中的介绍中，我们将会大量用到Excel对象模型中的各种方法和属性，但是Excel对象模型也是相当庞杂的，要想全部了解几乎是不太可能的，但是可以随时参阅Excel的文档。

在本节中将重点介绍在编程中常用的几种对象，包括Application对象、Workbook对象、Worksheet对象和Range对象。首先来看Application对象。

1.2.1 Application对象

Excel对象模型的一个重要特点就是，所有对象都属于一个对象集合或包含多个对象。而在这个对象模型金字塔的顶端就是Application对象。Application对象模型如图1-9所示。通过Application对象可以引用需要操作的所有元素，如下面这一行就是如何设置某个单元格的值的操作：

```
Application.Workbooks(1).Worksheets(1).Cells(1,1) = 100
```

从上面的代码中可以看到Application、Workbooks(1)、Worksheets(1)对象和Cells对象之间的从属关系。在理论上操作Worksheet对象或者Workbook对象都需要使用“.”符号明确对象之间的关系，但是因为Worksheet和Workbook等对象在实践中要大量使用，这样Excel就将类似的对象都进行了处理，使得这些对象在使用时不需要输入完全合格的名称（即类似上面代码清单的写法）。所以，上面的写法可以缩略为

```
Cells(1,1) = 100
```

Application属性

使用Application对象通常是为了对整个应用程序的环境进行设置，或者是保存配置等操作，在本书的第6章将会大量使用这些Application对象的属性完成这些操作。Application对象包含170个设置Excel应用程序的属性的property。不需要完全了解这些属性，下面给出10个最为常用的属性。

- ActiveCell
- ActiveSheet
- ActiveChart
- ActiveWindow
- ActiveWorkbook
- RangeSelection

- Selection
- ScreenUpdating
- StatusBar
- ThisWorkbook

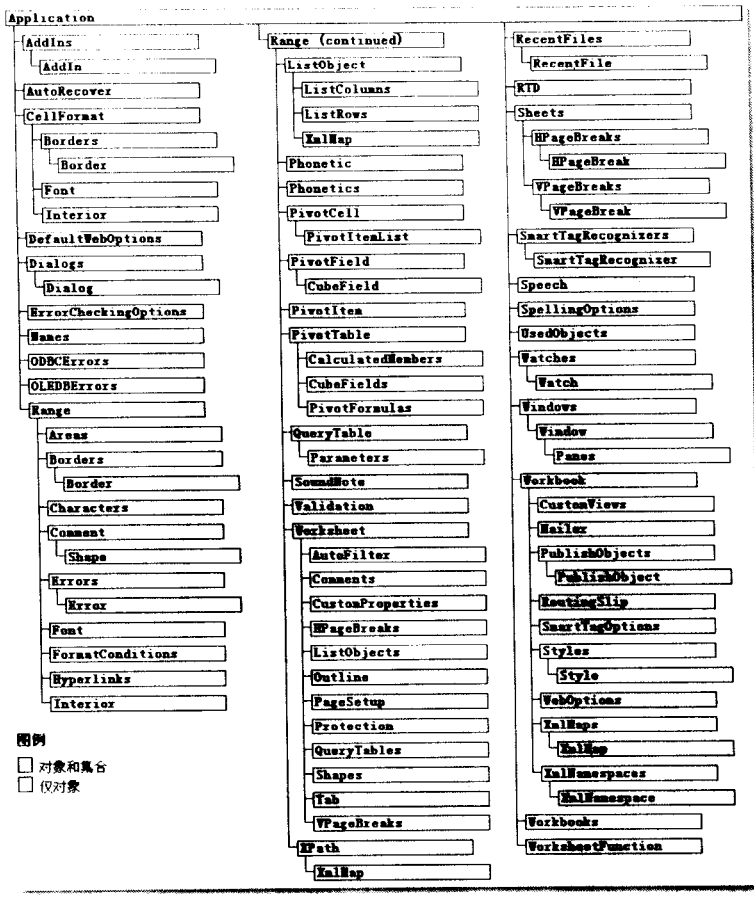


图1-9 Excel Application对象模型

因为这个属性非常常用，所以这里给出一些示例代码，来突出这些属性的实际用途。

1）ActiveCell

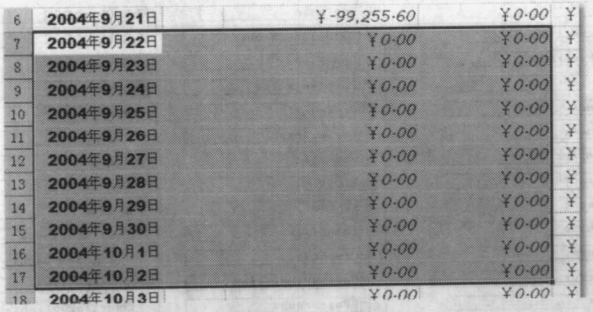
ActiveCell即当前的激活工作簿中激活工作表的当前激活单元格的引用，可以通过这个属性设置当前单元格的值、公式或者其他属性。代表清单1-1将设置当前激活单元格的字体为“黑体”，这是一个非常简单的例子，但是却能说明问题。

代码清单1-1 使用ActiveCell设置当前激活单元格的字体

```
Sub SetTheFont ( )
    With ActiveCell.Font
        .Name = "黑体"
        .Font Style = "黑体"
        .Size = 12
        .Strikethrough = False
        .Superscript = False
```

```
        .Subscript = False
        .OutlineFont = False
        .Shadow = False
        .Underline = xlUnderlineStyleNone
        .ColorIndex = xlAutomatic
    End With
End Sub
```

提示 通过录制宏生成的代码可以发现Selection对象似乎同ActiveCell属性类似，但是请注意Selection对象所代表的是选中的一个范围的单元格集合，在这个集合中只有一个单元格是ActiveCell（如图1-10所示：选中的单元格中左上角的为ActiveCell）。



6	2004年9月21日	¥-99,255.60	¥0.00	¥
7	2004年9月22日	¥0.00	¥0.00	¥
8	2004年9月23日	¥0.00	¥0.00	¥
9	2004年9月24日	¥0.00	¥0.00	¥
10	2004年9月25日	¥0.00	¥0.00	¥
11	2004年9月26日	¥0.00	¥0.00	¥
12	2004年9月27日	¥0.00	¥0.00	¥
13	2004年9月28日	¥0.00	¥0.00	¥
14	2004年9月29日	¥0.00	¥0.00	¥
15	2004年9月30日	¥0.00	¥0.00	¥
16	2004年10月1日	¥0.00	¥0.00	¥
17	2004年10月2日	¥0.00	¥0.00	¥
18	2004年10月3日	¥0.00	¥0.00	¥

图1-10 ActiveCell在一个Selection中

2）ActiveSheet属性

ActiveSheet属性返回当前激活的工作表。可以根据当前返回的工作表进行任意操作。在代码清单1-2中，我们将移动当前的工作表到当前工作表的后面。

代码清单1-2 工作表复制

```
Sub Mover1( )
    ActiveSheet.Move _
      After:=ActiveWorkbook.Sheets(ActiveWorkbook.Sheets.Count)
End Sub
```

3）ActiveChart属性

ActiveChart返回当前图表。

在代码清单1-3中，我们将打印当前工作表中所有嵌入的图表。

代码清单1-3 使用ActiveChart示例

```
Sub PrintEmbeddedCharts( )
    Dim ChartList As Integer
    Dim X As Integer
    ' 获取工作表中图表的个数
    ChartList = ActiveSheet.ChartObjects.Count
    For X = 1 To ChartList
        ' 选中图表对象
```

```
            ActiveSheet.ChartObjects(X).Select
            ' 将其变成激活图表
            ActiveSheet.ChartObjects(X).Activate
            ' 打印该图表
            ActiveChart.PrintOut Copies:=1
        Next
    End Sub
```

4）ActiveWindow属性

ActiveWindow属性返回当前激活的窗口，激活的窗口就是在最前面的窗口。

对于ActiveWindow需要注意的一点是，Caption属性等同于Name属性，即当设置了Caption属性之后，就可以使用这个名称来代替编号来引用这个窗口。

ActiveWindow在编程中会经常用来显示特定的单元格，或者用来调整窗口的显示比例。这一点在设计应用程序的界面上会常常用到。在代码清单1-4中，我们使用ActiveWindow的ScrollRow和ScrollColumn方法将窗口定位到特定对象的位置，只需在输入框中输入对象的名称即可。

代码清单1-4 使用ActiveWindow

```
Sub ScrollToObject( )
    Dim ObjName As String
    On Error GoTo Finish

    ObjName = Application.InputBox(Prompt:="请输入要查找对象的名称", _
              Title:="查找对象")
    ActiveWindow.ScrollRow = _
       ActiveSheet.DrawingObjects(ObjName).TopLeftCell.Row
    ActiveWindow.ScrollColumn = _
       ActiveSheet.DrawingObjects(ObjName).TopLeftCell.Column
     Exit Sub
Finish:
    MsgBox "对象名称无效"
End Sub
```

5）ActiveWorkbook属性

该属性返回当前激活的工作簿。

6）DisplayAlerts属性

此属性的主要作用是在宏的运行过程中，通过设置该属性的值（TruelFalse）达到阻止警告框出现的目的，这在自动化处理如删除无用的工作簿或覆盖文件等操作方面是十分有用的，该属性默认设置为True，当该属性设置为False的时候，Excel会使用默认的选择回答上述警告框。其他的例外情况是OverWrite警告框的默认值是Yes，但是当DisplayAlerts设置为False时，Excel默认为No。

7）RangeSelectionlSelection属性

当工作表中已选定一个图形对象时，Selection属性返回的是一个图形对象而不是一个