



超值光盘
辅助学习

赠送

一本**155**页的使用代码参考书

44个办公应用模板

兼容
Excel
2003/2007

EXCEL VBA

常用代码实战大全

袁竹平 编著

实例丰富，强调实践

精选**150**个典型范例，近**245**个技巧。重点强调实践性，书中的很多范例都来源于编者实际开发的程序，通过对这些范例的学习，可以增强读者的动手实践能力。

优化代码，解析详细

本书摒弃了VBA复杂理论的讲解，在选择知识讲解的重点和范例的设置上以需要为先，以实用为主，知识点针对性强，代码短小精炼。

循序渐进，难度适中

本书在结构上采用由浅入深的方式，站在初学者的立场，以解决初学者可能遇到的问题和急需掌握的知识为主线，来完成本书的编写。



化学工业出版社

EXCEL VBA

常用代码实战大全



化学工业出版社

· 北京 ·

本书重点强调实践性,旨在帮助 Excel 初学者和有一定 Excel VBA 编程基础、希望进一步提高的读者掌握 VBA 编程设计的技巧。全书精选 150 个典型范例,近 245 个技巧,以实用为主,知识点针对性强,并逐层对代码辅以深入浅出地剖析,让读者通过范例的学习,快速将 Excel VBA 编程应用到实际生活和工作中。

本书主要介绍 VBA 常用对象的应用技巧、控件及用户窗体的应用技巧、对话框的应用技巧、函数的应用技巧,使用 VBA 操作文件等技巧,最后介绍了制作员工花名册、收据打印系统、员工考勤系统三个综合实例。

本书内容丰富、图文并茂,适合初步掌握 VBA 的基本知识,但需要进一步提高编程能力的初学者,以及希望通过 VBA 提高工作效率的各类从业人员。也适合各大、中专院校相关专业师生和企业、相关单位的培训班学员阅读。

图书在版编目 (CIP) 数据

Excel VBA 常用代码实战大全 / 袁竹平编著.

北京: 化学工业出版社, 2010. 7

ISBN 978-7-122-08647-1

ISBN 978-7-89472-290-4 (光盘)

I. E... II. 袁... III. 电子表格系统, Excel
IV. TP391.13

中国版本图书馆 CIP 数据核字 (2010) 第 093341 号

责任编辑: 王思慧 张 敏

装帧设计: 王晓宇

责任校对: 周梦华

出版发行: 化学工业出版社 (北京市东城区青年湖南街 13 号 邮政编码 100011)

印 装: 化学工业出版社印刷厂

787mm×1092mm 1/16 印张 24¹/₄ 字数 583 千字 2010 年 7 月北京第 1 版第 1 次印刷

购书咨询: 010-64518888 (传真: 010-64519686) 售后服务: 010-64518899

网 址: <http://www.cip.com.cn>

凡购买本书,如有缺损质量问题,本社销售中心负责调换。

定 价: 59.80 元 (含 1CD-ROM)

版权所有 违者必究

前 言

Excel 2007 是 Office 家族成员中功能强大、技术先进、使用方便的数据管理和分析系统。它采用电子表格方式进行数据处理，工作直观方便。

为了使 Excel 2007 发挥最大功效，可以使用 VBA 开发电子表格应用程序。使用 VBA 可以为 Excel 应用程序提供新的功能或增强现有的功能，从而减少在 Excel 中进行操作所需的工作量。

VBA 语言简单易学，但功能非常强大。用户可以将 Excel 作为开发平台，建立各种应用程序。例如实现使重复的任务自动化，简化模板的使用，使其成为开发平台，创建报表以及对数据进行复杂的操作和分析等。通过将 VBA 与 Excel 结合起来，不仅可以提高工作效率，也使得对各种数据的处理得心应手。

本书特色

1. 实例丰富，强调实践

为了让读者易于理解概念，掌握编程设计的技巧，本书列举了大量范例。范例的选择力求做到与知识点密切结合并突出应用技巧；范例的设计以读者易于理解为目标，同时突出实用性，能够解决在实际操作中遇到的问题。通过这些范例，读者便可更加深入地理解相关概念和语法，从而达到灵活使用 VBA 编写应用程序的目的。

另外，本书重点强调实践性，书中的很多范例都来源于编者实际开发的程序，通过对这些范例的学习，可以增强读者的动手实践能力。

2. 优化代码，解析详细

基于 Excel 的 VBA 程序设计，代码并不复杂、算法并不高深，实用并简便易行才是真正需要的。本书摒弃了 VBA 复杂理论的讲解，在选择知识讲解的重点和范例的设置上以需要为先，以实用为主，知识点针对性强，代码短小精炼。这极大地优化了本书的内容结构，增强了本书的含金量。本书所涉及的程序源代码层次清楚，语句简洁，便于读者理解，代码解析内容丰富。

3. 循序渐进，难度适中

本书面向 Excel VBA 的初级用户，以让读者快速掌握 Excel VBA 程序设计为目标。本书在结构上采用由浅入深的方式，站在初学者的立场，以解决初学者可能遇到的问题和急需掌握的知识为主线，来完成本书的编写。本书采取循序渐进的讲解方式，层层推进，步步深入，学习难度适中，学习梯度设置科学，让读者非常容易掌握。

4. 超值光盘，辅助学习

为了帮助读者更直观地学习，本书所附光盘中提供了书中所有范例的 VBA 代码文件，

相当于 200 页的使用代码参考书。

本书内容

本书共分为 11 章，循序渐进地介绍了 Excel VBA 的编程技巧。

第 1 章~第 5 章讲解了 VBA 常用对象的使用技巧，包括单元格对象、工作表对象、工作簿对象、图表、图形对象和应用程序对象。

第 6 章讲解了控件及用户窗体的使用技巧。

第 7 章讲解了对话框的使用技巧，包括消息框、输入框及内置对话框。

第 8 章讲解了函数的使用技巧，包括 VBA 函数、工作表函数及自定义函数。

第 9 章讲解了使用 VBA 操作文件的技巧。

第 10 章讲解了 VBA 的一些其他应用技巧。

第 11 章讲解了综合应用 VBA 编程技巧制作员工花名册、收据打印系统和员工考勤系统。

本书读者

本书内容丰富、图文并茂，适合初步掌握 VBA 的基本知识，但需要进一步提高编程能力的初学者，以及希望通过 VBA 提高工作效率的各类从业人员。也适合各大、中专院校相关专业师生和企业、相关单位的培训班学员阅读。

编 者

目 录

第 1 章 Range (单元格) 对象	1
范例 1 单元格的引用方法.....	1
1-1 使用 Range 属性引用单元格区域.....	1
1-2 使用 Cells 属性引用单元格区域.....	2
1-3 使用快捷记号实现快速输入.....	2
1-4 使用 Offset 属性返回单元格区域.....	3
1-5 使用 Resize 属性返回调整后的单元格区域.....	4
范例 2 选定单元格区域的方法.....	4
2-1 使用 Select 方法选定单元格区域.....	4
2-2 使用 Activate 方法选定单元格区域.....	5
2-3 使用 Goto 方法选定单元格区域.....	6
范例 3 获得指定行的最后一个非空单元格.....	6
范例 4 使用 SpecialCells 方法定位单元格.....	8
范例 5 查找特定内容的单元格.....	9
5-1 使用 Find 方法查找特定信息.....	9
5-2 使用 Like 运算符进行模式匹配查找.....	12
范例 6 替换单元格内字符串.....	14
范例 7 复制单元格.....	14
7-1 复制单元格区域.....	14
7-2 仅复制数值到另一区域.....	17
范例 8 禁用单元格拖放功能.....	18
范例 9 设置单元格格式.....	19
9-1 设置单元格字体格式.....	19
9-2 设置单元格内部格式.....	20
9-3 为单元格区域添加边框.....	21
范例 10 单元格的数据有效性.....	24
10-1 添加数据有效性.....	24
10-2 判断是否存在数据有效性.....	26
10-3 动态的数据有效性.....	27
范例 11 单元格中的公式.....	28
11-1 在单元格中写入公式.....	28
11-2 判断单元格是否包含公式.....	30
11-3 判断单元格公式是否存在错误.....	31
11-4 取得公式的引用单元格.....	32

11-5 将公式转换为数值	33
范例 12 为单元格添加批注	33
范例 13 合并单元格操作	34
13-1 判断单元格区域是否存在合并单元格	34
13-2 合并单元格时连接每个单元格的文本	35
13-3 合并内容相同的连续单元格	36
13-4 取消合并单元格时在每个单元格中保留的内容	37
范例 14 高亮显示选定单元格区域	38
范例 15 双击被保护单元格时不弹出提示消息框	39
范例 16 单元格录入数据后的自动保护	40
范例 17 Target 参数的使用方法	43
17-1 使用 Address 属性	43
17-2 使用 Column 属性和 Row 属性	43
17-3 使用 Intersect 属性	44

第 2 章 Sheet (工作表) 对象

范例 18 引用工作表的方法	45
18-1 使用工作表名称	45
18-2 使用工作表索引号	45
18-3 使用工作表代码名称	46
范例 19 选择工作表的方法	46
范例 20 遍历工作表的方法	47
20-1 使用 For...Next 语句遍历工作表	48
20-2 使用 For Each...Next 语句遍历工作表	48
范例 21 工作表的添加与删除	49
范例 22 禁止删除指定工作表	52
范例 23 禁止更改工作表名称	53
范例 24 判断是否存在指定工作表	54
范例 25 工作表的深度隐藏	55
范例 26 工作表的保护与取消保护	56
范例 27 自动建立工作表目录	58
范例 28 循环选择工作表	60
范例 29 在工作表中一次插入多行	60
范例 30 删除工作表中的空行	61
范例 31 删除工作表的重复行	62
范例 32 定位删除特定内容所在的行	63
范例 33 判断是否选中整行	64
范例 34 限制工作表的滚动区域	65
范例 35 复制自动筛选后的数据区域	65

范例 36	使用高级筛选功能获得不重复记录	66
范例 37	获得工作表打印页数	67
第 3 章	Workbook (工作簿) 对象	69
范例 38	引用工作簿的方法	69
38-1	使用工作簿名称引用工作簿	69
38-2	使用工作簿索引号引用工作簿	70
38-3	使用 ThisWorkbook 属性引用工作簿	70
38-4	使用 ActiveWorkbook 属性引用工作簿	71
范例 39	新建工作簿	71
范例 40	打开指定的工作簿	72
范例 41	判断指定工作簿是否被打开	73
41-1	遍历 Workbooks 集合方法	73
41-2	使用错误处理方法	74
范例 42	关闭工作簿时不弹出保存对话框	74
42-1	使用 Close 方法关闭工作簿	75
42-2	单击工作簿关闭按钮关闭工作簿	76
范例 43	禁用工作簿的关闭按钮	76
范例 44	保存工作簿的方法	77
44-1	使用 Save 方法保存工作簿	77
44-2	使用 SaveAs 方法保存工作簿	77
44-3	使用 SaveCopyAS 方法保存工作簿	78
范例 45	保存指定工作表为工作簿	78
范例 46	不打开工作簿取得其他工作簿数据	79
46-1	使用公式取得数据	79
46-2	使用 GetObject 函数取得数据	80
46-3	隐藏 Application 对象取得数据	81
46-4	使用 ExecuteExcel4Macro 方法取得数据	81
46-5	使用 SQL 连接取得数据	82
第 4 章	Shape (图形)、Chart (图表) 对象	84
范例 47	在工作表中添加图形	84
范例 48	导出工作表中的图片	87
范例 49	在工作表中添加艺术字	89
范例 50	遍历工作表中的形状	90
范例 51	移动、旋转图形	91
范例 52	自动插入图片	93
范例 53	固定图片的尺寸和位置	95
范例 54	使用 VBA 自动生成图表	96
范例 55	批量制作图表	98

范例 56	导出工作表中的图表	101
第 5 章	Application (应用程序) 对象	102
范例 57	取得 Excel 版本信息	102
范例 58	取得当前用户名称	102
范例 59	实现简单的计时器功能	103
范例 60	屏蔽、更改组合键功能	104
范例 61	设置 Excel 标题栏	105
范例 62	自定义 Excel 状态栏	106
范例 63	灵活关闭 Excel	106
范例 64	暂停代码的运行	107
范例 65	防止用户中断代码运行	108
范例 66	隐藏 Excel 主窗口	109
66-1	设置 Visible 属性为 False	109
66-2	将窗口移出屏幕	110
第 6 章	控件与用户窗体	112
范例 67	使用文本框输入数值	112
范例 68	限制文本框的输入长度	113
范例 69	验证文本框输入的数据	114
范例 70	文本框回车后自动输入数据	114
范例 71	文本框的自动换行	115
范例 72	格式化文本框数据	117
范例 73	使控件始终位于可视区域	117
范例 74	高亮显示按钮控件	118
范例 75	为列表框添加列表项的方法	119
75-1	使用 RowSource 属性添加列表项	119
75-2	使用 ListFillRange 属性添加列表项	120
75-3	使用 List 属性添加列表项	121
75-4	使用 AddItem 属性添加列表项	122
范例 76	去除列表项的空行和重复项	123
范例 77	移动列表框的列表项	124
范例 78	允许多项选择的列表框	126
范例 79	多列列表框的设置	128
范例 80	加载二级组合框	130
范例 81	使用 RefEdit 控件选择区域	132
范例 82	使用多页控件	133
范例 83	使用 TabStrip 控件	134
范例 84	在框架中使用滚动条	136
范例 85	制作进度条	137

范例 86	使用 DTP 控件输入日期	138
范例 87	使用 spreadsheet 控件	140
范例 88	使用 TreeView 控件显示层次	142
范例 89	使用 Listview 控件	145
89-1	使用 Listview 控件显示数据列表	145
89-2	在 Listview 控件中使用复选框	147
89-3	调整 Listview 控件的行距	149
89-4	在 Listview 控件中排序	151
89-5	Listview 控件的图标设置	152
范例 90	使用 Toolbar 控件添加工具栏	154
范例 91	使用 StatusBar 控件添加状态栏	156
范例 92	使用 AniGif 控件显示 GIF 图片	159
范例 93	使用 ShockwaveFlash 控件播放 Flash 文件	160
范例 94	注册自定义控件	162
范例 95	不打印工作表中的控件	164
范例 96	遍历控件的方法	165
96-1	使用名称中的变量	165
96-2	使用对象类型	166
96-3	使用程序标识符	166
96-4	使用 FormControlType 属性	167
范例 97	使用程序代码添加控件	168
97-1	使用 Add 方法添加表单控件	168
97-2	使用 AddFormControl 方法添加表单控件	169
97-3	使用 Add 方法添加 ActiveX 控件	171
97-4	使用 AddOLEObject 方法添加 ActiveX 控件	172
范例 98	禁用用户窗体的关闭按钮	173
范例 99	屏蔽用户窗体的“关闭”按钮	174
范例 100	为用户窗体添加图标	175
范例 101	为用户窗体添加最大最小化按钮	176
范例 102	屏蔽用户窗体的标题栏和边框	177
范例 103	显示透明的用户窗体	178
范例 104	为用户窗体添加菜单	179
范例 105	自定义用户窗体的鼠标指针类型	182
范例 106	用户窗体的打印	183
范例 107	设置用户窗体的显示位置	185
107-1	调整用户窗体的显示位置	185
107-2	由活动单元格确定显示位置	186
范例 108	用户窗体的全屏显示	187
108-1	设置用户窗体的大小为应用程序的大小	187

108-2 根据屏幕分辨率设置	187
范例 109 在用户窗体中显示图表	188
109-1 使用 Export 方法显示图表	188
109-2 使用 API 函数显示图表	189
范例 110 用户窗体运行时调整控件大小	190
范例 111 使用代码添加用户窗体及控件	192
范例 112 以非模式显示用户窗体	196
第 7 章 使用对话框	197
范例 113 使用 MsgBox 函数显示消息框	197
范例 114 自动关闭的消息框	199
114-1 使用 WshShell.Popup 方法关闭消息框	199
114-2 使用 API 函数关闭消息框	200
范例 115 使用 InputBox 函数输入数据	201
范例 116 使用 InputBox 方法	203
116-1 输入指定类型的数据	203
116-2 获得选定的单元格区域	204
范例 117 使用内置对话框	205
117-1 调用 Excel 内置对话框	205
117-2 获取所选文件的文件名和路径	207
117-3 使用“另存为”对话框备份文件	208
范例 118 调用操作系统的“关于”对话框	210
第 8 章 函数的使用	211
范例 119 使用时间和日期函数	211
119-1 计算程序运行时间	211
119-2 获得当月的最后一天	211
119-3 计算某个日期为星期几	212
119-4 计算两个日期的时间间隔	214
119-5 获得指定时间间隔的日期	215
119-6 格式化时间和日期	215
范例 120 使用字符串处理函数	217
范例 121 判断表达式是否为数值	218
范例 122 自定义数值格式	219
范例 123 使用 Rond 函数进行四舍五入运算	220
范例 124 使用 Array 函数创建数组	221
范例 125 将字符串按指定的分隔符分开	222
范例 126 使用动态数组去除重复值	223
范例 127 调用工作表函数	225
127-1 使用 Sum 函数求和	225

127-2 查找工作表中最大、最小值	226
127-3 不重复值的录入	227
范例 128 使用个人所得税自定义函数	228
范例 129 使用人民币大写函数	230
范例 130 判断工作表是否为空表	231
范例 131 查找指定工作表	232
范例 132 查找指定工作簿	233
范例 133 取得应用程序的安装路径	234

第9章 文件操作 236

范例 134 导入文本文件	236
134-1 使用查询表导入	236
134-2 使用 Open 语句导入	237
134-3 使用 OpenText 方法导入	238
范例 135 创建文本文件	239
135-1 使用 Print # 语句将数据写入文本文件	239
135-2 使用 SaveAs 方法将数据另存为文本文件	240
范例 136 获得文件修改的日期和时间	241
范例 137 查找文件和文件夹	242
范例 138 获得当前文件夹	243
范例 139 创建和删除文件夹	244
范例 140 重命名文件或文件夹	244
范例 141 复制指定的文件	245
范例 142 删除指定的文件	246
范例 143 使用 WSH 处理文件	246
143-1 获取文件信息	247
143-2 取得文件基本名	248
143-3 查找文件	249
143-4 搜索文件	249
143-5 移动文件	250
143-6 复制文件	251
143-7 删除文件	251
143-8 创建文件夹	252
143-9 复制文件夹	252
143-10 移动文件夹	253
143-11 删除文件夹	254
143-12 导入文本文件	254
143-13 创建文本文件	256
143-14 取得驱动器信息	257

第 10 章 其他应用	260
范例 144 检查电脑名称	260
范例 145 定时关闭电脑	261
范例 146 保护 VBA 代码	262
146-1 设置工程密码	262
146-2 设置“工程不可查看”	263
范例 147 使用数字签名	264
范例 148 打开指定网页	267
范例 149 自定义“加载项”选项卡	268
范例 150 使用右键快捷菜单	271
150-1 使用右键快捷菜单添加菜单项	271
150-2 自定义右键快捷菜单	272
150-3 使用快捷菜单输入数据	274
150-4 禁用右键快捷菜单	275
范例 151 VBE 相关操作	276
151-1 添加模块和过程	276
151-2 建立事件过程	278
151-3 模块的导入与导出	279
151-4 删除 VBA 代码	280
范例 152 优化代码	282
152-1 关闭屏幕刷新	282
152-2 使用工作表函数	283
152-3 使用更快的 VBA 方法	284
152-4 使用 With 语句引用对象	285
152-5 简化代码	286
第 11 章 综合实例	288
范例 153 制作员工花名册	288
范例 154 制作收据打印系统	298
范例 155 制作员工考勤系统	325
附录 A Excel 2007 快捷键和功能键	353
附录 B Excel 2007 函数速查表	358

第 1 章 Range (单元格) 对象

范例 1 单元格的引用方法

在 VBA 中经常需要引用单元格区域，主要有以下几种方法。

1-1 使用 Range 属性引用单元格区域

使用 Range 属性以 A1 样式引用单元格区域，范例代码如下：

```
#001 Sub MyRng()  
#002     Range("A1:B4, D5:E8").Select  
#003     Range("A1").Formula = "=Rand()"  
#004     Range("A1:B4 B2:C6").Value = 10  
#005     Range("A1", "B4").Font.Italic = True  
#006 End Sub
```

【代码解析】 MyRng 过程使用 Range 属性以 A1 样式引用单元格区域。

应用于 Range 对象的 Range 属性代表一个单元格或单元格区域，语法如下：

表达式.Range(Cell1, Cell2)

参数 Cell1 是必需的，必须为采用宏语言的 A1 样式引用，可包括区域操作符（冒号）、相交区域操作符（空格）或合并区域操作符（逗号），也可包括货币符号（即绝对地址，如“\$A\$1”，但会被忽略掉），还可以使用局部定义名称，如 Range("MyRng")，其中 MyRng 为已定义的单元格区域名称。

参数 Cell2 是可选的，为区域左上角和右下角的单元格。可以是一个包含单个单元格、整行或整列的 Range 对象，或是一个局部定义名称。

第 2 行代码 使用 Select 方法选定工作表的 A1:B4、D5:E8 单元格区域。

第 3 行代码 在工作表的 A1 单元格中写入公式。


第 4 行代码 将工作表中 A1:B4 单元格与 B2:C6 单元格的相交区域 B2:B4 单元格赋值为 10。


第 5 行代码 将工作表中 A1:D4 单元格的字体样式设置为斜体。

【运行结果】 运行 MyRng 过程，结果如图 1-1 所示。

	A	B	C	D	E
1	0.025732				
2		10			
3		10			
4		10			
5					
6					
7					
8					
9					
10					

图 1-1 使用 Range 属性引用单元格区域

 **注意**：在没有使用对象识别符的情况下，Range 属性返回活动表的一个区域，如果活动表不是工作表，则该属性无效。

 **知识扩展**：在实际应用中，引用合并区域可以使用 Union 方法。引用相交区域可以使用 Intersect 方法。

1-2 使用 Cells 属性引用单元格区域

Cells 属性返回一个 Range 对象，代表指定单元格区域中的所有单元格。Cells 属性有一个 Item 属性，可以通过 Item 属性的行、列编号来引用单元格，范例代码如下：

```
#001 Sub MyCell()
#002     Dim i As Byte
#003     For i = 1 To 10
#004         Sheets("Sheet1").Cells(i, 1).Value = i
#005     Next
#006 End Sub
```

【代码解析】MyCell 过程在工作表的 A1:A10 单元格区域依次输入序号。

第 3 行代码 使用 For...Next 语句重复执行第 4 行代码 10 次。For...Next 语句是最常见的循环结构，在 For 和 Next 之间的语句将运行指定的次数。

第 4 行代码 使用 Cells 属性的 Item 属性。通过行、列编号来引用 A1~A10 单元格，其中行编号为变量，它的值为 For...Next 语句中计数器的值。

Cells 属性的 Item 属性的语法如下：


```
Cells.Item(Row, Column)
```

因为 Item 属性是 Range 对象的默认属性，所以可以在 Cells 关键字后面紧接着指定行和列编号，如 Cells(1, 1) 引用 A1 单元格。


	A	B	C	D
1	1			
2	2			
3	3			
4	4			
5	5			
6	6			
7	7			
8	8			
9	9			
10	10			
11				
12				

图 1-2 使用 Cells 属性引用单元格区域

在需要引用不确定的单元格区域时，Cells 属性的参数可以使用变量，它比使用 Range 属性更加方便。

 **注意**：在没有使用对象识别符的情况下，Cells 属性将返回一个 Range 对象，代表活动工作表中所有的单元格。

【运行结果】运行 MyCell 过程，结果如图 1-2 所示。

 **知识扩展**：用户可以将 Cells 属性作为 Range 属性的参数使用，下面的代码将引用 A1:B10 单元格区域：

```
Range(Cells(1, 1), Cells(10, 2))
```

1-3 使用快捷记号实现快速输入

在 VBA 中可以使用方括号作为 Range 属性的快捷方式，这样就不必输入单词“Range”

和使用引号, 范例代码如下:

```
#001 Sub FastMark()
#002     [A1] = "Excel 2007"
#003 End Sub
```

【代码解析】FastMark 过程使用快捷记号为 A1 单元格赋值。

第 2 行代码 使用快捷记号在工作表的 A1 单元格中输入字符串 “Excel 2007”。

【运行结果】运行 FastMark 过程, 结果如图 1-3 所示。

『注意』: 使用快捷记号引用单元格区域时, 只能使用固定字符串而不能使用变量。

『知识扩展』: 使用快捷记号同样可以利用 Range 属性的各种表示方法, 如表示交叉范围的[A1:B4 B2:C6]、合并区域的[A1:B4,B2:C6]等, 也可以使用局部定义名称, 下面的代码将工作簿中已命名为 “Rng” 的单元格区域赋值为 4。

```
Sheets("Sheet1").[Rng] = 4
```

	A	B	C	D
1	Excel 2007			
2				
3				
4				
5				
6				

图 1-3 使用快捷记号为单元格赋值

1-4 使用 Offset 属性返回单元格区域

在 VBA 中, 除了以上三种直接引用单元格的方法外, 还可以使用 Offset 属性来引用某个指定区域以外的单元格区域, 范例代码如下:

```
#001 Sub RngOffset()
#002     Sheets("Sheet1").Range("A1:B2").Offset(2, 2).Select
#003 End Sub
```

【代码解析】RngOffset 过程使用 Range 对象的 Offset 属性, 选中 A1:B2 单元格, 偏移两行两列的单元格区域。

应用于 Range 对象的 Offset 属性代表位于指定单元格区域的一定偏移量位置上的单元格区域, 语法如下:

```
表达式.Offset (RowOffset, ColumnOffset)
```

	A	B	C	D	E
1					
2					
3					
4					
5					
6					
7					
8					

图 1-4 使用 Offset 属性返回单元格区域

参数 RowOffset 是可选的, 表示区域偏移的行数, 正值表示向下偏移, 负值表示向上偏移, 默认值为 0。参数 ColumnOffset 是可选的, 表示区域偏移的列数, 正值表示向右偏移, 负值表示向左偏移, 默认值为 0。

【运行结果】运行 RngOffset 过程, 选中 A1:B2 单元格, 偏移两行两列的单元格区域, 如图 1-4 所示。

1-5 使用 Resize 属性返回调整后的单元格区域

用户还可以使用 `Resize` 属性调整指定单元格区域的大小并返回调整后的单元格区域，范例代码如下：

```
#001 Sub RngResize()
#002     Sheets("Sheet1").Range("A1").Resize(4, 4).Select
#003 End Sub
```

【代码解析】`RngResize` 过程使用 `Resize` 属性选中 A1 单元格扩展为四行四列后的单元格区域。

应用于 `Range` 对象的 `Resize` 属性调整指定区域的大小并返回调整后的区域，语法如下：

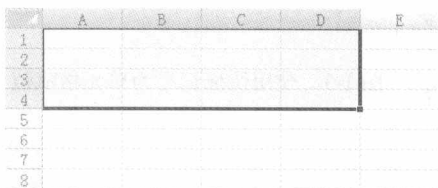


图 1-5 使用 `Resize` 属性调整单元格区域

表达式.`Resize`(`RowSize`, `ColumnSize`)

参数 `RowSize` 是可选的，为新区域中的行数，如果省略，则该区域中的行数保持不变。参数 `ColumnSize` 是可选的，表示新区域中的列数，如果省略，则该区域中的列数保持不变。

【运行结果】运行 `RngResize` 过程，选中 A1 单元格并扩展为四行四列后的区域，如 1-5 所示。

范例 2 选定单元格区域的方法

2-1 使用 Select 方法选定单元格区域

在 VBA 中一般使用 `Select` 方法选定单元格区域，范例代码如下：

```
#001 Sub RngSelect()
#002     Sheets("Sheet2").Activate
#003     Sheets("Sheet2").Range("A1:B10").Select
#004 End Sub
```

【代码解析】`RngSelect` 过程使用 `Select` 方法选定 Sheet2 工作表的 A1:B10 单元格区域。

第 2 行代码 利用 `Activate` 方法使 Sheet2 工作表成为活动工作表。使用 `Select` 方法选定单元格时，单元格所在的工作表必须为活动工作表，否则 `Select` 方法将发生运行错误，如图 1-6 所示。

第 3 行代码 使用 `Select` 方法选定 Sheet2 工作表的 A1:B10 单元格区域。

【运行结果】运行 `RngSelect` 过程，结果如图 1-7 所示。



图 1-6 `Select` 方法无效提示