

第 30 章 开发“Excel 百宝箱”

学习 VBA 通常就两个目的：解决临时问题和开发通用型插件解决一类问题。当然也可以利用 VBA 开发商业性质的插件。其中开发通用型插件可以将 VBA 发挥得淋漓尽致，也是部分读者学习 VBA 的最高目标。

本章将详述如何开发一个 Excel 百宝箱，读者可以从中学习大型插件的开发思路。

本章包括

- ◆ 程序规划
- ◆ 合并工具箱
- ◆ 图片工具箱
- ◆ 打印工具箱
- ◆ 不重复值工具箱
- ◆ 系统工具箱
- ◆ 开发函数
- ◆ 定制多级菜单并发布
- ◆ 财务工具箱
- ◆ 批注工具箱
- ◆ 安全工具箱
- ◆ 图表工具箱
- ◆ 文件工具箱
- ◆ 其他工具箱
- ◆ 定制百宝箱帮助

30.1 程序规划

对于大型插件必须先进行程序规划，然后再编代码，否则代码、模块会给人凌乱的感觉。

30.1.1 了解终端用户需求

如果要成为一个专业开发者，必须做的一件事是了解大众用户需求什么，以及 Excel 各版本都有哪些功能，其中无法完成用户需求的都体现在哪一方面。

可以这么说：开发插件就是弥补 Excel 功能的不足，或者对某个功能进行简化。开发者需要通过网络调查用户的日常需求都体现在哪一方面，有哪些需求是大众化需求，哪些需求是 Excel 内置功能无法完成的，以及虽然可以完成但效率却较差。有调查结果时，设计插件才会有的放矢。

例如，按照中国用户的习惯，将小写金额转为大写金额，将工资明细表生成工资条、设计底端标题、双面打印、批量导入图片等都是办公大众的日常需求，而 Excel 却没有对应的处理工具，那么程序员应该从这些方面去开发插件。

再如对工作表批量重命名，虽然 Excel 本身提供了命名工具，但有大量工作表需要命名时却显得效率低下，此时唯有借助插件完成。

本章所开发的 Excel 百宝箱包括 100 个功能，其中 80 个 Sub 过程，20 个 Function 过程。无

法确保所有功能都具有大众化需求，但大部分功能在工作中拥有较高的实用性。

30.1.2 确认插件功能表

Excel 百宝箱包括 100 个功能，按以下方式对其进行分类：

- ◆ 财务工具箱——包括若干财务制表相关工具
- ◆ 合并工具箱——包括单元格合并与拆分类的工具
- ◆ 批注工具箱——包括批量插入、修改、替换、查找批注的工具
- ◆ 图片工具箱——包括批量导入、导出图片的工具
- ◆ 安全工具箱——包括工作表与磁盘安全的若干工具
- ◆ 打印工具箱——包括与打印相关的若干工具
- ◆ 图表工具箱——包括批量操作图表标签与转换图表为图片的工具
- ◆ 唯一值工具箱——包括对重复值进行删除、标示以及产生唯一随机数等工具
- ◆ 文件工具箱——包括文件批量创建、命令、查找等工具
- ◆ 系统工具箱——包括查看磁盘信息、端口占用情况、开机时间、网卡与 CPU 序列号等工具
- ◆ 程序员工具箱——包括若干 VBA 程序员需要的工具
- ◆ 其他工具箱——包括以上工具以外的 10 多个工具
- ◆ 自定义函数——包括 20 个 Function 过程，提升 Excel 的运算能力



百宝箱的代码较长，本书不一一讲述每一个工具的设计过程。读者可以从随书光盘获取所有过程的源代码。

30.1.3 规划插件结构

由于插件包含 80 个 Sub 过程和 20 个 Function 过程，代码较多，为了让程序的启动速度不会太慢，必须进行合理地规划。

根据各功能的实现方式不同，将百宝箱分为两个文件。其中一个文件用于存放 80 个 Sub 过程，另一个文件用于存放 20 个自定义函数。可以有选择性地安装两者之一。

在 Sub 插件中根据需要设计以下 18 个窗体，如表 30-1 所示。

表 30-1 需要用到的窗体

合并工作簿	批量插入行	选区到图片
计算钞票张数	工作表批处理器	文件夹搜索器
报表拆分	批注管理	批量命名文件
复选框工具	图表标签修改	定位工具
拆分工作簿	图表标签位移	选择工具箱
批量加密码	图表到图片	磁盘安全

而根据各工具的功能类别，使用 15 个标准模块和一个类模块存放代码，如表 30-2 所示。

表 30-2 需要用到的模块

合并模块	菜单模块	图片模块
唯一值模块	其他模块	系统模块
财务模块	开发模块	文件模块
打印模块	批注模块	选择工具模块
安全模块	图表模块	SheetClass (类模块)



代码按类别存放在多个模块中并非必须的，只为方便阅读和后续维护。

限于篇幅，本章对财务工具箱、合并工具箱、批注工具箱和图片工具箱的相关工具进行详细讲解，其他工具不再一一罗列，读者可以从随书光盘中获取所有代码。

通过财务工具箱、合并工具箱、批注工具箱和图片工具箱的设计，读者完全可以了解设计一个大型插件的思路，对于其他工具，读者可以参考光盘代码，并逐步去摸索，其设计步骤与思路和财务工具箱、合并工具箱等大同小异。

30.2 财务工具箱

财务工具箱包括制作工资条、根据工资计算所需钞票张数、小写金额转大写、工作簿合并、工作表合并、工作表拆分、工作簿拆分、复选框工具、文本与数据转换等工具。本节分别讲述各工具的设计过程。

30.2.1 制作工资条头

案例要求：将工资明细表转换成工资条（注：工资条由标题行和每一个职工的工资明细组成，标题通常为 1 行、2 行或者 3 行。本例工具并非仅针对工资条，只要与工资条形式相近的需求都行，例如养老保险金打印等），且标题行可以由用户随意定义；工资条与工资条之间需要一个空行方便裁剪，且该行的高度也可以随意调整。

实现步骤：

step 1 插入一个标准模块，且将其命名为“财务模块”。

step 2 在模块中录入以下过程代码：

```
Sub 生成工资条()
    If ActiveSheet.ProtectContents Then MsgBox "工作表已保护,本程序拒绝执行!", 64,
    "Excel 百宝箱": Exit Sub
    On Error Resume Next
    Dim XRange As Range, YRange As Range, ZRange As Range, rownum As Integer, i As
    Integer, BtRow
    Star:
    BtRow = Application.InputBox("请指定标题行数!", "确定工资条头", 1, , , , 1)
```

```

If BtRow <= 0 Or BtRow > 10 Then MsgBox "只能 1 到 10 之间,请重新输入", 64, "Excel
百宝箱": GoTo Star
Set XRun = Cells(BtRow + 1, 2)
Application.StatusBar = "★★★★正在生成工资条,请稍候.....★★★★" '
在状态栏显示信息
Application.Calculation = xlManual '设置为手动计算
Application.ScreenUpdating = False '关闭屏幕更新
Do '开始循环
    XRun.Offset(1, 0).Rows("1:" & BtRow + 1).EntireRow.Insert Shift:=xlDown
    '插入行,行数为标题行加 1
    Set XRun = XRun.End(xlDown) '重置对象将 XRun 为其向下第一个非空单元格
    rownum = Range("B" & Rows.Count).End(xlUp).Row '记录数据区域的最大行号
号
Loop While XRun.Row < rownum '只要 XRun 的行号小于最大非空行号就一直循环下去
rownum = Range("B" & Rows.Count).End(xlUp).Row '记录数据区域的最大行号
Rows("1:" & BtRow).Copy '复制标题行
'合并所有用于存放标题的行
Set YRan = Rows(BtRow + 3 & ":" & 2 * BtRow + 2)
For i = 2 * (BtRow + 2) + 1 To rownum Step BtRow + 2
    Set YRan = Union(YRan, Rows(i & ":" & i + BtRow - 1))
Next
YRan.Select '选择区域
ActiveSheet.Paste '粘贴
'合并所有空行
Set ZRan = Rows(BtRow + 2)
For i = 2 * (BtRow + 2) To rownum Step BtRow + 2
    Set ZRan = Union(ZRan, Rows(i))
Next
Application.ScreenUpdating = True
'将空行的边框去掉
ZRan.Borders.LineStyle = xlNone
BtRow = Application.InputBox("请指定工资条之间空行的高度?", "间距", 20, , , ,
1)
If BtRow <= 0 Or BtRow > 409 Then MsgBox "行高必须在 0 至 409 之间,你输入的不
在此范围内,默认以 20 处理!", 64, "错误": BtRow = 20
ZRan.RowHeight = BtRow '将空白行设定高度
Cells(1, 1).Select '选择 A1 单元格
Application.Calculation = xlAutomatic '还原自动计算
Application.StatusBar = "" '原还状态栏
End Sub

```

以上过程的思路为:

- ◆ 首先确认工资条的标题行数,由用户指定。
- ◆ 用 Do…Loop 循环每隔一行插入空行,空行的数量等于标题行加 1,预留一个空行作为间隔行,方便用户裁剪。
- ◆ 利用 For 循环将第一行以外的所有空行合并为一个 Range 对象,并将标题行复制到空

行。

- ◆ 再次循环，选择作为间隔行的空行。并弹出输入框让用户指定间隔行的高度，默认高度为 20。程序会根据用户的输入值调整间隔行的高度，同时去除该间隔行的边框。

为了让用户可以还原工资明细表，需要再编写一个删除工资条标题的工具。代码如下：

```
Sub 恢复工资明细()
    Dim XRan As Range, rownum As Integer, i As Integer
Star:
    BtRow = Application.InputBox("请根据工资条头行数输入数据--只能是自然数!", "确认
    行数 ", 1, , , , 1)
    If BtRow <= 0 Or BtRow > 10 Then MsgBox "标题行数必须在 1 到 10 之间!", 64, "Excel
    百宝箱": GoTo Star
    Application.ScreenUpdating = False '关闭屏幕更新
    Application.Calculation = xlManual '设置为手动计算
    '合并所有空行和标题行
    Set XRan = Rows(BtRow + 2 & ":" & 2 * BtRow + 2)
    rownum = Range("B" & Rows.Count).End(xlUp).Row '以 B 列为标准,B 列的工资信
    息不能空白 (如果有可以替换为一个空格)
    For i = 2 * (BtRow + 2) To rownum Step BtRow + 2
        Set XRan = Union(XRan, Range("B" & i).Rows("1:" & BtRow + 1).EntireRow)
    Next
    XRan.Delete Shift:=xlUp '删除标题行
    Application.ScreenUpdating = True '恢复屏幕更新
    Application.Calculation = xlAutomatic '恢复自动计算
End Sub
```

以上过程的思路如下：

- ◆ 首先确认工资条的标题行数，由用户指定。
- ◆ 利用 For 循环合并第一个标题行以外的标题行与间隔行，然后整行删除。

功能测试：

step 1 假设工作表中有如图 30.1 所示工资明细表，执行过程“生成工资条”将弹出如图 30.2 所示对话框，默认值为 1，表示工资条的标题行数为 1 行。

1	个人信息	基本	奖惩		实发		
2	姓名	工号	工资	加班工资	奖金	扣款	工资
3	赵	1245	1200	270	250	25	1695
4	钱	3245	1200	240	100		1540
5	孙	2210	1200	240	250		1690
6	李	1000	1200	100	250	25	1525
7	周	1059	1200	140	100		1440
8	吴	3215	1200	100	250	100	1450

图 30.1 工资明细表

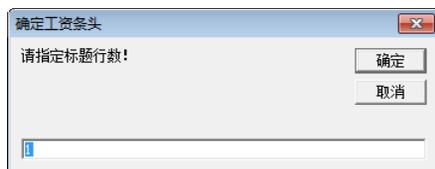


图 30.2 指定工资条的标题行数

step 2 在对话框中输入 2 并按回车键,程序会自动完成所有职员的工资条设计。然后弹出图 30.3 所示对话框,用于调整工资条之间的间隔行高度。该默认值为 20,通常使用 10 即可。

step 3 最后完成的效果如图 30.4 所示。如果执行过程“恢复工资明细”,将会询问工资条行数,输入 2 并回车可以还原为图 30.1 所示明细表。



图 30.3 指定间隔行的行高

1	个人信息	基本	奖金			实发
2	姓名	工号	工资	加班工资	扣款	工资
3	赵	1245	1200	270	25	1695
4						
5	个人信息	基本	奖金			实发
6	姓名	工号	工资	加班工资	扣款	工资
7	钱	3245	1200	240	100	1540
8						
9	个人信息	基本	奖金			实发

图 30.4 工资条

注意事项:

(1) 工作表保护状态下无法插入行,所以在代码中需要判断工作表是否受密码保护。

(2) 本工具不采用 Usedrange.rows.count 计算最后一行的行号是因为 Usedrange.rows.count 本身有限制,即 Usedrange.会将该空白行计算在内,那么使用工资条工具后将产生多余的行,而采用 End(xlUp).row 方法计算最后一个非空行的行号会更准确。

(3) 工资条以 B 列为基准计算最后非空行的行号,那么 B 列的已用区域中不能有空白单元格。

(4) 工资表中一定会有很多计算公式,在插入行时会导致这些公式重算,将延长工具的执行时间。为了解决这个 BUG,必须将计算模式调整为手动,执行完毕后再还原为自动。

30.2.2 根据工资计算所需钞票张数

案例要求: 部分工厂、企业在发放职工工资时采用现金发放,而这涉及一个零钱计算的问题,即每个职工工资需要多少张 100 元钞票、多少张 10 元钞票或者多少张 1 元钞票。利用 VBA 可以瞬间完成钞票张数的计算,而且钞票面值可以随意定义。

实现步骤:

step 1 插入一个窗体,将其名称修改为“计算钞票张数”。

step 2 在窗体中插入两个框架,在第一个框中添加一个复选框,并将其“Caption”属性修改为“100 元”。

step 3 将复选框复制 6 份,分别将其“Caption”属性修改为“50 元”、“20 元”、“10 元”、“5 元”、“2 元”、“1 元”,但名称属性保持不变。

step 4 在第二个框架中添加三个命令按钮,且将“Caption”属性分别修改为“全选”、“反选”和“计算钞票”。编辑完成后其控件布局如图 30.5 所示。

step 5 双击窗体进入窗体事件代码窗口,并输入以下 3 段代码:

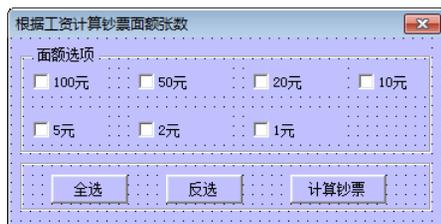


图 30.5 窗体控件布局

```
Private Sub CommandButton1_Click()
    Dim i As Byte
    For i = 1 To 7 '将所有复选框勾选
        Me.Controls("CheckBox" & i).Value = True
    Next i
End Sub
```

以上代码表示单击“全选”按钮时，将所有复选框打勾。

```
Private Sub CommandButton2_Click()
    Dim i As Byte
    For i = 1 To 7 '反选,将其勾选状态设置相反的状态
        Me.Controls("CheckBox" & i).Value = Not Me.Controls("CheckBox" & i)
    Next i
End Sub
```

以上代码表示单击“反选”按钮时，将每个复选框勾选状态设置为相反的状态。

```
Private Sub CommandButton3_Click()
    Dim rngg As Range, a As Long, b As Long, c As Long, d As Long, e As Long, f
    As Long, g As Long
    Dim cell As Range, rng As Range, i, j, add As String
    For i = 1 To 7 '循环检查每个复选框,如果7个都没有选中则退出程序
        j = j + Me.Controls("CheckBox" & i).Value
    Next i
    If j = 0 Then Exit Sub
    i = 0
    Me.Hide '隐藏窗体
    If TypeName(Selection) = "Range" Then add = Selection.Address Else add =
    "B2:B10" '如果选择的是单元格,则记录地址
    '记录选择的地址
    Set rngg = Application.InputBox("请输入待计算的区域地址,也可以手动选择区域。" &
    Chr(10) & "但不能从首行开始,切记!", "工资区域", add, 10, 10, , , 8)
    Set rngg = Range(rngg(1).Offset(-(rngg.Row = 1), 0), Cells(Rows.Count,
    rngg.Column).End(xlUp)) '作用是:1.防止选择多列时出错和选择整列时浪费时间
    Application.ScreenUpdating = False
    For Each cell In rngg '遍历选区
        If Not VBA.IsNumeric(cell) Then MsgBox "不能处理非数字区:" & cell.Address:
        GoTo Star
        Set rng = rngg.Item(1).Offset(-1, 0) '在数据区上方一行需要写入标题,例如 100 元
        /1 元等
        a = Int((cell.Value / 100) * -(CheckBox1.Value)) '100 元面值
        b = Int((cell.Value - a * 100) / 50 * -(CheckBox2.Value)) '50 元面值
        c = Int((cell.Value - a * 100 - b * 50) / 20 * -(CheckBox3.Value))
        d = Int((cell.Value - a * 100 - b * 50 - c * 20) / 10 * -(CheckBox4.Value))
        e = Int((cell.Value - a * 100 - b * 50 - c * 20 - d * 10) / 5 * -(CheckBox5.Value))
        f = Int((cell.Value - a * 100 - b * 50 - c * 20 - d * 10 - e * 5) / 2 *
        -(CheckBox6.Value))
        g = Int((cell.Value - a * 100 - b * 50 - c * 20 - d * 10 - e * 5 - f * 2)
```

```

* -(CheckBox7.Value)
  If CheckBox1.Value Then rng.Offset(0, 1) = "100元": cell.Offset(0, 1) = a
'如果勾选了100元就计算100元钞票张数
  If CheckBox2.Value Then rng.Offset(0, 2 - CheckBox1.Value - 1) = "50元":
cell.Offset(0, 2 - CheckBox1.Value - 1) = b
  If CheckBox3.Value Then rng.Offset(0, 3 - CheckBox1.Value - CheckBox2.Value
- 2) = "20元": cell.Offset(0, 3 - CheckBox1.Value - CheckBox2.Value - 2) = c
  If CheckBox4.Value Then rng.Offset(0, 4 - CheckBox1.Value - CheckBox2.Value
- CheckBox3.Value - 3) = "10元": cell.Offset(0, 4 - CheckBox1.Value -
CheckBox2.Value - CheckBox3.Value - 3) = d
  If CheckBox5.Value Then rng.Offset(0, 5 - CheckBox1.Value - CheckBox2.Value
- CheckBox3.Value - CheckBox4.Value - 4) = "5元": cell.Offset(0, 5 - CheckBox1.
Value - CheckBox2.Value - CheckBox3.Value - CheckBox4.Value - 4) = e
  If CheckBox6.Value Then rng.Offset(0, 6 - CheckBox1.Value - CheckBox2.Value
- CheckBox3.Value - CheckBox4.Value - CheckBox5.Value - 5) = "2元":
cell.Offset(0, 6 - CheckBox1.Value - CheckBox2.Value - CheckBox3.Value -
CheckBox4.Value - CheckBox5.Value - 5) = f
  If CheckBox7.Value Then rng.Offset(0, 7 - CheckBox1.Value - CheckBox2.Value
- CheckBox3.Value - CheckBox4.Value - CheckBox5.Value - CheckBox6.Value - 6)
= "1元": cell.Offset(0, 7 - CheckBox1.Value - CheckBox2.Value - CheckBox3.Value
- CheckBox4.Value - CheckBox5.Value - CheckBox6.Value - 6) = g
Star:
  i = i + 1
Next cell
j = rngg.End(xlToRight).Column '计算最后一列列号
Range(Cells(rngg.Row + i, rngg.Column + 1), Cells(rngg.Row + i,
j)).FormulaR1C1 = "=SUM(R[-" & i & "]C:R[" & -1 & "]C)" '设置公式计算合计
Range(Cells(rngg.Row + i, rngg.Column + 1), Cells(rngg.Row + i,
j)).EntireColumn.AutoFit '设置自动适应列宽
Range(Cells(rngg.Row - 1, rngg.Column + 1), Cells(rngg.Row + i,
j)).Borders.LineStyle = xlContinuous '添加边框
Application.ScreenUpdating = True
MsgBox Chr(13) + " ( ) ~ ^ " + Chr(10) + " Home ~ " +
Chr(10) + "|田|田田|_____ " + Chr(10) + "
" + Chr(13) + "_____ " + Chr(13) +
Chr(13) + "计算完毕! 最末行是钞票的合计张数!", 64, "Excel 百宝箱"
Unload Me '关闭窗体
End Sub

```

以上过程的思路为:

- ◆ 首先,判断是否所有复选框都没有选中,如果是则退出程序。
- ◆ 再判断当前选区是否为单元格,如果非单元格则将变量“add”赋值为“B2:B10”,否则赋值为选区地址。
- ◆ 然后,弹出对话框让用户选择待计算区域,默认区域为变量“add”的值,用户要从工作表中任意选择一列作为计算区域,如果选择多列,程序会将其重置为一列。
- ◆ 利用 For 循环分别计算 7 种面值的钞票各需要多少张。对于非数值单元格则跳过。如果

某面值的复选框已选中，那么在右上方显示该面值，同时在对应的单元格显示该面值的钞票张数。

- ◆ 最后利用 FormulaR1C1 对以上所有面值的钞票设置合计的公式。

step 6 在“财务模块”中输入以下过程调用窗体：

```
Sub 计算钞票 ()
    计算钞票张数.Show 0
End Sub
```

功能测试：

step 1 选择工资数据，然后执行过程“计算钞票”，将弹出窗体。

step 2 单击“全选”按钮，勾选所有面值的钞票，窗体中所有复选框将打勾，如图 30.6 所示。



图 30.6 全选所有代表钞票面额的复选框

step 3 单击按钮“计算钞票”，将弹出“工资区域”的对话框，可以保持默认值“\$B\$2:\$B\$7\$”，也可以重新选择 B 列，程序会自动将所选择的 B 列转换成有效区域“\$B\$2:\$B\$7\$”。执行完毕后，工资计算的结果如图 30.7 所示，其中最后一行是合计值。

step 4 再次运行窗体，仅仅选择 50 元和 2 元，那么执行完毕后工作表中仅仅计算所需 50 元和 2 元的钞票张数。

	A	B	C	D	E	F	G	H	I
1	姓名	工资	100元	50元	20元	10元	5元	2元	1元
2	赵	2160	21	1	0	1	0	0	0
3	钱	2268	22	1	0	1	1	1	1
4	孙	1646	16	0	2	0	1	0	1
5	李	1758	17	1	0	0	1	1	1
6	周	2470	24	1	1	0	0	0	0
7	吴	2020	20	0	1	0	0	0	0
8			120	4	4	2	3	2	3

图 30.7 计算结果

注意事项：

(1) 本工具仅仅对单列进行计算，所以如果用户选择了多列，在程序中需要使用 Resize 来重置区域为 1 列，而且去除选区下方的空白区域（如果有空白区域的话）。

(2) 表中的 Int 表示取整的函数，例如 2160 除以 100 等于 21.6，那么结果取 21。

(3) 为了美观，需要对钞票张数存放区域调整为自动列宽，并添加边框。

30.2.3 小写金额转大写

案例要求：作为财务人员，在工作中经常需要将小写的人民币金额转换成大写。本工具可以批量地将小写金额转换为大写形式。

实现步骤：

在“财务模块”中录入以下代码：

```
Sub 小写转大写()  
    If ActiveSheet.ProtectContents Then MsgBox "工作表已保护,本程序拒绝执行!", 64,  
    "Excel 百宝箱": Exit Sub  
    Dim RMBS As String, rng As Range, cell As Range  
    If Selection(1) = "" Or Not VBA.IsNumeric(Selection(1)) Then MsgBox "不要选  
    择非数字区及空白区.", 64, "Excel 百宝箱": Exit Sub  
    Application.ScreenUpdating = False '关闭屏幕更新  
    Application.Calculation = xlManual '设置为手动计算  
    For Each cell In Selection  
        If Len(cell) = 0 Or Not VBA.IsNumeric(cell) Then GoTo nextt '跳过非数值单  
        元格和空单元格  
        If cell = 0 Then cell = "零元整": GoTo nextt  
        RMBS = Replace(Replace(Application.Text(Round(cell, 2)), "[DBnum2]"), ".",  
        "元"), "-", "负")  
        RMBS = IIf(Left(Right(RMBS, 3), 1) = "元", Left(RMBS, Len(RMBS) - 1) & "  
        角" & Right(RMBS, 1) & "分", IIf(Left(Right(RMBS, 2), 1) = "元", RMBS & "角",  
        IIf(RMBS = "零", "", RMBS & "元整")))  
        RMBS = Replace(Replace(RMBS, "零元", ""), "零角", "")  
        cell = RMBS  
    nextt:  
    Next cell  
    Application.ScreenUpdating = True '恢复屏幕更新  
    Application.Calculation = xlAutomatic '恢复自动计算  
    Selection.EntireColumn.AutoFit  
End Sub
```

以上过程的思路为：

- ◆ 如果选区中第一个单元格为空白或者非数值，或者工作表保护则退出程序。
- ◆ 利用 For 循环逐个转换单元格的值，如果中间有非数值或者空白则跳过。
- ◆ 当遇到零值时，直接转换成“零元整”，否则利用工作表函数 Text 将数值转换成大写形式，再利用 Replace 将“.”、“-”和“零角”等不规范的字符替换成规范化字符。

功能测试：

step 1 打开图 30.8 所示收支表。

step 2 选择金额区域 B2:B7，执行过程“小写转大写”，结果如图 30.9 所示。

注意事项:

(1) 空白单元格不需要转换, 但仅仅利用“Not IsNumeric(cell)”进行判断是不行的。因为 IsNumeric 函数会将空值当作 0 处理, 所以必须加入“Len(cell) = 0”语句。

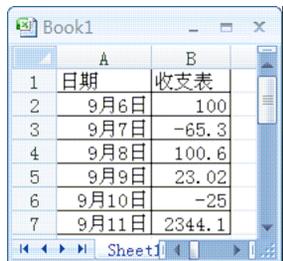


图 30.8 收支表



图 30.9 小写金额转大写

(2) 批量转换时会引起公式重算, 所以必须将计算模式调为手动, 同时关闭屏幕更新从而提高速度。

30.2.4 工作簿与工作表合并

案例要求: 如果有多个格式完全相同的工作簿的数据需要汇总或者进行数据分析, 通常需要将多工作簿的工作表合并到一起, 包括合并为一个工作表或者合并到一个工作簿中, 但以多个工作表形式保存。同理, 一个工作簿中的多工作表数据也可以合并到一个工作表中。

实现步骤:

- step 1** 插入一个窗体, 将其名称修改为“合并工作簿”。
- step 2** 在窗体中添加一个多页控件; 将第一页和第二页的名称分别修改为“多工作簿合并”和“多工作表合并”。
- step 3** 在第一页根据需要在添加六个命令按钮、一个文字框、四个标签、两个单选按钮和一个复选框, 且按图 30.10 方式布局。

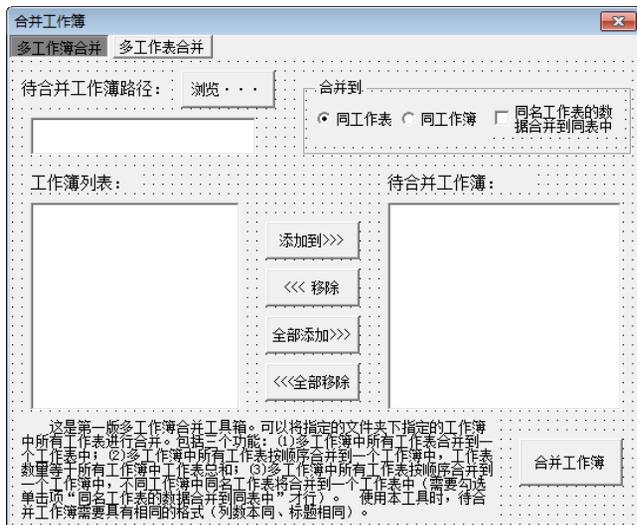


图 30.10 多工作簿合并窗口的控件布局

step 4 双击窗体进入代码窗口，并在窗口中录入以下代码：

```
Dim i As Byte, k As Byte, j As Integer, l As Integer, ans As Byte
Private Sub UserForm_Activate()
'打开窗体时，禁用五个按钮及复选框
    CommandButton2.Enabled = False
    CommandButton3.Enabled = False
    CommandButton4.Enabled = False
    CommandButton5.Enabled = False
    CommandButton6.Enabled = False
    CheckBox1.Enabled = False
End Sub
```

以上过程表示窗体激活时禁用“浏览”以外的五个按钮，其目的是避免列表框中空白而执行其他按钮时产生的错误，如果采用在五个按钮的单击过程中逐个加入判断语句来防错会显得更烦琐，本例的思路是最简洁的方式。

```
Private Sub OptionButton2_Click()
    CheckBox1.Enabled = True '允许使用复选框
    CheckBox1.Value = True '勾选复选框
    MsgBox "勾选右边的复选框则将不同工作簿中同名工表进行合并；" & Chr(10) & "否则不同工作簿中同名工作表将存放在合并工作簿的不同工作表中，以工作簿名加工作表名来命名工作表。"
End Sub
```

以上过程表示选择第一个单选框时禁止使用复选框，选择第二个单选框才允许使用复选框。

```
Private Sub CommandButton1_Click()
    Dim file() As String, str As String, n As Byte
    Set fd = Application.FileDialog(msoFileDialogFolderPicker)
    With fd '将选择文件夹的地址赋予文字框中
        If .Show = -1 Then TextBox1.Text = .SelectedItems(1) Else Exit Sub
    End With
    str = Dir(TextBox1.Text & "\*.xl*") '搜索所有 Excel 文件
    While Len(str) > 0 '如果文件存在
        n = n + 1
        ReDim Preserve file(1 To n)
        file(n) = str '将文件名存入数组中
        str = Dir()
    Wend
    If n = 0 Then MsgBox "未发现 excel 文件哦! ": Exit Sub
    ListBox1.List = WorksheetFunction.Transpose(file) '将数组中的值赋予列表框
    ListBox2.List = WorksheetFunction.Transpose(file)
    CommandButton2.Enabled = True
    CommandButton3.Enabled = True
    CommandButton4.Enabled = True
    CommandButton5.Enabled = True
    CommandButton6.Enabled = True
End Sub
```

以上过程表示单击“浏览”按钮时弹出对话框，让用户选择一个文件夹，然后将该文件夹中所有 Excel 文件名导入到两个列表框中，再启用五个按钮。其中左边列表框用于存放文件夹中的所有 Excel 文件名称，右边列表框中则用于存放实际参与合并的文件，可以随意增减。

```
Private Sub CommandButton2_Click()
    For i = 0 To ListBox1.ListCount - 1
        If ListBox1.Selected(i) Then ListBox2.AddItem ListBox1.List(i) '将选
        择的项目添加到列表框 2
    Next i
End Sub
Private Sub CommandButton3_Click()
    If ListBox2.ListCount > 0 Then
        For i = 0 To ListBox2.ListCount - 1
            If ListBox2.Selected(i) Then ListBox2.RemoveItem (i) '删除选择项目
        Next i
    End If
End Sub
Private Sub CommandButton4_Click()
    ListBox2.List = ListBox1.List
End Sub
Private Sub CommandButton5_Click()
    ListBox2.Clear
End Sub
```

前两个过程表示添加和删除实际参与汇总的工作簿名，后两个过程表示全部删除第二个列表框项目和导入第一个列表框所有内容到第二个列表框中。

```
Private Sub CommandButton6_Click()
    Dim Paths As String, nm As String, sh As Worksheet, temp As Byte, k As Byte
    Dim ans As Byte, tm As String, bool As Byte, newbook As String
    On Error Resume Next
    With Workbooks.add '添加一个工作簿
        Application.DisplayAlerts = False
        .Sheets(3).Delete '只需要一个工作表,删除多余的两个
        .Sheets(2).Delete
        .Sheets(1).Name = "目录"
        Application.DisplayAlerts = True
        nm = ActiveWorkbook.Name '提取当前工作簿名
        If OptionButton1 Then GoTo sht '如果选择合并到同工作表,则运行标签"sht"的语句
        If OptionButton2 And CheckBox1.Value = False Then '如果选择合并到同工作簿,
        且"同名工作表的数据合并到同表中"未选中
            Application.ScreenUpdating = False
            For k = 0 To ListBox2.ListCount - 1 '遍历列表框 2 中所有项目
                Paths = IIf(Right(TextBox1.Text, 1) = "\", TextBox1.Text, TextBox1.Text
                & "\") & ListBox2.List(k)
                namess = Dir(Paths)
                Workbooks.Open Filename:=Paths '打开工作簿
            For i = 1 To Sheets.Count
                Workbooks(namess).Activate
            End For
        End If
    End With
    GoTo sht
End Sub
```

```
'如果工作表不是空白则将其所有数据复制到当前工作簿的最后一个工作表中,并对当前工
作簿最后一个工作表命名,以工作簿名连接冒号再连接工作表名对工作表进行命名
If Not IsEmpty(Sheets(i).UsedRange) Then Sheets(i).Copy
After:=Workbooks(nm).Sheets(Workbooks(nm).Sheets.Count)
Workbooks(nm).Sheets(Workbooks(nm).Sheets.Count).Name = Left(namess,
Len(namess) - VBA.InStr(3, StrReverse(namess), ".")) & " : " &
Workbooks(namess).Sheets(i).Name
Next i
Workbooks(namess).Close False '关闭工作簿且不保存
Next k
l = 1
.Sheets("目录").Range("a1:b1") = Array("序号", "工作表") '添加目录
For Each sh In Worksheets
If sh.Name <> "目录" Then
sh.UsedRange.Borders.LineStyle = xlContinuous
tm = tm & sh.Name & Chr(10)
.Sheets("目录").Cells(l, 1) = l - 1
.Sheets("目录").Cells(l, 2).Value = sh.Name
End If
l = l + 1
Next sh
.Sheets("目录").Select '定位于目录工作表
Application.ScreenUpdating = True
MsgBox "本文件夹的工作簿已汇总到本工作簿!" & Chr(10) & "汇总后的工作表名分别为:"
& Chr(10) & tm, vbOKOnly + 64, "合并完成"
Else
'确定标题行数,该数据不参与合并 ans = Application.InputBox("请确认待合并工作簿的
标题行数,该行将产生在合并工作簿中作为新的标题行:", "标题行", 1, , , , 1)
If ans < 1 Then Exit Sub
For k = 0 To ListBox2.ListCount - 1
Paths = IIf(Right(TextBox1.Text, 1) = "\", TextBox1.Text, TextBox1.Text
& "\") & ListBox2.List(k)
namess = Dir(Paths)
Workbooks.Open Filename:=Paths '逐个打开工作簿
For i = 1 To Workbooks(namess).Sheets.Count '遍历当前工作簿中所有工作表
bool = 0
shn = Workbooks(namess).Sheets(i).Name
For j = 1 To Workbooks(nm).Sheets.Count
If Workbooks(nm).Sheets(j).Name = shn Then bool = 1: Exit For
'判断是否有同名工作表
Next j
If bool = 0 Then
Workbooks(nm).Sheets.add
After:=Workbooks(nm).Sheets(Workbooks(nm).Sheets.Count) '添加工作表
Workbooks(nm).ActiveSheet.Name = Workbooks(namess).Sheets(i).Name
'对新工作表命名,等于数据源工作表的名称
Workbooks(namess).Sheets(i).Rows("1:" & ans).Copy
Workbooks(nm).ActiveSheet.Cells(1, 1) '将标题复制到该表中
End If
```

```

        If Not IsEmpty(Workbooks(namess).Sheets(i).UsedRange) Then ' 如果有
数据
        If Workbooks(namess).Sheets(i).UsedRange.Rows.Count <= ans Then
Exit For '如果数据行小于设置的标题行数则退出循环
        '复制标题以外的数据区

Application.Intersect(Workbooks(namess).Sheets(i).UsedRange.Offset(ans, 0),
Workbooks(namess).Sheets(i).UsedRange).Copy
Workbooks(nm).Sheets(shn).Cells(65536, 1).End(xlUp).Offset(1, 0)
        End If
        ActiveSheet.UsedRange.Borders.LineStyle = xlContinuous '添加边框
    Next i
    Workbooks(namess).Close False '关闭工作表且不保存
Next k
l = 1
With Workbooks(nm).Sheets("目录") '建立目录
    .Range("a1:b1") = Array("序号", "工作表")
    For Each sh In Workbooks(nm).Worksheets
        If sh.Name <> "目录" Then
            sh.UsedRange.Borders.LineStyle = xlContinuous '对非目录工作表的
数据区域添加边框
            tm = tm & sh.Name & Chr(10)
            .Cells(l, 1) = l - 1
            .Cells(l, 2).Value = sh.Name
        End If
        l = l + 1
    Next sh
    .Select
End With
Application.ScreenUpdating = True
MsgBox "本文件夹的工作簿已汇总到本工作簿!" & Chr(10) & "汇总后的工作表名分别为:"
& Chr(10) & tm, vbOKOnly + 64, "汇总完成"
End If
Exit Sub
sht:
ans = Application.InputBox("请确认待合并工作簿的标题行数,该行将产生在合并工作簿中
作为新的标题行:", "标题行", 1, , , , 1)
If ans < 1 Then Exit Sub
For k = 0 To ListBox2.ListCount - 1
    Paths = IIf(Right(TextBox1.Text, 1) = "\", TextBox1.Text, TextBox1.Text
& "\") & ListBox2.List(k)
    namess = Dir(Paths)
    Workbooks.Open Filename:=Paths
    .Activate
    Workbooks(namess).Sheets(1).Rows("1:" & ans).Copy Cells(1, 1) '复制标题行
    For i = 1 To Workbooks(namess).Sheets.Count
        If Not IsEmpty(Workbooks(namess).Sheets(i).UsedRange) Then
            If Workbooks(namess).Sheets(i).UsedRange.Rows.Count <= ans Then Exit
For '如果已用数据行数小于或等于标题行则退出循环

```

```
'开始复制数据标题行以外的数据

Application.Intersect(Workbooks(names).Sheets(i).UsedRange.Offset(ans, 0),
Workbooks(names).Sheets(i).UsedRange).Copy Cells(.Sheets(" 目 录
").UsedRange.Rows.Count + 1, 1)
    End If
Next i
Workbooks(names).Close False
Next k
End With
Application.ScreenUpdating = True
End Sub
```

以上过程的思路为：

- ◆ 首先，新建一个工作簿，且删除两个多余的工作表，仅保留一个工作表。如果是合并多工作簿到同工作表名，那么此工作表用于存放合并后的数据；如果是合并多工作簿到同工作簿中，则此工作表用于存放合并后工作表的目录。
- ◆ 利用 IF 判断单选按钮的状态，如果选择“同工作表”则弹出对话框，让用户确认待合并工作簿的工作表标题的行数，再逐个打开工作簿，先将其标题行复制到当前工作表第一行中，再复制标题以外的已用区域到第一个非空行，直到循环完毕所有选择的工作簿。
- ◆ 如果用户选择“同工作簿”，那么区分两种情况，即是否勾选“同名工作表的数据合并到同表中”。如果未勾选该复框，那么逐个打开列表框中的工作簿，并在合并工作簿中对应创建新工作表，直接将目标工作簿中每个工作表的数据复制到合并工作表中，且将合并工作簿的工作表命名为源数据工作簿名加冒号再加工作表名，从而区分合并工作表与源工作簿的关系。最后对合并工作簿创建目录；如果将复选框打勾，那么同样首先确认标题行数，然后逐个打开工作簿，将复制标题行和非标题行。不同的是如果待合并工作表与当前工作簿中某个工作表重命名时，不再新建工作表，而是将它标题行以外的数据追加到同名工作表之后。最后再创建工作表目录。

step 5 进入第二页，添加四个标签、六个按钮、一个文字框和两个单选按钮，且按图 30.11 所示布局。

step 6 双击“浏览…”按钮打开代码窗口，并输入以下代码：

```
Private Sub MultiPage1_Change() '切换多页控制时执行
    Dim sht As Worksheet
    TextBox2 = ActiveWorkbook.FullName '显示工作簿路径
    ListBox3.Clear
    For Each sht In ActiveWorkbook.Sheets
        ListBox3.AddItem sht.Name '将工作表名添加到列表框
    Next sht
    OptionButton4.Enabled = False '禁用单选框
End Sub
```

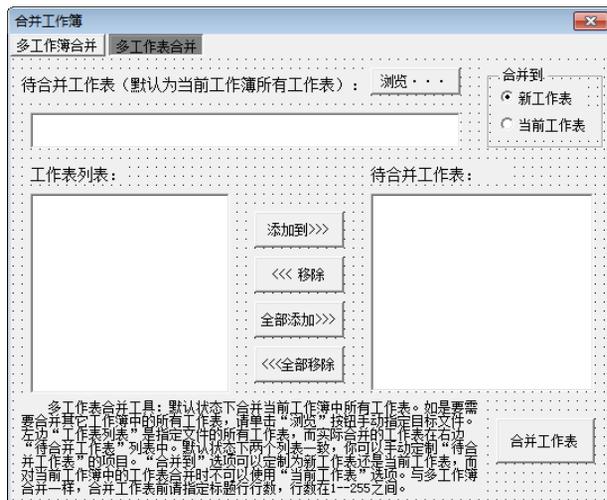


图 30.11 多工作表合并窗口的控件布局

以上过程表示进入多页控件的第二页时在文字框显示当前工作簿名,且将所有工作表罗列在列表框中,同时禁用单选框“OptionButton4”,表示只能合并到新表中。

```
Private Sub CommandButton7_Click()
    Dim sht As Worksheet, wkbook
    Set wkbook = ActiveWorkbook '将当前工作簿赋予对象变量
    With Application.FileDialog(msoFileDialogFilePicker) '选择工作簿
        .AllowMultiSelect = False '不能多选
    End With
    If .Show = -1 Then
        TextBox2 = .SelectedItems(1)
        Workbooks.Open (.SelectedItems(1)) '打开工作簿
        ListBox3.Clear '清空两个列表框
        ListBox4.Clear
        For Each sht In ActiveWorkbook.Sheets '将工作簿中所有工作表添加到两列表框中
            ListBox3.AddItem sht.Name '将工作表名添加到列表框
            ListBox4.AddItem sht.Name
        Next sht
        OptionButton4.Enabled = True
        wkbook.Activate '返回原工作簿
    End If
End Sub
```

以上过程的思路为:

- ◆ 单击“浏览…”按钮时弹出对话框让用户选择待合并的工作簿。
- ◆ 将工作簿路径显示在文字框中,同时打开该工作簿,将所有工作表名添加到两个列表框。
- ◆ 最后让“当前工作表”单选按钮可用,且返回原来的工作簿。

```
Private Sub CommandButton8_Click()
    For i = 0 To ListBox3.ListCount - 1
        If ListBox3.Selected(i) Then ListBox4.AddItem ListBox3.List(i) '添加选择
```

内容

```
Next i
End Sub
Private Sub CommandButton9_Click()
    If ListBox4.ListCount = 0 Then Exit Sub
    If ListBox4.ListCount > 0 Then
        For i = 0 To ListBox4.ListCount - 1
            If ListBox4.Selected(i) Then ListBox4.RemoveItem (i) '移除选择内容
        Next i
    End If
End Sub
Private Sub CommandButton10_Click()
    ListBox4.List = ListBox3.List '复制列表框数据
End Sub
Private Sub CommandButton11_Click()
    ListBox4.Clear '清除列表框
End Sub
Private Sub MultiPage1_Change() '切换多页控制时执行
    Dim sht As Worksheet
    TextBox2 = ActiveWorkbook.FullName '显示工作簿路径
    ListBox3.Clear
    For Each sht In ActiveWorkbook.Sheets
        ListBox3.AddItem sht.Name '将工作表名添加到列表框
    Next sht
    OptionButton4.Enabled = False
End Sub
```

前四个过程表示添加、删除列表框中的待合并工作表，第五个过程表示切换多页控件时执行对文本框和列表框赋值。

```
Private Sub CommandButton12_Click() '多工作表合并
    If VBA.InStr(1, TextBox2.Text, "\") = 0 Then MsgBox "请先保存当前工作簿再执行合并! ": GoTo err
    If ListBox4.ListCount = 0 Then Exit Sub
    ans = Application.InputBox("请确认待合并工作簿的标题行数,该行将产生在合并工作簿中作为新的标题行:", "标题行", 1, , , , 1)
    Application.ScreenUpdating = False
    If ans < 1 Then Exit Sub
    With Workbooks(Split(TextBox2.Text, "\")(UBound(Split(TextBox2.Text, "\"))))
        '以"\\"为分隔条件取出路径中的文件名,然后对该工作簿进行操作
        If .Sheets(1).UsedRange.Rows.Count <= ans Then MsgBox "指定的行标题数目过大! ": Exit Sub '如果已用数据区域行数小于标题行则退出"
        If OptionButton3 Then '如果选择"新工作表"
            With Worksheets.add '添加工作表
                On Error Resume Next
                .Name = "合并工作表" '命名并移动到最后一
                .Move After:=Sheets(Sheets.Count)
            End With
            Workbooks(Split(TextBox2.Text, "\")(UBound(Split(TextBox2.Text, "\")))).Sheets(1).Rows("1:" & ans).Copy .Cells(1, 1) '复制标题数据
        End If
    End With
End Sub
```

```

End With
For l = 0 To ListBox4.ListCount - 1
    If Not IsEmpty(.Sheets(ListBox4.List(l)).UsedRange) Then '如果有数据则
复制所有标题以外的数据
Application.Intersect(.Sheets(ListBox4.List(l)).UsedRange.Offset(ans,
0), .Sheets(ListBox4.List(l)).UsedRange).Copy Sheets(Sheets.Count).Cells
(65536, 1).End(xlUp).Offset(1, 0)
    End If
Next l
If OptionButton4.Enabled = False Then GoTo err
.Close , False '关闭工作簿且不保存
Else '否则
.Sheets(1).Rows("1:" & ans).Copy Cells(1, 1) '复制标题
For l = 0 To ListBox4.ListCount - 1
'复制标题以外的数据
Application.Intersect(.Sheets(ListBox4.List(l)).UsedRange.Offset(ans,
0), .Sheets(ListBox4.List(l)).UsedRange).Copy Cells(65536, 1).End(xlUp).
Offset(1, 0)
Next l
.Close , False
End If
End With
err:
TextBox2 = ""
ListBox3.Clear
ListBox4.Clear
Application.ScreenUpdating = True
CommandButton12.Enabled = False
End Sub

```

以上过程的思路为：

- ◆ 确认待合并工作表的标题行数。
- ◆ 如果选择“新工作表”，那么新建一个工作表，命名为“合并工作表”。然后将待合并工作表的标题复制到“合并工作表”的首行，再逐一合并每个工作表的已用区域到该工作表第一个非空行。
- ◆ 如果选择“当前工作表”，那么将待合并工作簿的每个工作表的已用区域复制到当前表中，按工作表顺序追加到空白区。

step 7 在“财务模块”中录入以下代码调用窗体：

```

Sub 工作簿合并()
合并工作簿.Show 0
End Sub

```

功能测试：

step 1 在 D 盘中存放了“一班.xlsx”、“二班.xlsx”和“三班.xlsx”三个工作簿，每个工作簿包括

“A 组”、“B 组”和“C 组”三个工作表，每个工作表的格式完全相同，仅仅姓名成绩存在差异。图 30.12 即为“一班.xlsx”工作簿的工作表分布情况。

step 2

执行过程“工作簿合并”，弹出“合并工作簿”窗体。

step 3

单击“多工作簿合并”选项卡中的“浏览”按钮，选择 D 盘，在窗体中的文字框中将显示路径“D:\”，同时在两个列表框中显示 D 盘中所有工作簿名称。

step 4

直接单击“合并工作簿”按钮，将会弹出对话框让用户指定标题行。输入 1 并回车即可把三个工作簿中所有工作表的数据复制到当前工作表中。

step 5

再次运行窗体，仍然将路径指定为 D 盘，但选中“同工作簿”单选按钮，且将复选框也取消选中状态，表示不合并同名工作表。当单击按钮“合并工作簿”，且指定标题行后，程序会将三个工作簿中九个工作表合并到新工作簿中，每个工作表以原工作簿名与工作表名作为名称，且在工作簿前创建一个工作表目录，如图 30.13 所示。

	A	B	C
1	姓名	成绩	
2	赵	97	
3	钱	90	
4	孙	95	
5	李	66	
6	周	70	
7	吴	82	

图 30.12 一班成绩表

	A	B	C	D	E	F
1	序号	工作表				
2	1	一班:A组				
3	2	一班:B组				
4	3	一班:C组				
5	4	三班:A组				
6	5	三班:B组				
7	6	三班:C组				
8	7	二班:A组				
9	8	二班:B组				
10	9	二班:C组				

图 30.13 多工作簿合并

step 6

切换到第二页“多工作表合并”，默认状态下将在文字框中显示当前工作簿名，且在左边的列表框罗列出所有工作表名。如果需要合并当前工作簿中所有工作表，则可以单击按钮“全部添加>>>”，然后单击“合并工作表”，程序会弹出对话框让用户指定工作表标题行数，最后将所有工作表的数据按顺序合并到一个新工作表中。

step 7

如果需要合并 D 盘中“三班.xlsx”，那么单击“浏览…”按钮，从对话框中选择目标工作簿并双击返回窗体，在文字框中将显示被选工作簿的完整路径，且在两个列表框中罗列出所有工作表名。单击“合并工作表”，那么 D 盘中“三班.xlsx”中的三个工作表数据会合并到一个工作表中，如图 30.14 所示。

	A	B	C	D	E	F
1	姓名	成绩				
2	魏	97				
3	陶	79				
4	姜	86				
5	戚	97				
6	谢	78				
7	邹	68				
8	喻	76				
9	柏	98				
10	水	80				

图 30.14 多工作表合并

注意事项:

- (1) 在编辑窗体完成并保存时必须确保默认显示第一页，否则运行程序时会出错。
- (2) 待合并工作簿中的工作表必须格式一致，否则合并无意义，包括列数据和标题行全部一致。
- (3) 工作簿合并或者工作表合并相对不同用户在使用上会有较大差别，设计程序时必须给用户最大的定义空间，工具才具有灵活性。本例中路径可以定义，参与合并的工作表可以定义，工作表标题行数可以定义，合并后的存放工作簿或者工作表可以定义。



本例测试文件存放目录参见光盘：..\ 第三十章\工作簿合并\

30.2.5 工作表拆分

案例要求: 将工作表以用户指定的列作为拆分条件，拆分成多个工作表。

实现步骤:

- step 1** 插入一个窗体，将其名称属性修改为“报表拆分”，将“Caption”改为“工作表”拆分。
- step 2** 在窗体中插入两个复合框、两个列表框、一个标签、五个命令按钮和两个单选按钮，且按图 30.15 所示布局排列。

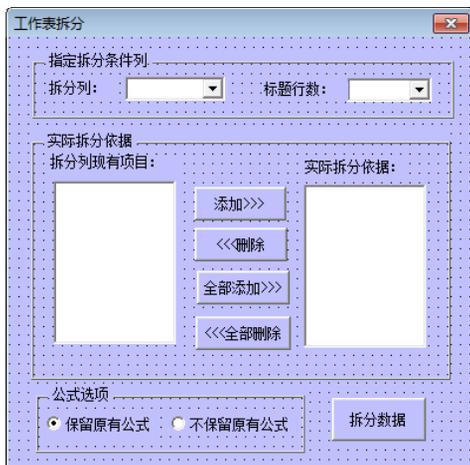


图 30.15 工作表拆分窗体控件布局方式

- step 3** 双击窗体打开代码窗口，并在窗口中录入以下代码:

```
Dim i As Integer, j As Integer, temp As Byte
Private Sub UserForm_Activate()
    For i = 1 To ActiveSheet.UsedRange.Columns.Count
        ComboBox1.AddItem WorksheetFunction.Substitute(Cells(1, i).Address(0, 0),
        1, ""), ComboBox1.ListCount '将已用列的编号填充到列表框
    Next i
    ComboBox1.Text = ComboBox1.List(0) '显示第一项
```

```
ComboBox2.List = Array(1, 2, 3, 4, 5, 6, 7, 8) '列表框 2 添加下拉列表
ComboBox2.Text = 2
End Sub
```

以上代码表示窗体激活时在第一个复合框中罗列出已用区域中每一列的英文列标,在第二个复合框中则添加 1 到 8 的自然数序列。

```
Private Sub ComboBox1_Change() '指定待拆分的列
    Dim cell As Range, only As New Collection, Item
    On Error Resume Next
    temp = Range(ComboBox1.Text & "1").Column '将控件中指定的字母转换成列号
    ListBox1.Clear '清除控件
    For i = ComboBox2.Value + 1 To Cells(Rows.Count, temp).End(xlUp).Row '遍历
        工作表中非标题行的数据
        If Cells(i, temp) <> "" Then only.add CStr(Cells(i, temp).Text),
        CStr(Cells(i, temp).Text) '提取不重复值到 only 中
    Next i
    For Each Item In only
        ListBox1.AddItem Item, ListBox1.ListCount '将不重复的数据罗列在列表框 1 中
    Next Item
    ListBox2.List = ListBox1.List '将列表框 1 的数据复制到列表框 2
End Sub
```

以上过程表示第一个复合框的值改变时将复合框所指定列的所有单元格的不重复值添加到两个列表框中。

```
Private Sub ComboBox2_Change() '指定标题行数
    Dim cell As Range, only As New Collection, Item
    On Error Resume Next
    temp = Range(ComboBox1.Text & "1").Column '将控件中指定的字母转换成列号
    If Cells(Rows.Count, temp).End(xlUp).Row <= VBA.CLng(ComboBox2.Value) Then
        MsgBox "你的工作表中" & Me.ComboBox1.Text & "列数据不足" & Me.ComboBox2.Text & "
        行! ": Exit Sub
    End If
    ListBox1.Clear
    For i = ComboBox2.Value + 1 To Cells(Rows.Count, temp).End(xlUp).Row
        If Cells(i, temp) <> "" Then only.add CStr(Cells(i, temp).Text),
        CStr(Cells(i, temp).Text) '提取不重复值到 only 中
    Next i
    For Each Item In only
        ListBox1.AddItem Item, ListBox1.ListCount '将不重复的数据罗列在列表框 1 中
    Next Item
    ListBox2.List = ListBox1.List '将列表框 1 的数据复制到列表框 2
End Sub
```

以上过程表示第二个复合框的值改变时将第一个复合框指定的列中除标题行以外的不重复的值添加到两个工作表中。作用与复合框 1 一样——更新列表框中的值。

```
Private Sub CommandButton1_Click()
    Dim k As Integer
```

```

For i = 0 To ListBox1.ListCount - 1 '遍历列表框 1
    If ListBox1.Selected(i) Then '如果呈选中状态
        For j = 0 To ListBox2.ListCount - 1 '遍历列表框 2
            If ListBox2.List(j) = ListBox1.List(i) Then MsgBox "已存在该项目!", 64,
"提示": Exit Sub
            k = k + 1
        Next j
        If k = j Then ListBox2.AddItem ListBox1.List(i) '将选中数据添加到列表框 2
    Exit For
    End If
Next i
End Sub

```

以上过程表示单击“添加>>>”按钮时将第一个列表框中选择的值添加到第二个列表框中。但在添加前需要检查是否已经存在相同值。

```

Private Sub CommandButton2_Click()
    If ListBox2.ListCount > 0 Then
        For i = 0 To ListBox2.ListCount - 1 '遍历列表框 2
            If ListBox2.Selected(i) Then ListBox2.RemoveItem (i) '删除选中数据
        Next i
    End If
End Sub

```

以上过程表示将列表框中选中的值删除。

```

Private Sub CommandButton3_Click()
    ListBox2.List = ListBox1.List '复制列表框数据
End Sub
Private Sub CommandButton4_Click()
    ListBox2.Clear '清除列表框数据
End Sub

```

以上过程表示全部删除拆分依据和添加全部拆分依据。

```

Private Sub CommandButton5_Click()
    Dim j As Integer
    j = ActiveSheet.Index '记录当前表地址
    temp = Range(ComboBox1.Text & "1").Column '将字母转数字
    On Error GoTo err
    Application.ScreenUpdating = False
    For i = 0 To ListBox2.ListCount - 1
        Sheets.add After:=Sheets(Sheets.Count) '添加工作表
        Sheets(Sheets.Count).Name = ListBox2.List(i) '工作表名等于列表框中项目名称
        Sheets(j).Rows("1:" & ComboBox2.Text).Copy Sheets(Sheets.Count).Cells(1,
1) '将源工作表的数据复制到新表中
    Next i
    Sheets(j).Select '返回原工作表
    On Error Resume Next
    For i = ComboBox2.Value + 1 To Cells(Rows.Count, temp).End(xlUp).Row

```

```
If OptionButton1 Then '如果保留公式
    Rows(i & ":" & i).Copy Sheets(Cells(i, temp).Text).Cells(Rows.Count,
temp).End(xlUp).Offset(1, 0).EntireRow '整行复制
Else
    Sheets(Cells(i, temp).Text).Cells(Rows.Count, temp).End(xlUp).Offset(1,
0).EntireRow = Rows(i & ":" & i).Value '否则复制其值
End If
Next i
For i = 0 To ListBox2.ListCount - 1
    Sheets(ListBox2.List(i)).UsedRange.Borders.LineStyle = xlContinuous '加
边框
    Sheets(ListBox2.List(i)).Cells.EntireColumn.AutoFit '自动调整列宽
Next i
Application.ScreenUpdating = True
MsgBox "拆分完毕!", 64, "提示"
Unload Me '关闭窗体
Exit Sub
err:
MsgBox "无法建立( " & ListBox2.List(i) & " )工作表!", 64, "提示" '重名时无法
建立新表,"!"这类特殊符号也不能
Application.DisplayAlerts = False
Sheets(Sheets.Count).Delete '删除新表
Application.DisplayAlerts = True
Application.ScreenUpdating = True
End Sub
```

以上过程的思路为:

- ◆ 首先,将第一个复合框中的英文列标转换成数值。
- ◆ 根据第二个列表框的项目个数创建对应数量的新工作表。
- ◆ 根据第二个复合框的值复制标题行到每个新工作表。
- ◆ 利用 For 循环在拆分列中逐一检查数据,如果与拆分条件中任意值相同,则复制该行到对应的工作表中,直到最后一行。

step 4 在“财务模块”中录入以下过程调用“报表拆分”窗体。

```
Sub 工作表拆分()
    报表拆分.Show 0
End Sub
```

功能测试:

step 1 打开待拆分的工作表,如图 30.16 所示。

step 2 执行过程“工作表拆分”,在打开的对话框中将拆分列调整为“B”,将“标题行数”调整 1,如图 30.17 所示。表示对 B 列第一行以外的数据作为标准进行拆分。

step 3 单击按钮“拆分数据”,执行结果如图 30.18 所示。

step 4 再次运行窗体，将拆分条件设置为 C 列，执行结果如图 30.19 所示。

	A	B	C	D
1	姓名	部门	性别	工资
2	赵	生产部	男	1800
3	钱	业务部	男	1800
4	孙	生产部	女	1560
5	李	人事部	男	1480
6	周	生产部	女	1600
7	吴	生产部	女	1650
8	郑	业务部	男	1650
9	王	生产部	男	1600
10	冯	人事部	女	1560

图 30.16 工资表

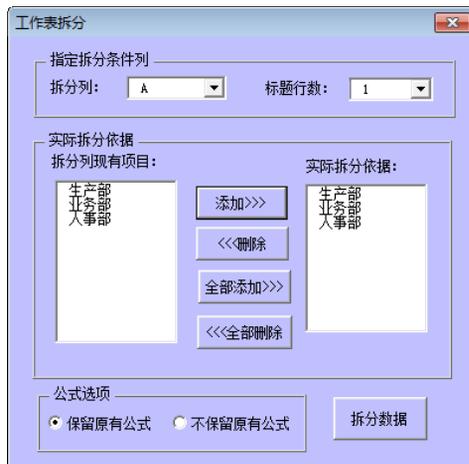


图 30.17 设置拆分条件

图 30.18 按部门拆分工作表

图 30.19 按性别拆分工作表

注意事项:

(1) 以一系列中的单元格数据为条件创建工作表并复制数据，必须确保条件没有重复，否则创建工作表会失败。所以将该列的值添加到列表框时就需要去重复值。

(2) 拆分工作表时提供的是否保留公式非常有用。如果每行中有单元格引用了本行的值进行计算，那么拆分后不影响公式结果；如果一行中任意单元格的公式引用了其他行或者其他工作表的数据，那么拆分时尽量将公式转换成值。

(3) 如果工作表中有某些特殊字符可能无法操作成功。因为“!”、“*”或者“\”等字符不可作为工作表名。

30.2.6 工作簿拆分

案例要求: 一个工作簿中有多个工作表，将每个工作表保存为一个单独的工作簿是工作中常有的事。本例利用 VBA 实现瞬间将数百个工作表保存为工作簿。

实现步骤:

step 1 插入一个窗体，将“Caption”属性改为“工作簿拆分”，将“名称”改为“拆分工作簿”。

step 2 在窗体中添加三个标签、一个文字框、六个命令按钮和两个单选按钮，且按图 30.20 方式对控件布局摆放。

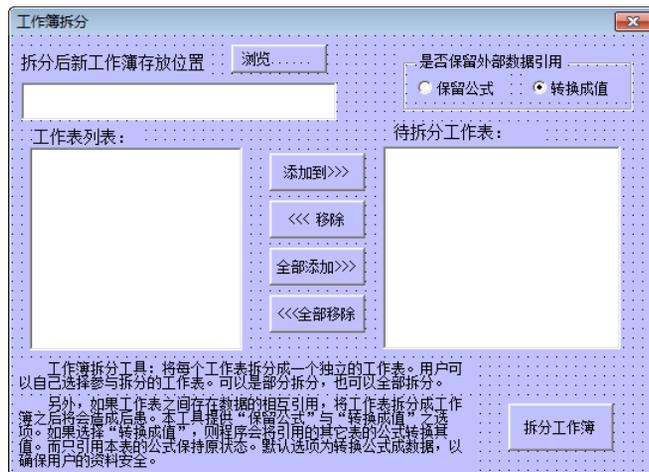


图 30.20 拆分工作簿窗体中的控件布局方式

step 3 双击窗体进入代码窗口，并录入以下代码：

```
Private Sub UserForm_Activate()
    CommandButton6.Enabled = False '激活工作簿时禁用"拆分工作簿"按钮
    On Error Resume Next
    For i = 1 To Sheets.Count
        Me.ListBox1.AddItem Sheets(i).Name '将所有工作表名添加到列表框 3 中
    Next
End Sub
```

以上过程表示激活窗体时禁用“拆分工作簿”向导，并将所有工作表名添到列表框中。

```
Private Sub CommandButton1_Click()
    Dim fd As FileDialog
    Set fd = Application.FileDialog(msoFileDialogFolderPicker)
    '如果选择了路径，则将路径显示在文字框中
    If fd.Show = -1 Then Me.TextBox2 = IIf(Right(fd.SelectedItems(1), 1) = "\",
    fd.SelectedItems(1), fd.SelectedItems(1) & "\")
    Set fd = Nothing
End Sub
```

以上过程表示单击“浏览…”按钮时选择一个文件夹，且将其路径存于文字框中。

```
Private Sub CommandButton2_Click()
    Dim i As Integer, j As Integer
    For j = 0 To ListBox1.ListCount - 1 '遍历列表框 1 的数据项
        If Me.ListBox2.ListCount > 0 Then '如果列表框 2 有数据
            If ListBox1.Selected(j) Then '如果列表框 1 中某数据呈选中状态
                For i = 0 To ListBox2.ListCount - 1 '遍历列表框 2 的数据项
                    '如果列表框 2 中选择的项目在列表框 2 中已经存在则运行 ERR 标签处的代码
                    If ListBox1.List(j) = ListBox2.List(i) Then GoTo err
                Next i
            End If
        End If
    Next j
```

```

End If
If ListBox1.Selected(j) Then ListBox2.AddItem ListBox1.List(j) '将列表框
3 中选择数据添加到列表框 4 中
Next j
err:
CommandButton6.Enabled = True '允许拆分工作簿
End Sub

```

以上过程表示将列表框 1 中选中的项目添加到列表框 2，如果重复则忽略。

```

Private Sub CommandButton3_Click()
If ListBox2.ListCount = 0 Then Exit Sub '如果列表框为空则退出
If ListBox2.ListCount > 0 Then
For i = 0 To ListBox2.ListCount - 1
If ListBox2.Selected(i) Then ListBox2.RemoveItem (i) '移除选择的项目
Next i
End If
If Me.ListBox2.ListCount = 0 Then CommandButton6.Enabled = False '列表框 2 为
空则禁止拆分工作簿
End Sub

```

以上过程表示将列表框 2 的选中项目删除。

```

Private Sub CommandButton4_Click()
ListBox2.List = ListBox1.List '复制数据
CommandButton6.Enabled = True '允许拆分工作簿
End Sub

```

以上过程表示将列表框 1 的所有项目复制到列表框 2。

```

Private Sub CommandButton5_Click()
ListBox2.Clear '清空列表框数据
CommandButton6.Enabled = False '禁止拆分工作簿
End Sub

```

以上过程表示清空列表框 2 的项目。

```

Private Sub CommandButton6_Click()
If Me.TextBox2 = "" Then MsgBox "请选择保存路径!", 64, "提示": Exit Sub '没有
存放路径则提示且退出程序
Application.ScreenUpdating = False
Dim i As Integer, sht As String
sht = ActiveWorkbook.Name '提取当前工作簿的名称
For i = 0 To Me.ListBox2.ListCount - 1 '遍历列表框 4
Sheets(ListBox2.List(i)).Select '选择与列表框中序号相同的工作表
ActiveSheet.Copy '整表复制
ActiveWorkbook.SaveAs Filename:=Me.TextBox2 & Me.ListBox2.List(i) & ".xls",
FileFormat:=xlNormal, CreateBackup:=False '工作簿另存
If Me.OptionButton2 Then '如果需要将公式转换成值
With ActiveSheet.UsedRange

```

```

Set cell = .Find("=*!", LookIn:=xlFormulas, SearchOrder:=xlByRows,
LookAt:=xlPart, MatchCase:=True) ' "=*!"表示有外部链接
On Error Resume Next
FirstAddress = cell.Address
Do '开始查找
    cell = cell.Value '将找到的公式转换成值
    Set cell = .FindNext(cell)
Loop Until cell Is Nothing Or cell.Address = FirstAddress
End With
End If
ActiveWindow.Close '关闭工作簿
Workbooks(sht).Activate '返回原工作簿
Next i
Application.ScreenUpdating = True
Shell "EXPLORER.EXE " & Me.TextBox2, vbNormalFocus '打开存放新工作簿的文件夹
End Sub

```

以上过程的思路为:

- (1) 检测文字框是否是空白, 没有存放路径则提示且退出程序。
- (2) 利用循环逐个复制列表框 2 中的工作表到新工作簿, 且另存到文字框所指定的路径。
- (3) 如果选择了“转换成值”单选按钮, 那么利用 Find 方法查找前四个字符等于“=*!”的单元格, 且将转换成值。公式中有“=*!”, 表示它引用了其他工作簿的数据。
- (4) 打开保存拆分工作簿的文件夹。

在“财务模块”中输入以下代码调用窗体。

```

Sub 工作簿拆分()
    拆分工作簿.Show 0
End Sub

```

step 1 打开待拆分的工作簿, 它包含七个工作表, 如图 30.21 所示。

	A	B	C	D	E	F
1	姓名	产量				
2	赵	4318				
3	钱	4374				
4	孙	4816				
5	李	4804				
6	周	5119				
7	吴	4201				
8	郑	5451				
9	王	4296				

图 30.21 包含七个生产表的工作簿

step 2 执行过程“工作簿拆分”, 单击“浏览…”按钮, 从对话框中选择 D 盘“生产表”目录, 返回窗体后在文字框中将产生路径。

step 3 单击按钮“全部添加”, 表示拆分所有工作表, 对于外部数据引用的处理方式选择“转换成值”, 如图 30.22 所示。

step 4 单击按钮“拆分工作簿”，拆分结果如图 30.23 所示。

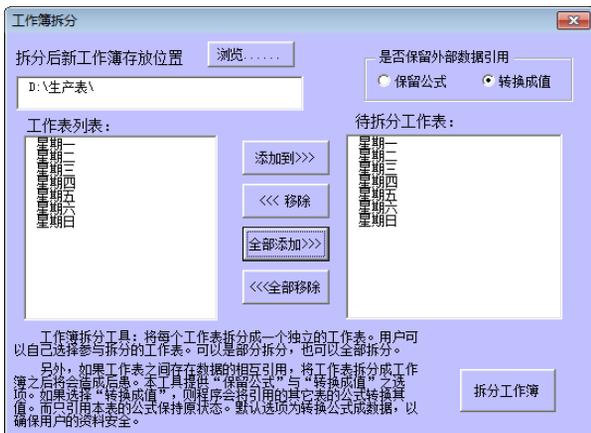


图 30.22 设置拆分方式

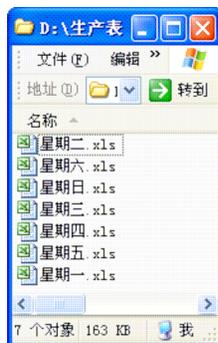


图 30.23 拆分结果

注意事项:

(1) 工作簿拆分时，如果单元格引用了其他工作簿的数据，尽量转换成值，否则工作簿移动后可能丢失引用的数据。

(2) 单元格是否引用其他工作簿的值，判断方法是公式中包括“=*!”，其中“*”是通配符，表示任意字符。

30.2.7 复选框工具

案例要求: 工作中常需要对某个选项进行“是”与“否”的判断，或者“合格”与“不合格”的判断，通常采用复选框来实现。本工具可以批量地生成、编辑、删除复选框。

实现步骤:

step 1 插入一个窗体，将其“名称”和“Caption”属性修改为“复选框工具”。

step 2 在窗体中插入四个命令按钮，两个单选框和两个复选框，以及一个文字标签对工具做说明。其布局摆放方式如图 30.24 所示。其中四个按钮的名称分别为“添加”、“删除”、“全选”、“不选”；两个单选框的名称分别为“选区操作”、“工作表操作”；复选框的名称分别为“关联单元格”和“打印复选框”。

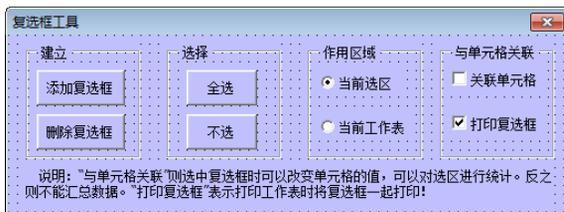


图 30.24 复选框工具控件布局方式

默认状态让生成的控件可以打印，但不关联到单元格。

step 3 双击窗体进入代码窗口，并输入以下代码：

```
Private Sub 添加_Click()  
    Dim cell As Range, rng As Range, 复选框文字 As String, i As Long  
    复选框文字 = Application.InputBox("请输入复选框需要显示的文字:", "提示", "", , , , ,  
    2) '确认复选框显示的文字  
    If TypeName(Selection) <> "Range" Then MsgBox "请选择单元格!", 64, "提示": Exit  
    Sub  
    If Selection.Rows.Count = Rows.Count Or Selection.Columns.Count = Columns.  
    Count Then MsgBox "不要整行整列操作,选择目标区域即可!", 64, "提示": Exit Sub  
    Set rng = Selection  
    Application.ScreenUpdating = False  
    For Each cell In Selection  
        ActiveSheet.CheckBoxes.Add(cell.Left, cell.Top, cell.Width - 8, cell.  
    Height).Select '添加一个复选框  
        With Selection  
            .Characters.Text = 复选框文字 '设置显示文字  
            If 关联单元格 Then .LinkedCell = cell.Address '链接单元格地址  
            .PrintObject = 打印复选框 '设置打印属性  
            .Name = "复选框" & i + 1 '指定名称  
        End With  
  
        i = i + 1  
        If 关联单元格 Then cell.Font.Color = cell.Interior.Color '如果关联单元格则将  
    单元格的字体色设置为单元格背景色  
    Next  
    Application.ScreenUpdating = True  
    rng.Select '选择单元格  
End Sub
```

以上过程的思路为:

- ◆ 首先,弹出对话框让用户确认复选框显示的文字。
- ◆ 如果全选工作表,或者选择整行整列时拒绝执行。
- ◆ 利用 For 循环在每个单元格添加复选框,复选框的显示文字由用户指定,同时根据两个复选框的状态决定选框是否与它所在单元格进行关联及是否能打印。
- ◆ 为了让单元格中显示的 true 和 false 不显示出来扰乱视线,将单元格文字的颜色设置为与单元格背景色一致。

```
Private Sub 删除_Click()  
    Dim cb As Shape, ShapCell As Range  
    On Error Resume Next  
    Application.ScreenUpdating = False  
    If 选区操作 Then '如果仅仅在选区操作  
        For Each cb In ActiveSheet.Shapes '遍历图形对象  
            Set ShapCell = Intersect(Selection, cb.TopLeftCell) '提取对象左上角单元格  
    与选区的交集  
            If Not ShapCell Is Nothing And Left(cb.Name, 3) = "复选框" Then cb.Delete:  
            ShapCell = "" '如果交集存在且名称前三个字符等于"复选框",那么删除它
```

```

Next
Else '否则
    ActiveSheet.CheckBoxes.Delete '清除所有复选框
End If
Application.ScreenUpdating = True
End Sub

```

以上过程的思路为：

- ◆ 声明一个 Shape 对象，利用它遍历工作表中所有图形对象。
- ◆ 如果窗体中勾选了“当前选区”，那么需要判断每个复选框是否在选区之间。通过 TopLeftCell 属性获取对象左上角的单元格，如果它与选区存在交集，而且名字中包括“复选框”，那么删除它；如果勾选了“当前工作表”，那么一次性删除所有复选框即可。

```

Private Sub 全选_Click()
    Dim cb As Shape, ShapCell As Range
    On Error Resume Next
    Application.ScreenUpdating = False
    Application.Calculation = xlManual '设置计算方式为手动
    If 选区操作 Then
        For Each cb In ActiveSheet.Shapes
            Set ShapCell = Intersect(Selection, cb.TopLeftCell)
            If Not ShapCell Is Nothing Then cb.ControlFormat.Value = xlOn '设置为选中状态
        Next
    Else
        ActiveSheet.CheckBoxes.Value = xlOn '所有复选框都选中
    End If
    Application.ScreenUpdating = True
    Application.Calculation = xlAutomatic '设置计算方式为自动
End Sub

```

以上过程与删除复选框的思路一致，唯一不同的是将删除改为对复选框打勾。

```

Private Sub 不选_Click()
    On Error Resume Next
    Application.ScreenUpdating = False
    Application.Calculation = xlManual '设置计算方式为动
    Dim cb As Shape, ShapCell As Range
    On Error Resume Next
    If 选区操作 Then
        For Each cb In ActiveSheet.Shapes '遍历所有 shape 对象
            Set ShapCell = Intersect(Selection, Range(cb.TopLeftCell.AddressLocal))
            If Not ShapCell Is Nothing Then cb.ControlFormat.Value = xlOff '如果在选区内，则将复选框设为 Off 状态
        Next
    Else '如果选择整个工作表

```

```

ActiveSheet.CheckBoxes.Value = xlOff '将所有复选框设为 Off 状态
End If
Application.ScreenUpdating = True
Application.Calculation = xlAutomatic
End Sub

```

以上过程与全选的思路完全一致。

step 4 进入“财务模块”，录入以下过程用于调用窗体。

```

Sub 复选框 ()
复选框工具.Show 0
End Sub

```

功能测试：

step 1 打开产品质报表，其数据如图 30.25 所示。

step 2 按住 Ctrl 键选择 B2:B7 和 D2:D7 两个区域，然后执行过程“复选框”。

step 3 所有选项保持默认状态，单击“添加复选框”按钮，在弹出的对话框中输入“合格”并回车，在工作表中会瞬间产生名为“复选框 1”、“复选框 2”等名称，显示文字均为“合格”的复选框，如图 30.26 所示。

	A	B	C	D
1	产品	质量	产品	质量
2	A		G	
3	B		H	
4	C		I	
5	D		J	
6	E		K	
7	F		L	

图 30.25 产品质量报表

	A	B	C	D
1	产品	质量	产品	质量
2	A	<input type="checkbox"/> 合格	G	<input type="checkbox"/> 合格
3	B	<input type="checkbox"/> 合格	H	<input type="checkbox"/> 合格
4	C	<input type="checkbox"/> 合格	I	<input type="checkbox"/> 合格
5	D	<input type="checkbox"/> 合格	J	<input type="checkbox"/> 合格
6	E	<input type="checkbox"/> 合格	K	<input type="checkbox"/> 合格
7	F		L	

图 30.26 批量产生“合格”复选框

step 4 重新执行过程“复选框”，将作用区域调整为“当前工作表”，然后单击按钮“删除复选框”，工作表中所有复选框被瞬间删除。

step 5 仍然选择 B2:B7 和 D2:D7 两个区域，然后将“关联单元格”选中再执行“添加复选框”，且在弹出的对话框中输入“OK”。对部分复选框打勾，再录入汇总的公式：

```
=SUMPRODUCT(--B2:B7)
```

公式可以根据复选框的状态汇总合格的数量，如图 30.27 所示。

step 6 复选框工具也可以应用到其他很多方面，如图 30.28 所示。

注意事项：

(1) 添加复选框时必须让复选框的边距与单元格为准，优点是多个复选框之间不会重叠，而且删除复选框时可以更准确地判断它的位置。

(2) PrintObject 表示对象是否可以打印，值为 true 时可以打印。而 LinkedCell 属性是指复选框打勾时，它所控制的单元格对象是哪一个，赋值时只能使用单元格地址，而非单元格对象。

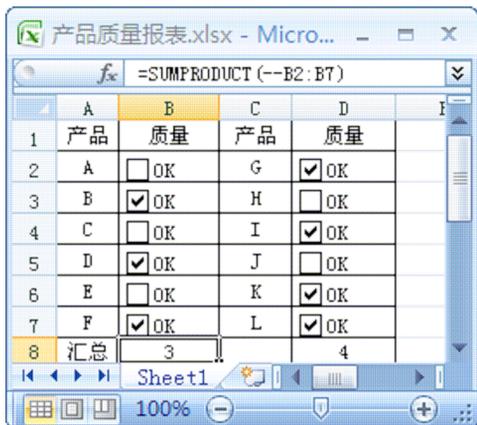


图 30.27 将复选框关联到单元格实现质量统计

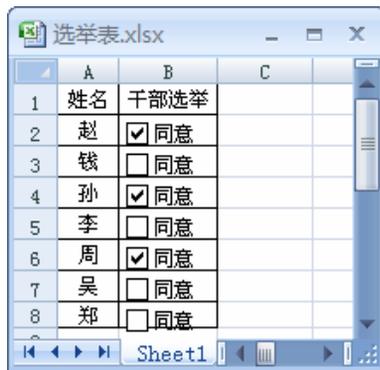


图 30.28 干部选举

30.2.8 文本与数据转换

案例要求：将工作表中的数字在值与文本之间转换，而且是批量转换。

实现步骤

step 1 在“财务模块”中录入以下过程：

```

Sub 文本与数值互换 ()
    On Error GoTo err
    Dim 转换方式, arr(), Arr2(), row As Long, column As Long
    转换方式 = Application.InputBox("输入 1: 将文本转换成数字;" + Chr(10) + "输入 2: 将数字转换成文本。", "操作选项", 1, , , , 1)
    Application.ScreenUpdating = False
    With Application.Intersect(Selection, ActiveSheet.UsedRange)
        If 转换方式 = 1 Then '如果选择将文本转换成数字
            .NumberFormatLocal = "G/通用格式" '修改数字格式
            .Value = .Value '转换值
        Else '否则
            arr = Application.Intersect(Selection, ActiveSheet.UsedRange) ' 将选区与已用区域的交集写入数组
            Arr2 = arr '再建立一个数组
            For row = 1 To UBound(arr) '逐行循环
                For column = 1 To UBound(arr, 2) '逐列循环
                    If IsNumeric(arr(row, column)) Then Arr2(row, column) = "" & arr(row, column) '如果是数字则前置""使用成为文本
                Next
            Next
            .Value = Arr2 '将数组一次性写入单元格中
        End If
    End With
err:
    Application.ScreenUpdating = True
End Sub

```

以上过程的思路为：

- ◆ 弹出一个输入框让用户指定转换方式。
- ◆ 如果输入 1，则将选区的格式设置为“G/通用格式”，然后将文本转换成数值。
- ◆ 如果输入 2，则将选区转换成数组再逐一添加前置单引号，最后将数组一次性存入选区中。

功能测试

- step 1** 打开职工资料表，如图 30.29 所示，图中工号是数字。
- step 2** 全选工作表，执行过程“文本与数字互换”，在弹出的“操作选项”对话框中输入 2，执行结果会如图 30.30 所示。所有数值已转换成文本，不能直接参与运算，且在单元格左上方有一个绿色小三角。
- step 3** 再次执行过程，在“操作选项”对话框输入 1，那么区域中的值将会还原为图 30.29 状态。如果需要数值转文本后不出现绿色小三角，可以有两种处理方式，一是选择带有小三角的单元格，手工选择“忽略”；二是在代码中加入以下语句，表示不显示该类错误提示。

```
Application.ErrorCheckingOptions.NumberAsText = false
```

图 30.31 是忽略错误提示后的工作表。

	A	B	C	D
1	姓名	工号	姓名	工号
2	赵	7578	王	7057
3	钱	3024	冯	4792
4	孙	9637	陈	9485
5	李	9619	褚	868
6	周	7574	卫	1124
7	吴	6777	蒋	6775
8	郑	1178	沈	5673

图 30.29 职工资料表

	A	B	C	D
1	姓名	工号	姓名	工号
2	赵	'7578	王	'7057
3	钱	'3024	冯	'4792
4	孙	'9637	陈	'9485
5	李	'9619	褚	'868
6	周	'7574	卫	'1124
7	吴	'6777	蒋	'6775
8	郑	'1178	沈	'5673

图 30.30 数字转文本

	A	B	C	D
1	姓名	工号	姓名	工号
2	赵	7578	王	7057
3	钱	3024	冯	4792
4	孙	9637	陈	9485
5	李	9619	褚	868
6	周	7574	卫	1124
7	吴	6777	蒋	6775
8	郑	1178	沈	5673

图 30.31 忽略错误提示

注意事项

(1) 本工具转换文本或者转换数字都会连带将公式转换公式结果，如果需要忽略公式，那么必须单元格逐个循环，利用 HasFormula 属性来判断是否带有公式。

(2) 代码中必须使用 Intersect(Selection, ActiveSheet.UsedRange)，否则用户全选工作表再执行过程将造成计算时间过长，如果在循环中再执行其他过程则可能造成死机。

30.3 合并工具箱

工作中常需要将符合某些条件的单元格合并，或者取消已合并的单元格。然而 Excel 自带的合并功能完全无法满足需求，通过自定义合并工具可以使合并更多样化。

30.3.1 合并数据并居中及其还原

案例要求：Excel 自带的合并工具具有一个 BUG——合并后仅仅保留左上角单元格的值。本例中的工具突破这个限制，让所有单元格的值都可以显示在合并区域中，而且可以定义分隔符。

实现步骤

step 1 插入一标准模块，且命名为“合并模块”。

step 2 在模块中录入以下过程代码：

```
Sub 合并数据并居中 ()
    If ActiveSheet.ProtectContents Then MsgBox "工作表已保护,本程序拒绝执行!", 64,
"Excel 百宝箱": Exit Sub
    Application.ScreenUpdating = False '关闭更新屏幕
    Application.DisplayAlerts = False '禁止提示
    Application.Calculation = xlManual '将计算方式改为手动
    ThisWorkbook.Sheets(1).[a10000] = ActiveWorkbook.Name & ActiveSheet.Name
    ThisWorkbook.Sheets(1).[a10001] = Selection.Address
    Selection.Copy ThisWorkbook.Sheets(1).[a1]
    Dim i As Long, Imax As Long, TempStr As String, arr, 间隔符号 As String
    If TypeName(Selection) <> "Range" Then MsgBox "请选择单元格!", 65, "Excel 百
宝箱 ": Exit Sub
    间隔符号 = Application.InputBox("请输入合并多行数据时需要使用的分隔符,可以保持空白。
", "确定分隔符", "", , , , 2)
    Set arr = Selection '将选区读入内存数组
    Imax = arr.Count '预先获取数组中元素个数
    For i = 1 To Imax '在数组中循环
        TempStr = TempStr & arr(i) & IIf(i Mod arr.Columns.Count = 0, Chr(10), 间
隔符号) '串联每个元素与间隔符,如果某元素在数组中位置编号除以数组的宽度(列数)余数为 0 时,
那么则串联换行符,而非间隔符
    Next
    With Selection
        .MergeCells = True '合并选区
        .HorizontalAlignment = xlCenter '居中
        .VerticalAlignment = xlCenter '居中
        .NumberFormatLocal = "@" '数字格式转为文本
        .Value = Left(TempStr, Len(TempStr) - 1) '将串联后的字符串直接写入合并后的单
元格
    End With
    Application.Calculation = xlAutomatic '将计算方式改为自动
    Application.DisplayAlerts = True '还原提示
    Application.ScreenUpdating = True '恢复屏幕更新
End Sub
```

以上过程的思路为：

- ◆ 首先，将选区的数据和选区地址及工作表名、工作簿名保存到加载宏的第一个工作表中，便于以后恢复。

- ◆ 让用户定义一个分隔符，用户可以将它定义为空白。
- ◆ 将选区读入数组中，然后利用 For 循环将数组中的值全部串联在一起，包括中间的分隔符。但是在最后一列时不串联分隔符，而是换行符。
- ◆ 最后合并区域，再将串联后的字符串赋值给合并区域，且去掉最后一个换行符。

step 3 再录入以下过程代码，用于还原数据：

```
Sub 还原()
    If TypeName(Selection) <> "Range" Then MsgBox "请选择单元格!", 65, "Excel 百宝箱 ": Exit Sub
    '如果工作簿名、工作表名和单元格地址都相同则将加载宏中的值复制回来
    If Selection.Address = ThisWorkbook.Sheets(1).[a10001].Text And ActiveWorkbook.Name & ActiveSheet.Name = ThisWorkbook.Sheets(1).[a10000].Text Then
        ThisWorkbook.Sheets(1).[a1].Resize(Selection.Rows.Count, Selection.Columns.Count).Copy Selection
    End If
End Sub
```

以上过程首先比较当前工作簿名、工作表名及区域地址与加载宏中保存的是否相同，如果相同则将该加载宏中的数据复制到当前选区中。

在工作簿 Open 事件中为还原过程设置快捷键：

```
Private Sub Workbook_Open() '工作簿启动时指定"工作表管理"的快捷键
    Application.OnKey "%q", "还原"
End Sub
```

功能测试：

step 1 在 A1:B2 单元格中按图 30.32 所示方式输入数据。

step 2 选择 A1:B2，再执行过程“合并数据且居中”，在弹出的“确定分隔符”对话框中输入“-”，如图 30.33 所示，执行的结果如图 30.34 所示。

	A	B
1	中国	广东省
2	东莞	长安镇

图 30.32 待合并数据



图 30.33 确定分隔符

	A	B
1	中国-广东省	
2	东莞-长安镇	

图 30.34 合并效果

step 3 保持选区不变，按下快捷键【Alt+q】执行过程“还原”，那么选区会还原为合并前状态。

step 4 如果执行过程在“确定分隔符”对话框中保持空白，那么在合并区域中将产生四个单元格的值的合并字符串。根据需要，也可以在定义分隔符时录入一个单词。

注意事项：

(1) 宏本身不支持撤销操作，要使合并区域前的数据恢复，唯一的方式就是使用辅助区保存这些值，但恢复次数仅限一次。

(2) 加载宏文件在关闭时不保存，所以如果重新打开 Excel 程序，合并前的数据不能再恢复。

30.3.2 合并数据并复制

案例要求：数据合并后存入剪贴板中，方便在其他任意单个单元格中粘贴。

实现步骤：

step 1 单击菜单【工具】→【引用】，勾选“Microsoft Form 2.0 Object Library”。如果没有该选项，则单击“浏览”按钮，选择“C:\WINDOWS\system32\FM20.DLL”，返回对话框后就会出现该选项。

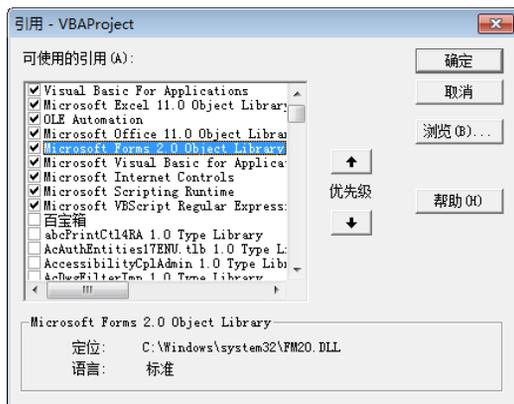


图 30.35 引用对象库

step 2 在“合并模块”中录入以下过程代码：

```
Sub 合并数据并复制() '必须引用对象库 FM20.DLL，它在对引用列表中显示为“Microsoft
Forms2.0 Object library”
    Dim 合并方式, 行数 As Integer, 列数 As Integer, i As Long, j As Long, 合并文本 As
String, TD As New DataObject, arr()
    If TypeName(Selection) = "Range" Then '如果选项是单元格
        If Selection.Count = 1 Then Exit Sub '选择一个单元格则退出程序
        arr = Selection '将选区读入数组
        行数 = UBound(arr, 1) '记录行数
        列数 = UBound(arr, 2) '记录列数
        If 行数 > 1 And 列数 > 1 Then '如果数组行数、列数都大于 1
            合并方式 = MsgBox("纵向合并点击是，横向合并点击否！", vbYesNo, "Excel 百宝箱") '确定
            合并方式
            If 合并方式 = vbYes Then '如果选择纵向合并
                For i = 1 To 列数
                    For j = 1 To 行数
                        合并文本 = 合并文本 & arr(j, i) '纵向串连数组中的所有数据
                    Next
                Next
            Else
                GoTo AA '选择“否”则执行标签 AA 处的代码
            End If
        Else '否则(指行数、列数大于 1 以外的情况以及选横向合并的情况)
            AA:
        End Sub
```

```

AA:
    For i = 1 To 行数
        For j = 1 To 列数
            合并文本 = 合并文本 & arr(i, j) '横向串连数组中的所有数据
        Next
    Next
End If
TD.SetText 合并文本 '将合并后的文本输出到 DataObject
TD.PutInClipboard '将 DataObject 输出到剪贴板
Else
    MsgBox "请选择单元格!", 64, "错误!"
End If
End Sub

```

以上过程的思路为:

- ◆ 首先, 选区读入数组, 且记录行数与列数。
- ◆ 弹出对话框让用户决定合并方式, 如果用户选择“是”则先列后行循环, 选择“否”则先行后列循环, 通过循环合并选区中所有单元格的值。
- ◆ 通过 PutInClipboard 方法将合并的文本存入剪贴板中。

功能测试:

step 1 分别在 A1:C1 单元格录入“2011年”、“5月”、“12日”。

step 2 执行过程“合并数据并复制”, 然后在 D1 单元格粘贴, 那么 D1 将包括 A1:C1 单元格所有内容, 如图 30.36 所示。

step 3 在 A1:B2 分别输入“A”、“B”、“C”、“D”, 然后执行过程“合并数据并复制”, 将弹

出如图 30.37 所示对话框。如果选择是则合并为“ACBD”, 否则合并为“ABCD”并存入剪贴板。

D1	A	B	C	D	E
	2011年	5月	12日	2011年5月12日	

图 30.36 合并前后

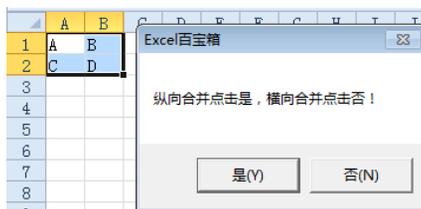


图 30.37 合并方式

注意事项:

(1) 如果不引用“Microsoft Forms2.0 Object library”, 那么变量声明为“New DataObject”将失败。

(2) 如果直接用 Selection 的索引号进行循环, 那么仅仅实现先行后列方式合并。所以代码中采用了两个变量来改变循环方式。

30.3.3 取消区域合并填充原合并值

案例要求: Excel 自带的取消合并工具在取消合并区域后仅仅保留左上角单元格有值, 其余单元格是空白。本例实现取消合并后让所有单元格都具有合并前的值。

实现步骤:

step 1 在“合并模块”中录入以下过程代码:

```
Sub 取消合并且填充数据()
    If ActiveSheet.ProtectContents Then MsgBox "工作表已保护,本程序拒绝执行!", 64, "Excel 百宝箱": Exit Sub
    If TypeName(Selection) <> "Range" Then MsgBox "请选择单元格!", 65, "Excel 百宝箱": Exit Sub
    Application.ScreenUpdating = False
    Application.Calculation = xlManual '将计算方式改为手动
    Dim rng As Range, val, cell As String
    For Each rng In Application.Intersect(Selection, ActiveSheet.UsedRange) '
    在选区与已用数据区域的交集中循环
        If rng.MergeCells Then '如果单元格已经合并
            cell = rng.MergeArea.Address '提取合并单元格的地址
            val = rng.Value '提取合并区域的值
            rng.UnMerge '取消合度
            Range(cell).Value = val '将取消合并的每个单元格赋值
        End If
    Next
    Application.Calculation = xlAutomatic '将计算方式改为自动
    Application.ScreenUpdating = True '恢复刷新
End Sub
```

以上过程的思路为:

- ◆ 首先, 判断选择的区域是否为合并区域, 非合并区域则退出程序。
- ◆ 提取合并区域的地址及值, 将合并区域取消合并, 然后对取消合并后的区域赋值, 让每个单元格都显示取消合并前的值。

功能测试:

step 1 在 A1 单元格录入“中国”, 并合并 A1:B2 区域, 如图 30.38 所示。

step 2 选择合并区域, 执行过程“取消合并且填充数据”, 那么 A1:B2 区域中每个单元格都将显示“中国”, 如图 30.39 所示。

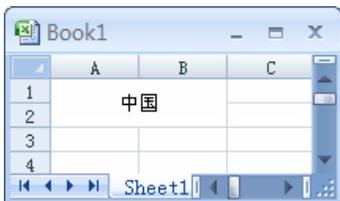


图 30.38 合并状态

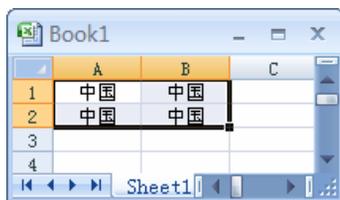


图 30.39 取消合并后的状态

step 3 如果存在多个合并区域，可以全选后执行过程“取消合并且填充数据”，那么所有取消合并区域都会取消合并，且填充合并时的值。如图 30.40 和图 30.41 所示即为取消合并前后的状态。

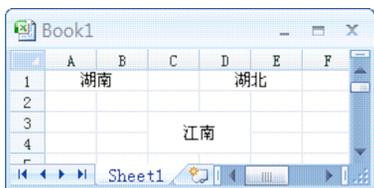


图 30.40 有多个合并区域的工作表



图 30.41 取消多个合并区域

注意事项:

- (1) MergeCells 用于判断单元格是否处于合并区域中。
- (2) 如果在合并区域中是公式而不是文本，那么取消合并后每个单元格仅显示公式的值。

30.3.4 可还原的合并居中

案例要求: 如果区域中有多个值，利用 Excel 自带的工具合并再取消合并后，仅仅左上角一个单元格有值，其余单元格空白。本工具可以实现取消合并后还原合并前的状态。

实现步骤:

step 1 在“合并模块”中录入以下过程代码:

```
Sub 可还原的合并 ()
    If ActiveSheet.ProtectContents Then MsgBox "工作表已保护,本程序拒绝执行!", 64, "Excel 百宝箱": Exit Sub
    If TypeName(Selection) <> "Range" Then MsgBox "请选择单元格!", 65, "Excel 百宝箱": Exit Sub
    Application.DisplayAlerts = False
    Application.ScreenUpdating = False
    Application.Calculation = xlManual '将计算方式改为手动
    Dim sht As Worksheet, ShtName As String, ActiveAdds As String, TmpShtName
    ShtName = ActiveSheet.Name '记录当前表名
    ActiveAdds = Selection.Address '记录当前选区地址
    Selection.Copy '复制选区
    Sheets.add '添加辅助工作表
    Range(ActiveAdds).Select '选择区域
    ActiveSheet.Paste '粘贴
    TmpShtName = ActiveSheet.Name '记录辅助表名称
    With Selection
        .HorizontalAlignment = xlCenter '居中
        .VerticalAlignment = xlCenter '居中
        .Merge '合并
        .Copy '复制
    End With
    Sheets(ShtName).Select '返回原表
End Sub
```

```

Selection.PasteSpecial Paste:=xlPasteFormats, Operation:=xlNone '仅仅粘贴格式
Selection.Borders.LineStyle = xlContinuous '添加边框
Sheets(TmpShtName).Delete '删除辅助表
Application.Calculation = xlAutomatic '将计算方式改为自动
Application.DisplayAlerts = True '还原提示
Application.ScreenUpdating = True '还原屏幕更新
End Sub

```

以上过程的思路为：

- ◆ 添加一个辅助工作表，将待合并区域复制到辅助表中。
- ◆ 对辅助表中的选区合并居中，并复制该合并区域，返回原表中粘贴格式。
- ◆ 最后删除辅助工作表。

功能测试：

step 1 在 A1:B2 中录入任意字符，如图 30.42 所示。

step 2 选择 A1:B2 区域，执行过程“可还原的合并”，合并后如图 30.43 所示。

step 3 单击功能中的合并居中按钮（图标：），A1:B2 区域将会返回图 30.42 的状态，合并前的所有数据都可以恢复。

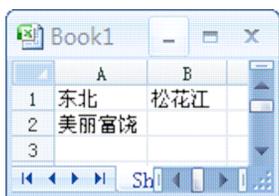


图 30.42 合并前

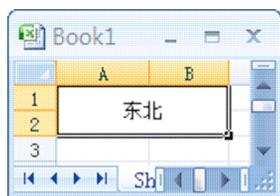


图 30.43 合并后

注意事项：

(1) 本例中技巧在于不是直接合并选区，而是将大小相同且已经合并过的区域的格式粘贴过来，从而使当前区域也具有合并性质。但却保留了所有数据。当取消合并后数据将显示出来。

(2) 删除工作表时会弹出提示，将 DisplayAlerts 属性赋值 false 可以禁止该提示。

30.3.5 合并列中相同数据及还原

案例要求：将一列中相同且相邻的单元格合并。

实现步骤：

step 1 在“合并模块”中录入以下两个过程的代码

```

Sub 合并列中相同数据()
    If ActiveSheet.ProtectContents Then MsgBox "工作表已保护,本程序拒绝执行!", 64, "Excel 百宝箱": Exit Sub
    If TypeName(Selection) <> "Range" Then MsgBox "请选择单元格!", 65, "Excel 百宝箱": Exit Sub

```

```
If Selection.Count = 1 Then MsgBox "请选择一个较大区域!", 64, "Excel 百宝箱": Exit Sub
Sub
  If Selection.Columns.Count > 1 Then MsgBox "只对单列数据进行操作!", 64, "Excel 百宝箱": Exit Sub
  Application.ScreenUpdating = False '禁止屏幕更新
  Application.Calculation = xlManual '将计算方式改为手动
  Dim rng As Range, rg As Range, rngs As Range
  Application.DisplayAlerts = False '禁止提示
  Set rngs = Intersect(ActiveSheet.UsedRange, Selection) '提取选区与已用数据区域的交集
  Set rg = rngs(1)
  For Each rng In rngs.Offset(1, 0).Resize(rngs.Count, 1) '在选区向下偏移一行的区域中循环
    If rng <> rng.Offset(-1, 0) Then '如果对象变量 rng 与其上一个单元不相等
      Range(rg, rng.Offset(-1, 0)).Merge '合并对象变量前面的相同数据区域
      Set rg = rng '重新指定对象变量
    End If
  Next
  Application.Calculation = xlAutomatic '将计算方式改为自动
  Application.DisplayAlerts = True '还原提示
  Application.ScreenUpdating = True '还原屏幕更新
  rngs(1).Select '选择原选区中第一个单元格
End Sub
```

以上过程的思路为:

- ◆ 提取选区与已用区域的交集。
- ◆ 利用 For Each 在交集向下偏移一行的区域中循环, 如果与上一个单元格相等则记录地址, 如果与上一个单元格的值不等则合并它前面的相同值区域。

```
Sub 取消列中合并还原合并数据()
  If ActiveSheet.ProtectContents Then MsgBox "工作表已保护,本程序拒绝执行!", 64, "Excel 百宝箱": Exit Sub
  If TypeName(Selection) <> "Range" Then MsgBox "请选择单元格!", 65, "Excel 百宝箱": Exit Sub
  If Selection.Count = 1 Then MsgBox "请选择一个较大区域!", 64, "Excel 百宝箱": Exit Sub
  Sub
    If Selection.Columns.Count > 1 Then MsgBox "只对单列数据进行操作!", 64, "Excel 百宝箱": Exit Sub
    Dim arr, rngs As Range, n As Long
    Set rngs = Application.Intersect(ActiveSheet.UsedRange, Selection) '取得选区与已用数据区域的交集
    With rngs
      arr = rngs '将区域读入数组
      Do While n < .Count '只要变量 n 小于数组中元素个数就一直循环下去
        If arr(n + 1, 1) = "" Then arr(n + 1, 1) = arr(n, 1) '如果数组中某个值为空文本则将其赋值是前一个非空文本
        n = n + 1 '累加变量
      End Do
    End With
  End Sub
End Sub
```

```

Loop
    .UnMerge '取消带有合并单元格的区域的合并属性
    .Value = arr '将数组的值赋与单元格
End With
rngs(1).Select '选择原选区中第一个单元格
End Sub

```

以上过程的思路为：

- ◆ 将选区与已用区域的交集赋值给数组。
- ◆ 利用 Do While 语句在交集中循环，每经过一个单元格则对变量 n 加 1。如果循环到数组中某个值为空值，则对它赋值为前一个值，直到数组中最后一个。
- ◆ 数组中的空值填充完毕后，将它赋值到区域中。

功能测试：

- step 1** 打开图 30.44，进入记录省市名称的工作表。
- step 2** 选择 A 列后执行过程“合并列中相同数据”，列中所有相同且相邻的单元格合并后效果如图 30.45 所示。
- step 3** 选择 A 列后执行过程“取消列中合并，还原合并数据”，那么 A 列将返回合并前的状态。
- step 4** 合并列中相同数据也可以应用到其他的很多行业中，例如图 30.46 所示的季度合并。

	A	B
1	省	市
2	山东省	枣庄市
3	山东省	淄博市
4	山东省	烟台市
5	辽宁省	铁岭市
6	辽宁省	营口市
7	辽宁省	沈阳市
8	辽宁省	辽阳市
9	福建省	三明市
10	福建省	厦门市
11	福建省	漳州市

图 30.44 合并前

	A	B
1	省	市
2	山东省	枣庄市
3	山东省	淄博市
4	山东省	烟台市
5	辽宁省	铁岭市
6	辽宁省	营口市
7	辽宁省	沈阳市
8	辽宁省	辽阳市
9	福建省	三明市
10	福建省	厦门市
11	福建省	漳州市

图 30.45 合并后

	A	B
1	时间	产量
2	第一季度	642
3	第一季度	603
4	第一季度	565
5	第二季度	667
6	第二季度	604
7	第二季度	755
8	第三季度	644
9	第三季度	649
10	第三季度	643

图 30.46 按季度合并

注意事项

- (1) 合并和取消合并都适用于非公式，对于公式它会转换成值。
- (2) 取消合并后如果需要每个单元格都显示边框，可以在倒数第二句前加一名代码：

```
rngs.Borders.LineStyle = xlContinuous
```

30.4 批注工具箱

批注通常用于对某个特殊的数据进行补充、说明，在工作中应用机率极高。而 Excel 自身的

批注工具功能有限,无法满足工作要求。百宝箱中集成的批注工具箱可以强化在批注方面的功能。

30.4.1 批注管理器

案例要求: 批注管理器具有四方面的功能: 可以查看前一条、下一条批注; 可以在管理器中修改、删除批注内容; 可以在管理器中显示每一批注的地址、可以批量替换批注中任意内容, 替换时支持是否区分大小写, 以及是否需要全字匹配等自定义设置。

实现步骤:

- step 1** 插入一个窗体, 将其“名称”和“Caption”属性均修改为“批注工具”。
- step 2** 在窗体中添加一个多页控件, 在第一页中添加五个标签、三个文字框、四个命令按钮, 其控件布局方式如图 30.47 所示。其中四个文字框的名称分别为“总数”、“当前”、“位于”和“内容”。
- step 3** 进入第二页, 添加三个标签, 两个文字框、两个复选框和一个命令按钮, 其布局方式如图 30.48 所示。其中两个文字框的名称分别为“原文本”和“新文本”, 两个复选框的名称分别为“区分大小写”和“完全匹配”。



图 30.47 批注工具第一页的控件布局方式

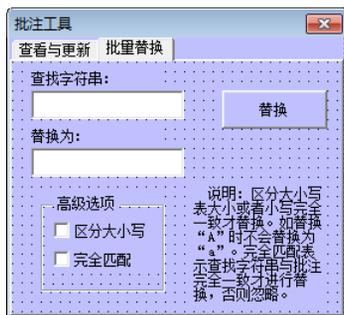


图 30.48 批注工具第二页的控件布局方式

- step 4** 双击窗体进入代码窗口, 并录入以下过程代码:

```
Private Sub UserForm_Activate()
    If ActiveSheet.Comments.Count = 0 Then MsgBox "当前表没有批注。": Unload Me: Exit Sub
    总数 = ActiveSheet.Comments.Count '获取当前表批注数量
    当前 = IIf(总数 = 0, "", 1)
    当前.Enabled = False '禁用“当前”按钮
    上一条.Enabled = False '禁用“下一条”按钮
    If ActiveSheet.Comments.Count < 2 Then 下一条.Enabled = False '如果没有批注则禁用“下一条”按钮
    位于 = ActiveSheet.Comments(1).Parent.Address(0, 0) '记录批注地址
    内容 = VBA.Replace(ActiveSheet.Comments(1).Text, Chr(10), " ") '记录批注内容
End Sub
```

以上代码在激活窗体时执行, 过程的思路为:

- ◆ 如果工作表中不存在批注，那么关闭窗口。
- ◆ 在窗体中显示工作表中的批注个数，以及第一条批注的地址、内容。
- ◆ 为了防错，根据批注的个数决定“上一条”和“下一条”按钮是否允许使用。

```
Private Sub 总数_Change()
    If 总数 = 0 Then 删除.Enabled = False Else 删除.Enabled = True
End Sub
```

以上过程表示“总数”标签的值变化时，根据批注个数决定“上一条”和“下一条”按钮是否允许使用。

```
Private Sub 当前_Change()
    On Error Resume Next
    If Me.当前 <= 1 Then 上一条.Enabled = False Else 上一条.Enabled = True '如果“当前”小于等于 1 则“上一条”按钮不可用，否则可用
    If Me.当前.Value * 1 >= 总数.Value * 1 Then 下一条.Enabled = False Else 下一条.Enabled = True '如果“当前”大于等于总数则“下一条”按钮不可用，否则可用
    If 当前 = 0 Then Me.位于 = "": 内容 = "" '如果“当前”等于 0，则“位于”按钮和“内容”按钮显示空文本
End Sub
```

以上过程仍然是根据批注总数和当前显示的批注个数来决定“上一条”和“下一条”按钮是否允许使用。

```
Private Sub 更新_Click() '
    If 内容 = "" Or 总数 = 0 Then Exit Sub '如果“内容”和“总数”按钮无数据则退出程序
    ActiveSheet.Comments(当前.Value).Parent.Comment.Text Text:=VBA.Replace(内容.Text, " ", Chr(10)) '更新批注，将“内容”中的字符导入到单元格
End Sub
```

以上过程表示未输入内容或者是工作表中无批注时禁用使用“更新”按钮；否则将“内容”文字框中的字符串代替当前批注。

```
Private Sub 上一条_Click()
    Me.当前 = 当前.Value - 1
    内容 = VBA.Replace(ActiveSheet.Comments(当前.Value).Text, Chr(10), " ") '显示“当前”按钮指定的批注（将回车符替换成空格）
    删除.Enabled = True
    位于 = ActiveSheet.Comments(当前.Value).Parent.Address(0, 0) '提取批注的地址
End Sub
```

以上过程表示单击“上一条”按钮时更新“内容”和“当前”、“位于”三个文字框的显示值，且让“删除”按钮可用。

```
Private Sub 下一条_Click()
    Me.当前 = 当前.Value + 1 '累加“当前”按钮的值
    内容 = VBA.Replace(ActiveSheet.Comments(当前.Value).Text, Chr(10), " ") '更
```

新内容

```
删除.Enabled = True  
位于 = ActiveSheet.Comments(当前.Value).Parent.Address(0, 0) '提取批注的地址  
End Sub
```

以上过程表示单击“下一条”按钮时更新“内容”和“当前”、“位于”三个文字框的显示值，且让“删除”按钮可用。

```
Private Sub 删除_Click()  
    ActiveSheet.Comments(当前.Value).Parent.ClearComments '清除批注  
    总数 = 总数 - 1 '“总数”按钮减1  
    Me.当前 = 当前.Value - 1 '“当前”按钮也减1  
    If 当前 = 0 Then '如果“当前”等于0则“删除”按钮不可用，“上一条”也不可  
        删除.Enabled = False  
        上一条.Enabled = False  
    Else  
        删除.Enabled = True '“删除”按钮可用  
    End If  
End Sub
```

以上过程表示单击“删除”按钮时，将当前查看的批注删除，同时更新“总数”、“当前”文字框的值，而“上一条”和“删除”按钮则根据“当前”文字框的值来决定是否可用。

```
Private Sub CommandButton1_Click()  
    Dim AA As String, bb As String  
    Application.ScreenUpdating = False  
    For i = 1 To ActiveSheet.Comments.Count '遍历批注  
        Cells(i, Columns.Count) = ActiveSheet.Comments(i).Text '将批注导出到最后一  
        列的单元格中  
    Next  
    If 完全匹配 Then '如果选择“完全匹配”，则以完全匹配模式替换字符串  
        Cells(1, Columns.Count).EntireColumn.Replace What:=原文本, Replacement:=  
        新文本, LookAt:=xlWhole, SearchOrder:=xlByRows, MatchCase:=区分大小写,  
        SearchFormat:=False, ReplaceFormat:=False  
    Else '否则以不完全匹配方式替换字符串  
        Cells(1, Columns.Count).EntireColumn.Replace What:=原文本, Replacement:=  
        新文本, LookAt:=xlPart, SearchOrder:=xlByRows, MatchCase:=区分大小写,  
        SearchFormat:=False, ReplaceFormat:=False  
    End If  
    For i = 1 To ActiveSheet.Comments.Count '再将字符还原到批注中  
        ActiveSheet.Comments(i).Parent.Comment.Text Text:=Cells(i,  
        Columns.Count).Text  
    Next  
    Cells(1, Columns.Count).EntireColumn.Clear '清除辅助区数据  
    Application.ScreenUpdating = True  
End Sub
```

以上代码的思路为：

- ◆ 利用 For 循环提取工作表中的所有批注到最后一列中。
- ◆ 直接对该辅助列进行替换。
- ◆ 将替换后的字符串再导入到对应的批注中，并删除辅助列。

插入一个标准模块，将其名称修改为“批注模块”，并录入以下代码调用窗体：

```
Sub 批注管理器 ()
批注管理.Show 0
End Sub
```

功能测试：

- step 1** 打开图 30.49 所示成绩表，其中包含 9 条批注。
- step 2** 执行过程“批注管理器”打开“批注工具”窗体，在第一页中将显示当前工作表功能测试中批注数量，以及第一条批注的内容和地址，如图 30.50 所示。
- step 3** 单击“下一条”按钮，窗体中将显示第二个批注的内容。
- step 4** 在内容批注框中输入新的内容后单击“更新”按钮，录入的字符将替代对应单元格的批注。

	A	B	C	D	E
1	姓名	成绩	姓名		
2	赵	55	不及格，补考		
3	钱	65	陈	84	
4	孙	100	褚	62	
5	李	93	卫	42	
6	周	100	蒋	98	
7	吴	62	沈	77	
8	郑	59	韩	49	
9	王	54	杨	46	

图 30.49 成绩表



图 30.50 利用批注管理器逐条查看批注

- step 5** 进入“批量替换”页，在查找文字框中录入“补考”，在存放替换文本的文字框中录入“仅颁发结业证”，“高级选项”保持不变，因为“区分大小写”选项仅用于英文，“完全匹配”表示查找字符与批注全部一致时才替换。单击“替换”按钮后，所有批注的“补考”二字都会替换成新的字符串，如图 30.52 所示。

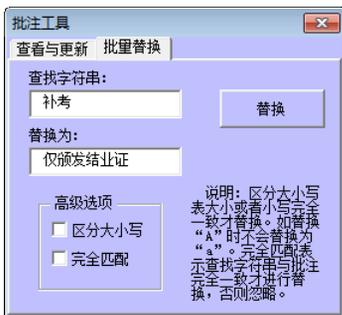


图 30.51 设置替换选项

	A	B	C	D	E
1	姓名	成绩	姓名		
2	赵	55	不及格，仅发结业证		
3	钱	65	陈	84	
4	孙	100	褚	62	
5	李	93	卫	42	
6	周	100	蒋	98	
7	吴	62	沈	77	
8	郑	59	韩	49	
9	王	54	杨	46	

图 30.52 批量替换结果

注意事项:

(1) 如果调用不存在的批注会产生错误,所以代码中需要严格限制“上一条”和“下一条”的 Enabled 属性。

(2) 逐条替换批注在操作上不如在单元格中替换字符串方便,所以将所有批注导入到辅助单元格中再执行替换会更好操作,替换后直接导入到批注框即可。

30.4.2 添加个性化批注

案例要求: Excel 添加的批注永远都是方块形,本例可以完成个性化的批注,产生 22 种批注外形。

实现步骤:

在“批注模块”中输入以下过程代码:

```
Sub 生成个性化批注()  
    If ActiveSheet.ProtectContents Then MsgBox "工作表已保护,本程序拒绝执行!", 64,  
        "Excel 百宝箱": Exit Sub  
    Dim mystr As String, mystr2 As String, Comment  
    ActiveCell.ClearComments '清除当前单元格批注  
    On Error GoTo err  
    mystr = InputBox("输入批注内容", "批注", Application.UserName, 10, 10) '确定  
    批注内容  
    '选择批注外观样式,提供 22 种外观可选  
    mystr2 = Application.InputBox("输入批注外形" & Chr(10) & "1 口哨型,2 书卷型,3 箭  
    头型,4 圆角矩形" & Chr(10) & "5 缺角矩形,6 菱形,7 五角星,8 云形标注,9 圆形,10 六边形,11  
    八边形,12 柱形,13 笑脸形,14 心形,15 八角星,16 横卷形,17 竖卷形,18 波形,19 双波形,20 十六角  
    星,21 二十四角星,22 文档.", "批注外型", 1, 10, 10, , , 1)  
    If mystr <> "" And mystr2 > 0 Then  
        Set Comment = ActiveCell.AddComment  
        With Comment  
            .Visible = False '不可见  
            .Text Text:=Replace(mystr, "chr(10)", Chr(10)) '设置批注文字  
            .Visible = True '可见  
            .Shape.Select True '选择批注,然后根据用户输入的值设置批注的外观  
            Selection.ShapeRange.AutoShapeType = Choose(mystr2, msoShapeFlowchart-  
            SequentialAccessStorage, msoShapeFoldedCorner, msoShapeRightArrow, msoShape-  
            RoundedRectangularCallout, msoShapePlaque, msoShapeDiamond, msoShape5point-  
            Star, msoShapeCloudCallout, msoShapeOval, msoShapeHexagon, msoShapeOctagon,  
            msoShapeCan, msoShapeSmileyFace, msoShapeHeart, msoShape8pointStar, msoShape-  
            HorizontalScroll, msoShapeVerticalScroll, msoShapeWave, msoShapeDoubleWave,  
            msoShape16pointStar, msoShape24pointStar, msoShapeFlowchartDocument)  
            .Visible = False  
        End With  
    AA:  
        ActiveCell.Select '选择单元格(否则当前选择对象是批注)  
    End If  
err:  
End Sub
```

以上代码的思路为：

- ◆ 清除活动单元格的原有批注。
- ◆ 弹出对话框让用户录入批注内容，可以使用 chr(10)来换行。
- ◆ 再弹出对话框让用户选择批注的外观。
- ◆ 根据两次的输入值在活动单元格产生对应的外观。

功能测试：

- step 1** 执行过程“生成个性化批注”，将弹出如图 30.53 所示对话框。
- step 2** 在对话框中录入字符串“停电 1 小时 chr(10)产量偏低”，并单击“确定”按钮。
- step 3** 在图 30.54 所示对话框中录入 6,单击“确定”按钮后工作表中批注外观如图 30.55 所示。如果批注内容中没有“chr(10)”，且外观选择 9，那么其结果如图 30.56 所示。



图 30.53 录入批注内容

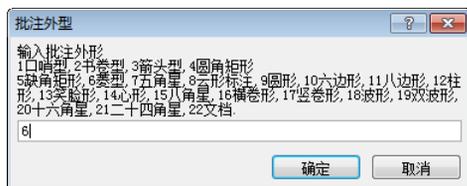


图 30.54 选择外观



图 30.55 菱形且强制换行批注



图 30.56 圆形不强强制换行批注

注意事项：

- (1) 设置批注的外观前必须使批注可见，且需要选择批注框，否则无法修改其外形。
- (2) 本例中提供了 22 种外形可选，读者可以加入更多的批注外形。可以通过查找“MsoAutoShapeType 枚举”帮助来获取其他可用的外形所对应的内置常量。
- (3) 在第 10 章中也有个性化批注的代码，在设计插件时，应尽量使插件更完善。本例相对于第 10 章的“添加个性批注”在两方面进行了完善：防错及添加换行符。

30.4.3 建立图片批注

案例要求：手工将一张图片插入到批注中是比较烦琐的。本例利用 VBA 操作可以快速完成，支持多种格式的图片。

实现步骤:

step 1 “在批注模块”中录入以下过程代码:

```
Sub 设置批注图片背景()  
    If ActiveSheet.ProtectContents Then MsgBox "工作表已保护,本程序拒绝执行!", 64,  
"Excel 百宝箱": Exit Sub  
    Const ImgFileformat = "Image Files (*.bmp;*.gif;*.tif;*.jpg;*.jpeg)," &  
"*bmp;*.gif;*.tif;*.jpg;*.jpeg"  
    Dim Com, pic As String, Sizes As String  
    Set Com = ActiveCell.Comment  
    If Not Com Is Nothing Then ActiveCell.Comment.Delete '如果有批注就删除它  
    pic = Application.GetOpenFilename(ImgFileformat) '打开一个图片选择对话框  
    If pic = "False" Then End '如果选择取消则退出程序  
Star:  
    Sizes = Application.InputBox("请指定图片的高度与宽度,中间有逗号隔开" & Chr(10) &  
"例如“300,260”或者“80,100”", "指定批注的大小", "100,120", , , , 2)  
    If InStr(Replace(Sizes, ",", "")) = 0 Then GoTo Star '如果没输入“,”  
则返回重新输入  
    '如果“,”前后都是数值则添加图片批注,否则返回重新输入  
    If IsNumeric(Split(Replace(Sizes, ",", ""), ",")(1)) And  
IsNumeric(Split(Replace(Sizes, ",", ""), ",")(2)) Then  
        With ActiveCell  
            .AddComment '添加批注  
            .Comment.Visible = True '可见  
            .Comment.Shape.Fill.UserPicture pic '设置图片填充  
            .Comment.Shape.Select True  
            .Comment.Shape.Height = Split(Replace(Sizes, ",", ""), ",")(0) '自  
定义的高度  
            .Comment.Shape.Width = Split(Replace(Sizes, ",", ""), ",")(1) '自定  
义的宽度  
            .Comment.Text Text:="" '用空格做为批注内容  
            .Comment.Visible = True '不可见  
        End With  
    Else  
        GoTo Star  
    End If  
    ActiveCell.Select '选择单元格  
End Sub
```

以上代码的思路为:

- ◆ 清除原有批注。
- ◆ 弹出对话框让用户选择图片,如果未选择图片则直接退出。
- ◆ 再弹出对话框让用户设定图片的高度与宽度,如果未指定或者录入不规范则返回重新录入。
- ◆ 插入一个窗体,将其批注背景设置为指定的图片,且需要设置高度与宽度。

功能测试：

- step 1** 执行过程“设置批注图片背景”，将弹出“打开”对话框，用户可以随意选择“*.bmp”、“*.gif”、“*.tif”、“*.jpg”、“*.png”五种格式之一。
- step 2** 在对话框中选择一幅图片并双击打开，程序会弹出对话框让用户指定图片的高度与宽度。在对话框中保持默认值不变，表示高度 100、宽度 120。
- step 3** 当单击“确定”按钮后，在活动单元格将产生以该图片为背景的批注，如图 30.58 所示。可以右键编辑批注、随意调整图片的高度与宽度。



图 30.57 指定批注的高寒度与宽度

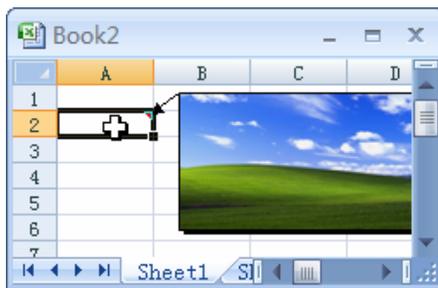


图 30.58 插入的图片批注

注意事项：

- (1) 在第 11 章也有一个设计图片批注的案例，但应用到插件中时不可照搬那段代码，而是需要有完善的防错机制，以及足够的自定义功能。这也是开发插件与编写普通过程的区别。
- (2) 为了防止用户录入全角状态的逗号造成数组函数 Split 无法识别，需要在代码中将全角转换成半角。

30.4.4 批量添加右列内容为批注

案例要求：为选择的区域批量添加批注，批注来自它左边或者右边的 N 列的内容，同时 N 可以随意定义。

实现步骤：

在本书第 11 章中有现成的代码，其章节与标题为“11.4.1 批量将数据导入批注”，直接复制该过程代码即可。该过程在设计上比较完善，不需要做修改。

30.4.5 批量导入同名照片到批注

案例要求：根据选区的字符，将同名照片批量地导入到选中的每一个单元格的批注中。而且照片的格式、宽度、高度可以定义。

实现步骤：

在“批注模块”中录入以下过程代码：

```
Sub 批量导入图片到批注() '批量导入同名图片到批注中
    Dim cell As Range, fd As FileDialog, 路径 As String, 宽 As Integer, 高 As Integer, rng As Range, TypeName As Integer, str As String
```

```
Restar:
    TypeName = Application.InputBox("输入 1: 插入 GIF 图片; " + Chr(10) + "输入 2: 插入 PNG 图片; " + Chr(10) + "输入 3: 插入 JPG 图片。" + Chr(10) + "输入 4: 插入 JPEG 图片。", "图片格式", 3, , , , 1)
    If TypeName < 1 Or TypeName > 4 Then MsgBox "输入错误": GoTo Restar
    str = VBA.Choose(TypeName, "*.GIF", "*.PNG", "*.JPG", "*.JPEG") '根据输入的数字决定选用图片格式
    Set rng = Application.Intersect(ActiveSheet.UsedRange, Selection)
    Set fd = Application.FileDialog(msoFileDialogFolderPicker)
    If fd.Show = -1 Then 路径 = fd.SelectedItems(1) Else Exit Sub '如果选择了路径提取路径全名否则退出程序
    If rng(1) = "" Then MsgBox "不能选择空白区。", 64, "提示": Exit Sub
    On Error Resume Next
    rng.ClearComments '清除选区的所有批注
    err.Clear
Star:
    宽 = Application.InputBox("您希望插入的图片显示多宽? " & Chr(10) & "Excel 默认宽度为 3.39, 你可以输入 1-15 之间的数据。" & Chr(10) & "小于 1 时当做 1 计算。", "确认宽度", 3.39, , , , 2)
    高 = Application.InputBox("您希望插入的图片显示多高? " & Chr(10) & "Excel 默认高度为 2.09, 你可以输入 1-15 之间的数据。" & Chr(10) & "小于 1 时当做 1 计算。", "确认高度", 2.09, , , , 2)
    If err <> 0 Then err.Clear: MsgBox "输入不规划, 请重新输入": GoTo Star
    If 宽 < 1 Or 高 < 1 Then 宽 = 3.39: 高 = 2.09
    If 宽 > 15 Or 高 > 15 Then MsgBox "原则上你的图片可以显示这么大, " & Chr(10) & "不过太大会有负面影响? 请重新输入 1-15 之间的数", 64, "提示": GoTo Star
    Application.ScreenUpdating = False
    For Each cell In rng '遍历选区
        If Dir(路径 & "\" & cell.Text & Mid(str, 2)) <> "" Then
            With cell.AddComment '遍历批注
                .Visible = True '可见性为 TRUE
                .Text Text:="" '不显示文本
                .Shape.Select True '选择批注
            With Selection.ShapeRange
                .Fill.UserPicture 路径 & "\" & cell.Text & Mid(str, 2) '填充批注
                .ScaleWidth 宽 / 3.39, msoFalse, msoScaleFromTopLeft '设置宽度
                .ScaleHeight 高 / 2.09, msoFalse, msoScaleFromTopLeft '设置高度
            End With
            cell.Offset(1, 0).Select '选择单元格
            .Visible = False '批注不可见
        End With
        GoTo nexts '运行 a 标签之语句
    End If
    MsgBox "未找到" & cell.Text & Mid(str, 2), 64, "提示"
    ActiveCell.ClearComments '清除批注
    cell.Offset(1, 0).Select '选择下一单元格
nexts:
Next
```

```
Application.ScreenUpdating = True
End Sub
```

以上代码的思路为：

- ◆ 首先，弹出对话框让用户选择图片的格式，包括 GIF、PNG、JPG 和 tif 四种格式。
- ◆ 再次弹出对话框让用户选择存放图片的文件夹，未选择文件夹则退出程序。
- ◆ 清除活动单元格的原有批注。
- ◆ 弹出两个对话框让用户指定批注的高度与宽度，如果输入不规范将返回重新输入。
- ◆ 最后遍历选中区中每一个单元格，将与单元格同名的图片引入每个单元格的批注中，且按用户需求设置其高度与宽度。

功能测试：

- step 1** 在磁盘中存放一些图片，如图 30.59 所示。
- step 2** 在工作表中 A 列录入同图片名称，即职工姓，然后选择姓名区域再执行过程“批量导入图片到批注”。
- step 3** 弹出的对话框保持默认格式的编号 3，表示导入 JPG 图片，如图 30.60 所示。



图 30.59 硬盘中的职工照片

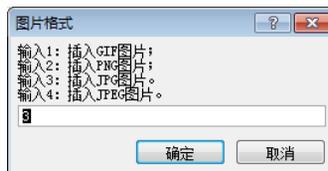


图 30.60 选择图片格式

- step 4** 单击“确定”按钮后将弹出如图 30.61 和图 30.62 所示对话框，保持默认值即可，表示批注的宽度与高度以批注的原来宽度与高度为准：

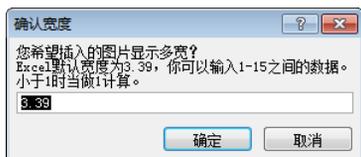


图 30.61 设置批注的宽度

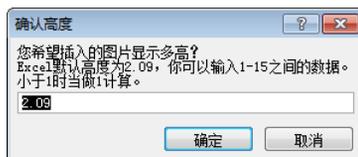


图 30.62 设置批注的高度

- step 5** 设置完所有选项后，执行结果如图 30.63 所示。如果与某单元格中字符串相同的图片未找到，将弹出如图 30.64 所示提示。

注意事项:

(1) 文件名称不区分大小写, 所以指定后缀名时可以仅仅采用大写格式。

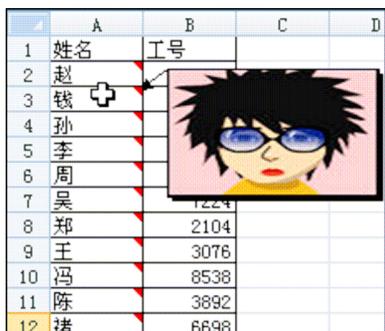


图 30.63 批量导入图片批注的效果



图 30.64 未找到图片之提示

(2) 导入图片到批注必须用 DIR 测试文件是否存在, 否则程序会中断。

(3) 插入批注不触发工作表事件, 那么仅仅需要修改 ScreenUpdating 属性来提速, 而不需要调整计算模式。



本例所用测试文件参见光盘: ..\ 第三十章\PIC\

30.5 图片工具箱

Excel 可以导入图片, 却无法批量调整图片的位置、大小, 对于图片批量导出就更无能为力了。“图片工具箱”可以实现图片批量导出、导入以及将单元格转换成图片。

30.5.1 将选区转换成图片

案例要求: 将工作表中选择区域转换成图片保存在硬盘中, 图片格式可选。

实现步骤:

step 1 插入一个窗体, 将其“Caption”修改为“输出到硬盘”, 将“名称”属性修改为“选区到图片”。

step 2 在窗体中添加三个标签, 两个文字框, 一个复合框及两个按钮, 并按图 30.65 所示布局。

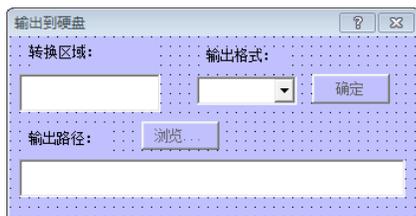


图 30.65 选区到图片窗体的控件布局方式

step 3 双击窗体打开代码窗口, 并在窗口中录入以下代码:

```

Private Sub UserForm_Activate() '激活窗体时执行
    If TypeName(Selection) <> "Range" Then MsgBox "必须选择单元格!", 64, "Excel 百宝箱": Unload Me: Exit Sub
    If Selection.Areas.Count > 1 Then MsgBox "只能选择一个区域!", 64, "Excel 百宝箱": Unload Me: Exit Sub
    If Selection.Rows.Count > 1000 Or Selection.Columns.Count > 1000 Then MsgBox "范围太大, 生成的图片会较模糊! 请重新选择", 64, "Excel 百宝箱": Unload Me: Exit Sub
    TextBox1.Text = Selection.Address '将选区的地址存入文字框
    ComboBox1.List = Array("JPG", "GIF", "BMP") '在列表框显示图片类型
    ComboBox1.Value = "BMP" '设置默认类型
    TextBox2 = CreateObject("WScript.Shell").SpecialFolders("Desktop") '设置默认路径
End Sub

```

以上过程代码表示激活窗体时在复合框中显示可选的三种图片格式，并在路径文字框中显示默认的路径——桌面。

```

Private Sub CommandButton1_Click()
    Dim fd As FileDialog
    Set fd = Application.FileDialog(msoFileDialogFolderPicker)
    If fd.Show = -1 Then TextBox1 = fd.SelectedItems(1) '将选择的文件夹地址写入文字框
    Set fd = Nothing
End Sub

```

以上过程表示从“打开”对话框中选择图片存放路径，并将该路径保存在文字框中。

```

Private Sub CommandButton2_Click()
    If Dir(Me.TextBox2.Text, vbDirectory) = "" Then Exit Sub
    Selection.CopyPicture Appearance:=xlScreen, Format:=xlBitmap '将选区复制成图片
    ActiveSheet.Pictures.Paste.Select '选择图片
    With Selection
        .Copy '复制图片
        With ActiveSheet.ChartObjects.add(0, 0, Selection.Width, Selection.Height).Chart '建立一个图表, 其高、宽等于图片的高与宽
            .Paste '粘贴图片到图表
            '将图表转成图片导出到设置的文件夹中
            .Export TextBox2.Text & "\" & VBA.Replace(VBA.Replace(TextBox1.Text, ":", "-"), "$", "") & "." & ComboBox1.Value
            .Parent.Delete '删除图表
        End With
        .Delete '删除图片
    End With
    Shell "explorer.exe " & TextBox2, vbMaximizedFocus '打开文件夹
    Unload Me '关闭窗口体
End Sub

```

以上代码的思路为:

- ◆ 如果指定的路径不存在, 那么退出程序。
- ◆ 将选择的区域复制为图片。
- ◆ 创建一个空图表, 且其大小与选区一致。再将剪贴板中的图片粘贴到图表中, 利用 Export 方法将图表导出为图片, 保存到指定的文件夹下, 将图片按区域地址命名。

step 4 插入一个模块, 并将其命名为“图片模块”, 并在模块中录入以下过程。

```
Sub 将选区另存为图片()  
    选区到图片.Show  
End Sub
```

功能测试:

step 1 选择如图 30.66 所示工作表中的 A2:B3 区域, 执行过程“将选区另存为图片”。

step 2 将窗体中的所有选项保存默认状态, 单击“确定”按钮。程序将会把选区导出到桌面。导出的图片效果如图 30.67 所示。



图 30.66 待导出的选区



图 30.67 导出的图片

注意事项:

(1) Excel 本身只能将选区转换成图片, 不可另存到硬盘中。但图表对象的 Export 方法可将图片导出到硬盘, 所以取巧的方式是将图片粘贴到图表中, 再将图表导出成图片。

(2) 图片的可选格式虽然提供了三种, 但只有 BMP 格式最清晰。

30.5.2 批量导入图片

案例要求: 根据选区的字符, 将包含单元格字符的照片批量地导入到选择中的每一个单元格右边的列中, 如果有多个图片符合条件, 则全部罗列出来。照片的格式、高度可以定义。

实现步骤:

在“图片”模块中录入以下过程代码:

```

Sub 批量导入图片到单元格()
    If ActiveSheet.ProtectContents Then MsgBox "工作表已保护,本程序拒绝执行!", 64,
    "提示": Exit Sub
    Dim fd As FileDialog, Paths As String, MyPath As String, MyFile$, cell As Range,
    rng As Range, row As Integer, JpgHeight As Integer, k As Byte, TypeName As Integer,
    str As String
    Set rng = Selection(1)
    If rng = "" Then MsgBox "不能选择空白区。", 64, "Excel 百宝箱": Exit Sub
    Set fd = Application.FileDialog(msoFileDialogFolderPicker)
    If fd.Show = -1 Then Paths = fd.SelectedItems(1) Else Exit Sub '如果选择
    了文件夹就返回文件夹地址,否则退出程序
    On Error Resume Next
    '确定插入的图片类型,有四种可选,GIF PNG JPG 和 JPEG
Star:
    TypeName = Application.InputBox("输入 1: 插入 GIF 图片;" + Chr(10) + "输入 2:
    插入 PNG 图片;" + Chr(10) + "输入 3: 插入 JPG 图片。" + Chr(10) + "输入 4: 插入 JPEG
    图片。", "图片格式", 3, , , , 1)
    If TypeName < 1 Or TypeName > 4 Then MsgBox "输入错误": GoTo Star
    str = VBA.Choose(TypeName, "*.GIF", "*.PNG", "*.JPG", "*.JPEG") '根据输入的
    数字决定选用图片格式
    row = ActiveCell.RowHeight
    JpgHeight = Application.InputBox("请输入你希望照片显示的高度(10-200 之间)," &
    Chr(10) & "当前单元格高度为" & row & ", 请根据需要按比例输入新高度。" & Chr(10) & "
    程序将把单元格的高度与宽度调整为同样大小。", "【确认照片高度】", row, , , , 1)
    If JpgHeight < 10 Then JpgHeight = 10 '限制高度在 10 到 100 之间
    If JpgHeight > 200 Then JpgHeight = 200
    Selection.RowHeight = JpgHeight '将单元格的高度设置为指定的图片高度
    Application.ScreenUpdating = False
    For Each cell In Selection
        If cell = "" Then GoTo A:
        MyFile = Dir(Paths & "\" & cell.Text & str) '获取文件
        While Len(MyFile) > 0 '如果文件存在
            ActiveSheet.Pictures.Insert(Paths & "\" & MyFile).Select '插入图片并选择
            图片
            With Selection.ShapeRange
                .Top = cell.Offset(0, k + 1).Top '设置图片上边距
                .Left = cell.Offset(0, k + 1).Left '设置左边距
                .Height = cell.Offset(0, k + 1).Height '设置高度
                '如果图片的宽度大于列宽,那么将列宽设置为图片宽度
                If (.Width * 4 / 3 - 5) / 8 > cell.Offset(0, k + 1).ColumnWidth Then
                cell.Offset(0, k + 1).ColumnWidth = (.Width * 4 / 3 - 5) / 8 '设置单元格列
                宽,让它等于图片的宽度(默认字体为 12 时才能用)
            End With
            Selection.Name = Left(MyFile, Len(MyFile) - 4) '指定图片在工作表中的名称
            MyFile = Dir
            k = k + 1
        Wend
        k = 0
    A:
    Next cell

```

```
Application.ScreenUpdating = True
rng.Select '选择单元格
End Sub
```

以上代码在激活窗体时执行，过程的思路为：

- ◆ 首先，弹出选择文件夹的对话框让用户选择图片路径，如果未选择对象则退出。
- ◆ 然后，再让用户确定图片格式，可以是 GIF、PNG、JPG 和 tif 四种格式之一。
- ◆ 再让用户确定图片的显示高度，只能在 10 到 100 之间。
- ◆ 利用 For 循环根据单元格的值插入图片，同时设定单元格的高度与宽度，且让图片的高度与宽度等于单元格的高度与宽度。如果同一列中不同图片的宽度不同，则以最宽者为准。
- ◆ 最后，将每个导入的图片按其原来的文件名进行命名。

功能测试：

假设在文件夹中有所有职工的照片，每个照片以职工姓名命名。而工作表中有所有职工的姓名，需要根据姓名导入照片。那么具体步骤如下：

- step 1** 选择 A 列的职工姓名区域“A2:A21”，执行过程“批量导入图片到单元格”。
- step 2** 将弹出一个选择图片存放路径的对话框，如果没选择则退出程序。
- step 3** 接着弹出如图 30.68 所示对话框，让用户选择图片的格式，单击“确定”按钮保持默认格式。
- step 4** 下一个对话框用于指定图片的高度，默认为活动单元格的高度，如图 30.69 所示。



图 30.68 选择图片格式

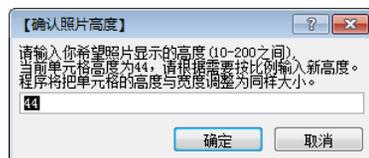


图 30.69 确定照片的高度

- step 5** 当单击对话框中的“确定”按钮后，在 B 列插入每个职工的照片，且照片等于单元格的高度。如果未找到某个职工的照片或者某单元格中空白则会跳过。导入结果如图 30.70 所示。
- step 6** 如果在对话框中将图片的高度设定为 40，那么其效果将如图 30.71 所示。

冯丽鹃			
	A	B	C
1	姓名	照片	
2	陈芳		
3	褚华		
4	冯丽鹃		
5	冯云		
6	李靖		
7	李师师		
8	钱丽		
9	钱秦		
10	秦通丽		

图 30.70 图片高度为 14 时的效果

褚华			
	A	B	C
1	姓名	照片	
2	陈芳		
3	褚华		
4	冯丽鹃		
5	冯云		

图 30.71 图片高度为 40 时的效果

step 7 本工具最主要的用途是创建产品目录。例如选择产品名称，批量导入该类型的产品图片，包括该类所有产品。例如图 30.72 中，A2 为“联想”，那么所在名为“联想 A”、“联想 B”或者“联想 E216”等牌子的产品照片都可以罗列出来。



图 30.72 罗列产品图片

注意事项:

(1) 表达式“对象.Width * 4 / 3 - 5) / 8”表示将以磅为单位的图形对象宽度转换为单元格列宽，一个列宽单位等于“常规”样式中一个字符的宽度。以上表达式仅仅在于工作表默认字号为 12 时可用，如果默认字号不是 12 将产生偏差。

(2) 如果不需要模糊匹配，即单元格的字符完全与图片名称匹配才导入图片，那么可以将格式中的“*”去掉，While 循环也可以去掉。



本例所用测试文件参见光盘：..\第三十章\批量导入图片\

30.5.3 批量导出图片到硬盘

案例要求：将当前工作表中的所有图形对象与图片导出到硬盘中，保存为图片。

实现步骤：

step 1 在“图片模块”中录入以下过程代码：

```
Sub 将当前表中的图形对象另存为图片文件() '将所有图形对象另存到本工作簿同路径下，成为图片文件
    If ActiveSheet.ProtectContents Then MsgBox "工作表已保护,本程序拒绝执行!", 64, "提示": Exit Sub
    Dim i As Byte, Paths As String, fd As FileDialog
    Set fd = Application.FileDialog(msoFileDialogFolderPicker)
    If fd.Show = -1 Then Paths = fd.SelectedItems(1) Else Exit Sub
    Paths = Paths & IIf(Right(Paths, 1) = "\", "", "\")
    If ActiveSheet.Shapes.Count = 0 Then Exit Sub
    For i = 1 To ActiveSheet.Shapes.Count '遍历图形对象
        ActiveSheet.Shapes(i).Select '选择图片
        If TypeName(Selection) = "ChartObject" Then '如果是图表
```

```
Selection.Chart.Export Paths & i & ".jpg" '导出为图片
Else
    Selection.Copy '复制图形对象
    With ActiveSheet.ChartObjects.add(0, 0, Selection.Width,
Selection.Height).Chart '添加图表
        .Paste '粘贴图片
        .Export Paths & "导出的图片" & i & ".jpg" '导出图片
        .Parent.Delete '删除图表
    End With
End If
Next i
Shell "EXPLORER.EXE " & Paths, vbNormalFocus '打开文件夹
End
End Sub
```

以上代码在激活窗体时执行，过程的思路为：

- ◆ 首先，弹出对话框让用户选择导出的图片需要保存在哪个文件夹中。
- ◆ 然后，利用 For 循环逐个复制图形对象，将其粘贴到新建的图表中，并将图片导出为图片文件。
- ◆ 最后删除表并打开存放图片文件的文件夹。

功能测试：

- step 1** 打开带有图片、艺术字和形状的（2003 中称自选图形）工作表，如图 30.73 所示。
- step 2** 执行过程“将当前表中的图形对象另存为图片文件”，将弹出“浏览”对话框，选择 D 盘后单击“确定”按钮，在 D 盘中将产生三个图片，分别为工作表中的图片、形状和艺术字，如图 30.74 所示。

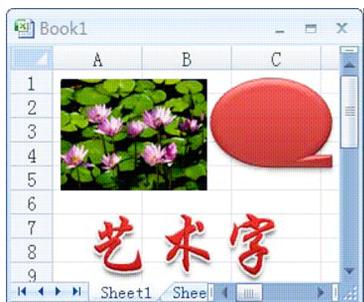


图 30.73 工作表中的图形对象



图 30.74 导出的图片

注意事项：

(1) 导出的图片的大小与工作表中的图片大小一致，如果需要更大的图片，可以修改代码，也可以将工作表中的图形对象扩大后再执行过程。

(2) 如果仅仅需要导出图片,那么可以使用 TypeName 判断对象的类型再决定是否导出。

限于篇幅,安全工具箱、打印工具箱、图表工具箱、唯一值工具箱、文件工具箱、系统工具箱、选择工具箱、程序员工具箱、其他工具箱从略。但所有工具箱的代码在随书光盘中供读者学习。

30.6 开发函数

Excel 有几百个函数,但远远不够完成工作需求,某些问题必须借助自定义的函数来处理。本章介绍 20 个 Function 的设计思路及其用法,读者可以直接将它应用于自己的工作中,也可以根据自己的需求对函数重新编辑,或者是借用该思路开发相似的功能。

30.6.1 开发自定义函数

百宝箱包括 20 个自定义函数,逐一介绍如下。

1. 不同项

函数要求: 提取第一个区域中有、第二个区域中没有的数据。

函数代码:

```
Function 不同项(rng1 As Range, rng2 As Range, Index As Integer) As String
    Application.Volatile
    Dim arr1() As Variant, arr2() As Variant, Item1 As Integer, temp As Integer,
    Counter As Integer '声明变量,
    arr1 = rng1 '将区域转换成数组
    arr2 = rng2
    On Error Resume Next
    For Item1 = 1 To UBound(arr1)
        temp = Application.Match(arr1(Item1, 1), arr2, 0) '逐个计算数组 1 中每个元素
        在数组 2 中的排位
        If Err <> 0 Then Err.Clear: Counter = Counter + 1 '如果有错误(不同值)则累
        加计数器
        If Counter = Index Then 不同项 = arr1(Item1, 1): Exit Function '如果计数器
        与第三参数相同则以该计数器做为索引号从数组 1 中取值,做为函数的结果
    Next Item1
End Function
```

功能测试:

step 1 打开如图 30.75 所示工作表,需要计算的是上届参赛而本届退出的人员名单。

step 2 在 D2 单元格录入以下公式:

```
=不同项(A$2:A$10,B$2:B$10,ROW(A1))
```

step 3 将公式向下填充到 D10 即可计算两组名单中的不同项目。

	A	B	C	D	E	F
1	上届参赛人员	本届参赛人员		不同项		
2	赵	周		赵		
3	钱	吴		钱		
4	孙	郑		孙		
5	李	蒋		李		
6	周	冯				
7	吴	卫				
8	郑	陈				
9	王	褚				
10	冯	李				

图 30.75 计算不同项

注意事项:

计算两个区域的不同项有很多方法，重点在于如何提速。本例采用了三个办法来提速：

(1) 将两个代表区域的参数转换成数组，遍历数组远远快于遍历区域。

(2) 判断不同项目时，不需要将所有不同的项目提取出来，最后通过索引号取值，而是仅仅利用 Match 函数计算排位，如果存在错误则表示它属于不同项。此时不需要将不同项提取出来，而是让计算器与第三参数 Index 比较，相同时才取值。所以整个过程仅需要取值一次。

(3) 当循环到需要的结果时，使用“Exit Function”及结束过程，防止它继续循环。

2. 共有项

函数要求: 计算两个区域中的相同项目。

函数代码:

```
Function 相同项(rng1 As Range, rng2 As Range, Index As Integer) As String
    Application.Volatile
    Dim arr1() As Variant, arr2() As Variant, Item1 As Integer, Item2 As Integer,
    Counter As Integer '声明变量,
    arr1 = rng1 '将区域转换成数组
    arr2 = rng2
    '嵌套循环, 逐个比较两个区域, 并记录相同的个数, 最后取出索引号与第三参数一致的不同值
    For Item1 = 1 To UBound(arr1)
        For Item2 = 1 To UBound(arr2)
            If arr1(Item1, 1) = arr2(Item2, 1) Then Counter = Counter + 1 '相同则累
            加计数器
            If Counter = Index Then 相同项 = arr1(Item1, 1): Exit Function '如果计数
            器等于参数 Index 则该值为结果
        Next Item2
    Next Item1
End Function
```

功能测试:

step 1 打开如图 30.76 所示工作表，需要计算的是上届参赛且而本届也参赛的人员名单。

step 2 在 D2 单元格录入以下公式：

```
=相同项(A$2:A$10,B$2:B$10,ROW(A1))
```

step 3 将公式向下填充到 D10 即可计算两组名单中的相同项目。

	A	B	C	D	E	F
1	上届参赛人员	本届参赛人员		相同项		
2	赵	周		李		
3	钱	吴		周		
4	孙	郑		吴		
5	李	蒋		郑		
6	周	冯		冯		
7	吴	卫				
8	郑	陈				
9	王	褚				
10	冯	李				

图 30.76 计算相同项

注意事项:

在本书第 12 章中讲解数组的筛选时也讲述过如何获取两个数组的相同项目，然而它仅仅适用于 Array 方式赋值的数组。本例采用区域转换成数组，则只能用循环来逐个比较。

3. 取工作表名

函数要求: 提取指定索引的工作表名称。

函数代码:

```
Function 工作表(序号)
    Application.Volatile '强制重新计算
    If 序号 > Sheets.Count Or 序号 < 1 Then '负数及大于工作表总数时返回“超出范围”
        工作表 = "超出范围"
    Else
        工作表 = Sheets(序号).Name '根据索引号取表名
    End If
End Function
```

功能测试:

step 1 在 A2 单元格录入以下公式:

```
=工作表(ROW(A1))
```

step 2 将公式向下填充，直到出现“超过范围”，公式可以将活动工作簿的所有工作表名罗列出来，如图 30.77 所示。

注意事项:

为了让新建或者删除工作表时函数的结果可以随时更新，必须使用 Volatile 语句。而为了防止用户录入非正数或者是大于工作表数量的数值，那么可以通过 IF 来判断录入是否规范。

4. 提取公式

函数要求: 将单元格中的公式提取出来，如果是数组公式必须使用花括号。

函数代码:

```
Function 公式(单元格 As Range)
    Application.Volatile
    If 单元格(1).HasFormula = False Then 公式 = "": Exit Function '不是公式则返回
```

空白

```

If 单元格(1).HasArray = True Then 公式 = "{" & 单元格(1).Formula & "}": Exit
Function '是数组是添加花扩号
公式 =单元格.Formula
End Function
    
```

功能测试:

step 1 待提取公式的列是 C 列, 在 E2 单元格中录入以下公式:

```
=公式(C2)
```

step 2 将公式向下填充到 E9, 从而计算 C 列所有公式, 如图 30.78 所示。

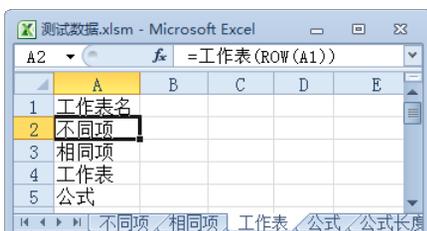


图 30.77 取工作表名

	A	B	C	D	E
1	姓名	成绩	排名		计算公式
2	赵	770	1		=IFERROR(RANK(B2, B\$2: B\$8), "")
3	钱	768	2		=IFERROR(RANK(B3, B\$2: B\$8), "")
4	孙				=IFERROR(RANK(B4, B\$2: B\$8), "")
5	李	740	3		=IFERROR(RANK(B5, B\$2: B\$8), "")
6	周	595	6		=IFERROR(RANK(B6, B\$2: B\$8), "")
7	吴	723	4		=IFERROR(RANK(B7, B\$2: B\$8), "")
8	郑	655	5		=IFERROR(RANK(B8, B\$2: B\$8), "")
9	参考人数	6			=COUNT(C2: C8)

图 30.78 提取 C 列公式

注意事项:

Formula 虽然代表公式, 如果单元格中不存在公式而是一个字符串或者数值时, 它就会将字符串或数值当成公式处理, 所以代码中需要使用 Formula 判断单元格中是否存在公式; 而 HasArray 则用来判断数组公式。

另一个重点是参数“单元格(1)”中的索引号 1 的作用。

5. 计算公式长度

函数要求: 计算包括公式的单元格的公式长度。

函数代码:

```

Function 公式长度(单元格 As Range)
Application.Volatile
If 单元格.HasFormula = True Then '如果是公式
公式长度 = Len(单元格(1).Formula) '计算公式长度
Else
公式长度 = Len(单元格(1).Text) '否则计算单元格显示值的长度
End If
If 公式长度 = 0 Then 公式长度 = "" '如果结果空 0 则不显示
End Function
    
```

功能测试:

step 1 在 B2 单元格中录入以下公式用于计算 A1:A9 区域中不重复值个数, 需要区分大小写:

```
=SUM(1/(MMULT(EXACT(A1:A9, TRANSPOSE(A1:A9))*1, ROW(A1:A9)^0)))
```

step 1 在 C2 单元格录入以下公式计算 B2 的公式长度，结果如图 30.79 所示。

```
=公式长度(B2)
```

注意事项：

Excel 有一个内置函数 Len 用于计算单元格中字符长度，但却无法获取公式的长度。例如本例中改用 Len(B2)，则计算结果为 1。如果需要计算公式的长度，只能对单元格的 Formula 进行计算。

6. 分隔取数

函数要求：按分隔符进行取数。

函数代码：

```
Function 分隔取数(单元格 As Range, 序号 As Byte, 分隔符 As String) As String
    Application.Volatile '强制重新计算
    '如果序号超出范围则返回空白
    If 序号 > UBound(Split(单元格(1), 分隔符)) + 1 Then 分隔取数 = "": Exit Function
    分隔取数 = Split(单元格(1), 分隔符)(序号 - 1) '利用 Split 将字符串转换成数组，再通过索引号取值
End Function
```

功能测试：

step 1 在 A1 单元格录入字符串“中国、广东、广州”。

step 2 在 B1 录入以下公式：

```
=分隔取数(A$1,ROW(A1),"、")
```

step 3 将公式向下填充，直到出现空值，结果如图 30.80 所示。

C2	A	B	C
1	A	不重单元格个数	公式长度
2	B	6	61
3	c		
4	a		
5	A		
6	B		
7	C		
8	D		
9	a		

图 30.79 计算公式长度

B1	A	B	C	D	E
1	中国、广东、广州	中国			
2		广东			
3		广州			
4					

图 30.80 按分隔符取字符串

注意事项：

利用 Split 将字符串转换成数组后，它的最小下界是 0，所以数组中元素个数要用 UBound(Split(单元格(1),分隔符)) + 1 来表示。

7. 合并数据

函数要求：将参数中所有元素合并成一个字符串，且可以指定分隔符号。参数可以是数值、文本、区域引用，也可以是数组。

函数代码：

```

Function 合并(符号 As String, ParamArray 区域() As Variant) As Variant
    Dim tmpText As Variant, i As Variant, cellv As Variant
    Dim cell As Range
    合并 = ""
    For i = 0 To UBound(区域) '遍历参数。有可能有多个参数
        If Not IsMissing(区域(i)) Then '如果有参数
            Select Case TypeName(区域(i))
                Case "Range" '如果参数是区域引用
                    For Each cell In 区域(i) '遍历区域
                        合并 = 合并 & cell & 符号 '合并
                    Next cell
                Case "Variant()" '如果是数组
                    For Each cellv In 区域(i)
                        合并 = 合并 & cellv & 符号
                    Next cellv
                Case Else '如果是其它参数
                    合并 = 合并 & 区域(i) & 符号
            End Select
        End If
    Next i
    合并 = Left(合并, Len(合并) - Len(符号)) '将合并后的字符串去除最右边的字符, 再做为运算结果
End Function

```

功能测试:

step 1 在 A1:A3 区域分别录入 “Word”、“Excel” 和 “PowerPoint”，在 C1 录入以下公式：

```
=合并(" ", " , OFFSET(A1, , , COUNTA(A:A)))
```

公式将会把 A 列中所有非空单元格的值合并成一个字符串，如图 30.81 所示。重点在于 A 列中添加或者删除数据时公式可以自动更新。例如 A4 中再录入 “Groove”，那么公式的计算结果会自动变成 “Word,Excel,PowerPoint,Groove”。

C1	A	B	C	D	E
	Word		Word, Excel, PowerPoint		
	Excel				
	PowerPoint				

图 30.81 合并 A 列的非空单元格

step 2 在 A1 单元格录入以下公式可以对内存数组进行合并：

```
=合并(" ", TEXT(ROW(2:8), "DDDD"))
```

其中 TEXT 的第二参数使用 “DDDD” 表示显示英文的星期，改为 “AAAA” 则成为中文星期。而 “ROW(2:8)” 表示 1900 年 1 月 2 号到 1 月 8 号，利用 TEXT 函数可以转换成包括 7 个英文星期的内存数组。合并函数可以将该内存数组以空格的分隔符合并为一个字符串，如图 30.82 所示。

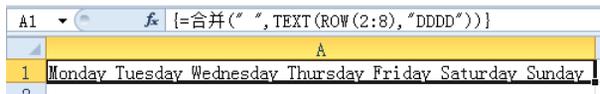


图 30.82 合并内存数组

注意事项:

函数的第二参数使用了 ParamArray，表示它代表不确定个数的可选参数。如果在 Excel 2003 中使用此函数，那么可以对 1 到 29 个参数合并，在 Excel 2010 中则可以对 1 到 254 个参数进行合并。例如可以采用以下公式合并三个工作表中的 A 列非空单元格：

```
= 合并(":",OFFSET(Sheet1!A1,,COUNTA(Sheet1!A:A)),OFFSET(Sheet2!A1,,COUNTA(Sheet2!A:A)),OFFSET(Sheet3!A1,,COUNTA(Sheet3!A:A)))
```

8. 将表达式转为值

函数要求: 将表达式转换成值。

函数代码:

```
Function 计算(单元格 As Range)
Application.Volatile
Dim temp As String, adds As String
adds =单元格.Address(0, 0) '记录参数引用的单元格的地址
If Left(单元格(1).Text, 1) = "{" And Right(单元格(1).Text, 1) = "}" Then '
如果首字符和末尾字符包括花括号
temp = VBA.Replace(VBA.Replace(单元格(1).Text, "{", ""), "}", "") '替换掉
Else
temp =单元格(1).Text
End If
'如果有 row() 或者 column() 则需要在括号中添加引用单元格的地址, 否则将产生错误值
temp = VBA.Replace(Replace(temp, "row()", "row(" + adds + ")"), "column()",
"column(" + adds + ")")
计算 = Application.Evaluate(temp) '计算结果
End Function
```

功能测试:

A2:A4 有三个表达式，包括有等号和无等号的表达式，以及包含花括号和无花括号的表达式。在 B2 输入以下公式可以计算表达式的值。

```
=计算(A2)
```

	A	B
1	表达式	计算结果
2	=sum(12,F100,sqrt(25))	17
3	100*5-20/round(123.456,)	499.3495935
4	{sum(row(1:100))}	5050

图 30.83 将表达式转换成值

注意事项:

如果表达式中的首尾存在花括号则无法计算，那么需要使用替换函数将其替换成空；而变量 adds 在本函数的中的作用是解决“row()”和“column()”在计算中的 BUG。例如在 A1 单元格中有表达式“row()+column()”，如果不使用替换，在 C3 单元格计算其值的时候将得到结果 6。因为表达式“row()+column()”

中有 C3 时表示计算 C3 的行号加列号；而将“row()”替换成“row(A1)”，将“column()”替换成“column(A1)”后则可以避过此 BUG。

9. 排名

函数要求：对成绩进行排名，但忽略重复的排名。

函数代码：

```
Public Function 排名(区域, 成绩) '两个参数可以随意交换位置
    Dim d As Object, e As Object, Rng, i As Integer
    Set d = CreateObject("scripting.dictionary") '创建字典对象
    For Each Rng In 区域 '遍历区域(也可以是内存数组)
        If Rng = 成绩 Then i = i + 1 Else If Rng > 成绩 Then d(Rng * 1) = 1
    Next
    If i > 0 Then 排名 = d.Count + 1 Else 排名 = "" '区域中不存在与“成绩”相同值就返回空值
End Function
```

功能测试：

图 30.84 中 B 列的成绩存在重复值，在进行排名时需要排除重复值而排名，不管有多少人成绩并列第一名时，都必须有第二名。

step 1 在 C2 单元格录入以下公式：

```
=排名(B$2:B$9,B2)
```

step 2 将公式向下填充至 C9，排名结果如图 30.85 所示。

step 3 相对于内置的 Rank 函数，本函数还有另一个优点——对内置数组进行排名。例如对大于 60 的成绩进行排名，Rank 无法完成。而本例函数可以利用 IF 将大于 60 的成绩转换成内存数组现排名，其公式如下：

```
=排名(IF(B$2:B$9>60,B$2:B$9),B2)
```

	A	B	C
1	姓名	成绩	中国式排名
2	赵	75	5
3	钱	89	3
4	孙	99	1
5	李	80	4
6	周	54	6
7	吴	90	2
8	郑	99	1
9	王	80	4

图 30.84 忽略重复值的排名

	A	B	C	D	E	F
1	姓名	成绩	大于60者排名			
2	赵	75	5			
3	钱	89	3			
4	孙	99	1			
5	李	80	4			
6	周	54				
7	吴	90	2			
8	郑	99	1			
9	王	80	4			

图 30.85 对大于 60 的成绩排名

注意事项：

CreateObject("scripting.dictionary")的特点是导入的项目不重复，本例中使用它来完成除重复值再排名。也可以改用 Collection 对象来剔除重复值。

10. 排序

函数要求：对具有重复出现性质的一组数据去除重复值，且按出现次数进行排序，可以通过参数调整升序或者降序。

函数代码：

```
Public Function 排序(数据, 名次, Optional Bl As Boolean = True) '第三参数可选,
默认为 true
    Dim dic As Object, Item, i As Byte, KeyCount, arr
    Set dic = CreateObject("scripting.dictionary")
    arr = 数据 '将数据转换成数组
    For Each Item In arr '遍历数组
        i = i + 1 '累加计数器
        '将序号添到字典对象中
        If dic(Item) = 0 Then dic(Item) = dic(Item) + 1 - i * 0.00001 Else dic(Item)
= dic(Item) + 1
    Next Item
    KeyCount = dic.Keys
    If 名次 > dic.Count Then '参数超出重复个数时则返回空值
        排序 = ""
    Else '否则根据降序的顺序从字典对象中取“名次”对应的值
        If Bl Then
            排序 = KeyCount(WorksheetFunction.Match(Application.Large(dic.Items, 名次),
            dic.Items, 0) - 1)
        Else
            排序 = KeyCount(WorksheetFunction.Match(Application.Small(dic.Items, 名次),
            dic.Items, 0) - 1)
        End If
    End If
End Function
```

功能测试：

图 30.86 所示的工作表是每一届的冠军榜，其中部分选手多次获得冠军。现需要根据选手夺冠的次数来排序，包括升序与降序。

step 1 在 C1 单元格录入以下公式并向下填充可以对冠军榜进行降序排列。获冠军次数越多，排序时越靠前：

=排序(B\$1:B\$10,ROW(),1)——第三参数 1 也可以改用 true 或者任意正数

step 2 在 D1 单元格产生以下公式并向下填充可以对冠军榜进行升序排列。

=排序(B\$1:B\$10,ROW(),0)——第三参数 0 也可以改用 false

	A	B	C	D	E
1	第一届	赵明明	王华	刘芳芳	
2	第二届	朱丽	赵明明	赵宏新	
3	第三届	赵明明	朱丽	朱丽	
4	第四届	王华	赵宏新	赵明明	
5	第五届	赵宏新	刘芳芳	王华	
6	第六届	王华			
7	第七届	王华			
8	第八届	朱丽			
9	第九届	刘芳芳			
10	第十届	王华			

图 30.86 去重复值并排序

注意事项：

本例的函数在排序时会自动剔除重复值，然后按出现的次序进行排序。如果两者出现的次数相

同,那么按它在原来区域出现的顺序排列。

11. 取数

函数要求: 从一个混合字符串中取值,包括取数值、文本、数值加运算值,以及计算数字与运算符的值。

函数代码:

```
Function 取数(区域 As Range, Optional 序号 As Integer = 0)
    Dim Rng As String, i As Integer, Str As String
    Select Case 序号
    Case 0 '如果参数为 0 或者忽略,那么提取数值
        For i = 1 To Len(区域(1))
            If Val(Mid(区域(1), i, 1)) > 0 Or Mid(区域(1), i, 1) = "0" Or Mid(区域(1), i, 1) = "." Then
                取数 = 取数 & Mid(区域(1), i, 1)
            End If
        Next i
    Case 1 '如果参数为 1,那么提取文本
        Rng = 区域(1)
        For i = 0 To 9
            Rng = Replace(Rng, i, "")
        Next i
        取数 = Rng
    Case 2 '如果参数 2,那么提取数值加运算符
        For i = 1 To Len(区域(1))
            If Val(Mid(区域(1), i, 1)) > 0 Or Mid(区域(1), i, 1) = "0" Or Mid(区域(1), i, 1) = "+" Or Mid(区域(1), i, 1) = "-" Or Mid(区域(1), i, 1) = "*" Or Mid(区域(1), i, 1) = "/" Or Mid(区域(1), i, 1) = "^" Or Mid(区域(1), i, 1) = "." Or Mid(区域(1), i, 1) = "(" Or Mid(区域(1), i, 1) = ")" Then
                取数 = 取数 & Mid(区域(1), i, 1)
            End If
        Next i
    Case 3 '如果参数为 3,那么提取数值加运算符并计算其结果
        For i = 1 To Len(区域(1))
            If Val(Mid(区域(1), i, 1)) > 0 Or Mid(区域(1), i, 1) = "0" Or Mid(区域(1), i, 1) = "+" Or Mid(区域(1), i, 1) = "-" Or Mid(区域(1), i, 1) = "*" Or Mid(区域(1), i, 1) = "/" Or Mid(区域(1), i, 1) = "^" Or Mid(区域(1), i, 1) = "." Or Mid(区域(1), i, 1) = "(" Or Mid(区域(1), i, 1) = ")" Then
                Str = Str & Mid(区域(1), i, 1)
            End If
        Next i
        取数 = Application.Evaluate("(" & Str & ")") '计算结果
    Next i
    End Select
End Function
```

功能测试:

在图 30.87 中, B 列记录每天的采购详情, 现需利用公式计算金额。

step 1 在 C2 单元格录入以下公式即可计算金额:

=取数(B2,3)——参数 3 表示取出数值与运算符并计算其值

f_x =取数(B2,3)			
	A	B	C
1	日期	购买记录	金额
2	9月3日	白菜5KG*1.2元+牛肉12斤*14元+油15元	189
3	9月4日		

图 30.87 计算金额

step 2 如果需要取出所有数值加运算符,那么可以修改参数。

=取数(B2,2)——计算结果为“5*1.2+12*14+15”

注意事项:

也可以将默认值改为 3,表示忽略第二参数时就计算表达式的值。

本函数在参数为 3 时在某个条件下无法正确计算,即“/”后面无数值时。例如“白菜 5KG*1.2 元/KG+牛肉 12 斤*14 元/斤”,那么它的表达式是“5*1.2/+12*14/”,这是无法计算的。所以如果单元格是字符并且以此方式存在的话,通常采用替换法将“/”替换成“/1”再计算。

12. 取唯一值

函数要求: 从区域中取出唯一值,可以是多个区域。而且支持先行后列或者先列后行。

函数代码:

```
Function 唯一值(模式 As String, 序号 As Integer, ParamArray 区域() As Variant)
    Application.Volatile True
    Application.ScreenUpdating = False
    If UCase(模式) <> "ROW" And UCase(模式) <> "COLUMN" Then — = "第一参数错误": Exit
Function
    If IsMissing(区域) Then 唯一值 = "": Exit Function '未指定第三参数则返回空值
    On Error Resume Next
    Dim cell As Range, rngs As Range, only As New Collection, i As Long, j As Byte
    Dim Cn As Byte, Rn As Byte, are As Byte
    For j = 0 To UBound(区域) '遍历第三参数开始的所有参数
        If rngs Is Nothing Then Set rngs = 区域(j)
        Set rngs = Application.Union(rngs, 区域(j))
    Next j
    Set rngs = Application.Intersect(rngs, ActiveSheet.UsedRange) '仅仅在已用区域
    的交集中循环
    With rngs
        If UCase(模式) = "ROW" Then '如果第一参数用“ROW”,那么先行后列
            For are = 1 To .Areas.Count
                Rn = .Rows.Count
                Cn = .Columns.Count
                For i = 1 To Cn
                    For j = 0 To Rn - 1
                        If .Areas(are).Item(i + j * Cn) <> "" Then only.Add .Areas(are).Item(i
+ j * Cn).Value, CStr(.Areas(are).Item(i + j * Cn))
                    
```

```

        Next j
    Next i
    Next are
    ElseIf UCase(模式) = "COLUMN" Then '如果为“COLUMN”那么先列后行
        For Each cell In rngs
            If cell <> "" Then only.Add cell.Value, CStr(cell.Value)
        Next cell
    End If
Endd:
    Application.ScreenUpdating = True
    唯一值 = only.Item(序号) '根据第二参数取对应的唯一值
    If 序号> only.Count Then 唯一值 = "" '索引号超出范围则返回空值
    End With
End Function

```

功能测试:

step 1 对图 30.88 中的 B 列冠军提取唯一值, 那么在 D2 单元格录入以下公式:

```
=唯一值("row",ROW(A1),B$1:B$10)
```

step 2 将公式向下填充至 D10, 并提取 B1:B10 的唯一值。

注意事项:

函数的第一参数仅仅在引用区域是多行多列时才有效, 它决定先行后列或者先列后行。而第三参数开始的所有参数都是可选的, 可以对 1 个、2 个……甚至 20 个区域提取唯一值。

13. 消除空值

函数要求: 返回一个区域中的非空单元格引用, 自动忽略所有空单元格。

函数代码:

```

Function 消除空值(区域 As Range, 序号 As Integer) As Variant
    Application.Volatile
    Dim Item, Item2, cell As Range
    Dim arr(1 To 10000) '最后计算范围是 10000 个单元格
    For Each cell In 区域 '遍历区域
        Item = Item + 1 '累加计数器
        If Len(cell) > 0 Then '如果非空值
            Item2 = Item2 + 1 '记录非空单元格个数
            If Item2 = 序号 Then 消除空值 = 区域(Item): Exit Function '如果等于“序号”则
从区域中对应的值做为函数结果
        End If
    Next
    消除空值 = "" '让序号超出非空单元格个数时返回空值
End Function

```

功能测试:

step 1 获取图 30.89 中 B 列的值且忽略空值, 可以使用以下公式:

=消除空值(B\$2:B\$11,ROW())

step 2 将公式向下填充到 B11, 那么公式可以取出 B 列所有非空值。

	A	B	C	D	E
1	第一届	赵明明		空班人员	
2	第二届	朱丽		赵明明	
3	第三届	赵明明		朱丽	
4	第四届	王华		王华	
5	第五届	赵宏新		赵宏新	
6	第六届	王华		刘芳芳	
7	第七届	王华			
8	第八届	朱丽			
9	第九届	刘芳芳			
10	第十届	王华			

图 30.88 取唯一值

	A	B	C	D
1	日期	补课		补课(去空值)
2	9月1日	语文		数学
3	9月2日			地理
4	9月3日	数学		化学
5	9月4日	地理		历史
6	9月5日			英语
7	9月6日	化学		
8	9月7日			
9	9月8日	历史		
10	9月9日			
11	9月10日	英语		

图 30.89 消除空值

注意事项:

将所有非空值加入一个数组中, 然后利用第二参数作为索引从数组中取值是比较常见的思路, 但为了函数的运算提速, 本例采用了不同的算法, 仅仅判断是否为空值, 而不用提取数值。直到循环到与第二参数“序号”一致的非空单元格个数时才执行一次取值, 然后退出循环。此思路在运算上较前者可以减少循环次数, 从而使函数的执行速度提高。

14. 颜色计数

函数要求: 计算区域中与参照单元格颜色相同者个数, 可用背景色或者字体色作为计算标准。

函数代码:

```
Function 颜色计数(参照颜色区 As Range, 计数区 As Range, Optional B1 As Boolean = True) '第三参数决字按单元格背景还是字体色计数
Application.Volatile
Dim cell As Range, Colors, Count
'如果第三参数是 True 则按背景色计数, 否则按字体色计数
Colors = IIf(B1, 参照颜色区(1).Interior.Color, 参照颜色区(1).Font.Color)
For Each cell In 计数区 '遍历计数区
If B1 Then
If cell.Interior.Color = Colors Then Count = Count + 1 '背景色相同则累加计数器
Else
If cell.Font.Color = Colors Then Count = Count + 1 '字体色相同则累加计数器
End If
Next cell
If Count = 0 Then 颜色计数 = "无此背景色": Exit Function
颜色计数 = Count '将个数赋与函数, 成为最终颜色计数结果
End Function
```

功能测试:

图 30.90 中有部分单元格存在特殊的背景色和字体色, 利用自定义函数可以计算其数量。

step 1 在 F2 单元格录入以下公式:

=颜色计数(F1,B2:D9,TRUE)——黄色背景单元格个数为 5

step 1 在 F4 单元格录入以下公式。

=颜色计数(F3,B2:D9,FALSE)——红色字体单元格个数为 3

注意事项:

在 Excel 2003 中可以使用 ColorIndex 获取单元格的顏色，而在 Excel 2010 中则会出错。为了使公式通用于 Excel 2003 和 Excel 2010，只能使用 Color。

15. 颜色求和

函数要求: 按字体色或者背景色对区域中的数据求和。**函数代码:**

```
Function 颜色求和(参照颜色区 As Range, 求和区 As Range, Optional B1 As Boolean = True) '第三参数决字按单元格背景还是字体色求和
Application.Volatile
Dim cell As Range, Colors, SUM
'如果第三参数是 True 则按背景色求和, 否则按字体色求和
Colors = IIf(B1, 参照颜色区(1).Interior.Color, 参照颜色区(1).Font.Color)
For Each cell In 求和区 '遍历求和区
If B1 Then
If cell.Interior.Color = Colors Then SUM = SUM + IIf(IsNumeric(cell), cell, 0) '背景色相同则累计数值
Else
If cell.Font.Color = Colors Then SUM = SUM + IIf(IsNumeric(cell), cell, 0) '字体色相同则累计数值
End If
Next cell
颜色求和 = SUM '将个数赋与函数, 成为最终颜色求和结果
End Function
```

功能测试:

图 30.91 中有部分单元格存在特殊的背景色和字体色，利用自定义函数可以计算其数据和。

step 1 在 F2 单元格录入以下公式:

=颜色求和(F1,B2:D9,TRUE)——黄色背景单元格之和为 268

step 2 在 F4 单元格录入以下公式。

=颜色求和(F3,B2:D9,FALSE)——红色字体单元格之和为 300

	A	B	C	D	E	F
1	姓名	成绩	姓名	成绩		黄色背景
2	赵	47	冯	100		5
3	钱	80	陈	84		红色字体
4	孙	41	褚	49		3
5	李	62	卫	66		
6	周	86	蒋	49		
7	吴	100	沈	49		
8	郑	90	韩	52		
9	王	100	杨	98		

	A	B	C	D	E	F
1	姓名	成绩	姓名	成绩		黄色背景
2	赵	47	冯	100		268
3	钱	80	陈	84		红色字体
4	孙	41	褚	49		300
5	李	62	卫	66		
6	周	86	蒋	49		
7	吴	100	沈	49		
8	郑	90	韩	52		
9	王	100	杨	98		

图 30.90 按指定背景色和字体色计算单元格个数 图 30.91 按指定背景色和字体色计算单元格中数据和

注意事项:

本函数与“颜色计数”的思路完全相同,不过如果第二参数所引用的区域存在文本,那么无法计算合计值,所以利用 IIF 函数将其文本转换成零值。另一个办法是利用工作函数 SUM 将文本自动转换成零值。

16. 去首尾求平均

函数要求: 对区域中的值去除最大值和最小值再计算平均数,支持多区域。

函数代码:

```
Function AVER(区域一 As Range, ParamArray 更多区域()) '去首尾后求平均
Application.Volatile
With WorksheetFunction
Dim Rng As Range, Sums
If UBound(更多区域) = -1 And .CountA(区域一) > 2 Then '如果参数“更多区域”忽略,而且第一参数中的值超过两个
AVER = (.SUM(区域一) - .Min(区域一) - .Max(区域一)) / (.CountA(区域一) - 2)
'结果为合计减去最大值和最小值
ElseIf UBound(更多区域) = -1 And .CountA(区域一) < 3 Then '如果忽略第二参数,且第一参数小于3两个单元格则结果为0
AVER = 0
End If
If UBound(更多区域) >= 0 Then '如果未忽略第二参数
For j = 0 To UBound(更多区域)
If Rng Is Nothing Then Set Rng = 区域一
Set Rng = Application.Union(Rng, 更多区域(j)) '合并所有参数引用的区域
Next j
If .CountA(Rng) > 2 Then
Sums = .SUM(Rng) - .Min(Rng) - .Max(Rng) '对合并后的区域求和,并减去最大值和最小值
AVER = Sums / (.CountA(Rng) - 2) '再求平均
Else
AVER = 0
End If
End If
End With
End Function
```

功能测试:

图 30.92 中 B 列和 D 列是 14 个评委对某选手的评分,在 F2 单元格中录入以下公式可以计算选手的最后得分。

```
=aver(B2:B8,D2:D8)
```

注意事项:

因计算最后得分前需要去除一个最高分和一个最低分,那么在计算之前需要确保区域中的单元格不少于 3 个,否则直接

1	评委	评分	评委	评分	洗头得分
2	赵	8.5	王	5.4	7.375
3	钱	6.7	冯	8.3	
4	孙	9.1	陈	5.9	
5	李	6.9	褚	9.4	
6	周	8.6	卫	7.6	
7	吴	6.6	蒋	6.8	
8	郑	8.1	沈	5	

图 30.92 计算选手得分

对结果赋值为 0。

17. 对 42 位超长数据求和

函数要求：对整数 28 位以内小数 14 位以内的数据执行求和运算。

函数代码：

```
'声明函数“Sums”，其类型为 String，有一个必选参数 Rng，参数类型为 Range 对象，用于指定单
元格区域
Function sums(Rng As Range) As String '此函数在 Rng.Count 不超过 10 的情况下计算结
果是正确的，否则因为进位的原因可以计算错误。
    Application.Volatile '声明为易失性函数
    Dim One, Two, Three, 点 As Byte
    On Error Resume Next '防错
    For Each cell In Rng '遍历所有单元格
        If Len(cell) > 0 Then '如果长度大于 0
            If IsNumeric(cell.Text) Then '如果是数值
                点 = VBA.InStr(1, cell.Text, ".") '计算小数点的出现位置
                If 点 > 0 Then
                    Three = Three + 1 * ("0." & Mid(cell.Text, 点 + 1, 14)) '如果有小数点，
                    那么取小数部数，不管小数多少位，取 14 位参与运算
                    '去掉整数后将单元格格式化为 28 位，然后取左边 14 位
                    One = One + Left(Format(Left(cell, 点 - 1), WorksheetFunction.Rept(0,
                    28)), 14) * 1
                    '将单元格格式化为 28 位，然后取右边 14 位
                    Two = Two + Right(Format(Left(cell, 点 - 1), , WorksheetFunction.Rept(0,
                    28)), 14) * 1
                Else
                    '去掉整数后将单元格格式化为 28 位，然后取左边 14 位
                    One = One + Left(Format(cell, WorksheetFunction.Rept(0, 28)), 14) * 1
                    '将单元格格式化为 28 位，然后取右边 14 位
                    Two = Two + Right(Format(cell, WorksheetFunction.Rept(0, 28)), 14) * 1
                End If
            End If
        End If
    Next
    If Three >= 1 Then
        Two = Two + Left(VBA.Format(Three, "#.00"), Len(VBA.Format(Three, "#.00"))
        - 3) '如果小数部分大于等于 1，那么整数部分右边 14 位加整数部分
        Three = Right(VBA.Format(Three, "#.00"), 3) '小数部分
    End If
    If Len(Two) > 14 Then One = One + Left(Two, Len(Two) - 14) '如果整数右边 14
    位相加后大于 14 位，那么左边 14 位加右侧的进位值
    If One = 0 And Two = 0 Then Two = "" '如果整数左边 14 位为 0，右边 14 位值为 0，那
    么将右边 14 位直接赋值 0
    '将整数部分左边 14 位、右边 14 位和小数串连起来作为函数的返回值。
    sums = One & Right(Format(Two, WorksheetFunction.Rept(0, 28)), 14) &
    VBA.IIf(Three >= 1, Three - 1, Three)
```

```
End Function
```

功能测试：

在 B2 单元格录入以下公式，用于计算图 30.93 中 A 列的数据之和。

=sums(A1:A5)——结果为整数 27 位、小数 12 位的文本型数值

	A	B
1	0.123457891243	超长数据求和
2	546578945612345789412345	200548126841073580391015993.249271291243
3	1547895461234578457891.1245684	
4	23145757.001245	
5	200000000000000000000000	

图 30.93 计算超长数值之和

注意事项：

工作表函数 SUM 仅仅能为 15 位数据进行求和，超过该限制就无法正确显示出来。本例自定义函数将超过 15 位的数据按每 14 位一段拆分成三段再分别计算，最后将三段的結果串联成文本。

18. 按颜色条件求和

函数要求：按条件区域的背景色对求和区域中对应的数据求和。

函数代码：

```
Function sumifcol(条件区 As Range, 颜色单元格 As Range, Optional 求和区) '按颜色
进行条件求和
    Dim i As Integer, SUM, counts As Integer, rng3 As Range
    Application.Volatile
    If IsMissing(求和区) Then Set rng3 = 条件区 '如果第三参数忽略则将“条件区”赋与变量“rng3”
    '如果未忽略第三参数则将按第一参数的高度与宽度重置第三参数，然后赋值给变量“Rng3”
    If Not IsMissing(求和区) Then Set rng3 = 求和区(1).Resize(条件区.Rows.Count, 条件区.Columns.Count)
    For i = 1 To 条件区.Count '遍历条件区
        If 条件区(i).Interior.Color = 颜色单元格(1).Interior.Color Then '如果颜色相同
            counts = counts + 1 '累加计数器
            SUM = WorksheetFunction.SUM(rng3(i)) + SUM '对符合颜色条件的区域求和
        End If
    Next i
    sumifcol = SUM
End Function
```

功能测试：

图 30.94 中 A 列的职工姓名中部分单元格有颜色标示，计算某颜色对应的工资之和，可以用公式：

```
=sumifcol(A2:A9,E1,C2)
```

其中 A2:A9 表示条件区域，它包含有背景颜色的单元格，而 E1 是条件参照区域，只能是一

个单元格，第三参数为可选参数，忽略该参数则表示对第一参数所在区域求和。本例中使用 C2，程序会自动将 C2 扩展为 C2:C9 再求和。

如果颜色标示在 C 列，那么可以忽略第三参数。

注意事项：

修改单元格的不会触发工作表事件以及公式重算。如果对 A 列的单元格修改背景色，需按下 F9 或者在任意单元格发生改变时公式才会更新运算结果。

19. 对区域中混合字符串的所有数字求和

函数要求：对包括英文、汉字和数字的混合字符串中的数字求和，可以批量执行。

函数代码：

```
Function SumNum(rang As Range)
    Dim i As Integer, cell As Range, Str As String
    For Each cell In rang '遍历区域
        For i = 1 To Len(cell.Text) '遍历单元格中每个字符
            If VBA.IsNumeric(Mid(cell.Text, i, 1)) Or Mid(cell.Text, i, 1) = "." Then '如果是数字或者小数点
                Str = Str & Mid(cell.Text, i, 1) '串连所有数字与小数点
            Else
                Str = Str & "+" '对于其它字符则转换成 "+"
            End If
        Next i
        Str = Str & "+" '在末尾再添加一个 "+"
    Next cell
    '如果末尾字符是 "+" 则添加一个 0，否则保持不变。最后将字符串计算结果并赋与函数 SumNum
    SumNum = Application.Evaluate(Str & IIf(Right(Str, 1) = "+", "0", ""))
End Function
```

功能测试：

图 30.95 中 B 列支出表，由于制表时将产品与金额录入在同一单元格中，无法采用 SUM 求和。那么可以使用自定义函数 SumNum 完成：

=sumnum(B2:B6)

E2		fx =sumifcol(A2:A9,E1,C2)		
	A	B	C	D
1	姓名	工号	工资	
2	赵	9567	1520	7130
3	钱	5445	2190	
4	孙	2050	1580	
5	李	6787	2140	
6	周	4649	1810	
7	吴	5480	1910	
8	郑	4196	1600	
9	王	7809	1600	

图 30.94 按颜色进行条件求和

B7		fx =SumNum(B2:B6)	
	A	B	
1	日期	食堂支出表	
2	9月1日	白菜35元大米43.2元	
3	9月2日	食用油45元猪肉29元	
4	9月3日	花菜30元	
5	9月4日	牛筋22.2元茄子14元	
6	9月5日	大米50元牛蒡18.9元	
7	合计		287.3

图 30.95 对区域中混合字符串的所有数字求和

注意事项：

Evaluate 方法计算表达式可以自动去除多余的、重复的运算符。例如以下三种表达式都可以

得到相同的计算 330。

```
Application.Evaluate("334 - --4")
Application.Evaluate("334 - - + - 4")
Application.Evaluate("334-4")
```

基于这个特点，定义函数时将所有非数值相关的字符转换成“+”再参与运算。

但如果最后一个字符是“+”将无法计算结果，所以需要判断最后一个字符是否是“+”，如果是则连接一个 0 使其成为一个完整的表达式，否则保持不变。

20. 批量命名工作表

函数要求：利用函数对工作表进行批量命名。

操作步骤：

step 1 首先在模块中插入以下代码：

```
Public Xlap, shts As Worksheet, Rng As Range, Star As Byte, Ends As Byte '
声明公共变量
Function RmSht(起始表 As Byte, 结束表, 新名称 As Range) '声明函数过程, 包含三个参数
On Error Resume Next
'前两个参数的范围进行限制
If 起始表>结束表 Or 起始表< 1 Or 结束表> Sheets.Count Then RmSht = "参数错误": Exit
Function
Set Rng = 新名称 '将第三参数赋与公共变量
RmSht = "改表名: " & 起始表 & "-" & 结束表 '函数的结果
Star = 起始表: Ends = 结束表 '将第一二参数赋与公共变量
Set Xlap = New APP '激活类模块
Set Xlap.sht = ActiveSheet
End Function
```

step 2 插入一个类模块，将其命名为“APP”，并在类模块中录入以下代码：

```
Public WithEvents sht As Worksheet
Private Sub sht_Change(ByVal Target As Range)
On Error Resume Next
Dim i As Byte, j As Byte
Set xlapp.sht = Nothing
Set xlapp = Nothing
For i = Star To Ends
Sheets(i).Name = Rng(j + 1).Text
j = j + 1
Next i
End Sub
```

功能测试：

在图 30.96 所示工作簿中，第 21、22 和 23 个工作表分别为“A”、“B”和“C”，要求利用函数批量命名为“星期一”、“星期二”和“星期三”。

step 1 在 A1:A3 区域录入“星期一”、“星期二”和“星期三”。

step 2 在 B1 单元格录入以下公式：

=rmsht(21,23,A1:A3)——表示用 A1:A3 的数据对 21 到 23 个工作表进行命名

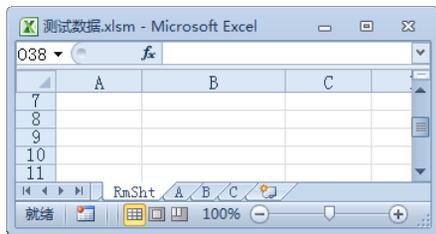


图 30.96 待命名的工作表

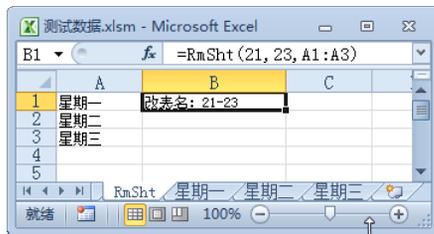


图 30.97 批量命名

注意事项：

函数的功能有很多限制，它的唯一作用是返回值，不能改变 Excel 的环境设置，主要体现在：

- ◆ 不能插入或者删除单元格、行、列、工作表
- ◆ 不能改变单元格格式
- ◆ 不能对工作表命名、移动、删除、添加
- ◆ 不能添加名称
- ◆ 不能改变工作簿的视图大小、切换窗口等

.....

如果一定要实现以上功能，通常配合类模块来实现。其思路为：处理函数获取用户需要设置的参数，通过类模块中的工作簿事件来完成设置。函数的参数通常借用公共变量来传递到类模块。

利用以上思路，也可以开发一个合并单元格的函数或者改变单元格字体的函数，读者可以多多练习。

30.6.2 设计函数帮助

自定义函数必须设计帮助，对函数的功能和参数进行详细说明，否则不仅用户不会用，部分开发者也无法明白各函数的含义。

设计函数的帮助可以利用 Application.MacroOptions 方式来实现。而 Application.MacroOptions 的代码也有两种处理方式：

- (1) 创建一个 SUB 过程，通过菜单或者快捷键执行，从而添加函数说明。
- (2) 在立即窗口执行，然后删除代码保存文件。

两种方法各有所长。方法一的优点在于可以随时修改代码，让函数的说明可以编辑，甚至修改成其他语言；第二种方法的优点在于添加一次说明，不需要反复加载、不需要单独一个添加说明的过程，那么插件的体积更小，启动速度也更快，两者差异不大。

本插件采用两种方法结合，即既通过立即窗口执行代码，为函数添加说明，又创建一个菜单调用过程，同时为过程指定一个快捷键。具体步骤如下：

step 1 在模块中录入以下过程：

```

Sub 加入函数说明 ()
    Dim counts As Byte
    If Application.Version * 1 <= 11 Then counts = 30 Else counts = 255
    With Application
        ThisWorkbook.IsAddin = False '必须取消 IsAddin 属性才可以设置宏的属性
        .MacroOptions Macro:="合并", Description:="将区域字符合并: 第一参数为分割符, 第二参数为引用区域。可以用""表示不使用分割符。第三参数开始为引用区域, 可以有 1 到" & (counts - 1) & "个区域。也还可以直接用文本做参数。", Category:="andysky 定义函数"
        .MacroOptions Macro:="取数", Description:="分离数字与文本: 第一参数为单元格; 第二参数为可选参数, 当为 0 或者省略时表示取出数字; 为 1 时表示排除数字; 为 2 时表示取出数字与运算符; 为 3 则提取数字与运算符且计算结果。", Category:="andysky 定义函数"
        .MacroOptions Macro:="唯一值", Description:="返回不重复值: 可以对 1 到" & (counts - 2) & "个区域取唯一值; 前三个为必选参数, 其余为可选参数。第一参数为 ROW 时先后行列取值, 为 COLUMN 时先列后行(不分大小写), 第三参数开始为引用区域。", Category:="andysky 定义函数"
        .MacroOptions Macro:="消除空值", Description:="消除空值函数: 可以选择多行多列, 按先后行列之方式返回值。两个参数, 一为区域一为序号。", Category:="andysky 定义函数"
        .MacroOptions Macro:="颜色求和", Description:="按背景颜色对区域值求和: 第一参数为参照值。", Category:="andysky 定义函数"
        .MacroOptions Macro:="颜色计数", Description:="按背景颜色计算区域中同背景之数据个数。第一参数为参照值。", Category:="andysky 定义函数"
        .MacroOptions Macro:="工作表", Description:="取工作表名: 一个参数, 输入工作表地址即返回工作表名。", Category:="andysky 定义函数"
        .MacroOptions Macro:="分隔取数", Description:="按分隔符取数: 第一参数为单元格, 第二参数表示取第 N 个, 第三个参数表示分隔符。", Category:="andysky 定义函数"
        .MacroOptions Macro:="相同项", Description:="返回两个区域共有项目: 前两个参数为区域, 第三参数为序号。", Category:="andysky 定义函数"
        .MacroOptions Macro:="不同项", Description:="返回第一区域中有而第二区域没有的项目: 前两个参数是区域, 第三参数为序号。", Category:="andysky 定义函数"
        .MacroOptions Macro:="公式", Description:="公式函数: 提取单元格中的公式, 只需要一个参数, 单元格引用。", Category:="andysky 定义函数"
        .MacroOptions Macro:="计算", Description:="计算函数: 计算单元格中的表达式, 如单元格中为“456+2”, 则计算其结果 458。只需要一个参数, 单元格引用。", Category:="andysky 定义函数"
        .MacroOptions Macro:="公式长度", Description:="计算单元格中公式的长度, 一个参数即单元格引用。", Category:="andysky 定义函数"
        .MacroOptions Macro:="sumifcol", Description:="按颜色进行条件求和: 有三个区域引用参数, 其中第三参数为可选参数。第一参数为参照区域, 第二参数为颜色条件, 第三参数为求和区域, 即第一参数中颜色与第二参数相同, 则将对第三参数的单元格进行求和。如果省略第三参数则对第一参数求和。", Category:="andysky 定义函数"
        .MacroOptions Macro:="AVER", Description:="评分函数: 计算参数区域中去除最大值与最小值之再求平均, 参数个数有" & counts & "个。", Category:="andysky 定义函数"
        .MacroOptions Macro:="排名", Description:="中国式排名函数: 即名次不间断, 当两人并列第二名时, 仍然存在第三名。有两个参数, 第一参数为成绩区域, 第二参数为待排名的成绩。参数不局限于区域引用", Category:="andysky 定义函数"
        .MacroOptions Macro:="排序", Description:="按出现次数排序: 对引用数据将出现次数

```

多的字符串排列在第一位, 然后依次降序排列所有数据。有两个参数, 第一参数为数据区域引用, 第二参数为名次, 可使用 ROW(a1)", Category:="andysky 定义函数"

```
.MacroOptions Macro:="SumNum", Description:="对区域中混合字符串中提取数字并求和:
对每个单元格取出数字, 并相加。有一个参数, 即引用区域。", Category:="andysky 定义函数"
```

```
.MacroOptions Macro:="Sums", Description:="按出现次数排序:对引用数据将出现次数
多的字符串排列在第一位, 然后依次降序排列所有数据。有两个参数, 第一参数为数据区域引用, 第二
参数为名次, 可使用 ROW(a1)。", Category:="andysky 定义函数"
```

```
.MacroOptions Macro:="RmSht", Description:="批量修改工作表名:批量地将工作表名
称进行重命名。第一参数表示起始表索引号, 第二参数为结束表索引号, 第三参数是新名称的存放区域
", Category:="andysky 定义函数"
```

```
ThisWorkbook.IsAddin = True '将 sAddin 属性设为 True 再保存
.DisplayAlerts = False
ThisWorkbook.Save
.DisplayAlerts = True
End With
End Sub
```

以上过程用于对每个函数及参数添加提示。其中变量 counts 的作用是区分 Excel 2003 和 Excel 2010, 因为两个版本中部分函数的参数个数不同。

step 2 继续录入以下过程, 用于创建一个菜单及指定快捷键:

```
Private Sub auto_open()
'创建一个新的按钮
With Application.CommandBars(1).Controls.Add(Type:=msoControlButton,
Temporary:=True)
.Caption = "函数说明 (&S)" '指定标题
.OnAction = "加入函数说明" '指定过程
.FaceId = 483 '指定图标
.Style = msoButtonIconAndCaption '指定显示方式
.TooltipText = "单击添加函数说明"
.ShortcutText = "Alt+S" '指定快捷键
End With
Application.OnKey "%s", "加入函数说明" '设定快捷键
End Sub
```

step 3 保存工作簿, 其格式选择 "*.xla", 工作簿名为 "自定义函数.xla", 表示此插件在 Excel 2003 和 Excel 2010 通用。

step 4 通过快捷键【Alt+t+i】加载该函数插件后, 其菜单如图 30.98 所示, 在菜单的提取信息中包括了快捷键【Alt+s】。

step 5 按下快捷键【Alt+s】, 然后使用快捷键【Shift+F3】打开“插入函数”向导, 函数类别中包括了新定义的函数类型。而选择任意函数时, 在其下方将显示该函数的功能说明和每个参数的含义, 如图 30.99 所示。

以上方式执行过程后函数可能有功能提示, 接下来利用立即窗口对函数添加帮助信息。

step 6 进入 ThisWorkbook 代码窗口, 按下快捷键【F4】打开属性窗口, 将 ThisWorkbook 的 IsAddin 属性设置为 false。

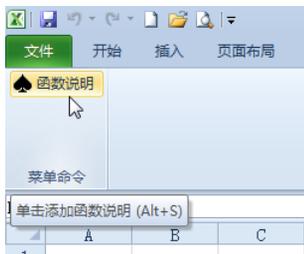


图 30.98 自定义的菜单和快捷键

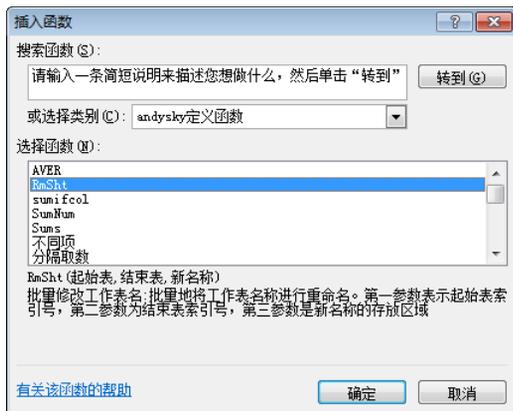


图 30.99 自定义函数与说明

step 7 使用组合键【Ctrl+G】调出立即窗口，并在立即窗口中执行下一语句：

```
If Application.Version * 1 <= 11 Then counts = 30 Else counts = 255: Application.MacroOptions Macro="合并", Description="将区域字符合并: 第一参数为分隔符, 第二参数为引用区域。可以用""表示不使用分隔符。第三参数开始为引用区域, 可以有 1 到" & (counts - 1) & "个区域。还可以直接用文本做参数.", Category:="andysky 定义函数"
```

以上语句可以对“合并”函数添加说明信息，修改内容并执行可以对其余函数添加帮助信息。

step 8 删除所有代码，将 ThisWorkbook 的 IsAddin 属性还原为 true，并保存加载宏。

经过以上设置后，“自定义函数.xla”文件不管在任何电脑上打开，都会在手动执行过程“加入函数说明”之前自动产生函数说明。而需要修改说明信息时，则对过程中的说明进行编辑，再执行过程即可。



本节所有插件及测试文件参见光盘目录：..\第三十章\自定义函数\

30.7 定制百宝箱帮助

Excel 百宝箱是一个大型插件，有必要对所有功能进行简单或者详细的说明。如果设计简单的说明，通常采用一个滚动的窗体对每个功能用文字进行描述；如果设计详细的说明，通常采用 CHM 格式的帮助用户文件和 HTML 格式的网页文件。

30.7.1 定制百宝箱帮助

本章以滚动式窗体设计百宝箱为例，展示一个具有鼠标跟随动画特效的窗体设计方法。本例的滚动式窗体需要具备以下特征：

- ◆ 具有百宝箱 100 个功能简介的文字
- ◆ 文字可以滚动，而鼠标移近文字时则停止滚动

- ◆ 带有鼠标跟随的动画特效
- ◆ 带有意见反馈的链接

具体代码如下:

```
Sub 关于百宝箱() '设计一个动画窗体说明
    On Error Resume Next
    Application.DisplayAlerts = False '不出现提示对话框
    Dim i As Integer, file As String, cell As Range, rng As Range, 地址 As String
    file = ThisWorkbook.Path & IIf(Right(ThisWorkbook.Path, 1) = "\", "hide.hta",
"\hide.hta") '临时文件地址(网页文件)
    地址 = "src=" & "" & ThisWorkbook.Path & IIf(Right(ThisWorkbook.Path, 1) =
"\", "", "\") & "Flash.swf" & "" '利用一幅Flash动画做背景
    i = FreeFile
    Open file For Output Access Write As #i '建立一个网页文件
    '向文件中写入数据
    Print #i, "<html>"
    Print #i, "<head>"
    Print #i, "<meta http-equiv=" + Chr(34) + "Content-Type" + Chr(34) + "
content=" + Chr(34) + "text/html; charset=gb2312" + Chr(34) + ">"
    Print #i, "<title>Alt+F4 关闭</title>" '定制网页的标题
    Print #i, "<embed style=" + Chr(34) + "LEFT: 15px; POSITION: absolute; TOP:
-20px" + Chr(34) + "align=left"
    Print #i, 地址
    Print #i, "width=700 height=550 type=application/x-shockwave-flash wmode="
+ Chr(34) + "transparent" + Chr(34) + "quality=" + Chr(34) + "high" + Chr(34)
+ "></embed>"
    Print #i, "<CENTER>"
    Print #i, "<FONT style=" + Chr(34) + "COLOR:#33FF00; FILTER:
shadow(color=black); FONT-FAMILY: 华文彩云; FONT-SIZE: 30pt; LINE-HEIGHT: 150%;
WIDTH: 100%" + Chr(34) + "><B>百宝箱第七版功能一览表</B></FONT></CENTER>"
    Print #i, "<script language=" + Chr(34) + "JavaScript" + Chr(34) + ">"
    Print #i, "var a_Colour='ff0000';"
    Print #i, "var b_Colour='ffffff';"
    Print #i, "var c_Colour='0000ff';"
    Print #i, "var Size=50;"
    Print #i, "var YDummy=new Array(),XDummy=new"
    Print #i, "Array(),xpos=0,ypos=0,ThisStep=0;step=0.2;"
    Print #i, "if (document.layers){
    Print #i, "window.captureEvents(Event.MOUSEMOVE);"
    Print #i, "function nsMouse(evt){
    Print #i, "xpos = window.pageYOffset+evt.pageX+6;"
    Print #i, "ypos = window.pageYOffset+evt.pageY+16; }"
    Print #i, "window.onMouseMove = nsMouse; }"
    Print #i, "else if (document.all)"
    Print #i, "{ function ieMouse()"
    Print #i, "xpos = document.body.scrollLeft+event.x+6;"
    Print #i, "ypos = document.body.scrollTop+event.y+16;}"
    Print #i, "document.onmousemove = ieMouse; }"
```

```

Print #i, "function swirl(){
Print #i, "for (i = 0; i < 3; i++)"
Print #i, "{ YDummy[i]=ypos+Size*Math.cos(ThisStep+i*2)*Math.sin
((ThisStep+i*25)/2);"
Print #i, "XDummy[i]=xpos+Size*Math.sin(ThisStep+i*2)*Math.sin((This-
Step+i*25)/2)*Math.sin(ThisStep/4); }"
Print #i, "ThisStep+=step;"
Print #i, "setTimeout('swirl()',10); }"
Print #i, "var amount=10;"
Print #i, "if (document.layers){
Print #i, "for (i = 0; i < amount; i++)"
Print #i, "{ document.write('<layer name=nsa'+i+' top=0 left=0
width='+i/2+' height='+i/2+' bgcolor='+a_Colour+'></layer>');"
Print #i, "document.write('<layer name=nsb'+i+' top=0 left=0 width='+i/2+'
height='+i/2+' bgcolor='+b_Colour+'></layer>');"
Print #i, "document.write('<layer name=nsc'+i+' top=0 left=0 width='+i/2+'
height='+i/2+' bgcolor='+c_Colour+'></layer>'); } }"
Print #i, "else if (document.all){
Print #i, "document.write('<div id=" + Chr(34) + "ODiv" + Chr(34) + " style="
+ Chr(34) + "position:absolute;top:0px;left:0px" + Chr(34) + ">"
Print #i, "+'<div id=" + Chr(34) + "IDiv" + Chr(34) + " style=" + Chr(34)
+ "; Position: relative " + Chr(34) + ">');"
Print #i, "for (i = 0; i < amount; i++)"
Print #i, "{ document.write('<div id=x style=" + Chr(34) +
"position:absolute;top:0px;left:0px;width:'+i/2+';height:'+i/2+';background
:'+a_Colour+';font-size: '+i/2+' " + Chr(34) + "></div>');"
Print #i, "document.write('<div id=y style=" + Chr(34) +
"position:absolute;top:0px;left:0px;width:'+i/2+';height:'+i/2+';background
:'+b_Colour+';font-size: '+i/2+' " + Chr(34) + "></div>');"
Print #i, "document.write('<div id=z style=" + Chr(34) +
"position:absolute;top:0px;left:0px;width:'+i/2+';height:'+i/2+';background
:'+c_Colour+';font-size: '+i/2+' " + Chr(34) + "></div>'); }"
Print #i, "document.write('</div></div>'); }"
Print #i, "function prepos(){
Print #i, "var ntscp=document.layers;"
Print #i, "var msie=document.all;"
Print #i, "if (document.layers){
Print #i, "for (i = 0; i < amount; i++)"
Print #i, "{ if (i < amount-1)"
Print #i, "{ ntscp['nsa'+i].top=ntscp['nsa'+(i+1)].top;ntscp['nsa'+i].
left=ntscp['nsa'+(i+1)].left;"
Print #i, "ntscp['nsb'+i].top=ntscp['nsb'+(i+1)].top;ntscp['nsb'+i].
left=ntscp['nsb'+(i+1)].left;"
Print #i, "ntscp['nsc'+i].top=ntscp['nsc'+(i+1)].top;ntscp['nsc'+i].
left=ntscp['nsc'+(i+1)].left; }"
Print #i, "Else"
Print #i, "{ ntscp['nsa'+i].top=YDummy[0];ntscp['nsa'+i].left=XDummy[0];"
Print #i, "ntscp['nsb'+i].top=YDummy[1];ntscp['nsb'+i].left=XDummy[1];"
Print #i, "ntscp['nsc'+i].top=YDummy[2];ntscp['nsc'+i].left=XDummy
[2]; } }"
Print #i, "else if (document.all){ "

```

```
Print #i, "for (i = 0; i < amount; i++) "  
Print #i, "{ if (i < amount-1) "  
Print #i, "{ msie.x[i].style.top=msie.x[i+1].style.top;msie.x[i].style.  
left=msie.x[i+1].style.left; "  
Print #i, "msie.y[i].style.top=msie.y[i+1].style.top;msie.y[i].style.  
left=msie.y[i+1].style.left; "  
Print #i, "msie.z[i].style.top=msie.z[i+1].style.top;msie.z[i].style.  
left=msie.z[i+1].style.left; } "  
Print #i, "else "  
Print #i, "{ msie.x[i].style.top=YDummy[0];msie.x[i].style.left=XDummy  
[0];"  
Print #i, "msie.y[i].style.top=YDummy[1];msie.y[i].style.left=XDummy  
[1];"  
Print #i, "msie.z[i].style.top=YDummy[2];msie.z[i].style.left=XDummy  
[2]; } } "  
Print #i, "setTimeout(" + Chr(34) + "prepos()" + Chr(34) + ",10); }"  
Print #i, "function Start(){"  
Print #i, "swirl(),prepos() }"  
Print #i, "window.onload=Start;"  
Print #i, "</script>"  
Print #i, "</head>"  
Print #i, "<body bgcolor=" + Chr(34) + "#000000" + Chr(34) + ">"  
Print #i, "<HR>"  
Print #i, "<MARQUEE id=help onmouseover=help.stop()"  
Print #i, "Style = " + Chr(34) + "FONT-WEIGHT: bold; COLOR: #00ff00;  
FONT-FAMILY: 宋体_GB2312" + Chr(34)  
Print #i, "onmouseout=help.start() scrollAmount=2 direction=up width=300"  
Print #i, "height=400>"  
Print #i, "<P></P> 交流论坛 :<a href='http://andysky.5d6d.com'  
target='_blank'>Excel 百宝箱论坛</a>"  
Print #i, "<P></P> 百宝箱 7.0 工具简易说明"  
Print #i, "<P></P> 目的: 节约计算量、提升工作效率"  
Print #i, "<P></P> 本工具集合功能简介: "  
Print #i, "<P></P> 菜单工具: 设计工资条 "  
Print #i, "<P></P> 菜单工具: 删除工资条头"  
Print #i, "<P></P> 菜单工具: 根据工资计算所需钞票张数"  
Print #i, "<P></P> 菜单工具: 小写金额转大写"  
Print #i, "<P></P> 菜单工具: 工作簿合并"  
Print #i, "<P></P> 菜单工具: 工作表合并"  
Print #i, "<P></P> 菜单工具: 工作表拆分"  
Print #i, "<P></P> 菜单工具: 工作簿拆分"  
Print #i, "<P></P> 菜单工具: 批量创建复选框"  
Print #i, "<P></P> 菜单工具: 文本与数据互换"  
Print #i, "<P></P> 菜单工具: 合并数据并居中"  
Print #i, "<P></P> 菜单工具: 还原 (快捷键 Alt+Q) "  
Print #i, "<P></P> 菜单工具: 合并数据并复制"  
Print #i, "<P></P> 菜单工具: 取消合并且填充数据"  
Print #i, "<P></P> 菜单工具: 可还原的合并"  
Print #i, "<P></P> 菜单工具: 合并列中相同数据"
```

```

Print #i, "<P></P>" 菜单工具: 取消列中合并还原合并数据"
Print #i, "<P></P>" 菜单工具: 批注管理器"
Print #i, "<P></P>" 菜单工具: 添加个性化批注"
Print #i, "<P></P>" 菜单工具: 建立图片批注"
Print #i, "<P></P>" 菜单工具: 批量添加右列内容为批注"
Print #i, "<P></P>" 菜单工具: 批量导入同名照片到批注"
Print #i, "<P></P>" 菜单工具: 将当前表中的图形对象另存为图片文件"
Print #i, "<P></P>" 菜单工具: 批量导入图片到单元格"
Print #i, "<P></P>" 菜单工具: 将选区另存为图片"
Print #i, "<P></P>" 菜单工具: 多工作表加密解密"
Print #i, "<P></P>" 菜单工具: 工作表反向加密"
Print #i, "<P></P>" 菜单工具: 保护公式"
Print #i, "<P></P>" 菜单工具: 禁用磁盘"
Print #i, "<P></P>" 菜单工具: 分页小计"
Print #i, "<P></P>" 菜单工具: 删除小计"
Print #i, "<P></P>" 菜单工具: 打印当前页"
Print #i, "<P></P>" 菜单工具: 双面打印"
Print #i, "<P></P>" 菜单工具: 预览底端标题"
Print #i, "<P></P>" 菜单工具: 打印底端标题"
Print #i, "<P></P>" 菜单工具: 批量修改标签"
Print #i, "<P></P>" 菜单工具: 批量移动标签"
Print #i, "<P></P>" 菜单工具: 图表输出为图片"
Print #i, "<P></P>" 菜单工具: 提取唯一值"
Print #i, "<P></P>" 菜单工具: 清除列中重复值"
Print #i, "<P></P>" 菜单工具: 不能输入重复值"
Print #i, "<P></P>" 菜单工具: 筛选唯一值"
Print #i, "<P></P>" 菜单工具: 圈释重复值"
Print #i, "<P></P>" 菜单工具: 标示重复值"
Print #i, "<P></P>" 菜单工具: 产生不重复随机数"
Print #i, "<P></P>" 菜单工具: 新建文件夹"
Print #i, "<P></P>" 菜单工具: 新建工作表"
Print #i, "<P></P>" 菜单工具: 搜索文件"
Print #i, "<P></P>" 菜单工具: 批量重命名"
Print #i, "<P></P>" 菜单工具: 建立文件目录"
Print #i, "<P></P>" 菜单工具: 锁定屏幕"
Print #i, "<P></P>" 菜单工具: 查看电脑使用时间"
Print #i, "<P></P>" 菜单工具: 查看磁盘信息"
Print #i, "<P></P>" 菜单工具: 网卡 IP 与 CPU 的 ID 号"
Print #i, "<P></P>" 菜单工具: 查看程序使用端口"
Print #i, "<P></P>" 菜单工具: 清理垃圾文件"
Print #i, "<P></P>" 菜单工具: 奇偶行列选择工具"
Print #i, "<P></P>" 菜单工具: 选择本表所有图片"
Print #i, "<P></P>" 菜单工具: 反向选择"
Print #i, "<P></P>" 菜单工具: 生成系统图标"
Print #i, "<P></P>" 菜单工具: 取得所有菜单"
Print #i, "<P></P>" 菜单工具: 提取所有代码到工作表"

```

```
Print #i, "<P></P>" 菜单工具: 删除所有代码"
Print #i, "<P></P>" 菜单工具: 一键修复 Excel"
Print #i, "<P></P>" 菜单工具: 隐藏非使用区"
Print #i, "<P></P>" 菜单工具: 字母与百家姓序列"
Print #i, "<P></P>" 菜单工具: 清除工作簿外部数据链接"
Print #i, "<P></P>" 菜单工具: 删除工作簿超级链接"
Print #i, "<P></P>" 菜单工具: 隔行插入行"
Print #i, "<P></P>" 菜单工具: 字符统计"
Print #i, "<P></P>" 菜单工具: 对单个字符批量上标"
Print #i, "<P></P>" 菜单工具: 七彩文字"
Print #i, "<P></P>" 菜单工具: 工作表目录"
Print #i, "<P></P>" 菜单工具: 工作表管理 (包括六个功能)"
Print #i, "<P></P>" 右键工具: 选择性粘贴链接"
Print #i, "<P></P>" 右键工具: 返回首页"
Print #i, "<P></P>" 共计 80 个工具"
Print #i, "<P></P>" 自定义函数: 合并"
Print #i, "<P></P>" 自定义函数: 取数"
Print #i, "<P></P>" 自定义函数: 唯一值"
Print #i, "<P></P>" 自定义函数: 消除空值"
Print #i, "<P></P>" 自定义函数: 颜色求和"
Print #i, "<P></P>" 自定义函数: 颜色计数"
Print #i, "<P></P>" 自定义函数: 工作表"
Print #i, "<P></P>" 自定义函数: 分隔取数"
Print #i, "<P></P>" 自定义函数: 相同项"
Print #i, "<P></P>" 自定义函数: 不同项"
Print #i, "<P></P>" 自定义函数: 公式"
Print #i, "<P></P>" 自定义函数: 计算"
Print #i, "<P></P>" 自定义函数: 公式长度"
Print #i, "<P></P>" 自定义函数: sumifcol"
Print #i, "<P></P>" 自定义函数: AVER"
Print #i, "<P></P>" 自定义函数: 排名"
Print #i, "<P></P>" 自定义函数: 排序"
Print #i, "<P></P>" 自定义函数: SumNum"
Print #i, "<P></P>" 自定义函数: Sums"
Print #i, "<P></P>" 自定义函数: RmSht"
Print #i, "<P></P>" 共计 20 个自定义函数"
Print #i, "<P></P>" 特别指出:函数存在 "andysky 定义函数列表中"
Print #i, "<P></P>" 本插件总计 100 个工具"
Print #i, "<P></P>" 有何不足之处请 Mail 至: excelbbx@163.com"
Print #i, "<P></P>"祝各位工作愉快-----andysky (罗刚君)</MARQUEE>"
Print #i, "<HR>"
Print #i, "<hta:application sismenu=no scroll=no border=yes>" '无滚动条、
有边框、无右键菜单
Print #i, "<script language=" + Chr(34) + "vbscript" + Chr(34) + ">"
Print #i, "Me.resizeto 800, 600" '窗体显示为 800*600
Print #i, "</script>"
Print #i, "</body>"
```

```
Print #i, "</html>"
Close (i) '关闭文本文件
Shell Environ$("comspec") & " /c "" & file & """, vbHide '利用 DOS 打开
网页文件
Application.DisplayAlerts = True '恢复提示
End Sub
```

以上代码的思路是：

- ◆ 创建一个 hta 格式的网页文件
- ◆ 向网页中逐句导入网页代码
- ◆ 运用 DOS 技术打开该网页

在网页窗体中有两种动画特效，一是网页代码产生的，一是 Flash 动画产生的。所以在相同的路径下必须放置一个名为“Flash.swf”的 Flash 文件。

执行过程“关于百宝箱”，其效果如图 30.100 所示。其中闪动的星星和蜻蜓由 Flash 动画产生，而跟随鼠标的三色旋转线条则由网页代码产生，向上滚动的文字也由网页代码产生。



图 30.100 Excel 百宝箱的动画帮助窗体

30.7.2 信息反馈

任何插件必须有一个与作者的联系方式，通常包括 QQ 交谈、邮件联系和论坛交流三种方式。本例中同时提供三种方式供插件的用户选用，代码如下：

```
Private Declare Function ShellExecute Lib "shell32.dll" Alias "ShellExecuteA"
(ByVal hWnd As Long, ByVal lpOperation As String, ByVal lpFile As String, ByVal
lpParameters As String, ByVal lpDirectory As String, ByVal nShowCmd As Long)
```

```

As Long
Sub 反馈() '向作者反馈信息
    Dim 方式 As Byte
    On Error Resume Next
    Star:
    方式 = Application.InputBox("输入 1: 与作者 QQ 联系" & Chr(10) & "输入 2: 与作者邮箱联系" & Chr(10) & "输入 3: 上论坛与作者交流", "确认反馈方式", 2, , , , 1)
    If err <> 0 Then err.Clear: GoTo Star
    If 方式<> 1 And 方式<> 2 And 方式<> 3 Then MsgBox "只能输入 1、2 或者 3", 64: GoTo Star
    Star
    Select Case 方式
    Case 1
        Shell "explorer.exe "Tencent://Message/?Menu=YES&Exe=&Uin=670218239
        ""
    Case 2
        Dim MyMail As String
        MyMail = "mailto:andy_qc@163.com?subject=关于 Excel 百宝箱 7.0 的建议
        &body=" & "关于 Excel 百宝箱 7.0 版, 我向您提出以下三点建议:%0A1:%0A2:%0A3:%0A%0A
        用户: " + Application.UserName
        ShellExecute 0&, vbNullString, MyMail, vbNullString, vbNullString, 1
    Case 3
        Shell "explorer.exe http://andysky.5d6d.com/bbs.php"
    End Select
End Sub

```

以上过程首先让用户选择三种方式之一, 如果输入数值不等于 1、2 或者 3 则返回重新输入, 然后利用 Select Case 语句根据用户的输入值决定反馈方式。如果是 1 则打开一个 QQ 与对方交流, 如果输入 2 则可向作者发送邮件, 如果输入 3 则打开交流论坛。

执行过程“反馈”, 将弹出图 30.101 所示对话框。如果输入 1 并单击“确定”按钮, 将弹出 QQ 聊天窗口, 立即与作者的 QQ 交流。

如果输入 2 并单击“确定”按钮, 那么将打开如图 30.102 所示窗口, 向作者发送邮件, 而且邮件的收件人与主题、正文等等都已在 VBA 中定义完成, 用户仅仅录入建议即可。



图 30.101 确认反馈方式

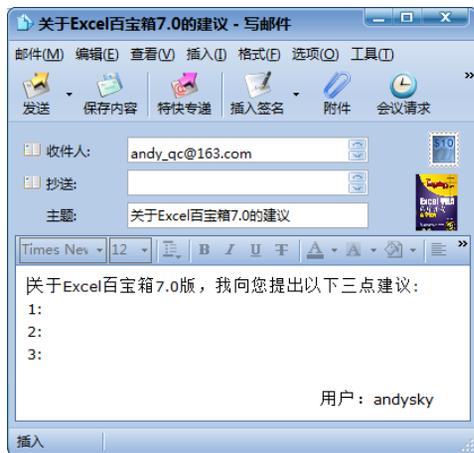


图 30.102 用邮件反馈信息

30.8 定制多级菜单并发布

Excel 百宝箱有 100 个功能，除自定义函数外，剩下的过程通常用快捷键和菜单来调用。

而快捷键不宜过多，用户无法记忆。本例中仅仅“还原”使用【Alt+S】调用，其他工具全部通过工作表菜单来调用。

30.8.1 定制菜单

Excel 百宝箱的菜单较多，主要分为四类：

- ◆ 工作表弹出式菜单
- ◆ 工作表右键菜单
- ◆ 分页预览状态下的工作表右键菜单
- ◆ 工作表标签右键菜单

创建菜单的代码如下：

```
Private Sub auto_open() '打开插件时创建菜单
    On Error Resume Next
    Dim Menu As CommandBarControl, SubMenu As CommandBarControl
    Set SubMenu = Application.CommandBars("Worksheet Menu Bar").
Controls.add(msoControlPopup, 1, , 3, 1) '创建工作表弹出式菜单
    SubMenu.Caption = "百宝箱(&B)" '指定名称。然后生成“财务工具箱”弹出式菜单
    With Application.CommandBars(1).Controls("百宝箱(&B)").Controls
        Set SubMenu = .add(msoControlPopup, 1, , , 1) '创建二级弹出式菜单
        SubMenu.Caption = "财务工具箱(&Financial_Tools)" '对二级菜单命名
        '从这一句开始，创建三级菜单，即“财务工具箱”的子菜单
        With SubMenu.Controls.add(msoControlButton, 1, , , True) '创建第三级子菜单
            .Caption = "制作工资条头(&Pay_head)" '指定子菜单对应的显示标题
            .OnAction = "生成工资条" '指定子菜单对应的宏
            .Style = msoButtonIconAndCaption '让子菜单同时显示文字与图标
            .FaceId = 374 '指定子菜单的图标
        End With
        .....
        '限于篇幅，不再罗列后面的菜单代码，其方法与第一个二级菜单和第一个三级子菜单完全一致，
        仅仅复制后修改对应的显示标题、宏与图标即可。完整代码见随书光盘中“百宝箱.xla”
    End With
End Sub
```

创建菜单时必须注意，其“OnAction”属性必须对应 Sub 过程名称，而编写 Sub 过程特别是调用窗体的 Sub 过程时，一定要确保过程不能与窗体同名，否则菜单无法调用过程，窗体也无法加载。例如窗体名称是“合并工作表”，那么调用窗体的 Sub 过程就不再能用该名称，通常改用“工作表合并”等意思相近的名称，或者改用英文。

执行以上过程后，百宝箱菜单的外观如图 30.103 所示。

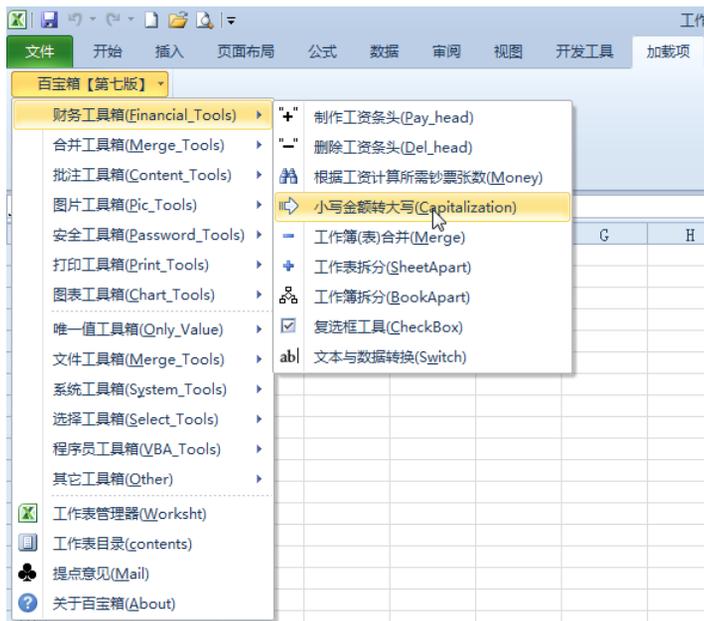


图 30.103 百宝箱外观

30.8.2 发布

Excel 百宝箱的代码是 Excel 2003 和 Excel 2010 通用的,而且所针对用户领域也是 Excel2003 和 Excel 2010 用户,那么插件必须使用通用格式——“*.xla”。具体步骤如下:

- step 1** 插件开发完成后对工程设置保护,使用最简单的密码即可(加密码是为了防止无意中破坏代码)。
- step 2** 单击“保存”,从“另存为”对话框中的“保存类型”下拉列表中选择“Excel 97-2003 加载宏 (*.xla)”,命名为“百宝箱.xla”。
- step 3** 关闭 Excel,在资源管理器中对插件单击右键,从菜单中选择【属性】,并进入“摘要”选项卡,将其标题设置为“Excel 百宝箱”;并在备注框中录入“Excel 百宝箱:包括 100 个功能,其中 79 个菜单工具,一个快捷键工具,还有 20 个自定义函数”,从而使“加载宏”对话框中可以展示插件的名称与功能。

由于 Excel 百宝箱插件包括“百宝箱.xla”、“Flash.swf”和“自定义函数.xla”三个文件,不适宜放在自启动文件夹中启动插件,用户需要通过快捷键【Alt+t+i】打开“加载宏”对话框的方式来安装插件。



Excel 百宝箱源代码参见光盘: ..\ 第三十章\百宝箱.xla

30.8.3 小结

虽然 VBA 的各种语法总是固定的,但开发插件与编写宏在思路上有较大的差异。特别是大型

的插件，必须具有足够的自定义空间、清晰整洁的界面、完善的防错机制以及统一风格的菜单，还有对当前主流版本的兼容性、完善的帮助系统，并提供论坛地址或者邮箱地址让用户反馈信息等。

本书通过多个大中型的插件讲述了设计插件的步骤和思路，特别在如何防错、兼容多版本以及自定义选项等方面需要读者多加琢磨。读者通过本书能开发自己的专业插件是作者的唯一心愿。

