

Excel 的高级应用

——物流与财务

杨密 杨乐 马轶雄 著



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

Excel 的高级应用

——物流与财务

杨 密 杨 乐 马轶雄 著

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书是针对当前不断提高的应用水平并根据多年的使用经验编著的。本书将 Excel 作为平台,详细介绍了如何构成完整的应用系统。全书共分 11 章,前 5 章主要介绍基本的使用方法,后 6 章给出一些具体的案例,这些案例都是企业管理中最常见、最需要的应用方法。这些案例涉及的领域较宽,包括财务管理、库存管理、人事管理、工程管理以及生产计划等。本书介绍了双文件的处理方式,即一个文件放数据、另一个文件放程序。这样的方法既方便使用,也方便软件开发。本书的另一个特点是 Excel 2003 和 Excel 2007 都可以使用。实际上,因为 Excel 2007 单元格数量的大大增加,更加适合用来做数据库的平台。本书中所有的举例都在本书随附的光盘中,读者可以随意使用,从而更加有助于学习与研究。

本书的读者对象是财务人员、物流管理人员、企事业管理人员或软件开发人员,也可作为高等院校的辅助参考书。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

Excel 的高级应用:物流与财务/杨密,杨乐,马轶雄著. —北京:电子工业出版社,2009.1

ISBN 978-7-121-07662-6

I. E… II. ①杨…②杨…③马… III. ①电子表格系统, Excel—应用—物流—物资管理②电子表格系统, Excel—应用—财务管理 IV. F252-39 F275-39

中国版本图书馆 CIP 数据核字(2008)第 168305 号

责任编辑:孙延真

印 刷: 北京牛山世兴印刷厂
装 订:

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1092 1/16 印张: 15.5 字数: 400 千字

印 次: 2009 年 1 月第 1 次印刷

印 数: 4000 册 定价: 36.00 元(含光盘 1 张)

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888。

质量投诉请发邮件至 zlts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010) 88258888。

前言

“Excel 有什么作用？”或者“什么样的工作需要使用 Excel？”等类似的问题现在很少听到了，然而，要言简意赅地给 Excel 下一个定义仍然不那么容易。一般认为 Excel 可以进行数据分析，财务人员最需要使用。笔者经过长期的使用和体会，可以给出这样的结论：Excel 更像一个平台，一个可以进行数据处理的平台。

平台的概念相当于一个环境，可以进行二次开发，本书就是根据二次开发的原则编写的。全书共分 11 章，前 5 章针对基本的使用方法，后 6 章给出一些具体的案例。像 Excel 这样的软件在应用时大体有 3 个过程：输入、处理（分析）、输出。本书以这 3 个过程为线索进行了系统的阐述，旨在将 Excel 构建成一个完整的系统。

本书介绍了 Excel 中数据的窗体输入方法，虽然 Excel 本身可以很方便地输入，但如果要应对复杂的输入（例如一对多的情况）就必须使用特定的窗体，本书的例子是财务管理的凭证输入。数据源既可以是自身输入的，也可以是来自其他软件（如 Access 等数据库）的。书中介绍了和数据库软件的交互使用，各取所长。有一部分数据取自金蝶财务软件。相信无论使用哪一种财务软件，都可以采用类似的方法进行有效的数据分析。

就数据的分析方面，本书讨论了多条件汇总的方法，很多人都对多条件汇总的方法感到困惑。Excel 的透视表功能是十分有用的，简单的使用可以按软件中提供的向导完成，而本书比较系统地介绍了用程序制作透视表的方法。针对固定的分析模型，用程序制作的透视表更加稳定适用。例如，有时候需要一项多列，用向导制作并不那么容易，程序才是更好的选择。

数据输出常见的方式包括屏幕和打印两种，打印输出需要考虑打印的效果，本书介绍了一些打印数据的技巧，有些特定的格式，如工资单采用了更合理的方法，直接在 Excel 环境中调用 Access 中的报表对象。

为了方便使用和开发，本书介绍了双文件的处理方式，即一个文件放数据、另一个文件放程序。使用的时候，用户只需要面对数据文件，程序文件隐含在内存中提供支持。这样的用法特别适合企业的物料管理，例如，在库存管理中，不同的物料应当由不同的部门、不同的人员来处理，但其处理方法应当基本相同，这时候，不同的物料使用不同的数据文件，但采用了相同的程序文件。这样一来，既方便使用，也方便软件开发。

本书的另一个特点是 Excel 2003 和 Excel 2007 都可以使用。就 VBA 的应用方面，2007 版比 2003 版并没有更多的不同，主要的不同是在操作界面和每个工作表的单元格数目上。每张工作表 2003 版中的单元格为 65536 行、256 列，而 2007 版的单元格为 1048576 行、16384 列，前者的单元格数目为 16777216，后者为 17179869184，增加了 1024 倍。这样一来，在编写程序时应当有所考虑。例如，在工作表中要得出某一列（例如 A 列）最后未使用单元格在哪一行，在 2003 版中应当写成：`[A65536].End(Xlup).row`，而在 2007 版中就应当写成：`[A1048576].End(Xlup).row`。实际上，因为单元格数量的大大增加，2007 版更加适合用做数据

库的平台。

为了帮助读者学习和研究,本书中所有的举例都配有电子文件,文件在本书随附的光盘中,读者可以随意使用,电子文件的文件名或目录名(双文件方式存放在单独的目录中)和书中的章节相关,很容易对照。

作者

2008年10月

目 录

第 1 章 从库存管理谈起	(1)
1.1 库存管理的基本要求	(1)
1.2 用软件计算出库单价	(1)
1.2.1 加权平均法	(2)
1.2.2 移动平均法	(2)
1.2.3 两种方法的比较	(2)
1.3 工作表的应用	(3)
1.3.1 工作表的准备	(3)
1.3.2 工作表改变事件	(6)
1.4 库存数据自动处理	(12)
第 2 章 工作表使用下拉列表	(17)
2.1 下拉列表的意义及方法	(17)
2.2 使用命名定义列表范围	(19)
2.3 使用事件生成列表	(23)
2.4 列表的使用方法	(25)
2.5 另一种类型的应用	(30)
2.6 控件的用法 (ComboBox)	(35)
2.7 控件的用法 (ListBox)	(39)
第 3 章 数据处理与窗体	(43)
3.1 窗体的作用	(43)
3.2 数据输入窗体	(44)
3.3 数据查询窗体	(52)
3.4 数据打印窗体	(59)
3.5 构成一对多关系的窗体	(63)
第 4 章 数据与程序	(74)
4.1 概述	(74)
4.2 工作簿事件取代工作表事件	(74)
4.3 数据与程序分离	(76)
4.3.1 数据文件的构成	(79)
4.3.2 程序文件的构成	(80)
4.4 数据文件与程序文件的结合	(87)
第 5 章 汇总报表	(88)
5.1 概述	(88)
5.2 数据汇总的函数方法	(88)
5.2.1 函数汇总方法之一	(91)
5.2.2 函数方法汇总之二	(93)

5.3	数据汇总的透视表方法	(96)
5.3.1	完整的数据透视表	(97)
5.3.2	常用的数据透视表	(100)
5.3.3	多列数据的透视表	(101)
5.4	独立文件的报表	(104)
第 6 章	财务报表	(107)
6.1	概述	(107)
6.2	当月数据	(108)
6.3	现金记录	(109)
6.4	汇总数据	(116)
6.5	报表与科目表(财务报表的过渡数据)	(118)
6.6	财务报表	(120)
6.7	规定格式的财务报表	(128)
第 7 章	考勤数据分析	(131)
7.1	概述	(131)
7.2	数据源与数据流	(131)
7.3	数据处理的步骤	(134)
7.4	数据分析结果	(147)
第 8 章	工资管理	(148)
8.1	概述	(148)
8.2	涉及的工作表	(148)
8.3	数据处理的步骤	(154)
8.4	打印工资单和银行数据	(176)
8.5	提供成本分类数据	(185)
第 9 章	多条件汇总	(193)
9.1	从 SUMIF 函数谈起	(193)
9.2	多条件汇总的方法	(195)
第 10 章	项目管理	(203)
10.1	概述	(203)
10.2	项目预算(项目工作表)	(203)
10.3	招标投标(乙方工作表)	(207)
10.4	签订合同(合同工作表)	(209)
10.5	合同付款(工程进度与付款工作表)	(214)
10.6	使用建议	(216)
第 11 章	生产计划的算法	(217)
11.1	概述	(217)
11.2	相关的术语及说明	(218)
11.3	数据结构	(222)
11.4	计算程序	(225)
11.5	生产计划任务单	(235)

第1章 从库存管理谈起

1.1 库存管理的基本要求

在企业管理中,库存管理是十分重要的,直接关系到企业的资产状况。无论是贸易型企业还是生产型企业,怎样使企业的存货做到账实相符,一向都不是个简单的问题。库存管理的基本要求可以分为两个部分:货物管理和账务处理。

先讨论货物管理。

库存管理就库存货物本身考虑首先应当详细记录货物入库和出库的数据,然后可以实时提供库存货物的信息。进一步的要求是在货物入库时记录供货人,货物出库时记录领用人。这些要求都是很容易做到的。在实际操作中还可能需要进行货龄分析,即提供货物在仓库的存放时间,其目的往往是为了做到先进先出,不要使货物在仓库中滞留太久。这项要求需要将货物和存放货物的场地相关联,为此,必须把场地用坐标标注。具体的操作方法将在后面结合软件使用一同介绍。

其次讨论货物账务处理。

库存管理如果就事论事地针对货物本身,并不复杂,但如果涉及账务处理,就麻烦了许多。两者的区别在于,货物管理只考虑数量,账务处理却要考虑金额。

下面以发出库存货物为例,分析发出商品会产生的账务分录。

- | | |
|-----------------|---|
| 1. 借: 相应的库存商品成本 | A |
| 贷: 减少的库存商品价值 | B |
| 2. 借: 应收账款 | C |
| 贷: 销售额 | D |

以上两笔分录, A 和 B 的金额相等, C 和 D 的金额相等。财务的意义不在此进行讨论,它们共同的关联是出库的数量。第 2 笔分录可以脱离库存管理单独考虑,只要记录库存商品的品种和数量,其单价及金额可能各不相同,由销售的时候确定。然而,第 1 笔分录的金额一般应当结合库存管理考虑。例如一个贸易企业的仓库,或者企业的原材料仓库,其库存商品的入库价值就是采购的金额,可是每一次采购的单价并不相同,这就需要采用某种方法计算出库时的商品价值。比较常见的方法是加权平均的方法。加权平均需要按月来计算,就是说到了月末,把所有出库的货物用加权平均方法得出其成本。然而,如果每一次出库时都想知道货物的成本,就必须采取移动加权平均的方法。

1.2 用软件计算出库单价

在设计库存软件的时候,库存货物的出库单价是个很值得研究的问题。

库存货物每次入库（采购）的单价往往不同，出库时的单价就必须采用某种计算方法计算得出，然后，再根据数量和单价的乘积得出出库金额，以此作为记账的依据。

虽然计算出库单价的方法有若干种，但一般企业普遍使用的是加权平均法，这种方法又分为全月一次加权平均（简称加权平均）和移动加权平均（简称移动平均）。这两种方法的共同特点是，在入库价格波动较大的情况下得出相对平稳的出库价格，从而避免计算出的成本起伏过大。

1.2.1 加权平均法

加权平均法的计算公式如下：

加权平均单价 = (期初结存存货实际成本 + 本期收入存货实际成本) ÷ (期初结存存货数量 + 本期收入存货数量)

本期发出存货成本 = 本期发出存货数量 × 加权平均单价

优点：核算的工作量比较少。

缺点：计价工作都在月末进行，过分集中，影响核算进度，而且平时不能看出发出材料的成本和结存存货成本。

1.2.2 移动平均法

移动平均法是指每次收货以后，立即根据库存存货数量和成本，计算出新的移动平均单价的一种方法。移动平均法与上面所讲的加权平均法的计算原理基本相同，只是要求在每次收入存货时重新计算加权平均单价。计算公式如下：

移动平均单价 = (前一次存货实际成本 + 本次收入存货实际成本) ÷ (前一次结存存货数量 + 本次收入存货数量)

本期发出存货成本 = 本期发出存货数量 × 移动平均单价

优点：材料计价工作可分散在月内进行，随时反映发出材料的实际成本。

缺点：每收进一批材料就要重新计算一次单价，核算工作繁重。

1.2.3 两种方法的比较

为了更加明确，用实际的例子进行一次对比。

我们的例子共有 5 次入库、出库操作。在表 1-1 中，出库金额和期末金额没有数值，只有到需要计算的时候（往往在月末）才得出一个出库单价，于是，在此计算时段中所有的出库使用统一的单价，出库金额和期末金额只有在汇总时计算出来。表 1-2 就不一样了，每一次入库或出库都需要期初值，每次出库都得出出库金额，并且计算出期末值作为下一次的期初值。

以上的比较不难得出结论，移动平均方法更能及时反映价格的变动情况。当然，其缺点是计算比较复杂。可是，如果用计算机处理，就可以克服这个缺点。

可以这样认为：加权平均和移动加权平均本质是一样的，加权平均是每月移动一次的移动加权平均。

表 1-1 加权平均方法的计算

日期	说明	期初		入库			出库		期末	
		数量	金额	数量	单价	金额	数量	金额	数量	金额
11/8	购入			300	3.00	900			500	
11/14	发出					-	400		100	
11/18	购入			400	1.15	460			500	
11/20	发出					-	300		200	
11/28	购入			100	1.20	120			300	
汇总		200	200	800	1.68	1,480	700	1,176	300	504

表 1-2 移动平均方法的计算

日期	说明	期初		入库			出库		期末	
		数量	金额	数量	单价	金额	数量	金额	数量	金额
11/8	购入	200	200	300	3.00	900			500	1,100
11/14	发出	500	1,100			-	400	880	100	220
11/18	购入	100	220	400	1.15	460			500	680
11/20	发出	500	680			-	300	408	200	272
11/28	购入	200	272	100	1.20	120			300	392
汇总		200	200	800		1,480	700	1,288	300	392

1.3 工作表的应用

现在讨论工作表的应用主要是针对库存管理的要求。本章十分重要，一方面介绍了 Excel 工作表使用的不同的思路；一方面为读者提出本书的基本构架。

1.3.1 工作表的准备

根据上面关于移动平均法的要求，可以得知每次入库时应当计算一次当前的单价，供出库时计算使用。入库时必须输入数量和金额，入库的数量和金额和当前库存的数量和金额分别相加，金额的和除以数量的和就是当前出库时的单价。每次出库时，只要输入数量，金额就可以自动计算出来。为了实现这样的目的，首先应当有两个区域：出库入库区域和库存区域。比较理想的方法是用两个工作表，一个解决库存，一个输入出库入库记录，参见图 1.1。

单证号	代码	名称	单位	入库数量	入库金额	出库数量	出库金额	出入	出入日期	月份	状态
2	M02	物品02	公斤	300	900.00			入库	2007-12-18	2007-12	1
3	M02	物品02	公斤			400	880.00	出库	2007-12-18	2007-12	1
4	M02	物品02	公斤	400	460.00			入库	2007-12-18	2007-12	1
5	M02	物品02	公斤			300	408.00	出库	2007-12-18	2007-12	1
6	M02	物品02	公斤	100	120.00			入库	2007-12-18	2007-12	1

图 1.1 用于输入入库、出库的工作表

我们希望在图 1.1 的工作表中输入信息,在图 1.2 的工作表中得出结果,必须用程序才可以做到,而程序都必须有一个执行触发器(或者说调用),否则程序不能起作用。一般情况下,事先写好的程序可以通过菜单的“工具/宏”调用,这种调用就是一种触发器。但这样做称不上自动,若要自动就必须采用另外的触发方式。在 Excel 中这样的触发器已经事先准备好了,那就是工作表的事件。工作表有若干事件,可以在不同的情况下发生,从而可以根据需要选择使用。从未使用过事件的人开始会觉得有些无从下手,实际上并非那么难,可以从比较简单、直观的事件着手,例如下面的事件。

	A	B	C	D	E	F	G	H	I	J	K	L	M
	库存项目				期初		入库		出库		期末		
1	分类	代码	名称	单位	数量	金额	数量	金额	数量	金额	数量	金额	移动单价
2	原材料	M01	物品01	公斤	0	0.00	0	0.00	0	0.00	0	0.00	
3	原材料	M02	物品02	公斤	200	200.00	800	1,480.00	700	1,288.00	300	392.00	1.31
4	原材料	M03	物品03	公斤	0	0.00	0	0.00	0	0.00	0	0.00	
5	原材料	M04	物品04	公斤	0	0.00	0	0.00	0	0.00	0	0.00	
6	原材料	M05	物品05	公斤	0	0.00	0	0.00	0	0.00	0	0.00	
7	原材料	M06	物品06	公斤	0	0.00	0	0.00	0	0.00	0	0.00	
8	原材料	M07	物品07	公斤	0	0.00	0	0.00	0	0.00	0	0.00	
9	原材料	M08	物品08	公斤	0	0.00	0	0.00	0	0.00	0	0.00	
10	原材料	M09	物品09	公斤	0	0.00	0	0.00	0	0.00	0	0.00	
11	原材料	M10	物品10	公斤	0	0.00	0	0.00	0	0.00	0	0.00	

图 1.2 用于汇总的工作表

首先解释怎样进入事件。

事件是针对工作表和工作簿的,每一个工作表都具有相同的事件。在 Excel 工作表的状态下单击菜单的宏/Visual Basic 编辑器就可以进入工作表或工作簿的事件状态,参见图 1.3。

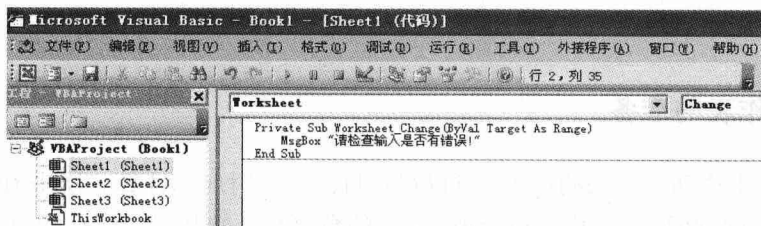


图 1.3 工作表的事件编辑

图 1.3 可以说明事件的名称以及属于哪个工作表。显然,当前的事件为工作表更改事件,针对工作表“Sheet1”。事件的基本语法构成:以 Private Sub 事件名称开始,以 End Sub 结束。在基本语法构成中间的内容就是根据需要写入的语句。可见,在使用事件的时候,首先要问一问为什么要使用事件,再了解需要什么样的事件,最后结合该事件写入执行指令。

结合图 1.3,本例的目的是在每次输入的时候提醒用户检查输入的内容是否正确。在每次输入数据后都将造成工作表的改变,那么,应当选什么事件呢?即应当使用本事件。写法如下:

```
Private Sub Worksheet_Change(ByVal Target As Range)
    MsgBox "请检查输入是否有错误!"
End Sub
```

当用户更改工作表中的任何单元格时产生此事件, Target 为更改的区域。输入、修改和删除都是更改。当在工作表中输入任何值的时候, 就会跳出一个对话框, 提醒用户检查输入的结果, 本例输入了“123”, 参见图 1.4。

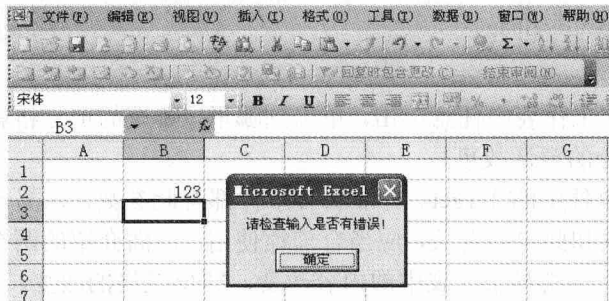


图 1.4 事件应用举例

如果把事件的语句写得复杂一些, 还可以有另外的效果。请参见下面的语句。

```
Private Sub Worksheet_Change(ByVal Target As Range)
    If MsgBox("输入的数据正确吗?", vbYesNo + vbQuestion) = vbNo Then
        Application.EnableEvents = False
        Application.Undo
        Application.EnableEvents = True
    End If
End Sub
```

以上的语句在用户输入数据后提出疑问, “输入的数据正确吗?” 如果回答是, 输入的数据将保留, 如果回答否, 输入的数据将删除, 恢复到数据输入前的状态, 参见图 1.5。

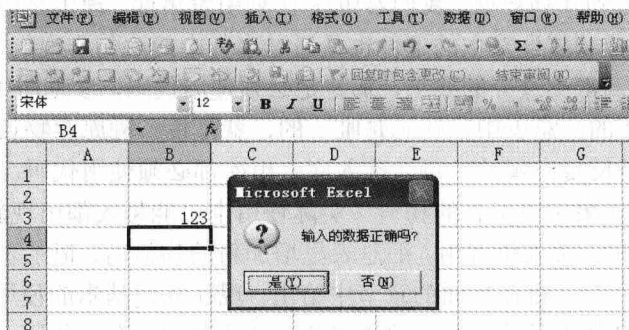


图 1.5 事件应用举例

为了更加明确, 不妨用另一个事件进行比较, 即工作表的选择改变事件。

```
Private Sub Worksheet_SelectionChange(ByVal Target As Range)
    ... ..
End Sub
```

工作表选择改变和工作表改变不同, 选择改变只要光标在单元格的位置有了变化就会激活该事件。例如原来是“A1”, 现在选择“C3”。而工作表改变则必须造成工作表的变化, 例如,

输入了内容或者删除了数据等。在工作表改变事件中, 参数 `Target` 是造成改变的区域, 而在工作表选择改变中是新选定的区域。用下面的程序做例子:

```
Private Sub Worksheet_SelectionChange(ByVal Target As Range)
    Target.Interior.ColorIndex = 3
End Sub
```

在上例中, 光标在工作表中任意单击, 单击到哪里, 哪里的单元格就被红色填满, 很容易让人想起那个点石成金的童话故事。

请特别注意以上事件中的 `Target`。并非所有事件都有此参数, 从另一个角度理解, 就是既然有该参数, 该事件就和此参数有关, 应当合理地使用。前面介绍的两个事件, 即工作表改变和工作表的选择改变应该都是工作表中的内容改变, 即单元格的改变, 也就是 `Target`。`Target` 可以是一个单元格, 也可以是多个单元格, 完全可以根据需要予以利用。

1.3.2 工作表改变事件

现在根据库存管理的需求讨论事件的应用。我们有两个工作表, 记录表 (图 1.1) 和汇总表 (图 1.2), 记录表是用户始终面对的表, 也可称之为输入窗体、输入界面等。也就是说, 事件应当出现在记录表上。记录表的数据输入将引起汇总表的数据变化。具体地说, 在记录表输入入库信息时汇总表将给出新的单价, 在记录表输入出库信息时, 只要输入数量, 金额将自动计算出来。请参照图 1.2 的 M 列, 移动单价。该单价在某种物料入库时就需要更新, 在某种物料出库时就需要取出来用于出库金额的计算。

虽然已经明确针对记录表使用工作表改变事件, 但还需要考虑究竟应当怎样用。在工作表改变事件中工作表上任何单元格的改变都会引发事件, 那么就要考虑应当在怎样的条件下使用。在图 1.1 中可以见到 9 列数据, 我们希望在 H 列引发事件, 理由是应当把所有的信息输入完成后再激活事件。控制事件发生的条件也是通过事件中的语句实现的。

还有一项怎么强调都不过分的要求, 那就是必须使用代码, 参见图 1.1 的“B”列和图 1.2 的“B”列。在图 1.2 的汇总表中, 代码是唯一的, 就是说一种库存物料对应一个代码。在图 1.1 的记录表中代码是反复出现的, 每一次入库、出库都必须使用代码。可以想象, 代码应当先出现在汇总表中, 如果出现了新的物料 (实际使用中可能每天都出现新的物料), 应当先在汇总表里增加, 然后在记录表里调用。汇总表里增加新的物料后, 记录表里就可以提供选择清单, 也是本书的重点之一, 将在后面的章节中给出。现在强调基本的使用规则, 代码和名称必须都是唯一的。输入代码时给出名称, 输入名称时给出代码。有了代码, 每次输入数据时记录表上的代码去寻找汇总表上的代码, 一旦吻合 (肯定可以找到), 就进行相应的处理。

有了上面的铺垫, 就可以针对事件进行更加详细的阐述。以下面的程序为例。该程序用于工作表改变事件, 每条语句后面都用序号注明, 便于解释。为了方便阅读, 程序分段说明, 但序号不中断, 读者可以根据序号得知其先后顺序。

程序 1-1: 计算出库单价。

```
Private Sub Worksheet_Change(ByVal Target As Range) 1
    Dim mRng As Range, r As Range 2
```

Dim i As Long, j As Long	3
Dim c As Currency, p As Currency	4
Dim n As Double	5
If Target.Row < 2 Or Target.Cells.Count > 1 Then Exit Sub	6

第1行和第125行语句是固定的语法格式。工作表事件的基本要求。

第2行到第5行的语句定义了内存变量，这些变量将在后面的过程中使用。应当特别注意第2行的变量，这两个变量是对象变量，对它们赋值的方法不同，必须以Set开始，在后面的介绍中将会涉及。而且，对象在Excel使用中是很重要的概念。首先应当记住3个对象：工作簿（每一个Excel文件）、工作表（工作簿中包括的工作表）、单元格（工作表中所有的单元格）。第2行就是定义的单元格对象，单元格对象可以是一个单元格，也可以是若干单元格组合。

为了程序结构更加清晰，可以在同一行中定义多个变量，例如第2、3、4行。需要注意的是，对于在同一行中定义的变量，应当在每个变量后面都加上As关键字和数据类型，不然系统将变量默认为Variant数据类型，而该类型的变量占用内存较大，需要更多的资源。

第6行是为了避免输入时可能出现的错误。也就是说，如果出现这样的情况，程序就不再继续向下执行。出现2种情况的任何一种，第6行以后的语句将不再执行：目标行小于2（第一行是表头）；目标单元格大于1（十分重要，可以保证复制操作正常实施）。

i = ActiveWorkbook.Worksheets("汇总信息").Range("B65536").End(xlUp).Row	7
j = ActiveWorkbook.ActiveSheet.Range("B65536").End(xlUp).Row	8

第7、第8行对两个变量赋值，采用了相同的方法得出两个工作表已经使用的最后一行的行坐标。使用这样的方法需要注意一点，所选用的列必须适当。该列中的数据应当为必填数据，本例用第2列作为选择列。“B65536”是B列最后一个单元格（由此可知一个工作表共有65536行），有时候，如果不能确定采用哪一列进行测试最合适，就要以使用数据最多的列为参照。

如果从A列到K列，不能确定哪一列的数据最多，可以采用其他方式，例如可以这样表达：

“Range(“A:K”).Find(“*”,,,,xlPrevious).Row”；

或者干脆所有的列“Range(“A1”).CurrentRegion.Rows.Count”，使用这样的方法需要注意中间不能有无数据的空列。

另外，还可以用“Range(“IV1”).End(xlToLeft).Column”得出被使用的最后一列的列坐标。

If Target.Row < j Then	9
MsgBox "不可以更改历史数据!"	10
Application.EnableEvents = False	11
Application.Undo	12
Application.EnableEvents = True	13
Exit Sub	14
End If	15

第9行到第15行，用来限制用户的不适当操作。本例的要求是能够处理移动平均的计算，那么，已经输入的数据是不可以更改的。第8行得出了最后一行的行坐标，本列为6，如果要对第5行的数据进行任何修改都会遭到拒绝，而且修改的内容也会恢复原样。

以下自第 16 行到第 124 行有一个 Select Case End Select 语句。该语句针对特定的条件采用相应的处理。一般说来, Select 语句可以用 If 语句所替代, 但 Select 语句的逻辑结构更加清晰, 或许也可以认为是编程人员的个人选择。本例是以目标单元格的列坐标为特定条件的, 当列坐标为 2、3 和 9 的时候, 需要有不同的处理, 否则就退出, 也就是说有 4 种情况。

Select Case Target.Column	16
Case Is = 2	17
Set mRng = ActiveWorkbook.Worksheets("汇总信息").Range("B3:B" & i)	18
For Each r In mRng	19
If Target.Value = r Then	20
Target.Offset(, 1) = r.Offset(, 1)	21
Target.Offset(, 2) = r.Offset(, 2)	22
Exit For	23
End If	24
Next r	25

第 16 行得出列坐标, 第 17 行到第 25 行是针对列坐标为 2 的处理。现在回到图 1.1, 本例希望在输入了物料代码后可以将物料名称和单位自动在第 3、第 4 列填入, 而有关的信息都在汇总表中。那么, 处理的方法是将第 2 列的目标单元格内容和汇总表上的单元格进行比对, 如果找到了相同的内容就把相关信息填入。自第 19 行到第 25 行是一个循环语句, 根据单元格对象 (第 18 行的定义) 每个单元格过一遍。如果某个单元的值和目标单元格的值相同, 就把相应的值取出来。应当特别注意 Offset(,1)表达式, 此含义是某个单元格向右偏移一格的另一个单元格, Offset(,2)意为向右偏移两格。本例中单元格的相对值都相同, 偏移一格是名称, 偏移两格是单位, 于是有了第 21 行和第 22 行的语句。

Case Is = 3	26
Set mRng = ActiveWorkbook.Worksheets("汇总信息").Range("C3:C" & i)	27
For Each r In mRng	28
If Target.Value = r Then	29
Target.Offset(, -1) = r.Offset(, -1)	30
Target.Offset(, 1) = r.Offset(, 1)	31
Exit For	32
End If	33
Next r	34

第 26 行到第 34 行的目的和前面第 17 到第 25 行相同, 但由于对比的目标单元格和对象单元格不同, 因此偏移量就不同了, 请特别注意第 30 行, 偏移值为-1, 就是说向左移一格。

以下从第 35 行到第 121 行是当列坐标为 9 时执行的代码, 也就是图 1.1 中的“1”列。

Case Is = 9	35
If Target.Offset(, 3) = 1 Then	36
MsgBox "输入已经完成, 不需要重复操作!"	37
Exit Sub	38
End If	39

在编程的时候需要考虑多种因素,有些需要考虑用户的操作方便,有些需要考虑避免错误。前面介绍怎样根据代码得到名称,就是考虑用户的方便;而第 35 行到第 39 行就是考虑避免错误。如果单纯考虑数量,程序就简单了许多,不需要考虑出库时采用何种价格。在图 1.1 中,“L”列为状态,如果数据已经输入,就令其值为 1 (参见第 110 行),为此第 36 行做一个检测,如果已经为 1,就意味着已经输入,提示用户不需要再输入,然后退出程序,第 39 行以后的程序就不会再执行了。

下面从第 40 行起到第 83 行,执行当输入的内容为“入库”时的指令,第 84 行到第 109 行执行“出库”的指令。

```

j = Target.Row                                     40
Set mRng = ActiveWorkbook.Sheets("汇总信息").Range("B3:B" & i) 41
If Target.Value = "入库" Then                       42
    If Target.Offset(-4) = 0 Or Target.Offset(-3) = 0 Then 43
        MsgBox "数量和金额不可以为零!"                44
        Application.EnableEvents = False                45
        Application.Undo                                46
        Application.EnableEvents = True                 47
        Exit Sub                                         48
    End If                                              49

```

第 43 行检测是否数量和金额为零,有一个为零都不行,没有这两个值就不能计算出新的移动平均单价。第 50 行到第 80 行是循环语句,循环的条件和前面第 19 行的条件相同,对比物料代码,但目的不同,这次的目的是为了得出移动平均单价的值。

```

For Each r In mRng                                  50
    If r.Value = Target.Offset(-7).Value Then        51
        If j = 2 Then                                52
            n = Target.Offset(-4) + r.Offset(9)      53
            C = Target.Offset(-3) + r.Offset(10)     54
        Else                                          55
            With r.Offset(5)                          56
                .Formula = "=SUMIF(进出记录!B$2:B$" & j - 1 _ 57
                    & ",汇总信息!B" & r.Row _
                    & ",进出记录!E$2:E$" & j - 1 & ")")
                .Value = .Value                      58
            End With                                  59
            With r.Offset(6)                          60
                .Formula = "=SUMIF(进出记录!B$2:B$" & j - 1 _ 61
                    & ",汇总信息!B" & r.Row _
                    & ",进出记录!F$2:F$" & j - 1 & ")")
                .Value = .Value                      62
            End With
        End If
    End If

```

End With	63
n = Target.Offset(-4) + r.Offset(9)	64
C = Target.Offset(-3) + r.Offset(10)	65
End If	66
p = C / n	67
r.Offset(11) = p	68
r.Offset(11).Style = "Comma"	69

从第 51 行到第 69 行要解决的问题是在汇总表的 M 列写入移动平均单价。此时考虑两种情况：第 1 条记录和第 1 条以后的记录。如果是第 1 条记录（在记录表中行坐标为 2），就应当直接将记录的数量和金额分别加上库存数量和金额，即程序的第 53 行和第 54 行。如果不是第一条记录，就必须重新计算汇总，而且计算时不考虑正在输入的记录，即第 57 行和第 61 行，然后再获得计算移动平均单价的金额和数量，即第 64 行和第 65 行。请特别注意两点，其一，在公式中有“j-1”，此时的 j 就是记录表中正在输入的记录的行坐标，这样就可以保证在汇总时排除正在输入的记录了；其二，第 58 行和第 62 行是很有意义的，把公式变成了数值。如果汇总数据很多，全部是公式，每次变动都计算将影响运算速度。

With r.Offset(5)	70
.Formula = "=SUMIF(进出记录!B\$2:B\$" & j _	71
& ",汇总信息!B" & r.Row _	
& ",进出记录!E\$2:E\$" & j & ")"	
.Value = .Value	72
End With	73
With r.Offset(6)	74
.Formula = "=SUMIF(进出记录!B\$2:B\$" & j _	75
& ",汇总信息!B" & r.Row _	
& ",进出记录!F\$2:F\$" & j & ")"	
.Value = .Value	76
End With	77
Exit For	78
End If	79
Next	80
With Target.Offset(-8).Resize(14)	81
.Font.ColorIndex = 0	82
End With	83

第 70 行到第 77 行是汇总的第二次运算，此时将正在输入的数据包含在内。可能有人会认为两次运算没有必要，只要输入时仔细，不出现重复输入就可以了，而实际情况是出现重复输入不可避免。经过两次运算，即使重复输入也不会造成重复累加的错误。第 81 行到第 83 行是设定字体的颜色，此时的设定为黑色，针对入库数据；第 109 行到第 111 行也是设定字体的颜色，此时的设定为红色，针对出库数据。请注意第 81 行的表达式，意思是以目标单元格对象