结合元数据（meta）和数据库技术的创新宏写法，全球仅有

GOTOP

李潜瑞 编著

作者感言：

- 找不到一本这样的书，只好自己写
- 有技术问题请联系我：
  chrita.lee@msa.hinet.net

Excel 宏魔法书

▶ 范例丰富

作者根据自己长期积累的经验，编写近百个范例，每一个范例又有若干种扩展，帮助读者解决跨表查询、网络资料抓取、彩票信息统计等等日常工作急需但Excel常规功能无法实现的工作。

▶ 使用方便

书中所有范例程序，均可从网站下载，读者只需调用范例或者对范例略作修改，即可轻松完成Excel无法完成的任务。

▶ 强大支持

本书作者提供网络在线答疑，如同请名师于身边，学习起来更轻松！
本书所有程序代码均可从网站下载：www.khp.com.cn

科学出版社

Excel 宏魔法书

李潜瑞 编著

科学出版社

内 容 提 要

你知道 Excel 可以直接跟数据库互动吗？你知道如何把 Excel 自己当作数据库，然后在多个工作表之间以数据库的方式整理数据吗？你知道如何把不像数据库结构的工作表转化为数据库吗？你知道如何直接从 Excel 撷取 Internet 上的数据吗？丰富内容尽在这本薄薄的小书中。

本书根据作者长期积累的经验，编写近百个范例，每一个范例又有若干种扩展，读者只需直接调用或略作修改，即可完成跨表查询、网络资料抓取、彩票信息统计等 Excel 常规功能无法完成的任务。

本书分为 4 章，第 1 章——概念架构，提供以数据库角度切入 Excel 宏程序设计的概念基础和基本编程技术，为全书之灵魂。第 2 章—— SQL 实例集，提供 SQL 实例程序，进一步强化 SQL 查询技术在实际工作中的应用。第 3 章——单元格自定义函数，提供 Excel 宏程序设计常会用到但并未内建的函数。第 4 章——综合技巧，提供 Excel 宏程序设计常会用到的、非数据库方面的技巧。

本书采用创新的宏写法，语言轻松活泼，实例丰富，实用性强，适合任何具备 Excel 宏编程基础的读者阅读。

图书在版编目（CIP）数据

Excel 宏魔法书/李潜瑞编著.—北京：科学出版社，2007

ISBN 978-7-03-019094-9

Ⅰ.E… Ⅱ.李… Ⅲ. 电子表格系统，Excel Ⅳ. TP391.13

中国版本图书馆 CIP 数据核字（2007）第 085496 号

责任编辑：陈跃琴 / 责任校对：科 海

责任印刷：科 海 / 封面设计：林 陶

科学出版社出版

北京东黄城根北街 16 号

邮政编码：100717

http://www.sciencep.com

北京市鑫山源印刷有限公司印刷

科学出版社发行 各地新华书店经销

*

2007 年 7 月第一版 开本：16 开

2007 年 7 月第一次印刷 印张：18.25

印数：0 001-4 000 字数：203 千字

定价：32.00 元

（如有印装质量问题，我社负责调换）

目　录

目录

contents

目录

contents

CHAPTER 1

概念架构

这是本不一样的 Excel 宏书，我假设你已经具备宏程序设计的基础。其实，一直看下去你会发现，要读懂本书，需要的宏基础也不必太深。如果你完全不会宏，市面上、网络上都有非常多的书籍参考资料；如果你刚学会基本的宏程序设计，我相信本书所介绍的整套思路及其产生的技巧会让你以后的日子轻松愉快；如果你已经“误入歧途”（说笑的啦!），是个已经有多年老练经验的超级用户（power user），把 Excel 拿来当数据库用，而且已经用它设计了许许多多好用的系统，相信你更可以从本书获得新的启发。

语法、语义与本书

在许多领域，人们都试图以最少的投入做最多的事。我们总希望“事半功倍”，不想做重复的事。所以，在企业的经营上，经理人想找出 80-20 法则；在科学的探索上，科学家想发现各种一劳永逸、完美无瑕的公式；在管理上，政治家希望建立制度；对于国际纷争，各国要建立机制或惯例；在股市里，分析师想找出各种技术线型；对于景气预测，经济学家则建立各种指标；软件的开发就更不用说了，尽管“改变”已经跃居软件开发的头号敌人，“改变是惟一的不变”已成为软件开发的无上真理，但软件工程专家仍孜孜不倦地希望在这个善变的人类活动中勾勒出一些可以倚赖的基本元素。我们所熟知的操作系统（OS）、数据库管理系统（DBMS），以及可能不怎么熟悉的面向对象（OO）、SQL、XML、CMM、RUP/XP/Agile 等，正是我们开发各种信息应用系统，在项目管理或程序技术上所倚赖的“大型架构”。

语法与语义的关系

小技巧发挥小功能，大架构则发挥大功能。在学习本书之前所学的那些 Excel 宏命令，与其说是“一行行的命令”，倒不如把它视为“一种语言”或“一种语义模型”更为贴切。但这语义不是你的，是 Excel 它自己的。

你大概觉得纳闷，命令跟语言或语义的区别在哪里？不都是一行行的“程序代码”吗？

你可以这么想，命令如同孤伶伶的字词，语义则像是一篇文章，有你的思想在里面。命令只有正确与否，语义却有丰富的生命，命令大家都可以用，语义则可以表现出差异性，命令是软件厂商对该软件的定位，它提供了一个最基本的语法平台，他只做到那里，剩下的事，要靠你对自己信息需求的想法，以你自己的语义，用它所提供的命令加以实现。同一个程序的需求，每个人用的都是一样的 Excel 语法，但语义（程序逻辑）却很难完全相同。甚至一个月前的你跟现在的你，对同一个程序的想法都会有所不同。

Excel 只能检查宏的语法，检查不了宏语义。如果写程序永远停留在语法的层次，那程序代码就成了一本本命令堆砌出来的流水账，不但难以维护、难以理解、也难以适应改变，提升到语义的层次，程序设计这件事才有乐趣跟艺术可言。

语法、语义与编程

这些东西好像不是写给你看的，你可能听一些程序设计师说过什么面向对象，报上登的程序设计课程也有一种叫“xxx 面向对象程序设计”的名词，我学的是 Excel，最多被人称作是“power user”（超级用户），懂这些做什么？

你当然不需要也可以不必懂面向对象（虽然你懂了之后可能会上瘾），就可以写出能运行的 Excel 程序，我在这本书谈的也不是面向对象（它只是行文至此的一个例子）。但程序设计这种唯心的纯脑力活动，能了解一下更深入的程序技巧，不也是一件好玩且很有意义的事吗？事实上，Excel 可以接受的宏写法，远超出绝

大多数用户的想象，买了一部保时捷不开，却把它当作是野狼在骑，不是也挺可惜？

一种语法，多种表述

自然语言的语法规则

自然语言尽管有所谓的语法在规范着它，但仍具有极大的自由度，所以作家或诗人可以用有限且枯燥的字词，构筑出无限而流传千古的文章和诗集，语言是我们跟其他动物最根本的不同。而这里所说的语言，当然不是猴群的示警声、鲸鱼的求偶声或迅恐龙召唤同伴的信号（尽管电影声称它们已经很聪明）这样单纯，它所指的当然是语义。

程序语言的语法规则

程序语言的规则就是语法，这很简单，不难学会，除了 on-line help，Excel 甚至可以把你的一举一动都录制下来（少数特殊的动作仍无法以宏录制，例如在线数据查询、设置单元格的批注字体、发送邮件等），但即使这样，也仍旧制造不出“聪明”的宏程序代码，原因还是那一个：计算机无法了解你的语义。这一能力，有时动物都比计算机强，我的宠物有时似乎真听得懂我在跟它讲什么呢！即使 Excel 能把你的一举一动全都记录下来，但无法知道你真正想做的事情是什么，所以它录制不出一个两层循环的九九表程序。

语言的自由度与语义

由于语言的这种自由度，所以一种语法可以产生出千千万万种不同的文章，同一个词汇可以孕育出好几种不同的意义，字有限，而人类想要表达的意义无穷，所以词汇的重载（overload）——一个词汇代表多重意义就成了一个必然现象。

程序语言也是如此，每个程序语言的关键词——所谓的“保留字”或“命令”，例如 if then、for loop、procedure 等，数量上都少得惊人，学会任何一个程序语言的这些保留字，绝不会像是我高中时英语老师要我背英文字典那样没完没了，而即使是号称最新式的所谓面向对象语言，保留字的增长幅度也绝不可能像自然语言那样，每年都会出现一大堆的新字。

既然程序语言的保留字这么少，又几乎都是中学程度的英文单词，那程序语言到底难在哪里？或者说，让很多人觉得它无趣在哪里？我为何这么问？因为如果不难又有趣，加上待遇也不算太差，为什么我们的程序设计师密度不能像美国、日本或以色列那样？程序设计师虽然不是当今职场上的显赫行业，但到底还算得上“热门行业”之列，虽然菜鸟的待遇有时还比不过卖茶鸡蛋的收入，但也总算得上“知识经济”一族，况且大家可别忘了，（曾经是、直到去年都还是）举世最有钱的人比尔·盖兹就是个写程序起家的。

原因就在于程序设计并非把语法或命令背熟就好，那大概花三炷香的时间就够了，程序设计的精髓乃是有内涵的语义，而非硬邦邦的语法。

语义、哲学与架构

觉得我好像在讲哲学？

语义与哲学的关系

也许你想得没错。而且，很多学问讲到最后往往都会自动升级成哲学。连张三丰教张无忌太极拳，都要他“只重其意，不重其招”，不要死背招式（甚至还要忘掉它们），而要把太极心法了然于胸。

我曾经上过一种叫面向对象（这名词我讲了好多遍了）的“计算机”课程，课时为 50 小时，要价 25 000 台币，都还有不少人自费参加，某些人（不包含我）上完之后的感觉大概跟你现在有点像：他们觉得好像上了一堂哲学课。那门课程全都是没有任何计算机的。这让我想起了美国投资大师巴菲特的“营运总部”，是在一个没有计算机的乡间，他刻意不让底下的分析师过份依赖计算机所提供的信息，以免影响他们的对某些事物的基本判断。

第五项修炼的作者，管理学大师——彼得圣吉，他开的一些管理课程，也在乡间举行，让经理人放松心情，有些经理人因为没有听到“实务的管理技巧”而气愤地认为，圣吉教的不是管理，而是哲学。

再举一个例子，形象颇受争议的国际金融基金经理人兼慈善家索罗斯在他的“全球资本主义危机”一书中曾经提及，他的生财之道及人生观“是建立在一些抽象的哲学概念之上”的。

为何我要提及哲学？因为它跟程序设计的关系非常密切，甚至可以这么说，程序设计的本质并非数学，而是哲学（或者也可以这么说，它是好多领域的综合体，包括工程、艺术、管理、数学、心理学、法律、甚至政治），套用一位我所钦佩的软件工程师的话："我们是在扮演虚拟世界的上帝角色"。

程序设计与数学的关系

常听到有人说，写程序要数学好，但写了这么多年的程序，我发现其实用到数学的时候并不多，用到的也不难。当然，这得看你对"数学好"的定义是什么。我的经验是，程序设计，尤其是商用的"一般"数据处理程序设计，大概只要高中数学程度就够了。具体来说，就是代数的概念，也可以说，它是一种将数字抽象化的能力。我记得有一次教一群小学生写程序（应补习班要求），我要他们计算长方形的面积，并在实际测试程序时，输入任意两个数字"代表"长方形的长与宽，那是 BASIC 吧！输入命令是 INPUT，照理说，程序是应该写成：

```
INPUT X, Y
```

而一位小学四年级的男生好不容易终于想出了程序的目的，但写出来的却是：

```
INPUT 2, 3
```

这时，一位六年级的男生告诉他说，要写成 INPUT X, Y，四年级的那个小男生不解，那个年长他两年的小朋友说出了一句

让我恍然大悟且至今仍记得的话："喔！那个X跟Y是'代词'啦！"

我这才告诉补习班的小学生，代数要中学才会教呢！缺乏将数字抽象化的最基本能力，连带也会缺乏将真实世界问题抽象化的能力，而这种能力正是进入程序设计世界的第一步。

语义是程序设计的核心精髓

先把你心中对哲学的刻板印象抛开（忘了人生哲学这等字眼），把它视作是一种"自圆其说的能力"。程序设计并不是要你去"证明"什么，而是要你去"创造"什么。所以数学不够好也没关系，大多数的时候，程序设计只需要中学程度的英语，数学呢？除非你要写的是反飞弹防御系统或F16战斗机上的程序，不然，前面已经提及，一般的商用数据处理最多高中程度的数学应该就可以搞定了。即使像保险系统中的精算功能，也有精算师（所谓的"领域专家"）会帮你搞定公式，你只需（在别人的帮助下）理解那些数学公式，将它翻译成数学公式的近亲——程序语言就行了。

答案已经很清楚，前面说了半天的"语义"，就是这里说的"哲学"。你对一个程序的想法，就是你日后向别人也向自己解释该程序的"语义模型"（程序的世界，一个月后的自己可能会完全不认识一个月前的自己，所以相信我，日前的你会有机会向日后的你解释的）。注意想法的"法"字，它在这里，等同于"制度"、"机制"、"模型"、"模式"、"原则"，以及我接下来要谈的"架构"。

程序的架构

简单程序中架构的作用

程序如果没有架构当然还是可以执行无误，一个九九表，不用两层循环，用 print 指令打印 81 次，除了多占一些硬盘空间跟 CPU 时间（反正因此而多花的空间跟时间大概比新鲜空气还便宜），结果并不会错误，如果还可以因此交出作业，虽然不会得到 A，但大概也不至于不及格，所以又有何不可？

当然可以，如果这样，我不必再说下去，你也不必再看下去，我们彼此谢谢，也不必再联络了。

但事情没这么单纯，之后如果老师要你交一份能产生八八乘法表或七七乘法表，或看准了你用print指令而要你交一份100*100乘法表的程序，甚至更狠，说任意输入两个数 m 跟 n，就输出 m*n 乘法表，那你的程序可就写不完了。由于缺乏一个最根本的架构（或“心法”），你的 print 招式禁不起一点点外在条件的变动。

复杂程序的架构

如果说这么简单而无聊、纯属学生交作业层次、对企业界毫无实际效益可言的程序，都存在着有架构没架构就有差别的状况，那稍微复杂一点的程序不就差更多了？

这一切之所以会如此，是因为软件的天敌是“改变”，程序代码不像电饭锅或电扇，可以一直用 30 年，它要一直变动、一直创