

编程范式与OOP思想

郑晖 著



电子工业出版社  
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY  
<http://www.phei.com.cn>



# 冒号课堂

编程范式与OOP思想

郑晖 著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

TP311.1  
Z432

## 内 容 简 介

本书以课堂对话的形式，借六位师生之口讲述编程要义。上篇对编程范式作了入门性的介绍，并对一些流行的编程语言进行了简评；下篇侧重阐发软件设计思想，其中在范式上以 OOP 为主，在语言上以 C++、Java 和 C# 为主。全书寓庄于谐，深入浅出，既可开阔眼界，又能引发思考，值得编程爱好者品读。

未经许可，不得以任何方式复制或抄袭本书的任何部分。

版权所有，侵权必究。

## 图书在版编目 (CIP) 数据

冒号课堂：编程范式与 OOP 思想 / 郑晖著. — 北京：电子工业出版社，2009.10

ISBN 978-7-121-09545-0

I. 冒… II. 郑… III. 程序设计 IV. TP311.1

中国版本图书馆 CIP 数据核字 (2009) 第 166928 号

责任编辑：徐定翔

印 刷：北京智力达印刷有限公司

装 订：北京中新伟业印刷有限公司

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：29.75 字数：570 千字

印 次：2009 年 10 月第 1 次印刷

印 数：4 000 册 定价：65.00 元

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 [zltz@phei.com.cn](mailto:zltz@phei.com.cn)，盗版侵权举报请发邮件至 [dbqq@phei.com.cn](mailto:dbqq@phei.com.cn)。

服务热线：(010) 88258888。

## 群英会“冒号”

### ◆ 结构篇

| 序号 | 阅读样章后的看法和建议  | 郑晔的回复  |
|----|--|--|
| 1  | <p><b>提出人：邹 飞</b></p> <p>该书上册对一些编程语言和范式做了介绍和比较，第二部分主要是集中在 OOP 的介绍。第一部分因为牵涉很多内容，显得有点散。</p> <p><b>霍 炬：</b>同意。</p>  | <p>第一部分是对各种编程范式的介绍，自然得牵扯到很多内容。但毕竟针对同一主题，形散而神不散。狭而深固然是一种学习方式（如书中的后七课），广而博同样也是一种（如书中的前六课）。</p>   |
| 2  | <p><b>提出人：徐 宥</b></p> <p>面向对象讲得很多，其他可能需要扩充。</p> <p>这本书第 5 章往后的内容主要集中在面向对象的 paradigm 上。如果本书的定位是 OOP 的话，这样是无可厚非的，只是前文的那些铺垫有重拳打棉花之感。</p> <p><b>霍 炬：</b>同意。</p>  | <p>本书的最初定位的确是 OOP，那是最广大的程序员使用的编程范式。但又不想让读者的眼光过于局限，故而在前面增加了各种编程范式和语言的介绍。</p> <p>只是铺垫越写越长，以致独立成篇了。后来准备对其他范式再作进一步深入，但时间有限，只能等以后有机会再说了。</p>  |
| 3  | <p><b>提出人：霍 炬</b></p> <p>概括来看，本书只讲了两部分，语言范式和面向对象。语言范式部分相当精彩。对各语言的描述，概括也恰当。</p> <p>面向对象部分个人认为过于拖沓。另外，如果不进行 GUI 开发，面向对象技术未必那么重要。用半本书来讲一种特定的技术，有点可惜。</p> <p>简言之，我认为两部分的跳跃太大。这两部分的立意高度有差距。</p> <p><b>徐 宥：</b>同意。</p> | <p>本书重点正是 OOP，尤其针对人们对 OOP 中常见误区和模糊区。介绍 OOP 的书汗牛充栋，用半本书来介绍不是太多，而是太少。至于是否拖沓，应以内容而非篇幅为标准。</p> <p>OOP 究竟有多重要，这是一个开放的话题，不是本书关心的问题。但可以肯定的是它不仅专长于 GUI，否则那么多的中间件（middleware）、Web Service 也不会用 OOP 来开发了。</p> <p>本书之所以以 OOP 为主题，是出于实用的角度，而不是理论的角度。毕竟现在的程序员大多数只知道 OOP。</p> <p>两部分的跳跃的确有些大，这与本书的写作过程有关。立意高度有差距也不假，原因前面已经解释过。</p> |

## ◆ 读者定位篇

| 序号 | 阅读样章后的看法和建议   | 郑晔的回复   |
|----|---|---|
| 1  | <p><b>提出人：邹 飞</b></p> <p>该书提到了很多范式，而且尝试对这些范式做出比较并给出自己的理解。个人认为，对一个语言或范式的理解更多依赖于对其特性的深入学习和把握，他人的说教较难让人有醍醐灌顶的感觉。个人对于这种思想甚或接近哲学理念的东西（比如语言之争）不很感兴趣，或者说语言的优劣很难三言两语表述清楚，而且这些观念很难说放之四海而皆准，每个人根据他学习和把握的知识可能都会有些自己不一样的看法，比如我个人就会觉得该书中有些观点有待商榷。</p> <p>如果该书定位于之前没有了解这些语言或范式的读者，对他们进行知识普及，我觉得还是很可取的，毕竟该书有着较广的知识面，而且对一些错误观念的批判也很有意义。但对于在某方面有所专长的工程人员，可能不是太适合，做技术书太泛了。</p> <p><b>霍 炬：</b>同意。</p> <p><b>徐 宥：</b>邹飞先生关于“观念很难说放之四海皆准”的意见我是同意的，但作者也说了，冒号也只是和大家探讨，而不是自己强迫大家接受，因此从这个角度看，不同的人对文本有不同的理解是正常的。事实上，有些文本需要看很多书做很多实践才能弄明白，这也是邹飞先生说的“更多是依赖对特性的深入学习”。这本书的内容就天生拒绝了那些不喜欢在思想层面反思的读者，也会让一部分持有不同思考的读者觉得读着不对劲。所以，我觉得邹先生给出的是正常的，客观的评价，也无需因此担心书的内容：)</p> | <p>本书反复说明编程范式是一个培养的过程，不是一朝一夕能领会的。但凡事总得有个开始的过程吧，本书的前半部分就是为了给不了解它们的读者一个编程范式的入门（但不是编程入门）。</p> <p>我认为思想甚或接近哲学理念的东西非常重要，关系到程序员的境界，对知识的融会贯通的把握。至于语言的优缺点当然很难三言两语表述清楚，本书特别说明是一家之言，并无定论。</p> <p>其实本书开宗明义就提出本课堂是开放式的，从来没有说哪一些观点是放之四海而皆准的。关键是给大家一些启发或触动，无论是引起共鸣还是争论，都是不错的结果。</p> <p>（当然这不是为可能出现的错误找借口。）</p> <p>这本书本就刻意与大多数书籍有所不同，开班发言中便已声明，请不要拿老眼光来看待。至于说到做技术书太泛了，这话应该只适用前半部分——那本就是一个泛泛的介绍，后半部分对 OOP 的最重要的特征进行了较为细节的讨论，恐怕算不得泛泛吧？而后半部分才是全书的重点，我相信对某方面有所专长的工程人员同样会有所帮助的。</p> |
| 2  | <p><b>提出人：霍 炬</b></p> <p>定位存在一定问题。本书涉及的方面过于宽泛。对缺乏足够实际经验的读者很难领会其内涵，恐怕也不易获得认同。</p>  | <p>缺乏足够实际经验的读者的确很难领会书中的内涵，但正如书中所说，学习是一个迭代提高的过程，需要多次创磨。另外，容忍无知也是学习的一个必经阶段，就怕许多人根本不知道自己无知。如果本书不能让一些人完全理解，让他们知道还有一些自己未知的领域，也是一种贡献。本书不强求认同，否则就不会如此观点鲜明地谈一些可能有争论的问题了。如果只是照本宣科，或只是机械地拼凑总结，那么本书的意义便完全丧失，本书所提倡的精神也完全被阉割。</p>  |



## ◆ 体例篇

| 序号 | 阅读样章后的看法和建议  | 郑晖的回复   |
|----|--|---|
| 1  | <p><b>提出人：邹 飞</b></p> <p>该书采用对话式来组织，形式很新颖，但缺点在于“废话”太多，很多时候需要用一些过场语言，可能会分散读者注意力。</p>  | <p>这是双刃剑，有人认为过场多余，有人认为过场有助于缓冲节奏，避免信息轰炸。如果句句无闲话，那就不是对话体了。</p>  |
| 2  | <p><b>提出人：蒋波涛</b></p> <p>在每一节的安排上紧凑些，不要搞预览、提问、讲解、插语、总结和参考这么复杂，因为它们会不断打断读者的思路。</p> <p><b>徐 宥：</b>我是非常同意的，但是作者也有放置这些的理由，我也很难评价哪种方法好。</p> | <p>这也是见仁见智的问题。只要书在排版上做些合理的安排，增强层次感，可以让读者很清楚哪是正文，哪是附文。</p> <p>预览是为了提起读者的兴趣；提问是为了让读者带着问题看书；插语是补充一些必要的知识，但又不妨碍正文的连贯性；</p> <p>总结是不少读者的要求，也是对较为分散的全文的再次整理；参考是对引用文献的尊重，更是必不可少的。</p> <p>最后，这六个部分正好暗合文中的六位主角的特征，也是一个创意。</p> |
| 3  | <p><b>提出人：霍 炬</b></p> <p>课后思考建议配参考答案，或者以后在网上补全。</p> <p><b>徐 宥：</b>同意。</p>  | <p>是一个好的建议。不过暂不在书中回答的主要原因是：有些问题是开放的，没有固定的答案，主要的目的是促使读者进一步思考。</p> <p>如果代答，似乎有违本书宗旨。</p>  |

## ◆ 其他

| 序号 | 阅读样章后的看法和建议  | 郑晖的回复  |
|----|--|--|
| 1  | <p><b>提出人：徐 宥</b></p> <p>metaphor 不要太多。</p> <p>P100 讲到大轮船开到 duck type 的池塘里面，具体是什么含义呢？是用 duck type 做了一个重量级的怪兽？第 6 页说小兵器一寸短一寸险，这个 metaphor 就变成了用低级语言难免风险，似乎思维上也很跳跃。过多的不落到实处的隐喻会让读者摸不着头脑。</p> <p><b>霍 炬：</b>metaphor 这个问题我很同意，很多书都是，用一个人們不熟悉的概念来讲解另外一个不熟悉的。</p> | <p>其实，本书曾多处（至少三四处）提到肤浅的比方反而会误人，必须进一步严谨地思考。比如书中的一句话：浅显的比方只是门槛前的台阶，借之或可拾级入门，却无法登堂入室。</p> <p>但比方的确会减少距离感。对于专注的主题，肯定不能只停留于表面的 metaphor。你提到的鸭子类型，因为是简介部分，故而没有多加解释，其实涉及 specification 的问题。在后面的章节，已经一再强调规范的重要性。</p> |
| 2  | <p><b>提出人：徐 宥</b></p> <p>建立社区，注意习题、思考的编号。</p>  | <p>很好的建议。</p>  |

| 序号 | 阅读样章后的看法和建议  | 郑辉的回复   |
|----|--|---|
| 3  | <p><b>提出人：徐 宥</b></p> <p>人物个性的塑造不够完整饱满。</p> <p>作者试图让冒号在禅宗棒喝的导师形象和循循善诱的老夫子形象之间求得平衡。比如 p39，机锋感很强，换成古文就是禅宗导师在教育弟子，第 13~14 页也是，可算是得道妙论，可是换到其他场合，冒号又成了一个“道貌岸然”的老师了 (p11)。有时候冒号又想成为大家的“同学”。</p>  | <p>这个有待改进，也是时间上过于仓促，没有进一步完善人物，更注重技术层面。但六个人物还是有他们的特点的：</p> <p>冒号寓庄于谐，深入浅出；引号博闻强记，引经据典；问号勤奋好学，刨根问底；逗号插科打诨，调节气氛；句号颖悟灵性，举一反三；叹号：浅尝辄止，多愁善感。</p>  |
| 4  | <p><b>提出人：邹 飞</b></p> <p>一些不一样的观点，如关于栈和堆的比较：</p> <p>a) 栈也可以再运行期决定空间大小 (alloca)。</p> <p>b) 关于堆是否 thread-safe，从 OS 角度看，heap api 基本都已经提供了 thread-safe 的版本，而语言层面，C/C++ 标准是 undefined，但并不是 no，事实上众多实现都是 thread-safe 的。</p> <p>c) “如果所需内存空间较小且较固定，则尽量采用栈分配”，采用堆还是栈，不是取决于对象大小而是生命周期吧？</p> <p><b>霍 炬：</b>其实 C 众多实现不是 thread-safe 的。而且我们现在都提倡尽量用多进程，分开数据共享区域。云风也有类似 blog。但是邹飞的说法也不能说不对，只是这种争论已经超过这个书的范围了。所以邹飞说的是“一些不一样的观念：”是观念，但不算错误。</p> | <p>a) 本书在 § 11.1 的插语 3 中已经对此进行了补充，特意还提到了 alloca。</p> <p>b) 栈一定是线程安全的，堆则不一定。既然 C/C++ 标准是 undefined，也可以说是不安全的，因为不在程序员的掌控之中便可理解为不安全的。哪怕对于某种实现是安全的，但谁也不能保证同样的源码一定在同样的 OS 或编译器下进行编译。如果确定了编译器和平台，有些 heap 可以是线程安全的，但那毕竟过于深入了，超出了本书的范围。</p> <p>c) 采用堆还是栈，当然与生命周期紧密相关，文中对此已作了说明。但与对象大小同样有关，栈空间是有限的，分配过大的对象将导致栈溢出。在嵌入式开发中，多用堆分配，就是出于这个考虑。</p> |
| 5  | <p><b>提出人：邹 飞</b></p> <p>关于实现继承和接口继承的定论上升到境界的层面，这个其实也是要看适用场景的。</p>   | <p>当然要看适用场景。但提倡接口继承而不是实现继承，这是一个一般的原则，书中已对此作了详细的说明。</p>  |
| 6  | <p><b>提出人：邹 飞</b></p> <p>把 VB 和 Delphi 称为前台语言不太合适吧？</p> <p>a) Delphi 也是可以脱离 IDE 而存在的。</p> <p>b) VB 和 Delphi 并不单纯用于 UI 开发，如果强调 IDE 的便利性，C# 也不差啊。</p>  | <p>在提到的 12 种语言中，VB 和 Delphi 显然是更侧重前台的，而且文中也专门提到它们 (尤其是 Delphi) 同样可胜任于后台编程。这似乎并不矛盾吧？</p>   |
| 7  | <p><b>提出人：邹 飞</b></p> <p>Python 效率为什么会成为最主要的问题？</p> <p><b>徐 宥：</b>技术层面的几个细节，邹飞先生说的都是对的，但是和原作者出发点不同，经历不同，所以认识不一样。我已经认真比较过文本了，基本我同意原作者的文本。作者这么写，其实背后是有很多经历的，所以从作者的角度看看，这些</p>  | <p>Python 的效率不仅比不上 C/C++/Java/C# 等编译语言，也不如 Perl 等解释型语言。在大多数场合下，这是公论，当然不排除 Python 解释器以后改进的可能。</p>  |

| 序号 | 阅读样章后的看法和建议  | 郑晖的回复                         |
|----|--|-------------------------------|
| 8  | <p>对技术的问题的叙述都是正确的，可以理解的。其实，对于技术问题的理解，没有一定对或者一定错的结论。</p> <p><b>提出人：霍 炬</b></p> <p>本书质量比较高，作者知识结构完整、广泛，实例和流程图严谨，确实是不可多得的参考书。但相应</p> <p>对读者的要求也会有所提高，缺乏足够实际经验的读者很难领会其内涵，恐怕也不易认同它。作为学习或提高用书有一定难度，但用来作高级的知识普及、开拓眼界是相当不错的。其实书中还是有大量观点和我一致，或者说，有经验的工程师，对此书大部分是认同的。</p> <p>阅读过程中，我也的确补充了一些以往不太清楚，或是不太确知的领域知识，也是有收获的。</p> | <p>这本书对读者的理论和实践水平确有一定的要求。</p> |

徐 宥 Washington University 就读（博士）

邹 飞 趋势科技（中国）研发中心 资深软件工程师

霍 炬 银杏搜索 创始人 技术 blogger

蒋波涛 宁波市规划与地理信息中心



## 交流·反馈

### “冒号”遇知音

徐宥：郑晖先生你好，刚从博文视点编辑那里拿到你联系方式，你的《冒号课堂》是上等作品。

郑晖：多谢夸奖。你好，刚看到你的评论，非常到位，建议也很中肯。

#### ◆ 书稿最后一章的内容（对话时，郑晖先生完成全书的前11章。）

徐宥：很好奇，最后一章你准备放什么内容？

郑晖：其实还有两章，一章讲设计原则，另一章讲设计模式。时间仓促，许多想写的还没写。

徐宥：对话体很难写好。之前要准备很多材料，所以我一拿到书稿看到是对话体，就知道这书是挑战。◎读完第一章，我就觉得这书太牛了。

郑晖：很高兴得到你的肯定。

#### ◆ 写书的缘由

徐宥：看得出来你是想倾心教授。

郑晖：写书是非常偶然的想法。有一天看到当年明月的事迹，便萌生了开博写书的念头，当晚便有了构思：用标点符号为名写成对话体。

徐宥：大家都一样。我也是看有些作者（例如李笑来）写博客写成系列了，我也就写了一个系列（《编程珠玑番外篇》）。只是我笔头慢，所以到现在还弄不成一本书。这种书肯定都是厚积薄发的，你一旦想写，马上构思，框架就全出来了。况且你的文字文学功底真地好，可以用这些去驾驭内容。

郑晖：一开始书的内容没想好，边写边整理。

徐宥：同感，常常信马由缰又是一篇。倒不一定是事先周详计划出来的。

#### ◆ 从数学转行到计算机的原因

徐宥：我看过你的简历。在做计算机前，你是大学数学老师？我也是数学系毕业的，能不能问一下是怎样的机缘使你转到计算机行业的？

郑晖：1996年我去美国读 math PhD，1997年被同学拉去上计算机课。你知道，一个纯粹的学数学的人，会觉得计算机远不如数学的挑战性大。

徐宥：是的，我上大学的时候也有同感。

郑晖：当年一心想当数学家，转行学计算机自己觉得是一种失败的标志。后来是被生活裹挟，不情愿地转行了。

徐宥：嗯，就像哈代一样。学数学的都对应用学科有天生的优越感，我大学时的老师老灌输我们这些。

郑晖：这是理论数学系学生的本能吧。数学系的人容易理论化、理想化，这是优点，也是缺点。

徐宥：不过计算机（任何学科）都是有其中精义的，就看读书仔细不仔细了。我到了大三大四，基本上就把这种心态收起，好好看计算机书了。

郑晖：一切学问都有其高妙之处。当然数学是科学之王。好心态很重要。学数学不等于高人一等，关键看个人修为。

#### ◆ 读书心得

徐宥：我看的第一本外文书，是 *Programming Languages: Concepts and Constructs*，这本书我看了好多好多遍，所以看到你的书真是百感交集。我知道中文世界没这样的书，谭浩强老师的书成吨成吨地卖，而上面那本我说的书一直没中文版。

郑晖：这也是我写书的一大缘由。寻章摘句、东拼西凑，既不能深入也不能浅出的书太多了。

徐宥：所以看书者，不能“通经脉”。你书中有一句话：好像一个经脉又通了。我就想到脂砚斋批《红楼梦》里的一句话，叫做“只叫批书人哭死”，难寻知己啊。

郑晖：我也有觅得知音的感觉。国内许多名气很大的人，写出的文章实在不敢恭维。博文有慧眼，寻到你这样的知音。

徐宥：客气了。

#### ◆ 书稿的定位

徐宥：这本书，浅读肯定读不出东西，要细读、深读。比如往往一句话冒号说出来，很简单，但其实里面的精妙，需要多次咀嚼，还要很深的阅历。我建议，这本书是一个起点，而不是一个终点，围绕这本书要有一个社区。个人觉得这本书是提升国内从业人员素质的一个里程碑，没多少书能有这个功底。不过我们象牙塔里面出来的，更关注“内功”层面的东西，不知道从业人员会不会关心。

郑晖：有一定技术追求的人应该会关心。

徐宥：不光要有技术追求，还要有技术视野。这鱼与熊掌，目前兼得的人不多。

郑晖：我们能产生共鸣与相似的背景有关。

徐宥：我们做科班的，视野自然是开阔的，因为这些东西无非就是客观的研究对象而已。可是一些从业人员，不要说 prolog 了，一个 MVC 就折腾不懂。我说的不是虚言。所以这本书是很有价值的，但是前提是读者要知道这书有价值。否则就是 unknown unknown（不知道自己不知道）。

郑晖：我书中有不少类似数学定理推理的段落，不知一般读者是否适应？

徐宥：我看行文很流畅，这个也看批书人背景了。

#### ◆ 书稿的构想

徐宥：你的文字功底很强，我要是能写出这么流畅的行文，至少要改稿几次。

郑晖：前面 6 课自己加工过，后面还没来得及。数学系的人，应该是追求完美的。我经常会为几个用词甚至题记而反复修改，虽然知道一般人不会注意它们。

徐宥：的确是，我读得出来。个人觉得，如果让这个书更加好读，听听其他人的审稿意见或许有帮助。我是恨不能几年前就有此书，不过就怕其他人觉得云里雾里的，没共鸣。

郑晖：其实开始我是想谈大家都关心的 OOP，后来感到说得不透，就从更广义的编程范式着手了。本来还想把其他范式再一一深入复习，但时间不够，只能等有机会写续集了。所以你会觉得有重拳打棉花之感，因为其他范式没有 review。

徐宥：其他范式怕是更加曲高和寡，书里讲 OOP，已经是很看内功了。

郑晖：是的，所以我后来还是以 OOP 为主，能把它说清楚就不错了。

徐宥：我觉得，最后能不能做到猪肚豹尾。前面草蛇灰线，自由发挥，最后收尾浑然一气（我知道这个很难写，我是随便想想）。比如，有没有一个好的餐馆 version 2。

郑晖：书的第 12 章是一个设计原则总纲，算得上是编程设计思想的大贯穿。从“数学上”证明了六对关键词的等价性，以及十几个编程原则的等价性。

#### ◆ 书稿的人物设置

徐宥：基本上你书里的冒号是很能讲问题的，可以考虑让一个标点特别爱插话，这样至少可以帮助读者理解冒号在做什么。

郑晖：为了让读者有些实在的感觉，还准备写设计模式，证明它们都是设计原则的推论。

徐宥：有时候不需要每一句都直指人心，这样或许会让文章好读一些。从 Design Principle（设计原则）到 Design Pattern（设计模式），这个 path 高屋建瓴。

郑晖：是的，逗号就是用来插话的。逗者，逗留之逗，逗趣之逗。越到后来这种角色越清晰。

徐宥：逗号的性格我看出来了。我也是仔细看了后面几章。

郑晖：开始逗号设计为最笨的，因为逗号永远得不出一个结论，后来才赋予逗号逗趣的性格；引号喜欢引经据典；叹号比较感性；句号喜欢下结论；问号自然问题特别多，但悟性不如句号。

徐宥：可否在适当的地方介绍这些人物性格？赋予人格化特征、性别，等等（博文有美编团队的）我其实没有读出你这些良苦构思（我还是读了好多遍稿子的）。或者说，问号、引号，基本上我是读出来的。

郑晖：不过，读者会在意这些吗？毕竟这是一本正儿八经的编程书，而不是小说。

徐宥：这个地方我倒是和你看法不大一样。邹欣（《编程之美——微软技术面试心得》、《移山之道——VSTS 软件开发指南》的作者）有一句话说：你在意，读者就在意了。☺

郑晖：我是想在适当的地方加入人物性格介绍，像《移山之道——VSTS 软件开发指南》那样。

徐宥：我能读出你的设计，我一直在注意这些标点符号，为什么呢？因为我想知道这些人物，他们怎么认识问题、怎么思考、怎么恍然大悟的。我看《移山之道——VSTS 软件开发指南》这本书，虽然我不懂微软技术，但是我特别记得里面的人物，就知道了这些情景。这么说吧，如果人物不鲜明，则对话无意义，因为你可以让冒号说完就放学。☺

#### ◆ 小花絮

郑晖：我的稿子你花多长时间看完的？

徐宥：如果让读者一直不释卷，最好还是能用问号带他求索，逗号引他发笑。我觉得稿子里面已经成功塑造了不少标点，但在叙述繁冗读者容易厌烦的地方，最好更加注意给读者“潜提示”。我大约看了五遍，第三遍是最认真的，花了一早上 5 个小时，后面两遍花了几天陆续看完的，前两遍有猎奇的味道，跳着看的。

郑晖：竟然读了五遍！难得这么认真审稿的人。多谢，比我自己读的都多☺

徐宥：不客气。其实还有一些小的文字细节，我怕冒昧，没写下来。

郑晖：非常欢迎。

徐宥：我看的时候记得，待我整理整理发给你。几乎每一页都有，有的写“有机锋之妙”。

郑晖：对了，你提到的“机锋感很强”是指哪里？

徐宥：如冒号听罢不语，直接问 6 个问题那一段，是标准的禅师带弟子顿悟。禅师不语，弟子问，禅师答，弟子不服，禅师再解，众弟子悟。

郑晖：哈哈，我倒没有刻意去追求这个，只是顺其自然。

徐宥：嗯，我是读到了，欣欣然。

郑晖：不过写对话体真的很累，明明能一口气写完的，不得不分成几段，但这样也留给读者思考的空间，避免信息的狂轰滥炸。

# 序

去年3月的一个下午，过于明媚的春光唤醒了一份久违的情怀，书摊上的一本《青年文摘》便成了合宜的载体。与其说是为了阅读，不如说是为了回忆——对20年前读书心境的回忆。孰料读罢开篇，怀旧之窗随即悄然关掩，一扇求新之门却戛然开启。那是一篇人物介绍，讲述一位籍籍无名的年轻人是如何因撰写博客而声名鹊起的。抚卷思之，网络平台已成大众舞台，人人皆可登台献技，自己何不前去一试？心念甫动，顿感技痒难耐，当晚寝不安席，于辗转反侧之中磨出了一本书的轮廓。

尽管钟书先生认为鸡与蛋应为松耦合关系，但一只来历不明的鸡确会招致人们对其产品可靠性的怀疑。故而在介绍《冒号课堂》的创作思路之前，先自我介绍一番。1986年我怀揣着成为数学家的梦想，考入武汉大学数学系。7年的大学生涯在浑浑噩噩中度过，毕业后在广州一所高校教了3年的高等数学。在混沌与迷茫中挣扎了10年，终于不堪蹉跎，1996年赴美攻读数学博士。始料不及的是，在大洋彼岸不仅没能一圆数学之梦，反倒从一个未曾碰过鼠标的电脑排斥者变成了一名IT工作者。“罪魁祸首”正是电脑和Internet，它们潜移默化地改变了人们的生活和思维方式，我亦未能幸免。1998年开始选修计算机课程，两年后拿到硕士学位，并在华尔街的一家软件公司找到了工作。2004年年底，选择回国发展，再度主导了人生的一次急转弯。回到广州后，顺利地进入了一家著名的外企。平淡而安逸的生活似乎注定与我无缘，不久又转去一家小公司作技术总监。如果用一句话来描述自己的职业生涯，那便是：数学是我的初恋情人，计算机是我的终生伴侣。无论成败，都是命运与人生双向选择的结果。

本书的创作虽出偶然，却也有其必然性。一方面，市面上的计算机书籍多为拼凑之作，且不少带有应试教育的痕迹。另一方面，论坛上充斥着各种谬言妄论，人们或目空一切，或人云亦云；每当争论一起，常常硝烟弥漫，出言无状者甚众。如此诸般，不忍卒睹。深感激浊扬清之必要，此念一直郁积于心，终至一朝爆发。自知虽无澄清玉宇之力，唯奢念带来一缕清风。

《冒号课堂》采用对话体，是为了借不同背景、不同水平、不同性格的人物之

口，多层次、多维度、多角度地展现知识的内涵与活性。人物皆以标点符号命名，是为了塑造让人过目不忘的形象：冒号善解释，引号善引用，问号善提问，逗号善缓冲，叹号善感叹，句号善总结。此外，6个标点符号还对应着每小节的6个部分：冒号是正文讲解，引号是文献参考，问号是问题列表，逗号是补充插语，叹号是精华预览，句号是本节总结。与一般纯技术类图书不同，本书非常强调学习方法和学习精神的重要性。在内容组织上也一反常规，以思想为主、以知识为辅，以抽象为主、以具体为辅，以范式为主、以语言为辅。人们常把书籍比作一种食物，其实书籍也是一种药物。一本书应当同时提供两种价值：一种是让人获取正确知识的食用价值，一种是让人抛弃错误观点的药用价值。《冒号课堂》更侧重后者，这多少给读者带来一定的阅读障碍，因为抛弃往往比获取更加困难。此外，全书涉及的知识点较多，覆盖的知识面较广，一些流行的语言或技术反被刻意地淡化。假如读者没有足够的计算机理论和实践基础，难免会感到一些困难和不适。古语有云：“学然后知不足”，认识到不足何尝不是学习的一种收获呢？从另一面说，假如读者发现书中疏谬，还请不要吝赐教，本人将不胜感激。坦而言之，随着写作的深入，自得之心日敛，惴惴之心日甚，正应了上面古语的后半句：“教然后知困”。

本书的完成首先需要感谢3位母亲：我的母亲、我太太的母亲、我女儿的母亲。没有她们默默无闻的支持和帮助，冒号课堂只能在梦中开班。还要感谢博文视点的周筠老师对本书的大力支持和关怀；白爱萍编辑负责而又耐心，不厌其烦地和我讨论一个个文字和版式的细节；博文的陈宜、杨小勤、陈琼、徐定翔、许莹、胡文佳等编辑也以同样的热情带给我很好的出版体验。我从其他亲友和网友那里也得到了许多热情的鼓励和有益的启示，是他们让虚拟的课堂变得真实和生动。

郑晖

2009年8月24日于广州



# 博文视点 重磅推荐

## 《编程之美》

★荣获中国书刊发行业协会组织的

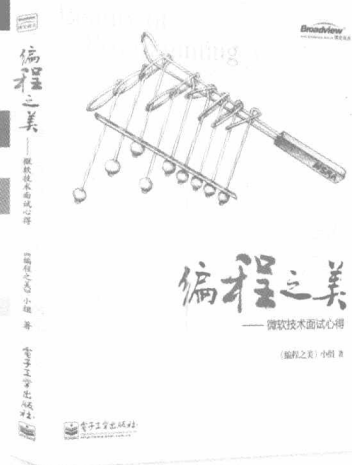
“2008年度全行业优秀畅销品种”奖

★荣获CSDN评选的“软件中国2008图书类编辑选择奖”

★51CTO读书频道和中国图书商报、中国互动出版网共同评选为

“2008年度最佳技术图书”

充满智慧与趣味、囊括大量有趣且有启发性的面试题日  
让您充分享受思考之乐、编程之美



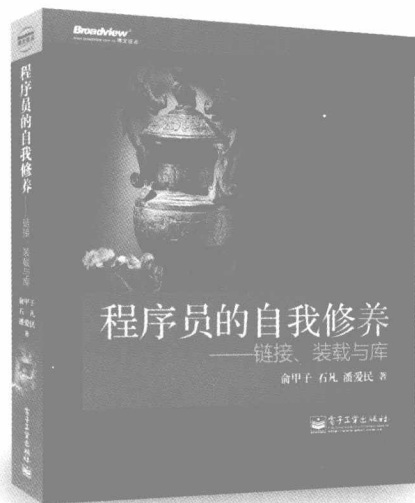
这本书表面上是讲解算法，实际上体现了一种面对困难、解决问题的心态……个人还是挺喜欢这类书的，把编程人性化了……

——拓荒者

《编程之美》中的算法以实例开篇，循序渐进地解决问题，一步步去剖析算法的本质，挖掘和发散算法功效，进而去淋漓尽致地体现算法的美妙！

——萝卜萝卜闪金光

## 《程序员的自我修养》



俞甲子、石凡、潘爱民原创精品

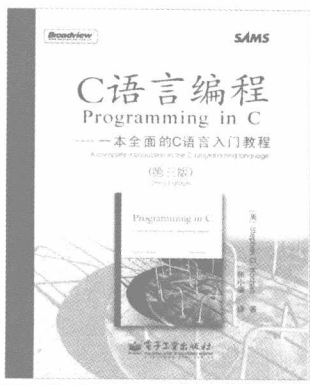
深入浅出地对系统软件的底层形成机制进行条分缕析  
了解程序的前世今生，

彻底理解敲入的代码如何变成程序

关于链接、装载等问题，是操作系统中很基础很重要的一个部分……一方面，我们理解系统如何去做，是为了悟到为何这样去设计。了解了为什么，反过来更能理解怎样去做……另一方面，我们对自己每天用的系统多一些了解，那是百利而无一害。

——云凤

# 博文视点好书推荐



## 《C语言编程：一本全面的C语言入门教程（第三版）》

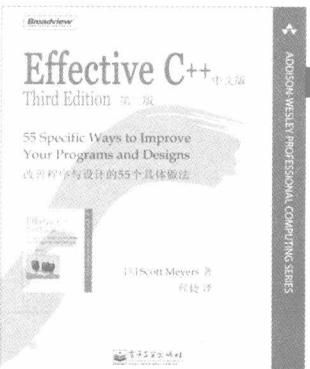
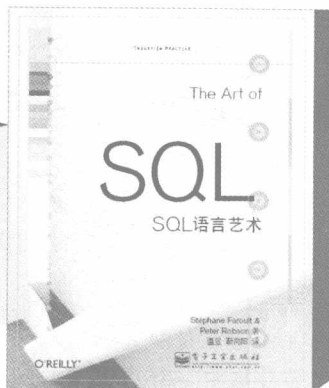
Stephen Kochan 著  
张小潘 译

- 极负盛名的C语言入门经典教材
- 内容详实全面，由浅入深，示例丰富

## 《SQL语言艺术》

Stéphane Faroult, Peter Robson 著  
温昱 靳向阳 译

- 巧妙借鉴《孙子兵法》的智慧结晶
- 深入探讨9种常见查询方案及其性能
- 传授25年的SQL性能与调校经验



## 《Effective C++中文版, 第三版》

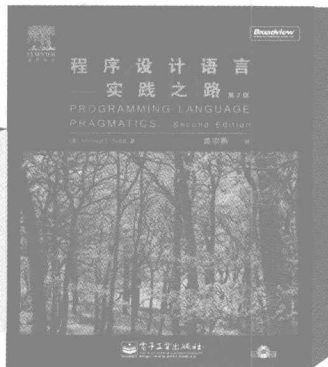
[美] Scott Meyers 著  
侯捷 译

- 自1991年《Effective C++》第一版之后，《Effective C++》第三版隆重出版
- 目标：在一本小而有趣的书中确认最重要的一些C++编程准则
- 作者Scott Meyers以一支生花妙笔将复杂的探索过程和前因后果写成环环相扣、故事性强的文字

## 《程序设计语言—实践之路（第二版）》

Michael I. Scott 著  
裴宗燕 译

- 创造性地将程序设计语言与编译器设计有机结合！修炼扎实的基本功做注重实效的程序员！
- 欧美多所大学“程序设计语言”或“软件系统”课程教材



# 目 录

|                              |     |
|------------------------------|-----|
| 上篇：编程范式与编程语言.....            | 1   |
| 第 1 课 开班导言.....              | 3   |
| 1.1 开班发言——程序员的 4 层境界.....    | 4   |
| 1.2 首轮提问——什么语言好？.....        | 7   |
| 1.3 语言选择——合适的就是好的.....       | 10  |
| 1.4 初识范式——程序王国中的世界观与方法论..... | 15  |
| 1.5 开发技术——实用还是时髦？.....       | 18  |
| 第 2 课 重要范式.....              | 25  |
| 2.1 命令范式——一切行动听指挥.....       | 26  |
| 2.2 声明范式——目标决定行动.....        | 31  |
| 2.3 对象范式——民主制社会的编程法则.....    | 37  |
| 2.4 并发范式——合作与竞争.....         | 43  |
| 第 3 课 常用范式.....              | 49  |
| 3.1 泛型范式——抽象你的算法.....        | 50  |
| 3.2 超级范式——提升语言的级别.....       | 55  |
| 3.3 切面范式——多角度看问题.....        | 63  |
| 3.4 事件驱动——有事我叫你，没事别烦我.....   | 69  |
| 第 4 课 重温范式.....              | 85  |
| 4.1 函数范式——精巧的数学思维.....       | 86  |
| 4.2 逻辑范式——当算法失去了控制.....      | 95  |
| 4.3 汇总范式——一张五味俱全的大烙饼.....    | 103 |
| 4.4 情景范式——餐馆里的编程范式.....      | 109 |
| 第 5 课 语言小谈.....              | 119 |
| 5.1 教学计划——接下来的故事.....        | 120 |
| 5.2 数据类型——规则与变通.....         | 125 |
| 5.3 动态语言——披着彩衣飞舞的脚本语言.....   | 131 |
| 5.4 语言误区——语言的宗教情结.....       | 137 |
| 第 6 课 语言简评.....              | 145 |
| 6.1 系统语言——权力的双刃剑.....        | 146 |
| 6.2 平台语言——先搭台后唱戏.....        | 154 |