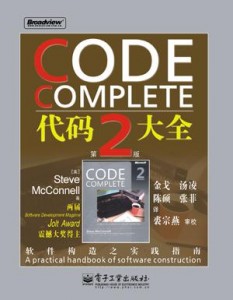
“如果能时光倒流，回到过去，作为一个开发人员，你可以告诉自己在职业生涯初期应该读一本，你会选择哪本书呢？我希望这个书单列表内容丰富，可以涵盖很多东西。”

很多程序员响应，他们在推荐时也写下自己的评语。以前就有国内网友介绍这个程序员书单，不过都是推荐数 Top 10的书。其实除了前10本之外，推荐数前30左右的书籍都算经典，伯乐在线整理编译这个问答贴，同时摘译部分推荐人的评语。下面就按照各本书的推荐数排列。

**1. 《代码大全》**史蒂夫·迈克康奈尔

推荐数：1684



“优秀的编程实践的百科全书，《**代码大全**》注重个人技术，其中所有东西加起来，就是我们本能所说的“编写整洁的代码”。这本书有50页在谈论代码布局。” —— Joel Spolsky

对于新手来说，这本书中的观念有点高阶了。到你准备阅读此书时，你应该已经知道并实践过书中99%的观念。– esac

**2. 《程序员修炼之道》**

推荐数：1504



对于那些已经学习过编程机制的程序员来说，这是一本卓越的书。或许他们还是在校生，但对要自己做什么，还感觉不是很安全。就像草图和架构之间的差别。虽然你在学校课堂上学到的是画图，你也可以画的很漂亮，但如果你觉得你不太知道从哪儿下手，如果某人要你独自画一个P2P的音乐交换网络图，那这本书就适合你了。—— Joel

**3. 《计算机程序的构造和解释》**

推荐数：916

****

就个人而言，这本书目前为止对我影响醉倒的一本编程书。

《**代码大全**》、《重构》和《设计模式》这些经典书会教给你高效的工作习惯和交易细节。其他像《人件集》、《计算机编程心理学》和《人月神话》这些书会深入软件开发的心理层面。其他书籍则处理算法。这些书都有自己所属的位置。

然而《计算机程序的构造和解释》与这些不同。这是一本会启发你的书，它会燃起你编写出色程序的热情；它还将教会你认识并欣赏美；它会让你有种敬畏，让你难以抑制地渴望学习更多的东西。其他书或许会让你成为一位更出色的程序员，但此书将一定会让你成为一名程序员。

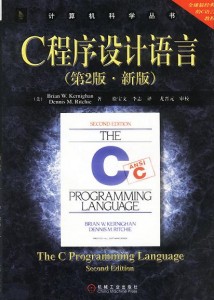
同时，你将会学到其他东西，函数式编程（第三章）、惰性计算、元编程、虚拟机、解释器和编译器。

一些人认为此书不适合新手。个人认为，虽然我并不完全认同要有一些编程经验才能读此书，但我还是一定推荐给初学者。毕竟这本书是写给著名的6.001，是麻省理工学院的入门编程课程。此书或许需要多做努力（尤其你在做练习的时候，你也应当如此），但这个价是对得起这本书的。

你还不确信么？那就读读第一版的前言或序言。网上有免费的电子版。-Antti Sykäri

**4. 《C程序设计语言》**

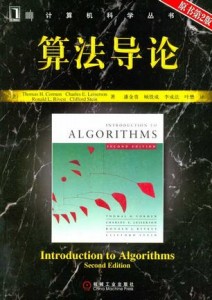
推荐数：774



这本书简洁易读，会教给你三件事：C 编程语言；如何像程序员一样思考；底层计算模型。（这对理解“底层”非常重要）—— Nathan

**5. 《算法导论》**

推荐数：671

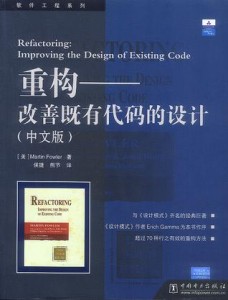


《代码大全》教你如何正确编程；《人月神话》教你如何正确管理；《设计模式》教你如何正确设计……

在我看来，代码只是一个工具，并非精髓。开发软件的主要部分是创建新算法或重新实现现有算法。其他部分则像重新组装乐高砖块或创建“管理”层。我依然梦想这样的工作，我的大部分时间（>50%）是在写算法，其他“管理”细节则留给其他人…… —— Ran Biron

**6. 《重构：改善既有代码的设计》**

推荐数：617

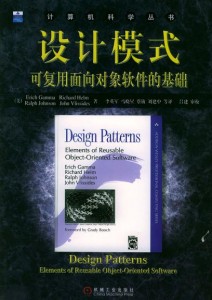


我想我不得不推荐《重构》：改进现有代码的设计。—— Martin

我必须承认，我最喜欢的编程语录是出自这本书：任何一个傻瓜都能写出计算机能理解的程序，而优秀的程序员却能写出别人能读得懂的程序。—— Martin Fowler

**7. 《设计模式》**

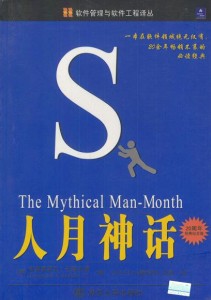
推荐数：617



就我而言，我认为四人帮编著的《设计模式》是一本极为有用的书。虽然此书并不像其他建议一样有关“元”编程，但它强调封装诸如模式一类的优秀编程技术，因而鼓励其他人提出新模式和反模式（antipatterns），并运用于编程对话中。—— Chris Jester-Young

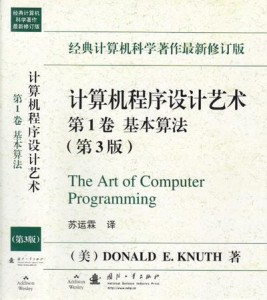
**8. 《人月神话》**

推荐数：588



**9. 《计算机程序设计艺术》**

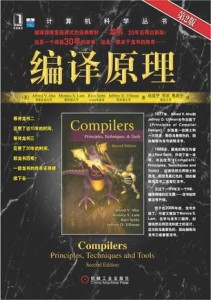
推荐数：542



这是高德纳倾注心血写的一本书。—— Peter Coulton

**10. 《编译原理》**（龙书）

推荐数：462



我很奇怪，居然没人提到龙书。（或许已有推荐，我没有看到）。我从没忘过此书的第一版封面。此书让我知道了编译器是多么地神奇绝妙。- DB

**11. 《深入浅出设计模式》**

推荐数：445

我知道四人帮的《设计模式》是一本标准书，但倒不如先看看这部大部头，此书更为简易。一旦你了解了解了基本原则，可以去看四人帮的那本圣经了。- Calanus

**12. 《哥德尔、艾舍尔、巴赫书：集异璧之大成》**

推荐数：437

如果下昂真正深入阅读，我推荐道格拉斯·侯世达（Douglas Hofstadter）的《哥德尔、艾舍尔、巴赫书》。他极为深入研究了程序员每日都要面对的问题：递归、验证、证明和布尔代数。这是一本很出色的读物，难度不大，偶尔有挑战，一旦你要鏖战到底，将是非常值得的。 – Jonik

**13. 《代码整洁之道》**

推荐数：329

虽然《代码整洁之道》和《代码大全》有很多共同之处，但它有更为简洁更为实际的清晰例子。 – Craig P. Motlin

**14. 《Effective C++》和《More Effective C++》**

推荐数：297

在我职业生涯早期，Scott Meyer的《Effective C++》和后续的《More Effective C++》都对我的编程能力有着直接影响。正如当时的一位朋友所说，这些书缩短你培养编程技能的过程，而其他人可能要花费数年。

去年对我影响最大的一本书是《大教堂与市集》，该书教会我很有关开源开发过程如何运作，和如何处理我代码中的Bug。 – John Channing

**15. 《编程珠玑》**

推荐数：282

尽管我不得不羞愧地承认，书中一半的东西我都没有理解，但我真的推荐《编程珠玑》，书中有些令人惊奇的东西。 – Matt Warren

**16. 《修改代码的艺术》**by Michael Feathers

我认为没有任何一本书能向这本书一样影响了我的编程观点。它明确地告诉你如何处理其他人的代码，含蓄地教会你避免哪些（以及为什么要避免）。- Wolfbyte

同意。很多开发人员讨论用干净的石板来编写软件。但我想几乎所有开发人员的某些时候是在吃其他开发人员的狗食。– Bernard Dy

**17. 《编码：隐匿在计算机软硬件背后的语言》**

我推荐Charles Petzold的《编码》。在这个充满工具和IDE的年代，很多复杂度已经从程序员那“抽取”走了，这本书一本开眼之作。 – hemil

**18. 《禅与摩托车维修艺术 / Zen and the Art of Motorcycle Maintenance》**

对我影响最大的那本书是 Robert Pirsig 的《禅与摩托车维修艺术》。不管你做什么事，总是要力求完美，彻底了解你手中的工具和任务，更为重要的是，要有乐趣（因为如果你做事有乐趣，一切将自发引向更好的结果）。 – akr

（编注：关于这本书，也可以看看阮一峰的读后感。）

**19. 《Peopleware / 人件集:人性化的软件开发》**

Demarco 和 Lister 表明，软件开发中的首要问题是人，并非技术。他们的答案并不简单，只是令人难以置信的成功。第二版新增加了八章内容。 – Eduardo Molteni

**20. 《Coders at Work / 编程人生》**

一本非常有影响力的书，可以从中学到一些业界顶级人士的经验，了解他们如何思考并工作。 – Jahanzeb Farooq

**21. 《Surely You’re Joking, Mr. Feynman! / 别闹了，费曼先生！》**

虽然这本书可能有点偏题，但不管你信不信，这本书曾在计算机科学专业课程的阅读列表之上。一个优秀的角色模型，一本有关好奇心的优秀书籍。 – mike511

**22. 《Effective Java 中文版》**

此书第二版教你如何编写漂亮并高效的代码，虽然这是一本Java书，但其中有很多跨语言的理念。 – Marcio Aguiar

**23. 《Patterns of Enterprise Application Architecture / 企业应用架构模式》**

很奇怪，还没人推荐 Martin Fowler 的《企业应用架构模式》- levi rosol

**24. 《The Little Schemer》和《The Seasoned Schemer》** nmiranda

这两本是LISP的英文书，尚无中文版。美国东北大学网站上也有电子版。

**25. 《交互设计之路》**英文名：《The Inmates Are Running The Asylum: Why High Tech Products Drive Us Crazy and How to Restore the Sanity》该书作者：Alan Cooper，人称Visual Basic之父，交互设计之父。

本书是基于众多商务案例，讲述如何创建更好的、高客户忠诚度的软件产品和基于软件的高科技产品的书。本书列举了很多真实可信的实际例子，说明目前在软件产品和基于软件的高科技产品中，普遍存在着“难用”的问题。作者认为，“难用”问题是由这些产品中存在着的高度“认知摩擦”引起的，而产生这个问题的根源在于现今软件开发过程中欠缺了一个为用户利益着想的前期“交互设计”阶段。“难用”的产品不仅损害了用户的利益，最终也将导致企业的失败。本书通过一些生动的实例，让人信服地讲述了由作者倡导的“目标导向”交互设计方法在解决“难用”问题方面的有效性，证实了只有改变现有观念，才能有效地在开发过程中引入交互设计，将产品的设计引向成功。

本书虽然是一本面向商务人员而编写的书，但也适合于所有参与软件产品和基于软件的高科技产品开发的专业人士，以及关心软件行业和高科技行业现状与发展的人士阅读。

他还有另一本中文版著作：《About Face 3 交互设计精髓》

**26. 《Why’s (Poignant) Guide to Ruby 》**

如果你不是程序员，阅读此书可能会很有趣，但如果你已经是个程序员，可能会有点乏味。

**27.《Unix编程艺术》**

It is useful regardless operating system you use. – J.F. Sebastian  
不管你使用什么操作系统，这本书都很有用。 – J.F. Sebastian

**28. 《Practices of an Agile Developer / 高效程序员的45个习惯：敏捷开发修炼之道》**

45个习惯，分为7个方面：工作态度、学习、软件交付、反馈、编码、调试和协作。

每一个具体的习惯里，一开始提出一个谬论，然后展开分析，之后有正队性地提出正确的做法，并设身处地地讲出了正确做法给你个人的“切身感受”，最后列出几条注意事项，帮助你修正自己的做法（“平衡的艺术”）。

**29. 《Test-Driven Development by Example. / 测试驱动开发》**

前面已经提到的很多书都启发了我，并影响了我，但这本书每位程序员都应该读。它向我展示了单元测试和TDD的重要性，并让我很快上手。 – Curro

我不关心你的代码有多好或优雅。如果你没有测试，你或许就如同没有编写代码。这本书得到的推荐数应该更高些。人们讨论编写用户喜欢的软件，或既设计出色并健壮的高效代码，但如果你的软件有一堆bug，谈论那些东西毫无意义。– Adam Gent

**30. 《Don’t Make Me Think / 点石成金:访客至上的网页设计秘笈》**

取决于你所追求的目标。我喜欢《代码大全》是因纯编程，《点石成金》是一本有关UI设计的卓越书籍。 – Justin Standard