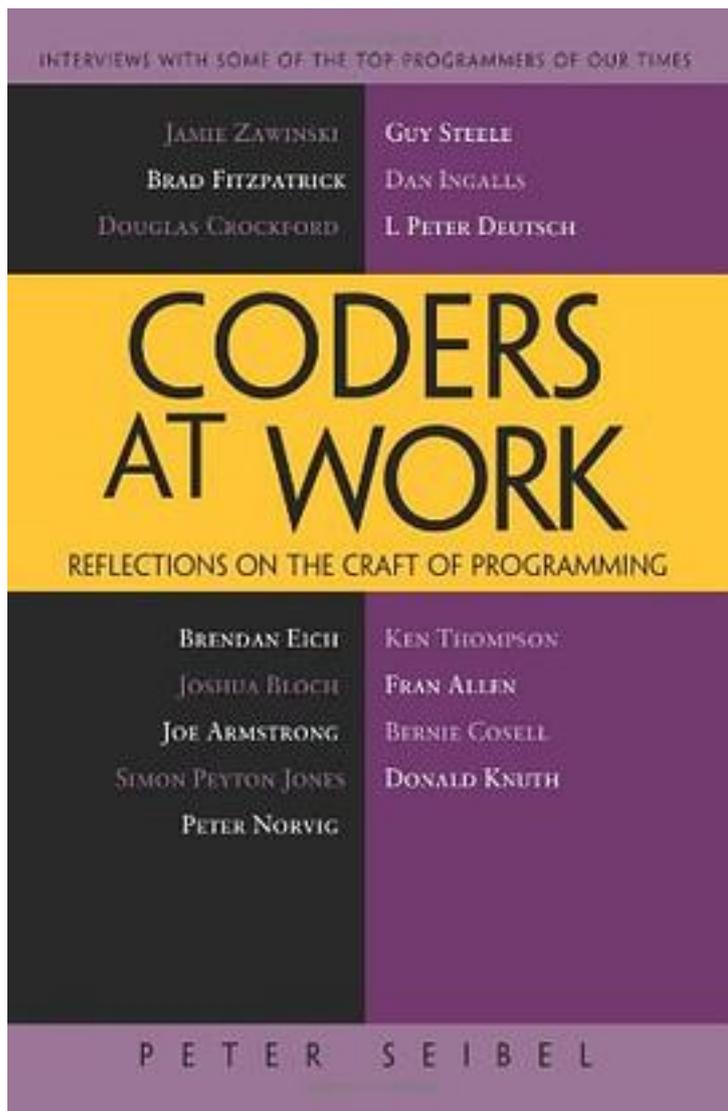# Coders at Work

[Coders at Work_下载链接1_](#)

著者:[美] Peter Seibel

出版者:Apress

出版时间:2009

装帧:electronic

isbn:9781430219491

Peter Seibel interviews 15 of the most interesting computer programmers alive today in Coders at Work, offering a companion volume to Apress's highly acclaimed best-seller Founders at Work by Jessica Livingston. As the words "at work" suggest, Peter Seibel focuses on how his interviewees tackle the day-to-day work of programming, while revealing much more, like how they became great programmers, how they recognize programming talent in others, and what kinds of problems they find most interesting.

Hundreds of people have suggested names of programmers to interview on the Coders at Work web site: www.codersatwork.com. The complete list was 284 names. Having digested everyone's feedback, we selected 15 folks who've been kind enough to agree to be interviewed:

Frances Allen: Pioneer in optimizing compilers, first woman to win the Turing Award (2006) and first female IBM fellow

Joe Armstrong: Inventor of Erlang

Joshua Bloch: Author of the Java collections framework, now at Google

Bernie Cosell: One of the main software guys behind the original ARPANET IMPs and a master debugger

Douglas Crockford: JSON founder, JavaScript architect at Yahoo!

L. Peter Deutsch: Author of Ghostscript, implementer of Smalltalk-80 at Xerox PARC and Lisp 1.5 on PDP-1

Brendan Eich: Inventor of JavaScript, CTO of the Mozilla Corporation

Brad Fitzpatrick: Writer of LiveJournal, OpenID, memcached, and Perlbal

Dan Ingalls: Smalltalk implementor and designer

Simon Peyton Jones: Coinventor of Haskell and lead designer of Glasgow Haskell Compiler

Donald Knuth: Author of The Art of Computer Programming and creator of TeX

Peter Norvig: Director of Research at Google and author of the standard text on AI

Guy Steele: Coinventor of Scheme and part of the Common Lisp Gang of Five, currently working on Fortress

Ken Thompson: Inventor of UNIX

Jamie Zawinski: Author of XEmacs and early Netscape/Mozilla hacker

作者介绍:

Peter Seibel is a serious developer of long standing. In the early days of the Web, he

hacked Perl for Mother Jones and Organic Online. He participated in the Java revolution as an early employee at WebLogic which, after its acquisition by BEA, became the cornerstone of the latter's rapid growth in the J2EE sphere. He has also taught Java programming at UC Berkeley Extension. He is the author of Practical Common LISP from Apress.

目录: Coders at Work—Reflections on the Craft of Programming

· · · · · · ([收起](#))

[Coders at Work_下载链接1_](#)

标签

software_engineering

programming

interview

in_English

computer_science

biography

评论

听牛人吹牛

书评

我要站出来批评一下那个名为"去他的算法内功基础，对于程序员，实用主义才是王道"的评论（我的评论对事不对人）。
首先我要说，这个观点绝对是错误的。表面上这句话好像抓住了"实用主义"的大旗，但却借此抨击算法等基本功的重要性，太误人子弟。就拿Google Fellow Jeff D...

绝世英才谈论他的童年，大学，软件开发的个人习惯，对其他语言的评价。讲到了他们是怎样开发出UNIX操作系统的，让读者如同置身于希腊神话的个人英雄主义时代，看到与现今的开发模式截然不同的做法。还可以看到Thompson对于新的软件开发思想的担忧，不免让人顿生英雄迟暮的感觉...

说是休闲书，是因为我没有从头到尾看的打算，我只是在上班无聊时打开随便看看。以我的英文能力，没有把任何英文读物当成休闲书看的本事，那要求一目十行，我还是只能一句一句地看，顶多发现下面的东西不感兴趣，就整体跳过。
牛人们陈述革命史是不爱看的，看了觉得自己当年的...

这两天每天中午午休前都会看一些《编程人生》。现在已经看了七八个人，这些人开始编程的时候，正好是计算机的起步期，所以对整个计算机的底层都比较了解。不知道这是不是这些人能够成为大师的一个必要条件，自己有没有必要在这方面努力一下。我想这也是很多程序员思考过的问题...

读完图灵俱乐部译的《编程人生》的前两章，给我第一感觉就是：听君一席话，胜读十年书。 Peter
Seibel先生对编程先驱Zawinski、Fitzpatrick的访谈非常精彩。从这两章访谈中，我收获到了以下几点： 1.
保持好奇心，充满激情，编程人生才精彩，编程人生才快乐。著名黑客Zawinski...

前些天和同事开玩笑的说，你愿意花10元钱去听对一位世界顶级大师的采访么？几乎所有的都表示愿意付更多的钱也去。
对呀，很便宜不是么？我读到了这本《编程人生》（英文版名称为Coders at
Work）有十五位编程大师的访谈，我在读书的时候大赚了一笔。
当然我读这本书不是赚...

首先感谢图书大厦给我们买书的冲动。刚看完了jwz, Ken, Peter Novig, Guy
Steele的，发现无论是科班出身还是从屏保程序自说自话，最终这帮人都能获得大牛的成就，感觉好像是当初刚入校门，像冰火里詹姆小时候对拂晓神剑、白牛、巴利斯坦等人的感觉一样："我可是将来要成为火影...

以访谈录的形式来将15位软件先驱的方方面面融合到一本书，起码对于采访者有非常高的要求，这点来说，Peter
Seibel做的非常成功，他对技术及程序员到软件先驱的成长路上的经验与挑战有很好的把握，也就是说，通过Peter的访谈，读者基本能找到自己想要的，也正是本书的一大特色，...

这是一本让人激奋又让人颓唐的书；这是一本让人学会狂妄，或者懂得谦卑的书；这是一本让人藐视编码，或者尊重编码的书；最终，它是一本教会我们从程序中收获乐趣的书，教师是这样一批让人高山仰止的牛人们。
正是因为这些牛人们不同寻常的经历，使得我们在阅读本书时，既充满...

读这本书，你不能指望从大师那学到什么可以立马上手的技能，也不能奢望读完了你就站在了大师的肩膀从此可以一览无遗。相反，这是一本介绍15位世界级编程大师的"发迹"史的。开放的国度和文化造就了先进的IT业，还有他们，这些中国读者熟悉不熟悉的名字。所以，换个角度看，...

这些大师基本上都是大学之前开始编程，并通过各种兼职、实习的方式参与大量的编程实践，又佐证了一万小时理论。
而对于编程语言、编程方式，每个人的看法都不同，甚至截然相反，所以口味的问题并不重要，重要的是深刻的理解。另外，这些程序员几乎都是五六十年代成长起来的那...

看完这本书，对这些伟大的程序员取得的成就无比仰慕。
我认为他们成功的必要条件有两点：
第一就是快乐，他们不是因为成功才快乐，而是因为快乐才成功，他们热爱编程，享受编程；而当你享受一件事情的时候，你就会花费很多时间在上面，这也就是第二个条件，积累...

Jamie Zawinski At the end of the day, ship the fucking thing! It's great to rewrite your code and make it cleaner and by the third time it'll actually be pretty. But that's not the point -- you're not here to write code; you're here to ship products. If y...

只读了Bloch, Knuth, Thompson, Crockford几个人的章节，最后实在是读不下去了，并不是大师们的言论或是表达有问题，而是觉得书里的内容和现在自己所处的状况相距太远了，好比是登泰山，大师们已经纷纷登泰山小天下了，指点江山、回忆过去走过的紧慢十八盘了，评说现在的登山线...

除了第一篇的Jamie还算年轻,其他都是老江湖了。访谈的话题话题,对于普通的计算机从业者来说比较晦涩,即便是能理解,那些上古的内容,泛泛一看没什么启发。觉得有些收获的是这些业界泰斗们对整个行业和他们职业生涯的看法,很难想象有真么多在计算机世界里浸淫几十年的精英。他们对...

多人翻译的，水平参差不齐，有些段落都读不通。

采访的全是硬核开发者。好几位都说自己会用 Emacs 编写代码，没有提到用 Vim 的。不知是不是年代特色，受 Lisp 熏陶所致。除了一位大量使用 C++，有好几位都表示对 C++ 的厌恶。基本都是从机器码开始学习编程，重视阅读源代码，...

多人翻译的，水平参差不齐，有些段落都读不通。

采访的全是硬核开发者。好几位都说自己会用 Emacs 编写代码，没有提到用 Vim 的。不知是不是年代特色，受 Lisp 熏陶所致。除了一位大量使用 C++，有好几位都表示对 C++ 的厌恶。基本都是从机器码开始学习编程，重视阅读源代码，...