

蔡学镛 著

Java 夜未眠

程序员的心声



电子工业出版社
Publishing House of Electronics Industry
<http://www.phei.com.cn>

名家作品

1267
C15d

Java 夜未眠

程序员的心声

蔡学镛 著



A1066332

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书是一本散文集。作为一名资深Java程序设计师，作者用清新自然的笔触记录下自己学习、工作、生活中的所见所思。书中收录的文章内容贴近程序员的生活，令读者产生强烈共鸣。此外，书中部分文章也以轻松的风格介绍了学习Java技术时的一些常见问题，并介绍了一些优秀的Java书籍，能够帮助读者兴趣盎然地学习Java技术。

本书适合于Java程序员、专业软件开发、程序设计爱好者阅读。

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

Java夜未眠/蔡学镛著. —北京: 电子工业出版社, 2003.4

ISBN 7-5053-8660-3

I.J... II.蔡... III.Java语言—程序设计 IV.TP312

中国版本图书馆CIP数据核字(2003)第026982号

责任编辑: 孟迎霞

特邀编辑: 熊 节

印 刷: 北京中科印刷有限公司

出版发行: 电子工业出版社 <http://www.phei.com.cn>

北京市海淀区万寿路173信箱 邮编100036

经 销: 各地新华书店

开 本: 148×210毫米 1/32 印张: 9.06 字数: 166千字

版 次: 2003年4月第1版 2003年4月第1次印刷

印 数: 5000册 定价: 20.00元

序

这本书的出现，纯粹是无心插柳的结果。

四年多前，我还在就读台湾的清华大学信息系（Computer Science）博士班时，和同学一起设计一套 Java MOD（Multimedia—On—Demand）系统，得到了 Java 应用软件竞赛校园组的冠军，也因此不久之后，知名的 O'Reilly 出版社台湾分公司邀请我在网站上开辟 Java 专栏，这开始了我的兼职作家岁月。

于是我在课业和工作之余，写写 Java 书评、技术心得、生活杂感，渐渐地，这些文章竟也得到大家的支持与喜爱，透过网络广泛地散播，这实在是我始料未及的。随着文章累积越来越多，我于是想将这些文章集结出版，以为纪念。

由于我的文章之前都已经在网络上公布，所以我并不预期此书可以卖得多好，幸好台湾 O'Reilly 仍愿意出版此书。此书在台湾出版之后，销售状况相当不错。有好一阵子，此书还高居天珑书局（台湾最大的计算机书籍门市）最畅销的书。

甚至有一次，我在台北市搭乘地铁时，我邻座的先生正在阅读此书，当时我可是开心极了，差一点把他的书抢过来强行签名。

书出版之后，我寄了几本给北京和武汉的朋友，《程序员》的社长蒋涛是其一。蒋涛于是决定出版此书的简体中文版。对我来说，这真是一件求之不得的事。本书得以顺利出版简体中文版，要归功于蒋涛兄与周筠姐，我还要谢谢熊节为此书进行简繁体转换以及文字编辑，也谢谢出版社所有同仁为此书所提供的协助。

我的专业是写程序和讲课，我不把写散文当成我的本业，有人喜欢看我的文章，对我来说已经是很开心的事了，所以我并不想借此书牟利，我将本书繁体中文版的所有版税捐赠给台湾的门诺医院，以为筹建老人安养中心之用。我也决定简体中文版照此原则处理：版税全数捐赠给社会机构。在蒋涛的建议下，我选择“北京青少年发展基金会（希望工程）”为捐赠对象，希望能帮助贫穷的儿童。由于我在台湾，并不方便处理这些捐赠事宜，所以委托蒋涛全权代为处理，非常谢谢他协助我完成此心愿。

希望大家都喜欢这本书。

蔡学镛

2003年2月26日，于台北



目 录

励志篇

学习，是一条漫长的道路.....	2
写程序，好好玩.....	7
如何进入程序设计的领域.....	14
职业敏感度.....	20
从美丽的菲奥莉娜说起.....	23
走出你的成功之路.....	26
从 A 到 E+.....	30

牢骚篇

香鸡排首部曲：程序与香鸡排.....	41
香鸡排二部曲：再论香鸡排.....	45
香鸡排三部曲：完结篇（数据修订版）.....	49
Software Rush.....	52
Ain' t No Sunshine Anymore?.....	55
e 化声声催，e 化路迢迢.....	59
我的座右铭.....	63
一个 Java 信仰者的告解.....	66

生涯篇

程序员的生涯规划.....	70
软件人才何处寻?	74
软件产业的知识经济.....	78
认证无用论?	81
认证有用论?	85
软件与性.....	89

图书篇

你说挑书就像挑老师一样, 我说你乱有思想的.....	94
苦恨年年压金线, 为他人作嫁衣裳.....	98
必也正名乎.....	101
偶像崇拜 (一).....	105
偶像崇拜 (二).....	109
偶像崇拜 (三).....	113
偶像崇拜 (四).....	118

程序设计学习篇

Java 标准简介.....	124
Java 学习之道.....	127
摩登原始人.....	130
你该学什么程序语言?	134
语言、平台、链接库.....	138
OOOO.....	142
Debug.....	146

软件工程篇

Design Pattern新解.....	151
软件产业的大洪水.....	156
没人在乎软件工程.....	160
软件工程的吊诡.....	164
版本控制.....	167

系统篇

不可能的任务.....	172
垃圾!	175
Java 相关的编译技术.....	179
实现 Java 平台的三种方式.....	183
变量的种类.....	187
Shit Happens, Part I.....	190
Shit Happens, Part II.....	193
用 XML 设计一个你自己的程序语言.....	196
Java 繁体中文处理完全攻略.....	201
编译、反编译、反反编译.....	210
Static 的意义与实现方式.....	214

杂感

An Open E-mail.....	221
散文随笔系列文章暂停.....	227
极短篇：王伯伯启示录.....	231
叫专家，太沉重.....	233

比赛杂感.....	237
-----------	-----

书评

Java in a Nutshell A Desktop Quick Reference.....	242
Java Examples in a Nutshell.....	245
Java Threads, 2 nd Ed.....	249
Database Programming with JDBC and Java, 2nd Edition.....	252
Java Swing, 2 nd Ed.....	256
Java 2D Graphics.....	259
Java Virtual Machine.....	263
Enterprise JavaBeans, 2nd Edition.....	267
Java Internationalization.....	270
Java Message Service.....	274

励志篇



- ☞ 学习，是一条漫长的道路
- ☞ 写程序，好好玩
- ☞ 如何进入程序设计的领域
- ☞ 职业敏感度
- ☞ 从美丽的菲奥莉娜说起
- ☞ 走出你的成功之路
- ☞ 从A到E+

学习，是一条漫长的道路

我在 Java 1.0 正式问世前就开始学习 Java，这么多年过去了，到现在我的 Java 学习历程还没有停过。我阅读原文书，研究原始代码，编写程序；自认为走得扎实，不奢望一步登天。像我这样老式的学习方式，显然和现在的快餐主义背道而驰。从许多读者的来信和学生的反应中，我发现大多数的人对于 Java 的学习历程都差不多是：因为公司需要使用 Java 来进行服务器的设计，所以急急忙忙地学习 Java 语言，然后就开始使用 J2EE 的 API，开始写起程序来了。如此急就章的学习方式，程序员基础能力根本就不够，对于面向对象精髓不能掌握，对于 Java 语言内部的运作机制毫无所悉，对 API 的整体连贯性懵懵懂懂。

当然，我们也不好因此责怪程序员，毕竟软件技术变动得

太快。公司不可能给程序员足够的训练之后才开始做项目。程序员一下子被指派使用 A 技术，还没弄懂 A 技术是怎么回事，又被指派使用 B 技术，而且都是缝缝补补的方式边学边用，每次都像是全新的开始，遑论技术能量的累积。

我很庆幸的是，我不太有这样的困扰。因为我是资讯工程系出身（而且我大学时上课一向很认真），所以理论基础稳固，学习新技术对我来说不是难事。我就读大学时，周遭许多同学都瞧不起数据结构、程序语言、操作系统这些所谓“学院派”的课程，以为这些课程一点都不实用。他们认为到了外面公司，这些信息科系所学的一切都派不上用场，“只要会 Visual Basic 和数据库就够了”。这种偏差的心态，恐怕会使得他们在知识经济时代吃足了苦头。

另外还有一派同学很瞧不起程序设计工作，他们告诉我，像我这样会写程序的人，未来进了社会“还不是被他们这些走管理的人踩在脚下”。所以，他们很轻忽理工课程的学习，甚至还有人相当热衷“成功学”，认为这是迈向成功的快捷方式，却因此把学校的课业弃之不顾。我不敢相信有人竟然如此地本末倒置，这就好像是不吃正餐和蔬果，光吃维他命和打点滴。

前一类的人太过于短视近利，后一类的人太过于好高骛远。我一直很不能理解这些人的想法为什么会这样，或许是因为社会环境的风气使然。我很庆幸我到目前还没被社会的大染缸给玷污了（最好这辈子都不要）。

我不认为我的学习方式是一种典范，但是一路走来，倒也颇有进展。许多读者来信问我的学习历程，虽然我个人的学习

方式不见得适用于每个人，但或许还有一些参考价值（特别是对于那些有志进入信息行业的年轻学子），我想透过本文简短地叙述一下。

我一向是采用先深后广（也称为 Bottom-Up、Deep-First）的学习方式。比方说，当我在学 A 技术的时候，学到一半发现需要 B 技术的基础，我会到书局找出一两本 B 技术的书，然后把 A 先搁着，开始看起 B 技术的书。甚至，我在技术书籍上看到不太熟悉的英文语句时，我会找出一本英文语法书详细读过。这种先深后广的学习方式，适合学生时代全面地自我能力提升，但不适合业界人士。试想，老板要你开发的 ERP 系统已经延迟了，你怎有空研究 J2EE 原文书中的英文语法？先深后广的好处是，学习很扎实；缺点是有时候会偏离主题太多。有一次我发现我原本是要学某软件技术，几次“先深后广”下来，我居然看起老子的道德经了。

在技术上，我一直都是一个喜新厌旧的人，很少有软件技术能让我持续研究半年以上，我几乎每隔几个月就要换一次领域。Java 能让我持续这么久，也正是因为 Java 的领域广。透过 Java，我的技术视野变开阔了。这些年来，我换过的 Java 相关领域包括了：虚拟机器、数据库、企业运算、多媒体、2D/3D 图形学、网络等等。

我的学习完全是兴趣导向的，所以压力并不大。因为有兴趣，所以我会很想充分理解一切细节。又因为理解，所以许多原本片片断断的知识都可以渐渐互相融会贯通，累积技术能量，理论和实务之间的藩篱被打破了，学习效率倍增。

我多年来的学习触觉很敏锐，我常常会的抢先一步学好有前

瞻性的技术。比方说，Java 还在 beta 时、UML 还在 0.8 时、XML 还在 draft 时，我都已经透过网络下载技术文件回来每天抱着猛读了。而在 Java、UML、XML 当红之后，我已经差不多把这些技术都摸熟了。

至于该学什么技术，我的判断方式是以技术的优劣来决定。优秀而有独到之处的技术是我的最爱，虽然这类的技术不见得会在市场上胜出，但学习这些技术所得到的启发，对于技术能量的累积与能力的提升会有相当大的助益。至于技术差，但市场需求甚殷者，我还是懒得碰。（好吧！我承认我曾因为市场需求的缘故而学过 MFC。越清楚 MFC 的技术细节，越是讨厌它，这真是个不堪回首的经验。）

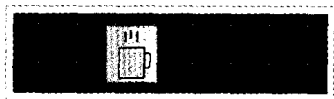
我通常只看英文技术资料，毕竟大部分第一手的技术信息都是以英文来传播的。所以我很早就开始阅读英文技术资料。读英文技术资料的好处是，就算没有学到书中的专业知识，至少也累积培养了英文阅读能力，我一直都是抱着这样的态度。一开始是正襟危坐地看英文技术书籍，字典、翻译机随侍在侧；几年下来，现在是躺着看、趴着看、很随性地看英文技术书籍，因为看英文技术书籍变成一种习惯了。现在，我可以用很快的速度吸收英文技术书籍的知识（有人叫我“吃书的机器”，我把这称号当做是一种恭维）。

近年来，我花在写程序的时间不多，因为时间对我来说很宝贵，而写程序很浪费时间。对初学者来说，大量地写程序是必要的，但过了某个阶段之后，写程序所带来的技术能力成长已经到了极限，还不如多花一点时间看书，学新技术和新观念。

我从国小时期开始学习写程序，迄今已有近十八年的时间；采取上述的方式密集学习，迄今也有近十年的光景。迩来数年，我接触的领域越来越广，而且学习速度正在加快，我认为以前那些努力植下的根苗开始成长了。看看现在的我，或许你会觉得羡慕，但回顾这段学习的岁月，何尝不是一条漫长的道路。

发表日期：April 25, 2001

技术难度：



写程序，好好玩

电视上透过 George&Mary 信用卡借钱的人，个个眉开眼笑，但我们都知道，利用信用卡预借现金又不是不用还，而且利息还高得吓人，那种鼓励大家透支的“理财工具”，还是少碰为妙。为了养家糊口，不敢使用 George&Mary 信用卡的我们，还是乖乖地写程序赚钱好了。写程序，若只是为了养活嗷嗷待哺的一家人，不免辛酸无奈，这时候我们必须想办法挖掘其中的快乐。佛家说：“一花一草，静观皆自得”；学镛说：“This bug that exception，静观皆自得”。只要用点心，其实你会发现，在程序代码中寻幽探访，也可以是很有趣的一件事。特别是在拨云见日，bug 尽除之后，更是有雨过天晴的畅快感受。本文章试图探讨写程序这项工作有趣的一面，其中不乏贼头贼脑、违法乱纪、不堪入目的行径。请各位参考即可，不要贸然实行，以免锒铛入狱。

开后门

如果没有在软件开发的过程中实施XP (eXtreme Programming) 所提倡的Pair Programming (Peer Review), 那么程序员就有机会在程序中开后门, 干些见不得人的勾当。

利用此手法, 你可以进行收集信息。此程序在客户的计算机上执行时, 你仿佛在客户的计算机中装入一个“内神”, 你最好再设计一个情报收集网站“外鬼”, 一旦“内神通外鬼”, 就可进行五鬼搬运, 客户的资料一五一十地通通落到你手中。

也有人利用此手法进行远程遥控, 甚至进行恶劣的破坏行为。程序员个人或者公司有计划在软件内植入后门, 这种情况可能比你想象中来得更多。从使用者的立场来看, 当然是令人感到很担心? 简单的因应之道有二:

- ◆ 少用来路不明的软件, 多用 Open Source 的软件。
- ◆ 少用 native 程序, 多用 Java 程序, 因为 Java 程序可以利用 policy tool 来关闭程序的某些权限。但是, 我发现 Java 软件使用者中, 会使用 policy tool 来自行进行安全设定的人少之又少, 比率接近零。

如果你对于 policy tool 感兴趣, 你可以参考 Java 2 SDK 的文件, 或者 O'Reilly 出版的《Java 安全防护》一书。

变花样

尽管你把程序很快地写完了, 也不要太早告诉主管, 否则主管一看到你这么上进, 又会丢一堆事给你。如果你的程序进度超前太多, 每天无事可干, 只会在办公室内上网阅读 Sleepless in Java, 这就太冒险了, 因为周遭的同事很快会发现你闲着, 并