

# Java 5.0 Tiger

## 程序高手秘笈

Brett McLaughlin  
David Flanagan 著  
O'Reilly Taiwan公司 编译

- Generics
- varargs
- Autoboxing
- printf()  
& format()

O'REILLY®

東南大學出版社

**Java 5.0 Tiger**

程序高手秘笈

## 图书在版编目 (CIP) 数据

Java 5.0 Tiger 程序高手秘笈 / (美) 麦克劳夫林 (McLaughlin, B),  
(美) 弗拉纳根 (Flanagan, D.) 著; O'Reilly Taiwan 公司编译. —  
南京: 东南大学出版社, 2005.8

书名原文: Java 5.0 Tiger: A Developer's Notebook  
ISBN 7-5641-0120-2

I. J... II. ①麦... ②弗... ③ O... III. JAVA 语言—程序设计 IV. TP312  
中国版本图书馆 CIP 数据核字 (2005) 第 057436 号

江苏省版权局著作权合同登记  
图字: 10-2005-086 号

©2004 by O'Reilly Media, Inc.

Simplified Chinese Edition, jointly published by O'Reilly Media, Inc. and Southeast University Press, 2005 Authorized translation of the English edition, 2004 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由 O'Reilly Media, Inc 出版 2004。

简体中文版由东南大学出版社出版 2005。英文原版的翻译得到 O'Reilly Media, Inc. 的授权。此简体中文版的出版和销售得到出版权和销售权的所有者——O'Reilly Media, Inc. 的许可。

版权所有, 未得书面许可, 本书的任何部分和全部不得以任何形式重制。

书 名 / Java 5.0 Tiger 程序高手秘笈

书 号 / ISBN 7-5641-0120-2

责任编辑 / 张烨

封面设计 / Edie Freedman, 张健

出版发行 / 东南大学出版社 (press.seu.edu.cn)

地 址 / 南京四牌楼 2 号 (邮政编码 210096)

印 刷 / 扬中市印刷有限公司

开 本 / 787 毫米 × 1092 毫米 16 开本 12.5 印张 235 千字

版 次 / 2005 年 10 月第 1 版 2005 年 10 月第 1 次印刷

印 数 / 0001-4000 册

定 价 / 28.00 元

## O'Reilly Media, Inc.介绍

为了满足读者对网络 and 软件技术知识的迫切需求,世界著名计算机图书出版机构 O'Reilly Media, Inc.授权东南大学出版社, 翻译出版一批该公司久负盛名的英文经典技术专著。

O'Reilly Media, Inc.是世界上在 UNIX、X、Internet 和其他开放系统图书领域具有领导地位的出版公司, 同时是联机出版的先锋。

从最畅销的《The Whole Internet User's Guide & Catalog》(被纽约公共图书馆评为二十世纪最重要的 50 本书之一) 到 GNN (最早的 Internet 门户和商业网站), 再到 WebSite (第一个桌面 PC 的 Web 服务器软件), O'Reilly Media, Inc.一直处于 Internet 发展的最前沿。

许多书店的反馈表明, O'Reilly Media, Inc.是最稳定的计算机图书出版商——每一本书都一版再版。与大多数计算机图书出版商相比, O'Reilly Media, Inc.具有深厚的计算机专业背景, 这使得 O'Reilly Media, Inc.形成了一个非常不同于其他出版商的出版方针。O'Reilly Media, Inc.所有的编辑人员以前都是程序员, 或者是顶尖级的技术专家。O'Reilly Media, Inc.还有许多固定的作者群——他们本身是相关领域的技术专家、咨询专家, 而现在在编写著作, O'Reilly Media, Inc.依靠他们及时地推出图书。因为 O'Reilly Media, Inc.紧密地与计算机业界联系着, 所以 O'Reilly Media, Inc.知道市场上真正需要什么图书。

# 目录

|                                          |    |
|------------------------------------------|----|
| 《程序高手秘笈》系列 .....                         | 1  |
| 序 .....                                  | 5  |
| 第一章 有什么新玩意? .....                        | 11 |
| 操作 Array .....                           | 11 |
| 使用 Queue .....                           | 15 |
| 使用 Comparator 将 Queue 排序 .....           | 17 |
| Override 返回类型 .....                      | 19 |
| 更好地运用 Unicode .....                      | 21 |
| 加入 StringBuilder .....                   | 22 |
| 第二章 Generic .....                        | 25 |
| 使用 Type-Safe 的 List .....                | 25 |
| 使用 Type-Safe 的 Map .....                 | 28 |
| Interating Over Parameterized Type ..... | 29 |
| 接受 Parameterized Type 作为参数 .....         | 31 |
| 返回 Parameterized Type .....              | 32 |
| 使用 Parameterized Type 作为类型参数 .....       | 33 |

|                                         |           |
|-----------------------------------------|-----------|
| 检查 Lint .....                           | 34        |
| Generic 与类型转换 .....                     | 36        |
| 使用类型通配符 .....                           | 39        |
| 撰写 Generic 类型 .....                     | 40        |
| 限制类型参数 .....                            | 42        |
| <b>第三章 Enumerated 类型 .....</b>          | <b>45</b> |
| 创建 Enum .....                           | 45        |
| 行内 (Inline) 声明 Enum .....               | 50        |
| Iterating Over Enum .....               | 51        |
| Switch 分支的 Enum .....                   | 52        |
| Enum 的 Map .....                        | 56        |
| Enum 的 Set .....                        | 58        |
| 在 Enum 中加入 Method .....                 | 61        |
| 实现有 Enum 的 Interface .....              | 64        |
| 值定的 Class 程序主体 .....                    | 65        |
| 自定义 enum .....                          | 67        |
| 对 Enum 作 Extend .....                   | 68        |
| <b>第四章 Autoboxing 与 Unboxing .....</b>  | <b>70</b> |
| 将 Primitive 转换成 Wrapper 类型 .....        | 70        |
| 将 Wrapper 类型转换成 Primitive .....         | 72        |
| Wrapper 类型的递增与递减 .....                  | 73        |
| Boolean 与 boolean .....                 | 74        |
| Conditional 与 Unboxing .....            | 75        |
| 流程控制语句与 Unboxing .....                  | 77        |
| Method Overload 的解析 .....               | 77        |
| <b>第五章 vararg .....</b>                 | <b>80</b> |
| 创建可变长度的 (Variable-Length) 参数 List ..... | 81        |
| Iterating Over 可变长度的参数 List .....       | 84        |

|                           |    |
|---------------------------|----|
| 允许零长度的参数 List .....       | 86 |
| 于 Primitive 处指定对象参数 ..... | 88 |
| 避免自动化的 Array 转换 .....     | 89 |

## 第六章 Annotation..... 92

|                                          |     |
|------------------------------------------|-----|
| 使用标准的 Annotation 类型 .....                | 92  |
| Annotating 一个被 Override 的 Method .....   | 95  |
| Annotating 一个被 Deprecated 的 Method ..... | 97  |
| 抑制警告 .....                               | 98  |
| 创建自定义的 Annotation 类型 .....               | 100 |
| 对 Annotation 作 Annotate .....            | 103 |
| 定义 Annotation 类型的 Target .....           | 104 |
| 设置 Annotation 类型的 Retention .....        | 105 |
| 制作 Annotation 类型的说明文件 .....              | 106 |
| 设置 Annotation 的继承 .....                  | 109 |
| Annotation 的 Reflecting .....            | 111 |

## 第七章 for/in 语句..... 117

|                                |     |
|--------------------------------|-----|
| 抛弃 Iterator.....               | 117 |
| Iterating Over Array .....     | 120 |
| Iterating Over Collection..... | 121 |
| 避免不需要的类型转换 .....               | 123 |
| 让你的 Class 可用于 for/in .....     | 125 |
| 判别 List 的位置与变量值 .....          | 129 |
| 在 for/in 循环中移除 List 项目 .....   | 131 |

## 第八章 静态的 Import..... 133

|                                |     |
|--------------------------------|-----|
| Importing 静态 Member .....      | 133 |
| 在静态的 Import 中使用通配符 .....       | 135 |
| Importing Enumerated 类型值 ..... | 136 |

|                                   |            |
|-----------------------------------|------------|
| Importing 多个同名的 Member .....      | 138        |
| 遮蔽静态 Import .....                 | 140        |
| <b>第九章 格式化 .....</b>              | <b>142</b> |
| 创建 Formatter .....                | 142        |
| 撰写格式化的输出 .....                    | 143        |
| 使用便利的 format() Method .....       | 149        |
| 使用便利的 printf() Method .....       | 150        |
| <b>第十章 Threading .....</b>        | <b>152</b> |
| 在 Thread 中处理未捕获的异常 .....          | 152        |
| 使用 Thread-Safe 的 Collection ..... | 156        |
| 使用 Blocking Queue .....           | 159        |
| 指定 Blocking 的 Timeout .....       | 163        |
| 从执行逻辑中抽离 Thread 逻辑 .....          | 164        |
| 以 Executor 作为 Service .....       | 166        |
| 使用 Callable 对象 .....              | 167        |
| 无 ExecutorService 地执行任务 .....     | 170        |
| 任务安排 .....                        | 171        |
| 高等 Synchronizing .....            | 173        |
| 使用 Atomic 类型 .....                | 175        |
| Locking 与 Synchronization .....   | 177        |
| 索引 .....                          | 183        |

# 《程序高手秘笈》系列

噢，你挑中了这本书。酷！我真感到兴奋！当然啦，你一定会怀疑为何这些书有着不一样的外观，封面做得好像是在学校用的笔记本（所以，本系列也称为笔记本系列）。我的意思是说，这是奥莱理的书对吧？那动物到哪里去了？还有，说真的，有必要弄出另外一个系列吗？这不能就归作cookbook吗？还是说nutshell，或者是那一种五花八门又很酷的hacks？事实上，我们的构想是要给出包括了封面、字体、甚至是页边空白的笔记这样全新的外观与感受。这都是为了要给你一些在手边真正能用得着的东西。

我深深地认为20世纪90年代被描述成每个人想学会每件事（有何不可？反正我们都有来自网络公司的六位数收入），而千禧年则是“信息痛苦”（information pain），人们没有时间（或者金钱）去读完六百页的书，定期学习两百件事务，但其中只有四种可以用到目前的工作上。如果有办法坐在程序高手（uber-coder）的旁边看着他做事，问问看这些专家们为什么要选择某个特定方法，那个绝招是怎么办到的，或者说他在处理管道流的时候是怎样避免线程的问题，那不是比较好吗？这样的特殊想法是没有书本可以帮上忙的。书本可以指导你，让你决定，但一个程序设计师的思路最终还是无法真地就用只字片纸来抓住。

我们要用这个系列来证明这个想法的假设是错的。

程序高手秘笈如同它所宣称的：书中的潦草手稿与注记正是王牌技客（alpha geek）在玩新语言、API或做项目时的心血结晶。那是解决问题直指要害的程序代码，标注多到当纸镇的成分还比做记号的成分要高。有意思的是，这里并

没有把焦点放在怎样才是漂亮的或者是在你有闲功夫来搞的时候可能还挺有趣的事情上（上一次是什么时候了？），而是着眼于你就是得“交卷”的需求。各位，这可不是在上课，这是实战。如果你想看的是一大堆概念、体系结构与UML图表，本人在此郑重为您推荐我们的动物封面与nutshell系列丛书；如果你要的是太阳底下所有新鲜事的答案，我们的cookbook系列是个中极品；再如果你觉得身陷云里雾里而技术既曲折又坎坷，hacks系列就是你的救星；但如果你是个程序设计师，你写程序，并且你就是想搞定它，那么你需要的就是程序高手秘笈。书中的字字句句是一个开发者想要传达到你心中的想法，而这些只能被厘清到勉强够格出书。我们写得很享受……也希望你能喜欢它。

## 笔记本系列是……

### 范例导向的指引

如同你于稍后的“本系列的编排”一节中会看到的，开发者的笔记本系列全书都是以范例程序代码为中心。你会看到几乎每一页都有程序代码，并且程序代码都是有作用的，而不是那种无聊到根本不值得用纸张去印的“Hello World!”程序。

### 以开发者为目标

曾经读过那些充斥着时髦术语，感觉上不像是程序设计，反倒比较像广告宣传，似乎是专门给呆伯特式（译注1）的老板看的书吗？我们也读过——本系列丛书正是那些书的反方。事实上，一个好的笔记本对不会写程序的人来说是本无字天书（别说我们没有警告你！），并且本来就应该就是这样。但是对于开发者……真是棒到不行。

### 真正值得体会玩味的

你真地有时间坐下来阅读完全无趣的东西吗？如果有的话，可能你正在看的是大部分的语言参考手册。但是如果你跟我们其他人一样，还是看笔记本比较好。实用的程序代码范例，针对应用范例的精要讨论，甚至还掺杂些幽默成分，这才是组成优良笔记本的好材料。

---

译注1 原文并未直接提到呆伯特这个人物，但 pointy-haired bosses 也无其他人选能出其右。

起而行，不坐而谈

如果你在夜深人静的时候想远离计算机好好看本书，这些书帮不了什么忙。这是打算给你在写程序，跟 `bytecode` 奋战的时候用的。因此，笔记本讲的除了程序还是程序。使用前请先启动好你的编辑程序。

## 笔记本系列并不是……

讲义

我们可不随便让任何人都写出一本程序高手秘笈。你得是货真价实的程序设计师，最好是写起程序来废寝忘食的那一种。全职作家、学术研究与理论大师在某些方面是一流的，但本系列是关于真枪实弹的程序设计，里面所谈的都是指令，而不是一些破碎的理论概念。

充斥着概念图形与类继承

这可不是 `nutshell` 系列（老早就跟你说过了）。你不会找到一百页的索引条目列出所有的 `method`，你也别想看到整页都是 UML 图表与 `method`、继承树与流程图，你能看到的都是一页又一页的程序源代码。你有没有开始认识这个循环的主题？

一大堆的解释，一点点的应用程序

目前，许多的程序设计书在给你看到有用的程序代码之前，似乎都有三四章以上的章节。我不确定到底是哪个家伙被作者说服，以为可以让读者等这么久，但铁定不是这系列里面的人。我们认为如果你在十页以内没有写程序的话，一定是出了什么问题。本系列丛书也满载了实用的应用程序，以最快的速度将你从书中范例带到实际工作中。

## 本系列书的编排

开发者的笔记本系列试着要传达与绝大多数其他书籍不同的信息，因为这样，组织编排也不相同。书中是有章节的安排，但是其间的差异也如同程序高手秘笈与传统程序设计书所为相去之远是一样的。首先，你会发现每个章与节中的标题都被安排围绕着一个特定任务，要注意我们说的是“任务”而不是“概念”。这可是攸关于本系列的重点，首要工作就是去完成某件事。每个标题都代表着一个实作（lab）。实作的意思如同它的名字所说的那样，是指完成特定目标的步骤。事实上，在每个实作之下，你会看到的第一个标题就是：“我该怎么做？”

这是每个实作的核心问题，你也会在相关章节中看到有够低级的（down-and-dirty）程序代码与细节。

有些实作中的事务是不要你去做（你在高中的化学实验课曾经把玩过钾吗？），只是为了帮你避免落入同样的陷阱。有些实作告诉你需要在一开始就得注意某些议题的充分理由，因此我们称此段落为“我为何要注意？”的原因显而易见。你会在“发生什么事？”段落中找到一些无法依靠程序范例就能清楚沟通的事物。在这些段落中你会发现概念与理论也是牢牢地盯住手上正在解决的任务，而不是一些用来增加页数的解释文字。最后，有些实作提出了替代方案，以及采用不同手法对付类似问题的疑难解答。这些都属于“关于……”段落，能够帮助交代清楚每个任务于程序设计大方向下的一些来龙去脉。

还有最后一件事情——你会在页边发现草写的眉批。那不是装饰用的，而是来自于产品开发者的提示、绝招与顿悟，有时也可能是个帮你解闷的冷笑话。这些注记代表着整体对话过程的一部分，尽可能让你贴近开发者的想法。希望这可以让你明显地感受到你确实是在向大师讨教（译注2）。

— Brett McLaughlin, *Series Creator*

---

译注2：本系列原文经常使用俚语与市井典故来凸显此系列之与众不同，故译文也以此为乐，读者应不至于误解为译者水平低下。

# 序

*Professional Java*

*Enterprise Java*

*Commercial Java*

这些都是最近在程序设计讨论中常常会谈到的名词。Java被认为是用来开发网页小游戏的玩具语言,想要赶上 C、C++ 这些“老大哥”而又徒劳无功的那种日子已经过去了。当 AWT 与 Swing (现在是 SWT) 已是 Java 语言很重要的一部分时, Java 同时也更向前大步扩展所涉及的范围——操作数据库、财物管理应用、电子商务等。它的速度已可与 C 相比,且 API 更为广泛。影响所及,核心语言也经历过重大的审定过程,然后有了 Java 1.3, 接下来是 1.4, 这些都是让平台迈向成熟的步骤,而不是以激进的方式来变化。

版本发展到代号为“Tiger”的 Java 5.0。事实上,它是“Java 5, version 1.5”。呃,它也是“J2SE”,我当它是“Java 2, Standard Edition, 5, version 1.5”。够混淆了吧? 谢天谢地,不管它叫什么,有了进步这一切纷扰都是值得的,现在这可不再是老爸那一代在用的 Java (或者更精确地说,已经不是你老哥在用的 Java) 了。

Tiger 比旧版本的更新版看起来还要像是个全新产品,满载着比你所知道的 Java 更多的极富戏剧性的变化,你无法光从阅读 release note 就能领会出来。也因为新功能比诡异的版本名称重要多了,所以在本书中我直接称他为“Tiger”,而不管这些 Java 2 version 5……嗯……version 1.5……啊……反正就是 Tiger。

无论 Tiger 最后的官方正式名称叫什么，光这语言所引进的新功能就要花将近两百页才能涵盖到，而你却会发现本书密布着程序代码、范例与洗练的释义。这可不是在浪费空间啊。事实上，那就是你所得到的——Tiger，下一代 Java 的简短密集训练课程。在阅读本书的同时，你会构造你的 `List` 类型、把 `overloading` 带入全新的境界、写出编译期的 `annotation`、将 `threading` 变得比以前更有效率。而以上这些都还没把你在源代码中使用各种新字符的乐趣算进去。没有把 `@`、`<`、`>` 与 `%` 弄到编辑器里面到处都是，那你可算是白活了！好吧，那或许只是我想在工作中增加的一点小趣味。不管是为了什么而接触 Tiger，你会发现有比以前更多的工具随你运用，而且变化比起 Java 1.0 版发行以来的任何版本还要多得多。启动你的编辑器，系好安全带，准备好马上行动。

让我们来驯服这头老虎。

## 本书的编排

本书被设定为交错于学习锻炼（你会从头开始看到尾）与实作参考（你会任意的跳读）之间。大体上来说，你可以自由地浏览目录与索引，去找你想要找的。然而，书中有许多主题是交互关联的（像是 `generic`、`for/in` 语句与 `autoboxing`），你可能会发现你正在阅读一篇假设你已读过前面章节的文章。在这种情况下，只要去浏览参考章节就应该没问题了。小小的额外学习怎么说都不错，是吗？

## 本书是如此写成的

此书是 David Flanagan 与 Brett McLaughlin 的不寻常但成果丰硕的合作结果。David 正在进行《Java in a Nutshell，第五版》的写作，但却渴望能以比该书制作时间还要快的进度来报道 Tiger 主要的语言异动。同一时间，Brett，“开发者的笔记本”这一创新系列的幕后编辑者，非常想要在该系列中来本 Tiger 的书。

接下来事情是这么发生的：

- David 研究 Tiger 的新功能并用它们来为“Java in a Nutshell”写作。他将新题材的草稿寄给 Brett 看。

- Brett带着热情把章节给分解开,几乎将所有内容重新写过,加上新的范例,然后把内容重新组合成笔记本系列的格式。

结果整本书几乎都是 Brett 根据 David 的研究所写的。本书的写作语气与迷人风格来自 Brett,而本书与《Java in a Nutshell》的读者会发现两者间很难得有相同的内容。Brett 只有在极少数的情境下使用与《Java in a Nutshell》相同的程序代码范例,而这些情形都会在页边加以说明。

## 关于范例

本书有遍布各处的数百个范例。文中有些是完整的程序代码,某些则仅显示部分。有些读者可能喜欢自己输入程序,但也有许多像我们这样的人没有时间。因此,每个完整的范例以及几乎所有片段的范例都有已经能够编译的 Java 程序代码可供下载。

另外,编译的过程(特别是 class 的 path 问题)仍然还是 Java 最麻烦的问题。为此,范例也附上了命名为 *build.xml* 的 Ant 编译文件来帮忙。你需要下载并安装 Ant (位于 <http://ant.apache.org>) 以便使用编译文件,我也极力建议你这么做。Ant 很容易安装,如果你有需要辅助的时候也可以参考《Ant: The Definitive Guide》(O'Reilly 出版)。目录的结构应该如下所示:

```
<basedir>
|
+--src (内有 build.xml)
|
+--classes
```

---

### 注意

范例文件已经安排好,直接下载并解压就行了。

---

切换到本机的 *src* 目录下,键入 *ant*。如果你的 Ant 没有正确设置的话你会得到错误信息。不然的话你会看到类似下面的输出:

```
${basedir}\code\src>ant
Buildfile: build.xml

compile:
[echo] Compiling all Java files...
[javac] Compiling 41 source files to code\classes
```

```
[javac] Note: code\src\com\oreilly\tiger\ch06\DeprecatedTester.java
      uses or overrides a deprecated API.
[javac] Note: Recompile with -Xlint:deprecation for details.
[javac] Note: Some input files use unchecked or unsafe operations.
[javac] Note: Recompile with -Xlint:unchecked for details.
```

```
BUILD SUCCESSFUL
Total time: 9 seconds
```

我把 *build.xml* 中其余的 target 留给你自己去探索。大部分的章节中会有关于适用该章或特定范例而标的注记。所有程序代码都详尽测试过，并适度地加上批注。请确定 Tiger 是你的 classpath 中第一个 Java 编译器，不然你会看到一堆乱七八糟的错误。

你可在 <http://www.oreilly.com/catalog/javaadn/> 下载范例程序代码、查阅勘误表、阅读相关资源与在线文件及本书的最新信息。请经常造访这个网址，这里随时会有新的内容或范例更新上线。

## 排版约定

斜体 (*Italic*)

- 路径名、文件名称、程序名称、编译器、选项与命令
- 第一次出现的名词定义
- 网址，例如域名与 URL

黑体字 (**Boldface**):

- 计算机键盘上的特定键
- 用户界面中按钮与菜单的名称

等宽字 (Constant width):

- 在 JSP 网页与 Java 程序中使用的文字与数字，包括了关键字、数据类型、常量、method 名称、变量、class 名称与 interface 名称
- 应该要逐字不差被输入到屏幕的命令行或选项
- 所有的 JSP 与 Java 程序代码
- HTML 文件、标记与属性

等宽斜体字 (*Constant width italic*):

- 需要用自有程序的实际值来取代的地方

等宽黑体字 (**Constant width bold**):

- 程序范例中用户输入的文字

---

#### 提示

这个标示代表对于一旁相邻段落的重点注记。

---

---

#### 警告

这个标示代表涉及相邻段落的注意提示。

---

## 建议与评论

本书的内容都经过测试, 尽管我们做了最大的努力, 但错误和疏忽仍然是在所难免的。如果你发现有什么错误, 或者是对将来的版本有什么建议, 请通过下面的地址告诉我们:

美国:

O'Reilly Media, Inc.  
1005 Gravenstein Highway North  
Sebastopol, CA 95472

中国:

100080 北京市海淀区知春路 49 号希格玛公寓 B 座 809 室  
奥莱理软件 (北京) 有限公司

询问技术问题或对本书的评论, 请发电子邮件到:

*info@mail.oreilly.com.cn*

O'Reilly 的每一本书都有专属网页, 你可以在此找到关于本书的相关信息, 提供包括范例程序的下载、勘误表与相关资源的链接。

*<http://www.oreilly.com/catalog/javaadn>*