

重视大脑的学习指南

第二版
包含全新的
模拟测验

Head First Servlets & JSP™

通过SCWCD考试之路

(中文版)

提供200余个实际的
模拟测验题让
你自测

全面更新以涵盖针
对J2EE 1.5的最新版
SCWCD考试



Watch it!
避开1.5考试中的
致命陷阱



使用c:out向世界
传达你的消息



了解Ted如何利用动态
属性增加魅力

掌握定制
标记库



O'REILLY® 中国电力出版社

Bryan Basham, Kathy Sierra & Bert Bates 著
荆涛 林剑 等译

Head First Servlets and JSP™

第二版

如果有一本Servlet书能比从收件箱删垃圾邮件更刺激，那该多好！
这可能只是个梦想吧……



Bryan Basham,
Kathy Sierra &
Bert Bates 著
荆涛 林剑 等译

O'REILLY®

Beijing • Cambridge • Köln • Sebastopol • Taipei • Tokyo

图书在版编目 (CIP) 数据

Head First Servlets and JSP: 第2版: 中文版 / (美) 巴萨姆 (Basham,B.), (美) 西拉 (Sierra,K.), (美) 贝茨 (Bates, B.) 著; 荆涛等译.—北京: 中国电力出版社

书名原文: Head First Servlets and JSP, Second Edition

ISBN 978-7-5083-8897-7

I. H… II. ①巴…②西…③贝…④荆… III. JAVA语言—程序设计 IV. TP312

中国版本图书馆CIP数据核字 (2009) 第087140

北京版权局著作权合同登记

图字: 01-2009-1959号

©2008 by O'Reilly Media, Inc.

Simplified Chinese Edition, jointly published by O'Reilly Media, Inc. and China Electric Power Press, 2010. Authorized translation of the English edition, 2008 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2008。

简体中文版由中国电力出版社出版, 2010。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。

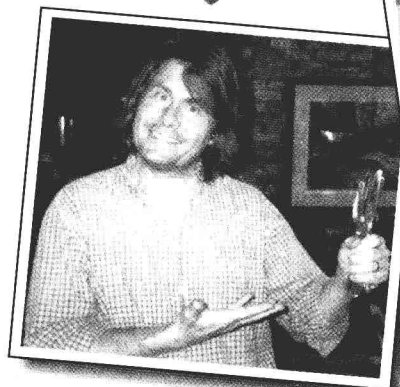
版权所有, 未得书面许可, 本书的任何部分和全部不得以任何形式重制。

书 名/	Head First Servlets and JSP: 中文版
书 号/	ISBN 978-7-5083-8897-7
责任编辑/	刘焜
封面设计/	Edie Freedman, Louise Barr, Steve Fehler, 张健
出版发行/	中国电力出版社
地 址/	北京市东城区北京站西街19号汇置通大厦 (邮政编码100005)
印 刷/	航远印刷有限公司
开 本/	880毫米×1230毫米 20开本 46印张 1265千字
版 次/	2010年8月第1版 2010年8月第1次印刷
印 数/	0001—3000册
定 价/	118.00元 (册)

作者

Head First系列（和这本书）的缔造者

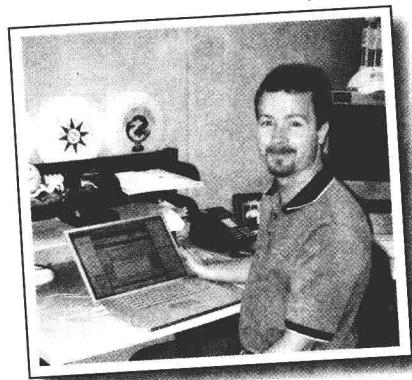
Bert Bates



Kathy Sierra



Bryan Basham



Bert很早就是一位软件开发人员和架构师，不过由于在人工智能领域有近十年的经历，使他对学习理论和基于技术的培训发生了兴趣。在他软件生涯的最初十年，他在全世界游历，为诸如Radio New Zealand、Weather Channel和Arts & Entertainment Network (A & E)等诸多客户提供帮助。他现在是Sun的Java证书考试开发小组中的一员，参与开发了许多证书考试，其中包括新的SCWCD考试。

Bert可以长时间地玩go游戏，无可救药地上了瘾，而且为go程序投入了很多精力。最后，还是Java语言的影响力让他终于罢手。他是一个不错的吉他手，现在正在努力学五弦琴。最近他买了一匹冰岛马，这也是他培训生涯中的一个新体验……

Kathy从开始设计游戏（她为Virgin、MGM和Amblin等都编写过游戏）和开发AI应用以来，一直对学习理论很感兴趣。Head First系列的大多数格式都出自她之手，具体来说，都是她为UCLA Extension（加利福尼亚大学洛杉矶分校）的“Entertainment Studies”研究项目讲授“New Media Interactivity”（新媒体交互）课程时完成的。最近，她成为Sun Microsystems公司的一名高级培训人员，负责教Sun的Java讲师如何讲授最新的Java技术，并参与开发了多个Sun的认证考试，其中就包括SCWCD考试。与Bert Bates一道，她积极地使用Head First概念培训了成千上万的开发人员。她还是世界上最大的Java群体网站的创始人之一，即javaranch.com，这家网站赢得了2003年和2004年《软件开发》杂志生产力大奖。她的爱好包括跑步、滑雪、骑马、玩滑板，还有超自然科学。

Bryan有20多年软件开发的经验，曾经在NASA使用AI技术开发过高级自动化软件。他还曾任职于一家开发定制OO企业应用的顾问公司。目前，Bryan成为Sun课程开发小组的一员，主要关注Java和OO设计原则。他曾参与开发过Sun的许多Java课程的开发，包括JDBC、J2EE、Servlets和JSP，以及OO软件开发。他也是原来和最新版本SCWCD考试的首席设计者。

Bryan很热衷佛教，喜欢玩飞盘，他还是一个音乐发烧友，另外滑雪水平相当高超。

我们的E-mail地址：

terrapin@wickedlysmart.com

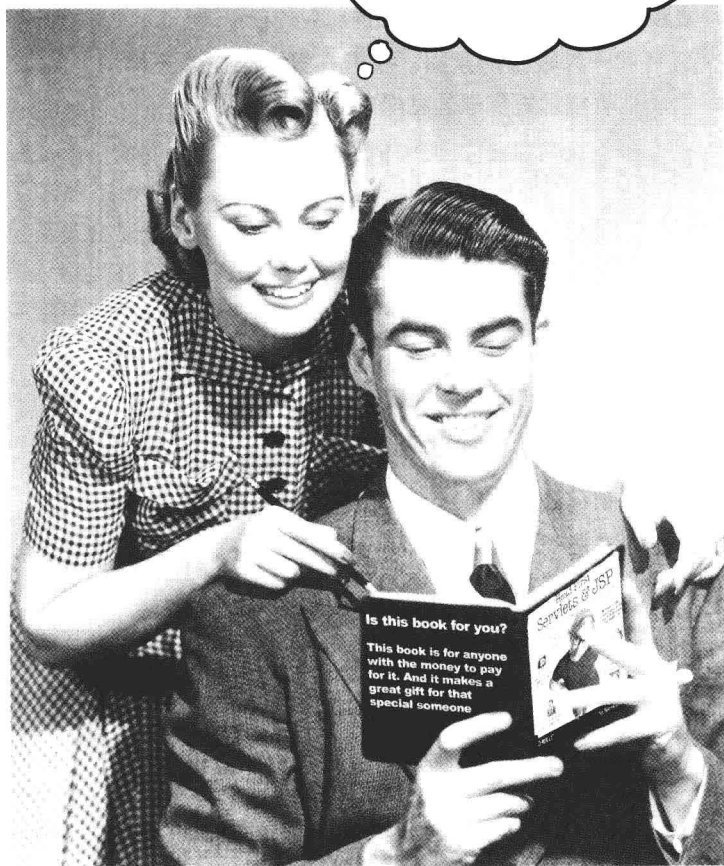
kathy@wickedlysmart.com

bryan@wickedlysmart.com

如何使用这本书

引子

真是无法相信，这样一些东西也能放在一本编程书里！



有一个问题真是听得我们耳朵都磨出茧了，这就是：“你们为什么要把这样一些东西放在一本编程书里呢？”这一节正是要回答这个问题。

谁能看这本书？

如果对下面的所有问题你都能肯定地回答“是”：

- ① 你知道如何用Java编程吗（不过不要求精通）？
- ② 你喜欢挑战困难吗？你是不是愿意在做中学，而不只是纸上谈兵？你想学习并通晓servlets和JSP，把它牢牢记住吗？是不是还希望顺利通过面向Java EE 1.5的SCWCD考试？
- ③ 你是不是更喜欢一种轻松的氛围，就像在餐桌上交谈一样，而不愿意被动地听技术报告似的枯燥乏味的说教？

那么这本书正是你需要的。

谁暂时还不适合看这本书？

如果满足下面任何一种情况：

- ① 你是不是对Java一无所知？虽然不要求你是一位高级Java程序员，但起码有一些经验才行。如果确实一点都不了解Java，先买一本《Head First Java》看看吧，不错，就是现在，事不宜迟，看完了那本书以后，再回来看这本书吧。
- ② 你是不是一个一流的Java开发人员，正在找一本参考书？
- ③ 你本身已经是一个经验丰富的Java EE专家，需要了解一些超高级的服务器技术、特定于服务器的有关问题以及企业体系结构，另外希望得到大量复杂、健壮的实际应用代码，是这样吗？
- ④ 你是不是对新鲜事物都畏头缩尾？只喜欢简单的直条，不敢尝试把条纹和格子混在一起看看？你是不是觉得，如果把Java组件都拟人化了，这样的一本书肯定不是一本正儿八经的技术书？

那么，太遗憾了，这本书不适合你。



[来自市场的声音：只要买得起，就都能看这本书]

我们知道你在想什么。

“这算一本正儿八经的编程书吗？”

“这些图用来做什么？”

“我真能这样学吗？”

我们也知道你的大脑在想什么。

你的大脑总是渴求一些新奇的东西。它一直在搜寻、审视、期待着不寻常的事情发生。大脑的构造就是如此，正是这一点才让我们不至于固步自封，能够与时俱进。

我们每天都会遇到许多按部就班的事情，这些事情很普通，对于这样一些例行的事情或者平常的东西，你的大脑又是怎么处理的呢？它的做法很简单，就是不让这些平常的东西妨碍大脑真正的工作，那么什么是大脑真正的工作呢？这就是记住那些确实重要的事情。它不会费心地去记乏味的东西；就好像大脑里有一个筛子，这个筛子会筛掉“显然不重要”的东西，如果遇到的事情枯燥乏味，这些东西就无法通过这个筛子。

那么你的大脑怎么知道到底哪些东西重要呢？打个比方，假如你某一天外出旅行，突然一只大老虎跳到你面前，此时此刻，你的大脑里会发生什么呢？

看到这只大老虎，你的神经元会“点火”，情绪爆发，释放出一些化学物质。

好了，这样你的大脑就会知道……

这肯定很重要！可不能忘记了！

不过，假如你正待在家里，或者坐在图书馆里。这里很安全，很温暖，肯定没有老虎。你正在刻苦学习，准备应付考试。也可能想学一些比较难的技术，你的老板认为掌握这种技术需要一周时间，最多不超过十天。

这就存在一个问题。你的大脑很想给你帮忙。它会努力地把这些显然不太重要的内容赶走，保证这些东西不去侵占本不算充足的脑力资源。这些资源最好还是用来记住确实重要的事情。比如大老虎，再比如火灾险情。如果你曾经只是身着短衣裤被大雪围困，这件事肯定不会忘却，你的大脑会记住绝不要让这种情况再发生第二次。

我们没有一种简单的办法来告诉大脑，“嘿，大脑，真是谢谢你了，不过不管这本书多没意思，也不管我对它是多么的无动于衷，但我确实希望你能帮助我把这些东西记下来。”

你的大脑想着：这真的很重要。



噢，又是800多页没意思的文字，枯燥又乏味。

你的大脑认为，这些根本不值得去记。

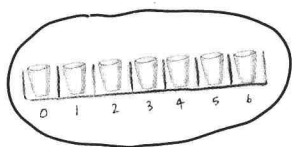


我们认为，“Head First”的读者就是学习者。

那么，怎么来学习呢？首先，必须了解，然后要保证自己确实不会忘记。这可不是填鸭式的硬塞。根据认知科学、神经生物学和教育心理学的最新研究，学习的途径相当丰富，绝非只是通过书本上的文字。我们很清楚怎么让你的大脑兴奋起来。

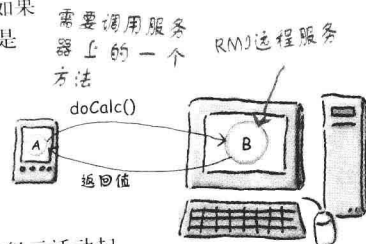
下面是一些Head First学习原则：

看得到。与单纯的文字相比，图片更能让人记得住，通过图片，学习效率会更高（对于记忆和传递型的学习，甚至能有多达89%的效率提升），而且图片更容易看懂。以往总是把图片放在一页的最下面，甚至放在另外的一页上，与此不同，如果把文字放在与之相关的图片内部，或者在图片的周围写上相关文字，学习者的学习能力就能得到多至两倍的提高，从而能更好地解决有关的问题。



采用一种针对个人的交谈式风格。最新的研究表明，如果学习过程中采用一种第一人称的交谈方式直接向读者讲述有关内容，而不是用一种干巴巴的语调介绍，学生在学习之后的考试中成绩会提高40%。正确的做法是讲故事，而不是做报告。要用通俗的语言。另外不要太严肃。

如果你面对着这样两个人，一个是你在聚会上结识的一位有意思的朋友，另一个人学究气十足，喋喋不休地对你说话，在这两个人中，你会更注意哪一个呢？



抽象方法真是没意思。
这些方法没有方法体。



让学习的人想得更深。换句话说，除非你很积极地让神经元活动起来，否则你的头脑里什么也不会发生。必须引起读者的好奇，促进、要求并鼓励读者去解决问题、得出结论、产生新的知识。为此，需要发出挑战，留下练习题和拓宽思路的问题，并要求读者完成一些实践活动，让左右脑都动起来，而且要利用到多种思维。

引起读者的注意，而且要让他一直保持注意。我们可能都有过这样的体验，“我真的想把这个学会，不过看过一页后实在是让我昏昏欲睡”。你的大脑注意的是那些不一般、有意思、有些奇怪、抢眼的、意料之外的东西。学习一项有难度的新技能，并不一定枯燥。如果学习过程不乏味，你的大脑很快就能学会。

`abstract void roam();`

没有方法体！
用一个分号结束。

影响读者的情绪。现在我们知道了，记忆能力很大程度上取决于所记的内容对我们的情绪有怎样的影响。如果是你关心的东西，就肯定记得住。如果你感受到了什么，这些东西就会留在你的脑海中。不过，我们所说的可不是什么关于男孩与狗的伤心故事。这里所说的情绪是惊讶、好奇、觉得有趣，想知道“什么是……”，还有就是一种自豪感，如果你解决了一个难题，学会了所有人都觉得很难的东西，或者发现你了解的一些知识竟是那些自以为无所不能的傲慢家伙所不知道的，此时就会有一种自豪感油然而生。



元认知：有关思考的思考

如果你真的想学，而且想学得更快、更深，就应该注意你怎样才会专注起来，考虑自己是怎样思考的，并了解你的学习方法。

我们中间大多数人长这么大可能都没有上过有关元认知或学习理论的课程。我们想学习，但是很少有人教我们怎么来学习。

不过，这里可以做一个假设，如果你手上有这本书，你确实想学习如何用Java构建Web应用。另外因为你要参加考试，所以需要记住你读到的所有内容。为此必须理解这些内容。要想最大程度地掌握这本书或其他任何一本书中介绍的知识，就要让你的大脑负起责来，要求它记住这些内容。

怎么做到呢？技巧就在于要让你的大脑认为你学习的新东西确实很重要，对你的生活有很大影响。就像老虎出现在面前一样。如若不然，你将陷入旷日持久的拉锯战中，虽然你很想记住所学的新内容，但是你的大脑却会竭尽全力地把它拒之门外。

那么究竟怎样才能让你的大脑把servlet看作是一只饥饿的老虎呢？

这有两条路，一条比较慢，很乏味。另一条路不仅更快，还更有效。慢方法就是大量地重复。你肯定知道，如果反反复复地看到同一个东西，即便再没有意思，你也能学会并记住。如果做了足够的重复，你的大脑就会说，“尽管看上去这对他来说好像不重要，不过，既然他这样一而再、再而三地看同一个东西，所以我觉得这应该是很重要的。”

更快的方法是尽一切可能让大脑活动起来，特别是开动大脑来完成不同类型的活动。如何做到这一点呢？上一页列出的学习原则正是一些主要的可取做法，而且经证实，它们确实有助于让你的大脑全力以赴。例如，研究表明，把文字放在所描述图片的中间（而不是放在这一页的别处，比如作为标题，或者放在正文中），这样会让你的大脑更多地考虑这些文字与图片之间有什么关系，而这就会让更多的神经元点火。让更多的神经元点火 = 你的大脑更有可能认为这些内容值得注意，而且很可能需要记下来。

交谈式风格也很有帮助，当人们意识到自己在与“别人”交谈，往往会更加关注，这是因为他们总想跟上谈话的思路，并能做出适当的发言。让人惊奇的是，大脑并不关心“交谈”的对方究竟是谁，即使你只是与一本书“交谈”，它也不会在乎！另一方面，如果写作风格很正式、干巴巴的，你的大脑就会觉得，这就像坐在一群人当中，被动地听人做报告一样，很没意思，所以不必在意对方说的是什么，甚至可以打瞌睡。

不过，图片和交谈风格还只是开始而已，能做的还有很多。

我想知道
怎样才能骗过我
的大脑，让它记住这
些东西……



我们是这么做的：

我们用了**很多图**，因为你的大脑更能接受看得见的东西，而不是纯文字。对你的大脑来说，一幅图确实能胜过一千个字。如果既有文字又有图片，我们会把文字放在图片当中，因为文字处在所描述的图片中间时，大脑的工作效率更高，倘若把这些描述文字作为标题，或者“湮没”在别处的大段文字中，那么就达不到这种效果了。

我们采用了**重复手法**，会用不同方式，采用不同类型的媒体，运用多种思维手段来介绍同一个东西，目的是让有关内容更有可能储存在你的大脑中，而且能够在多个区中都有容身之地。

我们会用你**意想不到的方式**运用概念和图片，因为你的大脑喜欢新鲜玩艺；在提供图和思想时，至少会含着一些**情绪因素**，因为如果能产生情绪反应，你的大脑就会投入更大的关注。而这会让你感觉到这些东西更有可能要被记住，其实这种感觉可能只是很幽默，让人奇怪或者比较感兴趣而已。

我们采用了一种针对个人的**交谈式风格**，因为当你的大脑认为你在参与一个交谈，而不是被动地听一场演示汇报时，它就会更加关注。即使你实际上在读一本书，也就是说在与书“交谈”，而不是真正与人交谈，但这对你的大脑来说并没有什么分别。

在这本书里，我们加入了40多个**实践活动**，因为与单纯的阅读相比，如果能实际做点什么，你的大脑会更乐于学习，更愿意去记。练习都是我们精心设计的，有一定的难度，但是确实能做出来，因为这是大多数人所希望的。

我们采用了**多种学习模式**，因为尽管你可能想循序渐进地学习，但是其他人可能希望先对整体有一个全面的认识，另外可能还有人只是想看一个示例。不过，不管你想怎么学，要是同样的内容能以多种方式来表述，这对每一个人都会有好处。

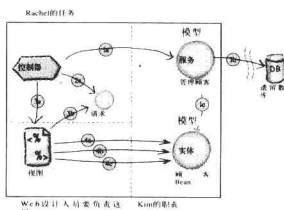
这里的内容不只是单单涉及左脑，也不只是让右脑有所动作，我们会让你的**左右脑都动起来**，因为你的大脑参与得越多，你就越有可能学会并记住，而且能更长时间地保持注意力。如果只有一半大脑在工作，通常意味着另一半有机会休息，这样你就能更有效率地学习更长时间。

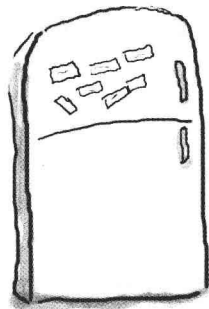
我们会**讲故事**，留练习，从**多种不同的角度来看**同一个问题，这是因为，如果要求大脑做一些评价和判断，它就能更深入地学习。

你会看到我们给出的一些练习，还要回答一些**问题**，这些问题往往不是直截了当就能做出回答，通过征服这些挑战，你就能学得更好，因为让大脑真正做点什么的话，它就更能学会并记住。想想吧，如果只是在健身房里看着别人流汗，这对于保持你自己的体形肯定不会有什么帮助，正所谓临渊羡鱼，不如退而结网。不过另一方面，我们会竭尽所能不让你钻牛角尖，把劲用错了地方，而是能把功夫用在点子上。也就是说，你不会为搞定一个难懂的例子而耽搁，也不会花太多时间去弄明白一段晦涩难懂而且通篇行话的文字，我们的描述也不会太过简洁而让人无从下手。

我们用了**拟人手法**。在故事中，在示例中，还有在图中，你都会看到人的出现，这是因为你本身是一个人，不错，这就是原因。如果和人打交道，相对于东西而言，你的大脑会投入更多的关注。

我们充分利用了80/20方法。我们认为，如果你真的要攻读JSP博士的话，这本书肯定不是你唯一的JSP书。所以我们不打算面面俱到……这里只提供了你真正需要的东西。





可以用下面的方法让你的大脑就范

好了，我们该做的已经做了，剩下的就要看你自己的了。这些提示只是个开头：听一听你的大脑是怎么说的，弄清楚对你来说哪些做法可行，哪些做法不能奏效。还可以做些新的尝试。

把这一页撕下来，贴到你的冰箱上。

① 慢一点。你理解的越多，需要记的就越少。

不要光是看看而已。停下来，好好想一想。书中提出问题的時候，你不要直接去翻答案。可以假想成真的有人在问你问题。你让大脑想得越深，就越有可能学会并记住。

② 做练习，自己记笔记。

我们给你留了练习，但是如果这些练习的解答也由我们一手包办，那和有人替你参加考试有什么分别？不要只是坐在那里看着练习发呆。拿出笔来，写一写，画一画。大量研究都证实，学习过程中如果能实际动动手，将改善你的学习效果。

③ 阅读“*There are No Dumb Questions*（这里没有傻问题）”部分。

顾名思义。这些问题可不是可有可无的旁注，它们绝对是核心内容的一部分！千万不要跳过去不看。

④ 上床睡觉之前不要再看别的书了，或者至少不要再看其他有难度的东西。

学习中有一部分是在你合上书之后完成的（特别是，要把学到的知识长久地记住，这往往无法在看书的过程中做到）。你的大脑也需要有自己的时间来再做一些处理。如果在这段处理时间内你又往大脑里灌输了新的知识，那么你刚才学的一些东西就会被丢掉。

⑤ 要喝水，而且要多喝点水。

如果能提供充足的液体，你的大脑才能有最佳的表现。如果缺水（可能在你感觉到口渴之前就已经缺水了），学习能力就会下降。

⑥ 说出来，大声地说出来。

说话可以刺激大脑的另一部分。如果你想看懂什么，或者想更牢固地记住它，就要大声地说出来。更好的办法是，大声地解释给别人听。这样你会学得更快，而且可能会有一些新的认识，而这是以前光看不说的时候无从发现的。

⑦ 听听你的大脑怎么说。

注意一下你的大脑是不是负荷太重了。如果发现自己开始浮光掠影地翻看，或者刚看的东西就忘记了，这说明你该休息一会了。达到某个临界点时，如果还是一味地向大脑里塞，这对加快学习速度根本没有帮助，甚至还可能影响正常的学习。

⑧ 要有点感觉！

你的大脑需要知道这是很重要的东西。要真正融入到书中的故事里。为书里的照片加上你自己的说明。你可能觉得一个笑话很蹩脚，不太让人满意，但这总比根本无动于衷要好。

⑨ 等完全看完这本书之后再做最后的模拟测验。

如果太早做这个模拟测验，你就不清楚自己为真正的考试准备得怎么样。等你觉得准备得差不多了，再做测验。而且一定要在180分钟的时间内完成这个模拟测验，SCWCD考试的时间正好180分钟。

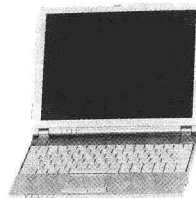
看这本书需要些什么：

要把你的大脑动起来，另外需要一支笔，除此以外，你还需要Java、Tomcat 5和一台计算机。

其他的开发工具就不需要了，比如，不是非要有一个集成开发环境（Integrated Development Environment, IDE）。我们强烈建议，在学完这本书之前，只用一个基本编辑器就可以了，其他的工具先不要用。支持servlet/JSP的IDE会隐藏一些细节，你将无法切身体会这些细节，而这些内容又确实很重要（可能正是要考的），所以，最好完全手工地开发servlet/JSP代码，而不要借助某个工具的力量。等你真正理解了到底会发生什么情况之后，可以再转而采用某个工具，让它自动完成创建和部署servlet/JSP的一些步骤。如果你已经知道怎么使用Ant，那么读完第3章之后就可以使用Ant来帮助你完成部署。不过在你完全记住了Web的部署结构之前，还是建议你不要使用Ant。

得到Tomcat

- 你需要有Java SE v1.5或更高版本。
- 如果还没有Tomcat 5，可以通过以下方式得到：
<http://tomcat.apache.org>
在主页左侧的下载菜单中选择“Tomcat v5.5”。
- 把页面向下滚动到“Binary Distributions”，下载合适的版本。如果不知道需要哪一个版本，可以选择“Core”发行包，这正是你需要的。
- 把安装文件保存在一个临时目录。
- 安装Tomcat。
- 如果是Windows，要双击install.exe文件，然后按照安装向导的指令完成安装。
- 如果是其他系统，你想把Tomcat安装到硬盘上的哪个目录上，就把安装文件解开到那个位置。
- 要想更容易地运行书中给出的指令，请把Tomcat主目录命名为“tomcat”（或者为实际的Tomcat主目录建一个“tomcat”别名）。
- 根据不同的系统，适当地设置环境变量JAVA_HOME和TOMCAT_HOME。
- 你手上应该有一份规范，不过不是非得有一个规范才能通过考试。写这本书的时候，可以从以下地址获得规范：
Servlet 2.4 (JSR #154) <http://jcp.org/en/jsr/detail?id=154>
JSP 2.0 (JSR #152) <http://jcp.org/en/jsr/detail?id=152>
JSTL 1.1 (JSR #52) <http://jcp.org/en/jsr/detail?id=52>
访问JSR，点击最新版本的下载链接（Download Page）。
- 测试Tomcat，为此对于Linux/Unix/OS X，启动tomcat/bin/startup脚本（startup.sh）。在浏览器上访问<http://localhost:8080/>，就会看到Tomcat欢迎页面。



Java 2标准版1.5

Tomcat 5

考试涵盖以下规范：

- Servlets 2.4
- JSP 2.0
- JSTL 1.1

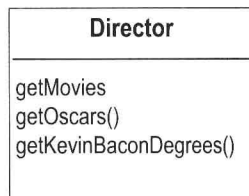
需要了解的最后几点：

要把这看做是一个学习过程，而不要简单地把它看成是一本参考书。我们在安排内容的时候有意做了一些删减，只要是对有关内容的学习有妨碍，我们会毫不留情地把这些部分一律删掉。另外，第一次看这本书时，要从头看起，因为书中后面的部分会假定你已经看过而且学会了前面的内容。

我们对UML有所修改，使用了一种更简单的“UML”。 ↴

我们用了“类”UML图（注意，可不是UML类图，而是指与UML图很相似）。

你很可能早就会UML了，不过这个考试对UML没有要求，所以在学这本书之前不必非要先学UML。你不用担心在学servlet、JSP、JSTL的同时还要学UML。



我们没有把规范中的细枝末节一网打尽。

SCWCD考试是很重细节的，当然我们也一样。但是，如果规范中的某个细节不在考试范围内，我们就不会谈到这个内容，除非它对大多数组件开发人员都很重要。开始开发Web组件(servlet和JSP)需要知道些什么？想通过考试又需要知道些什么？其实这二者85%的内容都是重叠的。我们也谈到了一些考试不要求的内容，不过会特别指出这是考试不要求的，如果只是要通过考试，你就不必去记这些内容。实际的SCWCD考试就是我们出题，所以我们知道你应该把劲往哪里使！如果考试中的某个问题问到了一个过于“吹毛求疵”的细节，但是我们觉得学习这一点有些得不偿失，事倍功半，我们就会忽略这个细节，或者轻描淡写地提一下，也可能只是在一道模拟测验题中问到。

书里的实践活动不是可有可无的。

这里的练习和实践活动可不是可有可无的装饰和摆设，它们也是这本书核心内容的一部分。其中有些练习和活动有助于记忆，有些能够帮助你理解，还有一些对于如何应用你所学的知识很有帮助。千万不要忽略任何实践活动。

我们有意安排了许多重复，这些重复非常重要。

Head First系列的书有一个与众不同的地方，这就是，我们希望你确实实地学会，另外，希望在学完这本书之后你能记住学过了什么。尽管重复很有必要，不过大多数参考书不一定认为重复和回顾是一个重要的环节，但是在这本书里，你会看到一些概念会一而再、再而三地出现很多次。

代码例子尽可能短小精悍。

有读者告诉我们，如果查了200行代码才能找到要理解的那两行代码，这是很让人郁闷的。这本书里大多数例子往往都开门见山，作为上下文的代码会尽可能的少，这样你就能一目了然地看到哪些东西是需要你学习的。别指望这些代码很健壮，要知道这里的代码甚至是不完整的。要开发健壮而且完整的代码，这正是你学完这本书以后要做的工作。书里的示例特意写得很简单，以便于你学习，这些例子的功能往往不太完备。本书的部分代码示例可以在www.headfirstlabs.com得到。

关于SCWCD考试（针对Java EE 1.5）

更新后的SCWCD考试现在叫作“面向Java平台企业版5的Sun认证Web组件开发人员考试（Sun Certified Web Component Developer for the Java Platform, Enterprise Edition 5）”（CX-310-083），不过不要被这个名字吓到。更新后的考试还是针对Java EE v1.4、servlet v2.4和JSP v2.0规范设计的。

我得先通过SCJP吗？

是的。不管是Web组件开发人员考试（Web Component Developer exam）、业务组件开发人员考试（Business Component Developer exam）、移动应用开发人员考试（Mobile Application Developer exam），还是Web服务开发人员考试（Web Services Developer exam），实际上所有开发人员考试都要求你首先是一个Sun认证Java程序员，也就是必须先通过SCJP（Sun认证Java程序员考试，Sun Certified Java Programmer）。

考试中有多少个问题？

参加考试时，你要回答69个问题。每个人回答的问题可能不一样，考试提供了很多套题目。不过，每个人拿到的题目难度都一样，而且每套题都会全面地覆盖所有内容，没有特别的侧重。在实际的考试中，针对大纲中的每个考试要求都至少会有一道题，有些考点可能还涉及不只一个问题。

我要在多长时间之内完成考试？

考试时间是三个小时（180分钟）。对大多数人来说，这都不成问题，因为这些问题都不会很长、很复杂，不会让你太为难。大多数问题都是很简短的多项选择，所以一下子就能确定是不是知道答案。

有哪些题型？

与我们最后提供的模拟测验题几乎一样，只有一个比较大的差别，因为题目是多项选择，实际的考试会告诉你每道题有几个正确答案，但是模拟试题中没有明确指出这一点。另外，你在实际考试中可能会遇到这样一些题目，要求把什么东西拖放到某个位置，这种问题在模拟测验里没有出现。不过，这种拖放问题实际上就是用交互的方式来回答匹配问题，也就是一个东西和另外哪个东西匹配。

要正确回答多少个问题才算通过？

必须答对49个问题（70%）才算通过考试。回答完所有问题之后，不要急匆匆地就按下done按钮，可以让鼠标在这个按钮上停一会，等有足够的勇气后再点击。因为一旦按下，几乎立刻（大约6纳秒）就能知道你是否通过（当然，我们相信你一定能通过）。

为什么这本书的模拟试题不指出有多少个正确选择？

我们希望模拟试题比实际的考试要稍微难一点，这样你就能对自己是否已经做好准备有一个最真实的认识。一般来说，人们在做书上的模拟试题时得分总是高一些，因为这套题目可能做了不只一次，这样一来，你可能对自己的准备程度有过高的估计，我们可不希望你在估计不足的情况下草率参加考试。读者后来都反馈说他们最后的考试得分和做这套模拟题的得分相当接近。

考试之后我能拿到什么？

离开考试中心之前，记得要拿到你的测验报告。你在每一部分得分多少，会在这个测验报告中有一个总结，另外报告中还会指出你是否通过了考试。要留好这份报告！这是你通过认证的第一份证明。考试完再过几周，你会收到Sun教育服务机构（Sun Educational Services）寄出的一个小包裹，这里面有你真正的纸质证书，Sun给你的一封祝贺信，还有一个可爱的胸针，上写Sun Certified Web Component Developer（Sun认证Web组件开发人员）几个“小字”，这几个字实在是太小了，所以就算是你凭这个胸针谎称自己得到了其他认证，可能也能蒙混过关，没人能看出来有什么分别。如果你想喝点酒庆祝自己通过了考试，别指望Sun教育服务机构会随包裹给你寄瓶酒。

考试费用是多少，怎么注册呢？

考试的费用不便宜，需要\$200。所以你需要这本书……，确保一次就能通过考试。你要通过Sun教育服务机构注册，向他们提供你的信用卡号。相应地，你会得到一个准考证号，用这个准考证号就可以与离你最近的考试中心（Prometric Testing Center）约一个考试时间了。

要想从网上了解有关的详细情况并购买一个准考证，可以先访问：<http://www.sun.com/training/certification/>。如果你在美国，访问这个网页就足够了。如果你不在美国，可以从右边的菜单条选择你所在的国家。

考试软件是什么样子？

这个软件用起来极其简单，就是向你问一个问题，再由你回答。如果你还不想回答，可以先跳过去，等以后再答。如果确实回答了，但是没有把握，希望在有时间的情况下再检查一下，可以对这个问题“做个标记”。做完之后，你会看到一个界面，其中显示出哪些问题还没有回答，另外哪些问题加了标记，这样就能直接返回去再完成或检查这些题目。

在考试刚开始的时候，会给你提供一个简单的教程，告诉你怎么使用这个软件，还会让你做一个小测验（与servlet无关）练练手。学习这个教程用的时间不算在SCWCD考试时间内。等你学完这个考试软件教程，而且确实准备好了，时钟才开始计时。

到哪能找到一个有关这个考试的学习小组，要准备多久才能参加这个考试？

要说关于这个考试的在线讨论组，最好的莫过于本书作者维护的论坛了（当然了，这有什么奇怪的）。你可以访问www.javaranch.com，然后浏览Big Moose Saloon（所有论坛都在这里）。这个讨论组很有用，可别错过了。在那里肯定会有人回答你的问题，没准回答问题的正是我们自己。JavaRanch是互联网上最友好的Java群体，所以不管你的Java水平如何，都不用害怕，这里欢迎每一个人。如果你还需要通过SCJP考试，同样可以在这里获得帮助。

要准备多久才能参加这个考试，这要取决于你已经有多少servlet和JSP经验。如果你以前从未接触过servlet和JSP，可能需要6~12周的时间，具体要多久，这要看你每天在这上面花多少时间。如果你已经有丰富的servlet和JSP经验，往往只需要3个星期的时间就能做好准备。

Beta版测试人员&技术审校



Dave Wood

没有附照片 (但同样
令人敬畏): Amit
Londhe

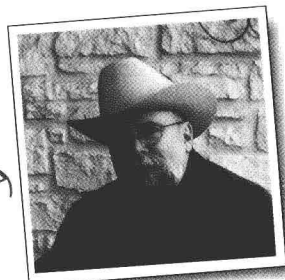
Philippe Maquet



Joe Konior



Bear Bibeault

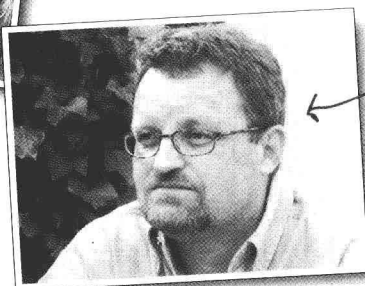


由于本书多了
两根白头发

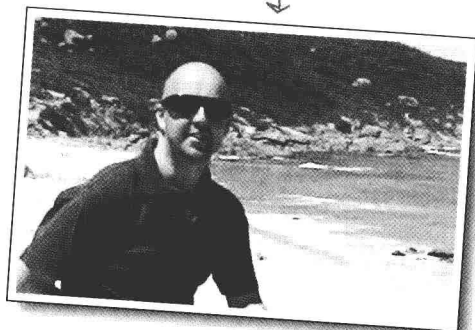
Neeraj Sing



Johannes dedong



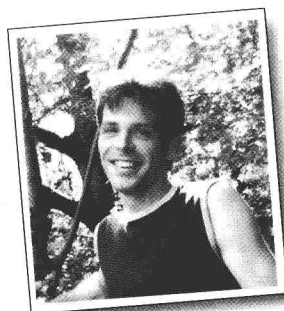
Andrew Monkhouse



Jason Menard



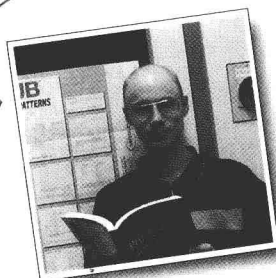
Jet Camps



Dirk Schreckmann



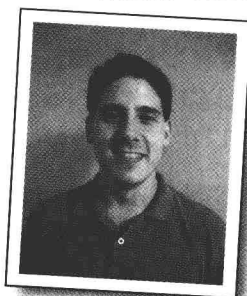
Oliver Roell



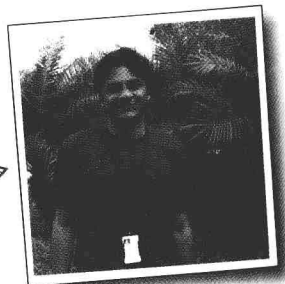
Ulf Dittmer



Theodore Casser



Preetish Madalia



Sergio Ramirez



Collins Tchoumba



感谢 其他要批评的人：

致O'Reilly:

最要感谢的要算O'Reilly的Mike Loukides, 所有策划都归功于他, 是他的大力帮助才使Head First概念形成了一个系列。而且我们很庆幸, 这位编辑确实对Java相当精通。另外, 非常感谢Tim O'Reilly, 是他在幕后推动Head First走到今天。真的很幸运, 他总是能高瞻远瞩, 而且极富开创意识。非常感谢你, 聪明的Kyle Hart, 你是当之无愧的“Head First系列之母”, 是你发现了Head First正满足计算机图书领域所需, 并促使了这个系列的诞生。

致勇敢无畏的技术审校:

不错, 这本书比我们预想的要厚一些。不过, 如果没有JavaRanch审查主管Johannes deJong, 肯定会推迟很多。Johannes, 你是我们的英雄。还要特别感谢Joe Konior, 你对每一章的反馈篇幅都刚好合适。而且非常非常感谢Philippe Macquet, 你的一丝不苟, 你的高超技术, 还有乐观的态度, 让我们三个作者都迷恋不已, 恨不能和你结婚……不过这当然有些怪异。我们特别感谢Andrew Monkhouse提供的技术反馈, 而且帮助我们调整澳大利亚式的英语。Jef Cumps, 你唱的“setHeader”真是不怎么样(不过倒是很有感情), 但你的技术评论确实很有帮助。

Dave Wood总是喜欢打击我们, 最喜欢指着前几页说, “看看, 这不太Head First嘛”。另外, JavaRanch审查官Jason Menard、Dirk Schreckmann(前面的那个鱼脸)、Rob Ross、Ernest Friedman-Hill和Thomas Paul都提供了绝妙的反馈。一如平常, 要特别感谢javaranch Trail的老板, Paul Wheaton。

还要特别感谢第二版的以下技术审校: Bear Bibeault、Theodore Casser、Ulf Dittmer、Preetish Madalia、Sergio Ramirez、Oliver Roell、Neeraj Singhal和Collins Tchoumba。



Marc Peabody

模拟测验

如果你发现一道JSP模拟题实在太难, 害你苦思冥想也不得要领, 不要怪我们, 要怪就怪Marc Peabody! 感谢Marc对我们的帮助, 让参加SCWCD考试的人能做好充分的准备。Marc把大量空闲时间都用来审查JavaRanch, 在这里他因鼓动程序员用普通的Java EE技术构建出人意料的mashup应用而著称。