十大事件与Java相关的四十个名字

## 十大事件

### 1990－1994：Java缘起

文/孟岩

Larry Wall说，优秀程序员应有的三个特点:懒惰、急躁和傲慢。Java就是诞生在一群懒

惰、急躁而傲慢的程序天才之中。

1990年12月，Sun的工程师Patrick Naughton被当时糟糕的Sun C++工具折磨的快疯了。

他大声抱怨，并威胁要离开Sun转投当时在Steve Jobs领导之下的NeXT公司。领导层为了

留住他，给他一个机会，启动了一个叫做Stealth（秘密行动）的项目。随着James

Gosling等人的加入，这个项目更名为Green。其目标是使用C++为嵌入式设备开发一种新

的基础平台技术，James Gosling本人负责开发一个SGML编辑器。正如人们事后分析的那

样，这位天才的程序员太懒惰，所以没有把C++学好，开发中碰了一头包;太急躁??所以

不愿意停下来读读Scott Meyers的新书《Effective C++》;太傲慢??所以轻易地决定开

发一中新的编程语言。他把这种语言命名为C++++--，意思是C++“加上一些好东西，减

去一些坏东西”。显然这个糟糕的名字不可能长命百岁，很快这种颇受同伴喜爱的小语

言被命名为Oak。

到了1992年9月，Oak语言连同Green OS和一些应用程序一起发布在称做Start 7的小设备

上，从而使之有了第一次精彩的亮相。随后，Sun开了一家名为FirstPerson的公司，整

个团队被转移到这家公司里研发机顶盒，以投标时代华纳公司的一个项目。这帮天才被

技术狂热所鼓舞，开发出了一个高交互性的设备，结果没想到时代华纳公司和有线电视

服务商并不愿意用户拥有那么大的控制权，从而在竞标之战中败给了SGI。Oak的锋芒之

锐，竟然把客户都给吓懵了。Sun沮丧地关闭了FirstPerson，召回了整个团队。事实证

明，传统行业中那些脑满肥肠的保守主义者是腐朽没落的。回去！回到激情澎湃的IT产

业，抓住互联网的大潮，这才是出路！1994年，Oak被命名为Java，针对互联网的新一轮

开发如火如荼，一切已经就绪，熔岩在地下奔流，火山即将喷发。

### 1995: Java香浓世界

文/马伟

1995年，Sun正式对外公布了Java，并且发布了JDK 1.0。这种外形酷似C++，却包含一颗

Smalltalk般纯洁的面向对象之心的全新程序设计语言及其平台，几乎在一夜之间就成为

软件产业的新宠儿。Java当时仅仅被用来为网站制作一些动态应用,诸如动画图片之类，

但这仍然引起了很多Web开发者们的注意，他们非常渴望有一种安全的语言，可以在静态

的HTML网页上制作动画图片。Sun最终把Java集成到NetScape浏览器。同时因为它具有

“只写一次，随处运行”的特性，而引起了很多开发者的注意，他们可以再也不用为了

使程序能够在不同型号的硬件上运行而耗费大量的时间来编译代码了。

当时的Web浏览器的出现也为Java的出现起到了很好的推动作用，通过Java和Web浏览器的结合，人们似乎看到了什么，有人甚至预言PC将在一两年内退出历史的舞台，取而代

之的是基于Java的浏览器应用程序，通过网络计算设备来进行应用。Java的出现为当时

的软件产业带来了无限的遐想。

### 1996：Java大跃进，盟主地位就此定

文/马伟

SUN在1996年一开始首先成立了JavaSoft组织，并在1月23日正式发布自己的Java 1.0，

作为20世纪业界出现的最重要的技术之一，Java引起了编程世界的革命。直到现在，

Java仍然是互联网上最流行的语言。

在Sun正式发布Java 1.0之后，Java这门新生的语言就拥有了自己的会议??JavaOne，这

次会议初试啼音就吸引了600多名参与者。除了拥有这么多的积极参与者来进行Java的开

发之外，各大知名公司也纷纷向Sun申请Java的许可。一时间，NetScape、惠普、IBM、

Oralce、Sybase甚至当时刚推出Windows 95的微软都是Java的追随者。

Java的应用就像是世界上的顶级玩家们组成的一个公开联盟，告诉全世界我们大家就是

都在用着Java。也正是因为如此，Java也找到了自己的归宿。现在的J2EE已经成为中大

型企业级应用的标准，成为承接数据库和Web之间的一个重要桥梁。

当年Java的机会实在太多了，以至于很难知道到底该做什么。最终Java在应用服务器市

场获得了难以取代的地位，也确定了J2EE的发展方向，并且仍将延续下去。

### 1997-2001: 微软与Sun的Java官司

文/孟岩

Java诞生的1995年，正是微软在软件产业地位达到巅峰的时代，Windows 95发布时的风

光场面给人们留下的深刻印象至今难忘。尽管如此，作为最卓越的技术领袖，比尔?盖茨

仍然敏锐地注意到Java。当他了解了Java的一些细节之后，给予了这样的评价:“Java是

很长时间以来最优秀的程序设计语言。”基于此，微软于1996年3月申请并获得了Java许

可证。微软对于Java的这一热情态度在当时大大提高了人们对Java的兴趣和信心，但也

有不少人担心微软会依靠自己强大的影响力在标准之外另立标准，从而破坏Java的纯洁

性。

果然，从1997年发布Visual J++的第一个版本开始，微软就开始在Java中掺入自己的私

有扩展。这毫无疑问引起Sun的高度重视。1997年10月，Sun向美国加州地方法院起诉微

软公司违反两公司就微软使用Java技术所签定的合同，指控微软公司在自己的Java产品

中做了“不恰当的修改”，违反了合同中承诺向用户提供Java兼容产品的条款。这一官

司旷日持久，直到2001年1月双方达成和解，微软将继续提供采用Sun开发的Java技术的

现有产品（包括测试版）。不过，Sun有限制地仅对包括Java 1.1.4的微软产品提供许

可。到了2001年7月，微软公布新版的Windows XP将不再支持Sun的JVM，并且推出了.NET

平台与Java分庭抗礼。

现在回过头去看，当时的这一场官司对Java世界产生了深远的影响。如果没有这一场官

司，也许很多Java程序员都在使用Visual J++，基于WFC开发Windows客户端程序，同时

不得不面对被两个不同的事实标准所分裂的Java世界。

### 1998：Java 2平台发布

文/陶文

1998年，Java 2平台正式发布。经过了三年时间的发展、热热闹闹的攻关宣传、红红火

火的众厂商的热情参与，Sun终于知道Java适合干什么了。对比Java刚发明时的技术定

位，与Java的戏剧性触“网”的那段历史，Java 2平台的发布可真算得上是有的放矢

了。根据官方的文档，Java 2是Sun意识到“one size doesn’t fit all”之后，把最

初的Java技术打包成三个版本的产物，也就是著名的J2ME、J2SE、J2EE。

之所以说Java自从Java 2平台发布之后，进入了现代。那是因为之前的历史怎么看来都

和现在程序员日常开发使用的技术无什么关系，比如Applet，已经很少有人使用了。

Java 2之后的历史就不一样了，至少人们在推崇轻量级开发，猛批EJB时还不时会引用

J2EE这个词是如何诞生的。而Java 2的三大版本中，除了J2EE得到了长足发展和广泛使

用之外，J2ME也在手机市场上取得了遍地开花的结果。相较之下，J2SE难免落寞，只剩

SWT这个血统不纯的家伙在Rich Client回归的时代吸引着人们的眼球了。无论今天看来

当时的Java 2有多么的不成熟，至少经过市场和时间的检验，Java 2规划出来的三大方

向把Java技术指向了光明的方向是勿庸置疑的。

### 1998：JCP成立并正式运作，Java开源社群开始蓬勃发展

文/黄海波

1998年，JCP组织成立，并且开始把握Java的发展方向。JCP组织的开放性，不但使得所

有对Java感兴趣的商业公司可以参与Java的发展，更重要的是JCP允许个人、非盈利组

织、学校等加入，这就给Java带来了巨大的活力。随之兴起的Java开源运动的最大贡献

是实现和鼓励了知识共享，在众多热情的开源程序员们的努力和分享下，很多原先只被

商业公司掌握的技术、思想和产品可以被所有需要的开发人员免费或者以较低的价格获

得使用权, 并通过开放源代码更容易的获得反馈和改进意见从而进一步演化发展。我们

知道，所谓知识不是孤立发展认知，而是人们的经验，认识是思考交流和积累的产物。

而开源运动所带来的开放、反馈、交流的风气正是符合人类社会知识形成和发展的规

律。

开源运动起源于西方的发达国家，有其现实背景和文化根源。1990年代可以说是IT产业

的一个黄金时代。信息时代的兴起对IT人员，特别是软件人员有着巨大的需求。而软件

开发又是一种类似艺术创作的脑力活动，和所有的艺术家、作家们一样，在作品打上自

己的印记并流传在世界上是每一个创作人员的梦想。互联网时代下的高收入的舒适生

活，早九晚五的编写公司的代码并不能满足很多有激情的软件开发人员的梦想，再加上

西方传统的基督教文化中十分推崇的分享和交流，开源的出现和兴起也就水到渠成了。

今天，开源运动已经不仅仅是一些个人天才程序员们的游乐园地，而是发展成为一项开

源软件产业。

### 1998：WebLogic打开J2EE的魔匣

文/霍泰稳

Java语言的出现使得互联网络有了良好的交互性能，但这些很“酷”的技术仅被人们认

为是一些小花招，它还无法消除企业级用户对它的怀疑。1998年，BEA公司宣布收购

WebLogic公司，并接着推出由Sun公司第一个授权使用J2EE许可证的WebLogic Server应

用服务器，这个Java版的AppServer一推出就引起业界极大的兴趣。WebLoigc Server以

其对标准的支持、强悍的运算能力和安全的架构设计等特性也很快征服了那些怀疑J2EE

应用的人们。推出市场后不到一年，WebLogic Server就成为业内第一Java应用服务器。

这里我们援引一些当时著名咨询公司的调查数据来说明问题，“在IDC的报告中，BEA在

应用服务器和交易服务器领域市场份额第一；在Gartner的报告中，BEA WebLogic

Server拥有业内最广泛的EJB应用安装基础；在Giga Group的报告中，BEA WebLogic

Server市场份额占32%”。

因为应用服务器市场极大的发展潜力，在WebLogic Server之后，其它的很多公司也推出

了自己的AppServer，如IBM的WebSphere、Sun公司的iPlanet等，逐渐地应用服务器取代

了传统意义上的各类中间件，成为企业应用的基础平台。应用服务器的出现使得Java有

了真正意义上的发展。

### 2002-2004: Sun与微软的法律碰撞最终以喜剧收场

文/恶魔

2003年4月2 日，Sun与微软达成16亿美元的法律和解。如果不是晚了一天，许多人会以

为这是一个在4月1日愚人节开的玩笑。尽管当时所有人都像是看到“太阳从西边出来

了”那样张大了嘴巴，但这的确是事实。

根据两家公司达成的版权协议，双方会为采用对方的技术而支付专利费用，微软向Sun提

前支付3.5亿美元使用费，Sun则承诺，如果Sun集成微软的某些技术，也会向微软付款。

毫无疑问，“私下了结”的方式对双方而言都是最好的结果。就在协议签署的当天，在

美国旧金山由Sun和微软为“抛弃十年恩怨、携手合作“举行的新闻发布会上，尽管比

尔?盖茨没有到场，但这并没有防碍现场看起来异常轻松的气氛。麦克尼利和鲍尔默各自

穿了一件密歇根州底特律“Red Wings”曲棍球队的运动服，并谈及了一起在哈佛大学读

书的经历，麦克尼利还说:“当时我们两人是非常要好的朋友，当然我们也有吵架的时

候。”人与人当然可能成为终生的知己，但是公司与公司之间有的只能是利益上的分分

合合。

### 2000-2004: JBoss和Eclipse

Java开源软件的王者

文/莫映

Java和开源几乎就是天生的一对，这可以从无比兴盛繁荣的Java开源软件社区得到佐

证。目前最有影响力的Java开源软件项目，要数JBoss和Eclipse。可以说，几乎所有的

Java开发人员都获多或少的听到过或接触和使用过它们。前者是目前最优秀、应用最为

广泛的企业级开源J2EE应用服务器，后者是功能完全可以替代商业产品的Java IDE。二

者的覆盖功能之全、支持工具之广、子项目之多，几乎可以仅凭借它俩来完成企业应用

的开发构建到部署实施的全过程，而软件开发者和客户也都可以最大程度上享受高质

量，高可靠Java开源软件所带来的低成本优势。

JBoss和Eclipse的巨大成功，几乎令各自领域的商用竞争者抓狂，其中BEA的WebLogic和

IBM的WebSphere在商业利润上受到JBoss的巨大侵蚀，而Borland的JBuilder、JetBrains

的IDEA等诸多优秀的商用开发工具也不得不面对Eclipse独大的现实。JBoss的CEO兼创始

人 Marc Fleury曾直言不讳地表示，希望占据市场主导地位。“我们希望打败IBM，成为

中间件领域里最大的厂商。”JBoss在4.0以前还只是以一个Group存在，盈利手段主要靠

服务和销售文档。但在最近，JBoss已经发展成为一个有限公司，并吸纳多家风险投资，

专注于获取利润为目标之一的第二代开源软件模式（JBoss自己称为“Professional

Open Source”）的创新和运营。这区别于以理论研究为爱好的学院型开源或大公司为基

础的非盈利组织开源，如Linux和Apache。当然JBoss的这种运营方式势必会导致更多的

代码控制和专有修改权，但按JBoss的说法是这样更能获得企业客户的信赖。JBoss的这

种模式是否能获得成功还要我们拭目以待。

不管JBoss和Eclipse的未来发展如何，JBoss和Eclipse的成功已经让我们看到了Java开

源软件的威力，祝愿它们一路走好。

### 2004：Java 5.0

文/莫映

2004年9月30日，代号为“Tiger”，研发历时近三年的J2SE 5.0发布正式版本，这是

Java平台历来发布版本中改动面波及最大的一次。

纵观Tiger，“Ease of development”是其核心主题，这一点着重体现于语言特性上的

改进，这在很大程度上，简化了开发人员日常的编程任务，以往一些琐碎的手工劳动都

代之以轻松自然，而又安全可靠的自动化实现。其中的注解功能，以及随之而来的声明

式编程，还对构筑于J2SE 5.0之上的J2EE 5.0产生了巨大影响。尽管Tiger在语言特性上

做了很大的动作，但作为Java技术的基础支撑，这些改动都是深思熟虑的结果。

Tiger发布至今也有大半年了，那么Sun又是如何规划J2SE的未来蓝图的呢？据悉，J2SE

的下两个版本分别是代号为“Mustang”的J2SE 6.0和代号为“Dolphin”的J2SE 7.0，

预计Mustang将于明年发布。在吸取了Tiger研发周期过长的教训之后，Sun副总裁

Graham Hamilton表示，Mustang的发布周期将不会那么长。并且，Sun还将“Becoming

more open” 作为Mustang的主题之一。未来JCP对Java技术的影响将会愈加深入，而整

个研发过程也将会愈加透明。Mustang在正式发布前的内部版本也会陆续见诸于众，如

此，广大Java开发者便可以更加及时的了解到Java发展的最新情况。在语言层面上的扩

展依然会比较谨慎，比如像AOP这样的当下热门技术，依然不太可能会见诸其中。据

Hamilton所言，一个有可能被引入的语法特性被称作“friends”import机制，它将使由

多个包组成的大型项目变得易于管理。

## 十大人物

### James Gosling : Java之父

文/陶文

作为Java之父，James Gosling的名字可谓是耳熟能详。当人们评论一种编程语言时，总

喜欢捎带着把下蛋的母鸡一起带上。Java做为中国的编程语言学习者餐桌上有限的那么

几样餐点中的流行款式，自然是让James Gosling风光不已。虽然James Gosling现在已

经不是领导Java发展潮流的领军人物了，做为Sun的开发者产品组的CTO，怎么算来也是

身居高位了，俗事缠身吧，但是这并不妨碍其对于Java一如既往的爱护，表达着各式各

样鲜明的观点，引发一场又一场的争论。

James Gosling是很爱Java的??是啊，哪有当父母的不爱自己的孩子的呢。James

Gosling也是很爱Sun的??是啊，哪有当领导的不爱自己的公司的呢。于是我们在批

评.NET的安全性的队伍前头，在褒扬Java性能的队伍前头，在抨击SWT开倒车的队伍前

头，在给NetBeans大唱赞歌的队伍前头，我们都看到了James Gosling的身影。无论对

错、偏见或者固执，至少说明了Gosling的鲜明个性丝毫没有受到年龄的影响。也许也只

有这种天才而偏执的人物才能创造出Java这般伟大的语言来吧。

### Bill Joy : 软件业的爱迪生

文/徐昊

Joy生于1954年，1982年与Vinod Khosla, Scott McNealy和Andy Bechtolsheim一起创建

了Sun Microsystems，并从那时起担任首席科学家，直到2003年离开。他是一位令人崇

敬的软件天才，他在软件和硬件的历史上留下了无数令人仰止的传奇。

在上个世纪80年代早期，DARPA与BBN达成协议，准备将Vinton Cerf和Bob Kahn设计的

TCP/IP协议添加到Berkeley UNIX中。Bill Joy被委派来完成这项任务，然而他却拒绝将

BBN的TCP/IP协议栈添加到BSD中，因为在他的眼中BBN的TCP/IP实现还远不够好，于是

就写了一个高性能的TCP/IP协议栈。John Gage回忆道，“BBN和DARPA签署了巨额合同来

实现TCP/IP协议，然而他们的员工所编写的代码远没有一个研究生所做的好。于是他们

邀请Bill Joy参加他们的一个会议，这位研究生穿着一件T-Shirt就出现了，他们询问

他，‘你是如何做到的呢？’Bill回答说，‘这是非常简单的一件事，你读一下协议然

后就可以编码了’”。除了TCP/IP协议，基于分页的虚拟内存系统最早也是由Bill Joy

添加到Berkeley UNIX内核当中的。同时他还是vi、csh、早期Pascal编译器的作者。

关于Bill Joy惊人的软件才能流传最广的一个传奇是，据说他在上研究生的时候，想看

看自己能不能写一个操作系统出来，于是就在三天里写了一个非常简陋，但是可以使用

的Unix系统, 传说就是BSD的前身。虽然如此夸张的才情令人难以置信，但是考虑到主角

是Bill Joy，还是有一定的可信度的。Bill Joy硕士毕业之后，决定到工业界发展，于

是就到了当时只有一间办公室的Sun, 他作为主要设计者参与了SPARC微处理器的设计，

负责设计最为关键的一部分电路。这样兼精软硬件的天才实在是让人不得不佩服啊。

1995年，Sun发布了轰动世界的Java语言。当然，Bill Joy对Java也作出了不少的贡献，

首先是JINI??一种针对分布式服务的基础连接技术。任何可以内嵌JVM的电子设备都可以

通过JINI相互连接；JXTA是基于Java的P2P协议，允许互联网上的软件进行点对点交流和

协作。

这个其貌不扬的瘦高个，有着凌乱的亚麻色头发，被《财富》杂志誉为“网络时代的爱

迪生”的技术狂人，在短短的二十年间，创造了无数令人心动的软件。在MIT的BBS上曾

有一个帖子，说微软电话面试有一道题，问“Who do you think is the best coder,

and why?”虽然回复的帖子中大家都声明列举的best coder排名不分先后，然而大多数

人仍把Bill Joy列在第一位，或许可以从一个侧面验证Bill Joy在广大Programmer心目

中的地位吧。

### Joshua Bloch : Java 2 元勋

文/莫映

早在1996年，适逢Java刚刚崭露头角，年内好事连连。先是1月份发布JDK 1.0，然后是5

月底在旧金山召开首届JavaOne大会，年末又是JDK 1.1紧跟其后。正是在Java技术如火

如荼、大展拳脚的背景之下，Joshua Bloch来到了Sun，开始了他带领Java社区步入“迦

南美地”的漫长历程。

很快，他被从安全组调入核心平台组，从事底层API设计。至此以后，每逢JDK的重大版

本发布，总能在其中见到Joshua的“妙笔”。JDK 1.1中的java.math、1.4中的

assertions，还有大家所熟识的Collections Framework皆是Joshua一手打造。其中的

Collections Framework还获得了当年的Jolt大奖。到了J2SE 5.0研发阶段，身为平台组

构架师的Joshua接掌了Tiger大旗，其核心地位已然无人可以替代。作为Tiger的代言人

和领路人，没有谁比Joshua更清楚Tiger。相信大家一定还记得Joshua当年仿效英国诗人

William Blake所做的咏Tiger诗八首，优雅的笔调，透出大师深厚底蕴的同时，也道出

了Tiger的几大重要特性，这些特性是自JDK 1.1引入Inner Class以来，Java最大的语法

改进。

Java风雨十年，从JDK 1.1到J2SE 5.0，Joshua实在功不可没。难怪有人戏言，假如将

James Gosling比作Java之父，那么Joshua就是一手将Java “哺育”成人的Java之母。

Joshua对Java的贡献还不止于JDK，提起他的大作《Effective Java》（Addison

Wesley, 2001），相信Java粉丝们一定耳熟能详。该书荣膺2002年度Jolt大奖，且备受

James Gosling推崇。书中57条颇具实用价值的经验规则，来自Joshua多年来在JDK开发

工作中，尤其是Collections Framework设计中的实践心得，各个有理有据，剖析深入，

也足见其深厚功力。该书对Java社群的影响，犹如C++社群中的《Effective C++》。

Joshua对JCP的贡献也不小。他是JSR201和JSR175的领导者，前者包含了Tiger四大语言

特性，后者则为Java提供了元数据支持。此外，他还是JSR166的发起人之一（该JSR由

Doug Lea领导），并且是许多其他JSR的参与者。Joshua目前是JCP为数不多的几个执行

委员会成员之一。

Joshua Bloch给人的印象是谦逊平和，行事低调而不喜抛头露面，一个典型的技术人员

和实干家。不过即便如此，也丝毫不会减弱他对Java技术的卓越贡献和对Java社区的绝

对影响力。有人说，如果他能更彰显一些，就很有可能成为Java开发者中的领军人物，

就有如Don Box之于微软社群。

2004年7月初，就在Tiger发布在即之时，就在Jusha Bloch刚刚荣获Sun“杰出工程师

（Distinguished Engineer）”的称号之时，他突然离开Sun而去了正值发展态势迅猛的

Google。当他离开Sun的消息在TSS发布之后，众多拥趸表达了怀念与不舍之情。一年过

去了，我们还没有获知Joshua的任何近闻，似乎又是他行事低调的一贯作风所致，不知

他在Google状况如何。希望Joshua依然能继续“摩西未尽的事业”，以他的影响力推动

Java社群继续前行。据称，《Effective Java》的下一版会加入Java 5.0的部分，让我

们翘首以待吧。

### Bruce Eckel : 功勋卓著的机会主义分子

文/孟岩

Bruce Eckel原本是一位普通的汇编程序员。不知道是什么因缘际会，他转行去写计算机

技术图书，却在此大红大紫。他成功的秘诀不外乎两点：超人的表达能力和捕捉机会的

能力。他最早的一本书是1990年代初期的《C++ Inside & Out》，随后，在1995年他写

出了改变自己命运的《Thinking in C++》。如果说这本书充分表现了他作为优秀技术作

家的一面，那么随后他写作《Thinking in Java》并因此步入顶级技术作家行列，则体

现了他作为优秀的机会主义分子善于捕捉机会的另一面。写作中擅长举浅显直接的小例

子来说明问题，语言生动，娓娓道来，特别适合于缺乏实践经验的初学者。因此

《Thinking in Java》俨然成为天字第一号的Java教科书，对Java的普及与发展发挥着

不可忽略的作用。不过公允地说，Bruce Eckel的书欠深刻。比如在“Thinking in…”

系列中对设计模式的解说就有失大师水准。这一方面是因为书的定位非常清晰，另一方

面也是因为Bruce太过分心赶潮流，未能深入之故。TIJ之后，他预言Python将火，就匆

匆跑去写了半本《Thinking in Python》。后来Python并未如期而旺，于是他也就把书

稿撂在那里不过问了，机会主义的一面暴露无遗。我们也可以善意的猜测一下，他的下

一个投机对象会是什么呢？Ruby？.NET？MDA？总之，是什么我都不奇怪。

### Rickard Oberg :J2EE奇才

文/熊节

Oberg的作品很多，流行的代码生成工具XDoclet和MVC框架WebWork都出自他的手笔。这两个框架有一个共同的特点，即它们的功能虽然简单，但设计都非常优雅灵活，能够很

方便地扩展新功能甚至移植到新环境下使用。优雅的设计源自Oberg的过人才华，简单的

功能则折射出他玩世不恭的人生态度。正是这两种特质的融合，才造就了这个不世出的

奇才。

1999年，JDK 1.3发布，其中带来了一个重要的新特性：动态代理（Dynamic Proxy）。

当所有人都还在对这项新技术的用途感到迷惑时，Oberg发现用它便可以轻松攻克EJB容

器实现中的一些难关。这一发现的产物就是一本《Mastering RMI》，以及大名鼎鼎的

JBoss应用服务器。但Oberg很快又让世人见识了他的玩世不恭。由于和总经理Marc

Fleury在经营理念上不合，Oberg抱怨“法国的天空总让我感到压抑”，甩手离开了自己

一手打造的JBoss。此后的几年里，他和老友Hani Suleiman不断地对JBoss的“专业开

源”模式和Marc Fleury的商人味道冷嘲热讽，让众人为他的孩子气扼腕叹息。

2002年10月，微软推出Petstore示例应用的.NET版本，并宣称其性能比Java Petstore高

出数倍。正是Oberg深入分析这个示例应用的源代码，在第一时间指出它大量运用了SQL

Server专有的特性，性能对比根本不具参考价值。后来Oberg又先后关注了AOP和IoC容

器，两者都成为了J2EE架构的新宠。

### Doug Lea : 世界上对Java影响力最大的个人

文/KIT

如果IT的历史，是以人为主体串接起来的话，那么肯定少不了Doug Lea。这个鼻梁挂着

眼镜，留着德王威廉二世的胡子，脸上永远挂着谦逊腼腆笑容，服务于纽约州立大学

Oswego分校计算器科学系的老大爷。

说他是这个世界上对Java影响力最大的个人，一点也不为过。因为两次Java历史上的大

变革，他都间接或直接的扮演了举足轻重的脚色。一次是由JDK 1.1到JDK 1.2，JDK1.2

很重要的一项新创举就是Collections，其Collection的概念可以说承袭自Doug Lea于

1995年发布的第一个被广泛应用的collections；一次是2004年所推出的Tiger。Tiger广

纳了15项JSRs(Java Specification Requests)的语法及标准，其中一项便是JSR-166。

JSR-166是来自于Doug编写的util.concurrent包。

值得一提的是: Doug Lea也是JCP (Java小区项目)中的一员。

Doug是一个无私的人，他深知分享知识和分享苹果是不一样的，苹果会越分越少，而自

己的知识并不会因为给了别人就减少了，知识的分享更能激荡出不一样的火花。

《Effective JAVA》这本Java经典之作的作者Joshua Blosh便在书中特别感谢Doug是此

书中许多构想的共鸣板，感谢Doug大方分享丰富而又宝贵的知识。这位并发编程的大师

级人物的下一步，将会带给Java怎样的冲击，不禁令人屏息以待。

### Scott McNealy :SUN十年来的掌舵者

文/KIT

McNealy，Sun的CEO、总裁兼董事长。他曾经狂傲的说:“摧毁微软是我们每个人的任

务。”这位英勇的硅谷英雄，似乎带头起义，试图组织一个反微软阵线联盟，以对抗微

软这股庞大的托拉斯恶势力。他时常口出惊人之语，在公开场合大肆的批评微软，并曾

经说微软的.NET是.NOT。

Scott McNealy先后毕业于哈佛大学及史丹佛大学，分别持有经济学学士学位及企管硕

士。1982年MBA毕业的他和三个同学共同合伙创建了Sun，并于1984年成为Sun的执行官。

“要么吞了别人，不然就被别人吞了”是Scott McNealy的名言录之一。他擅长以信念带

动员工，鼓舞士气。极富自信的他，对于认定的事，总是坚持自己的想法，因此有人形

容他是一个刚愎自用的决策者。

身为Sun这艘船的掌舵者，Scott McNealy能够看多远，Sun就能走多远。Scott McNealy

认为将来软件界是一个只有服务，没有产品的世代。他希望打造出Sun不是一个纯靠硬件

赚钱的公司。从Open Source到Open Solaris，Sun希望可以成为提供整合性解决方案的

服务厂商。Solaris 10 + UltraSPARC是否可以像Scott McNealy希望的是下一匹世纪黑

马呢？Sun是否能以股价来证明华尔街分析师及普罗大众的诽短流长？Scott McNealy是

否能带领着Sun成为继微软之后的下一个巨人，一场场IT界的争霸战值得我们拭目以待。

### Rod Johnson : 用一本书改变了Java世界的人

文/ 刘铁锋

Rod在悉尼大学不仅获得了计算机学位，同时还获得了音乐学位。更令人吃惊的是在回到

软件开发领域之前，他还获得了音乐学的博士学位。有着相当丰富的C/C++技术背景的

Rod早在1996年就开始了对Java服务器端技术的研究。他是一个在保险、电子商务和金融

行业有着丰富经验的技术顾问，同时也是JSR-154（Servlet 2.4）和JDO 2.0的规范专

家、JCP的积极成员。

真正引起了人们的注意的，是在2002年Rod Johnson根据多年经验撰写的《Expert One-

on-One J2EE Design and Development》。其中对正统J2EE架构的臃肿、低效的质疑，

引发了人们对正统J2EE的反思。这本书也体现了Rod Johnson对技术的态度，技术的选择

应该基于实证或是自身的经验，而不是任何形式的偶像崇拜或者门户之见。正是这本书

真正地改变了Java世界。基于这本书的代码，Rod Johnson创建了轻量级的容器Spring。

Spring的出现，使得正统J2EE架构一统天下的局面被打破。基于

Struts+Hibernate+Spring的J2EE架构也逐渐得到人们的认可，甚至在大型的项目架构中

也逐渐开始应用。

Rod Johnson的新作《Expert One-on-one J2EE Development without JEB》则更让人吃

惊，单单“Without EJB”一词就会让大多数J2EE架构师大跌眼镜了。不过Rod Johnson

可能仅仅是想通过“Without EJB”一词表明应该放开门户之见。这也是Rod Johnson一

贯的作风，。也许正是这种思想，促使得Rod Johnson创建了Spring，真正改变了Java世

界。

### Alan Kay :Java的精神先锋

文/徐昊

Sun的官方Java教材中有一句话，说Java是“C++的语法与Smalltalk语义的结合”。而

Smalltalk的创造者就是Alan Kay。

Alan Kay于1970年加入Xerox公司的Palo Alto研究中心。早在70年代初期，Alan Kay等

人开发了世界上第二个面向对象语言Smalltalk，因此，Alan Kay被誉为Smalltalk之

父。2003年，Alan Key因为在面向对象程序设计上的杰出贡献，获得了有计算机界的诺

贝尔奖之称的ACM Turing Award。

Alan Kay成名于Smapltalk和OOP，而Java虽然在语言上类似于C，但是在语义上非常接近

Smalltalk，很多Java中的设计思想在Alan Kay的文献中找到根源，也有些人将Alan Kay

尊为Java思想的先驱。不过遗憾的是似乎Alan Kay老先生对Java并不买账，反倒攻击说

Java是存在致命缺陷的编程语言，Java的成功不是由于Java本身的内在价值，而是其商

业化的成功。Alan Kay欣赏的是Lisp，他认为Lisp是软件的麦克斯韦方程，其中的许多

想法是软件工程和计算机科学的一部分。看来拥有Alan Kay这样一位重量级的Java先驱

仍是我们Java一厢情愿的单恋吧。

### Kent Beck : 领导的敏捷潮

文：刘铁锋

Beck全家似乎都弥漫着技术的味道。生长在硅谷, 有着一个对无线电痴迷的祖父，以及

一个电器工程师父亲。从小就引导Kent Beck成为了业余无线电爱好者。

在俄勒冈州大学读本科期间，Kent Beck就开始研究起模式。然而在他最终拿到计算机学

位之前，他却是在计算机和音乐中交替学习。似乎Java大师都能够有这样的能耐，另一

Java大牛Rod Johnson同样也拥有音乐学的博士学位。

Kent Beck一直倡导软件开发的模式定义。早在1993年，他就和Grady Booch（UML之父）

发起了一个团队进行这个方面的研究。虽然著有了《Smalltalk Best Practice

Patterns》一书，但这可能并不是Kent Beck最大的贡献。他于1996年在

DaimlerChrysler启动的关于软件开发的项目，才真正地影响后来的软件开发。这次的杰

作就是XP（极限编程）的方法学。

和软件开发大师Martin Fowler合著的《Planning Extreme Programming》可谓是关于XP

的奠基之作。从此，一系列的作品如《Test Driven Development: By Example》，

《Extreme Programming Explained: Embrace Change》让更多的人领略到了极限编程的

精髓，也逐步导致了极限编程的流行。

Kent Beck的贡献远不仅如此。对于众多的Java程序员来说，他和Erich Gamma共同打造

的JUnit，意义更加重大。也许正式这个简单而又强大的工具，让众多的程序员更加认可

和信赖极限编程，从而引起了Java敏捷开发的狂潮吧。

## 十大产品

### Sun JDK :Java的基石

文/莫映

众所周知，流传于市的JDK不单Sun一家，比如IBM的JDK、BEA的JRocket、GNU的GCJ，以及如Kaffe这样的开源实现，不一而足。但是，根正苗红的Sun官方JDK一直以来都是备受瞩目的主流，它对Java社区的影响也是举足轻重。

1996年1月，Sun在成立了JavaSoft部门之后，推出了JDK 1.0，这是Sun JDK（Java

Development Kit）的首个正式版本;当年12月，JDK1.1出炉。该版除了对前序版本部分

特性做了改进以外，重写了AWT，采用了新的事件模型。1998年12月，JDK 1.2正式发

布。此时的类库日臻完善，API已从当初的200个类发展到了1600个类。在1.2版本中引入

了用100%纯Java代码写就的Swing，同时，Sun将Java更名为Java 2。

1999年，Java 技术形成了J2SE、J2EE和J2ME三大格局。Sun向世人公布了Java HotSpot

性能引擎技术的研究成果。HotSpot旨在进一步改善JVM性能，提高Java ByteCode的产生

品质，加快Java应用程序的执行速度。J2SE 1.3发布于2000年；2002年2月间，J2SE 1.4

问世，这是有JCP参与以来首个J2SE的发行版本。2004年9月30日，代号为“Tiger”的

J2SE 5.0终于出笼了，这次发布被誉为Java平台历来发布中特性变动最大的一次。包括

泛型在内的若干重大语法改进、元数据支持，包括多线程、JDBC在内的多项类库改进，

都令广大Java程序员激动不已。自此，Sun的官方JDK（J2SE Development Kit）已经步

入了一个新的高度。

### Eclipse :以架构赢天下

文/恶魔

IBM是在2001年以4000万美元种子基金成立Eclipse联盟，并且捐赠了不少程序代码。如

今，该组织有91个会员，包含许多全球最大的软件商。根据Evans Data公司的资料，

Eclipse是目前最受欢迎的Java开发工具。

Java厂商若要共同对抗微软，彼此之间就要有共同的开发工具才行。

在Eclipse平台上，程序员可使用好几种不同的语言。在前端方面，用户可整合多种工具

来撰写Plug-in程序或Unit Test。Eclipse最大的特色就在于其完全开放的体系结构，这

代表任何人都可下载并修改程序代码，给Eclipse写插件，让它做任何你能想到的事情，

即所谓“Design for everything but nothing in particular”。

Eclipse基金会的架构比较特别，反映出企业现今对于开放原始码计划也越来越积极主

动。Eclipse不像一般开放源码软件容许个人的捐献程序，该基金会是由厂商主导。不论

是董事会成员或者是程序赞助者几乎都来自于独立软件开发商(ISVs)的员工。

Eclipse首席执行官Mike Milinkovich说，这种厂商会员制是特意设计的；他说Eclispe

软件开发快速就是因为会员制的关系，同时又加上开放源码开发模式的临门一脚。这与

一般透过标准组织的做法全然不同。 这其实正好验证了一句老话:“开放即标准”。

### JUnit/Ant : 让Java自动化的绝代双骄

文/刘铁锋

在Java程序员必备的工具中，共同拥有且交口称赞的恐怕就非JUnit、Ant莫属了。一个

是单元测试的神兵利器，一个是编译部署的不二之选，它们让Java的开发更简单。

JUnit由XP和TDD的创始人、软件大师Kent Back以及Eclipse架构师之一、设计模式之父

Erich Gamma共同打造。名家的手笔和理念使得JUnit简单而强大，它将Java程序员代入

了测试驱动开发的时代。JUnit连任了2001、2002年“Java World编辑选择奖”以及2003

年“Java World最佳测试工具”和2003年“Java Pro最佳Java测试工具”等众多奖项，

深受Java程序员好评。

Ant是开源项目的典范，它让IDE的功能更加强大，从Sun的NetBeans到JBuilder,主流的

IDE中处处都有它的身影。“Another Neat Tool”原是它的本名，但这已经渐渐不为人

知。它彻底地让部署自动化，而程序员需要做的仅仅是几条简单的配置命令。和JUnit一

样，Ant也荣获了众多的殊荣:2003年JavaWorld“最有用的Java社区开发的技术编辑选择

奖”, 2003年Java Pro“最有价值的Java部署技术读者选择奖”，2003年“JDJ编辑选择

奖”，也让Ant受到的多方的认可。

Ant对JUnit的全面集成,则使得一切都变得更加完美。只需简单地配置，从自动测试到报

告生成，从编译到打包部署均可自动完成。强大的功能，简单的配置，让Java程序员高

枕无忧。实可谓让Java自动化的绝代双骄。

### Websphere : 活吞市场的大鲸

文/jini

1999年， IBM与Novell签订合作协议，成功地提供电子商务的解决方案给予原先使用

NetWare的用户。同年更是推出了WebSphere Application Server 3.0，并且推出

WebSphere Studio与VisualAge for Java让工程师可以快速开发相关的程序。2001年，

IBM更是宣布将应用服务器、开发工具整合在一起，与DB2、 Tivoli及Lotus结合成为一

套共通解决方案，如今、IBM更是并入了Rational Rose ( UML tools )让开发流程更是

完整化。

Sun在Web Services的策略方面远远落后于微软与IBM, 当他们手拉手在研订Web

Services规范, 加上IBM买硬件送软件或是买WebSphere送DB2的策略让企业大佬们纷纷转

向IBM的阵营, Sun才惊觉大势已去。WebSphere复杂的安装，深奥的设定，难以理解的出

错讯息不断地挑战开发者的耐心与毅力。

IBM如今已经不是将WebSphere定义为单一产品，它已经是一个平台的代名词。它里面的

产品目前包含了应用服务器、商业整合、电子商务、 数据讯息管理、网络串流、软件开

发流程、系统管理、无线语音等等。非常多样化,也让企业界愿意相信WebSphere可以带

给他们一套完整的解决方案。同时, IBM也在推广SOA的概念, 简单来说, 利用Web

Service的耦合性与工作流程的整合, 为企业内部打造以服务为导向的架构。

IBM捐献出Eclipse带给Java开发人员对IDE的重新掌握。未来是否会捐献出WebSphere的

哪一个部分成为OpenSources, 或许, 又是改写Java世界的时刻了。

### WebLogic : 技术人的最爱

文/jini

1995年, BEA成立了, 初期以Tuxedo数据转换的产品为基础, 成长之迅速是历年来最强的

企业。 1998年, BEA推出以Java为基础的网络解决方案, 提供了完整的中间层架构, 更

同时支持EJB 1.0 及微软的COM组件, 方便的管理接口掳掠了工程师的心。 在IBM和

Oracle尚未准备好迎击的时候, BEA已经席卷企业应用平台的市场。 WebLogic无论在市

场领先度与技术领导性与策略远观性都优于当年的所有应用服务器厂商。

如今WebLogic不仅仅是应用平台服务器的名称, 而是BEA对于整个企业解决方案的总称,

无论是WebLogic Portal或是WebLogic Integration配合着Workshop开发环境, 来自微软

的UI开发团队让Workshop几乎达到所见即所得。 接着, 在下一个版本之中, BEA的

BeeHive开放源代码计划将释出中间层控件的开发模块, 并且与Eclipse合作共同打造新

一代的开发环境。 如此强而有力的技术支持, 更是让顾客愿意使用WebLogic平台的最大

原因。

代号为“Diablo”的 WebLogic Server 9.0小恶魔已经出现了, 目前虽然仅仅是BETA

版, 以Portlet 方式打造的管理接口与完整且美妙的WebServices支持, 实在很难找到可

以挑剔的地方, 虽然去年被IBM的技术性推销超越了市场占有率, 不过接下来SOA的平台

竞争现在才开始, BEA的LOGO也加入“Think liquid”并且推出新的AquaLogic平台做为

数据服务平台, 可见, Java的应用服务器的战争, 还会继续进行着。

### JBuilder : Java开发工具的王者

文/刘铁锋

Java的开发工具中,最出名的莫过于Borland公司的JBuilder了。对于一些没有弄清楚开

发工具与JDK的区别的Java入门者来说，JBuilder就如同Visual C++之于C++，以为

JBuilder就是Java的全部。比起捆绑在服务器上销售的JDeveloper，JBuilder应该是唯

一的仅靠自身的实力而占领了大部分市场的Java商用开发工具了。而JBuilder作为Java

开发工具的王者，其夺冠之路并非一帆风顺。直到Java的天才Blake Stone成为JBuilder

的Architect之后，JBuilder 2.0以及3.0才逐渐推出。2000年3月14日，JBuilder 3.5的

推出别具意义，它成为了业界第一个用纯Java打造的开发工具，也风靡了整个Java开发

工具市场。在同年11月份推出的JBuilder 4.0乘胜追击，冲破了50%的市场占有率，成为

了真正Java开发工具的王者。

Borland以每半年左右推出一个新版本的速度，让众多的对手倒在了沙场。而Microsoft

因为与Sun的官司，也使得一个强大的对手退出了战争。2001年，加入了对企业协作支持

的JBuilder 5以及强化了团队开发工具的JBuilder 6打败了最后一个对手Visual Age

For Java。2002年JBuilder 7推出之后，再也没有其他厂商与JBuilder竞争。

孤独的王者并没有停下脚步，在2003年到2005年间，JBuilder也仍然延续了其半年一个

版本的速度，推出了8、9、10、2005四个版本。强大的功能以及持续的改进，也让Java

程序员多了一分对能够在开发工具市场上与Microsoft血拼十数年的Borland的敬仰。

### Oracle : Java人永远的情结

文/熊节

在林林总总的数据库之中，有一种尤其令人又爱又恨、印象深刻，那就是关系型数据库

市场的“大佬”??Oracle。

从公司的角度，Oracle和Sun有着诸多相似之处，例如：两家公司都拥有一位个性鲜明的

CEO。早在Java诞生之初的1995年，Oracle就紧随NetScape从而第二个获得了Java许可

证。从那以后，Oracle对Java的鼎力支持是Java能够在企业应用领域大获成功的重要原

因之一。

所有J2EE程序员都知道，Oracle的JDBC驱动虽然与Oracle数据库配合良好，但在不少地

方使用了专有特性。其中最为著名的就是“CLOB/BLOB问题”，诸如此类的问题给开发者

带来了很多麻烦。为了同时兼顾不同的数据库，他们不得不经常把自己的一个DAO（数据

访问对象）写成两份版本：针对Oracle的版本和针对其他数据库的版本。有不少人为了

开发便利，舍弃了数据库之间的可移植性，将自己的产品绑定在Oracle的专有特性上。

Oracle提供的Java开发工具也与此大同小异。不管是数据库内置的Java支持还是

JDeveloper IDE， Oracle的Java工具都和Oracle数据库有着千丝万缕的联系。看起来，

只要Oracle还是数据库市场上的“头牌”，了解、学习Oracle的专有特性，周旋于

Oracle特有的问题和解决方案之中，就将仍旧是J2EE程序员在数据库基础和SQL之外的必

修功课。对Oracle的爱与恨，也将仍旧是Java人心头一个难解的情结。

### Struts、Hibernate : 让官方框架相形失色的产品

文/刘铁锋

好的框架能够让项目的开发和维护更加便捷和顺利。相比Sun官方标准的迟钝以及固执，

开源框架也更得到Java程序员的共鸣。Struts以及Hibernate就是这样一类产品，它们简

单、优雅，更让官方的产品相形失色。

谈起Struts，不可避免地就要提及MVC（Model-View-Controller）的理念。而准确地

讲，MVC的提出却最早源于JSP的标准。在1998年10月7号，Sun发布的JSP的0.92的规范中提出的Model 2就是MVC的原型。在1999年12月Java World的大会中，Gavind Seshadri的文章最早阐述了Model 2就是一种MVC的架构，同时也提及了MVC架构是一种最好的开发方法。2000年3月，由Craig McClanahan发布的Struts成为了最早支持MVC的框架。Struts

在设计上虽然存在一些诟病，但是不可否认的是，它使得Java Web应用的开发更加简洁

和清晰，也让更多的程序员爱上了Java，并开始遗忘官方的JSP。时至今日，比起如

WebWork、Tapestry以及Sun官方的JSF，Struts或多或少存在些不足，但是众多成功项目

的实施，仍然使其牢牢占据的Java Web应用框架的首位。

Hibernate则在某种程度上改变了人们对构建J2EE的思路。相比其EJB的Entity Bean的映

射技术，Hibernate则显得更加简洁和强大。五分钟就能把Hibernate跑起来，让更多的

Java程序员享受到了开发的乐趣。第15届Jolt大奖中，最优秀数据库、框架以及组件的

奖项中，Hibernate当仁不让获得头筹；不仅如此， Hibernate甚至还影响了官方的标

准。在众多Java程序员翘首以待的EJB 3.0的规范中，Hibernate得到了支持。

Java开源的繁荣不仅让众多Java的开发者享受到了更多的便利，甚至影响了官方的标

准。恐怕这也是作为Java人独有的乐趣之一吧。

### PetStore : J2EE人的必修课

文/陶文

很少有一个例子项目如PetStore这 般广为人知，而这很大程度上要归功于Sun很“英

明”地把PetStore做成一个只展示架构而在性能调优上留下了大大余地的例子。围绕着

性能话题，产生了颇为有趣的厂商之间以及平台之间的Pet Wars。除去这些关于性能的

流言蜚语乃至中伤，PetStore在展示J2EE1.3平台的架构、演示什么叫分层方面还是有着

很大的功劳的。而且PetStore在架构方面的丰富性使得其成为J2EE的那些轻量级小兄弟

们展示自身的一个必选科目。

不谈那些围绕PetStore的口水，那些数不尽的盗版，PetStore给开发新手带来的最重大

的影响，我想应该是架构的观念而不是性能，也不是业务。做为一种技术的Demo，这无

可非议。但是如果你是一个新手，跟着PetStore亦步亦趋地学习J2EE开发，难免会陷入

过度设计、华而不实之类的困境。围绕着.NET的PetStore的克隆PetShop展开的架构与性

能的大讨论，是不是也在促使我们学习新技术时应该以解决问题为导向呢？特别是当你

想把一个如PetStore这般的Sample Project的技术照搬到你的现实世界的Real Project

来时。

## 十大组织

### Sun : 因为Java而永被荣光

文/孟岩

Sun是1980年代初期由斯坦福大学三位年轻学生创立的公司。与一般人的印象不同，

“SUN”的本意并不是企图剽窃天上那颗温暖的恒星的威名，而是“斯坦福大学网络”的

意思。Sun在“前Java”时代就因为SPARC芯片、Solaris操作系统和“网络就是计算机”

的口号而为人所知。1990年12月，Sun启动了一个看上去没什么意思的嵌入式软件项目。

然而，基于C++的开发很快遇到了麻烦。一个创新型技术公司的特色立刻显示出来，一群

天才不是去深入C++，而是另辟蹊径，发明了Java。这个传奇故事已经尽人皆知，但是其

中所包含的精神却始终令人望空凝思。

Java的发明，使得Sun真正有机会在软件的历史天空中放射出太阳的光芒。Sun发明了

Java，并且在长达十年的时间里始终走在Java大潮的最前端。Sun是Java的老家，是Java

慈爱的母亲，这一切任何人都改变不了。虽然Sun似乎没能够从Java中获得应有的金钱回

报，但这丝毫没有挫伤Sun对于Java的母爱，还有对于Java大潮的舍我其谁的领导气概。

所有人都迷恋富有的感觉，但是也迟早会意识到钱不是世上最宝贵的东西。这个世界并

不缺少会赚钱的公司，但是能够靠着创新型技术推动整个世界进步的公司却是凤毛麟

角。Sun应该感到骄傲，他们将因为Java而在历史的天空里发射出太阳的光芒。

### IBM : Java经济的最大受益人

文/恶魔

Sun公司是Java的发明人，但IBM却是Java最大的受益者。是IBM抢占了利润丰厚的应用服务器市场的头把交椅，是IBM在Java技术上投入最多的金钱，拥有最大的影响力和最好的开发者社区。可以毫不夸张地说，Java使IBM的软件体系得到复兴，在某种意义上，甚至可以说，是Java创造了这种复兴。Java之后又来了Linux，这种建造在不属于自己的平台

上以获得成功的理念更是变得非常有影响力。正是这种理念铸就了今天IBM “按需计

算，服务为王”的王者风范。

2004年三月，IBM以Java的解放者的姿态借机向Sun发难。IBM公司负责新兴技术的副总裁史密斯在一封公开信中表示，IBM愿意与Sun合作成立一个项目，意在通过开放源代码开

发模式管理Java的开发工作。

墙内开花，墙外香。面对IBM的成功，到底是谁妒嫉呢？或许去程序的社区中逛逛聊聊，

明眼人是不难发现事实真相的。也许Sun应该好好向IBM学习经营之道。尽管利润额不如

硬件及服务部门，但IBM软件部门的利润率是最高的??高达85%的利润率足以令人惊叹。

在最近的一个季度里，IBM软件部的利润率上升了8%，其中WebSphere产品组的利润率上

升了14%。

正是IBM在开源和Java上的全身心地投入又秉承开放性的原则，今日的Java才能以日进千

里的速度将许多竞争对手远远抛在后面。Java 10年，IBM功不可没。

### BEA : 用AppServer影响Java阵营

文/霍泰稳

十年前诞生的Java并不是一开始 就那么引人注目的，虽然用Applet也曾为互联网络带来

一抹亮色，但毕竟只是Toy。在企业级应用市场上，Java一直没有什么起色，虽然Java的

支持者一直在鼓吹它有着大型企业级应用的强悍功能。过高的期望与低能的产品，一时

间颇让人怀疑Java的路是否已经走到了尽头？可以说是WebLogic Server的出现逐渐打消

了人们的顾虑，BEA公司慧眼独具在2001年收购的这个产品将人们的目光吸引到电信、金

融、政府等Java企业级应用方面，WebLogic Server以其优良的性能让人们看到Java应用

广阔的未来。虽然随后在Java应用服务器方面出现了像IBM公司的WebSpere、开源软件

JBoss等Java应用服务器，但WebLogic Server几乎占领世界前500强所有企业的应用服务

器市场地位依然无法撼动。

Java现在已经不单纯是一个语言，从另一方面它也代表着开放与创新。很多以Java产品

为基础的公司或者从事Java开发的程序员骨子里都有着开放与创新的烙印，BEA公司的发

展深深地印证了这一点。与合作伙伴的密切合作向Java社区贡献产品基础源代码、加入

权威开源组织参与Java标准的制定等证实着BEA的开放，而其产品从WebLogic Server一

种拓展到WebLogic Platform、WebLogic Portal、WebLogic Workshop等其它领域又证实

着它的创新能力。

### Oracle : 早起的鸟儿有虫吃

文/孟岩

Oracle的老板拉里?艾利森是有名的混世魔王和花花公子，所以尽管他也是软件产业成功

人士的代表，却绝不是程序员们心目中的英雄，程序员们毕竟不是央视《对话》节目里

群众演员，没必要为了节目需要而对权贵财阀们做出一副贱骨头状。但是，任何人都不

能不钦佩Oracle在技术上的前瞻性和坚决性。Oracle是1996年获得Java许可证的，紧接

着就大胆地将Java作为战略性的发展方向而予以全面支持。要知道当时Java的前景并不

是十分确定的，而Oracle的坚决投入，使得它在后来的Java世界中抢得一席之地。1998

年9月发布的Oracle 8i为数据库用户提供了全方位的Java支持。Oracle 8i成为第一个完

全整合了本地Java运行时环境的数据库，开发者用Java就可以编写Oracle的存储过程，

这意味着可以仅在Oracle数据库中就完成几乎全部的应用开发。J2EE兴起后，Oracle更

是有心进入开发工具市场，因而购买了JBuilder的源码，并在此基础上开发出

JDeveloper。如今Oracle除了数据库稳居第一之外，在Java开发工具世界里也自成一

派。这一切不能不归功于当初的眼光远大。

### Apache : 开源软件的品牌保证

文/陶文

Java程序员的日常工具箱中，我们可以发现Ant、Tomcat、Log4、Lucene这些鼎鼎大名的

开源产品。而它们的共同点在于，都是由Apache Software Foundation社群中杰出的开

发者开发的开源项目。Apache这个名字在Java的世界中实在太出名了，以至于

“Apache”这六个字母成为开源项目品质保证的代名词。Apache是自由开源的一面旗

帜，其Apache License更是成为商业友好的License的首选，只SourceForge上就有1000

多个以Apache License授权的项目，其流行程度可见一斑。

但是，如我们所知，Apache最早闻名IT界是靠高性能的Web服务器，其历史甚至和Java一样长。Apache对于Java的偏爱，以及其发展的速度也映射出了Java繁荣的一角。现在去

它的主页上看看，满目望去全部都是Java的开源项目，早就不光是其C服务器的老本行

了。Apache对Java最大的贡献就是提供了这么一个精品的开放舞台，让杰出的开发者和

成熟的开源项目走到一起，共同给Java语言提供一个丰富的工具仓库。对于一种语言、

一个平台来说，其库的丰富程度对于开发者来说的重要性再怎么强调也不为过。勿庸置

疑，Aapache上会出现越来越多的Java开源项目，而我们开发者也将更多地得益于这令人

目不暇接的繁荣。

### TheServerSide : 论坛的专业精神

文/刘天北

成立于2000年5月，TSS最初以一本书而广为人知。它的创始人Ed Roman同时也是J2EE名著《Mastering EJB》的作者;Roman运营着一个J2EE咨询/培训公司

TheMiddlewareCompany（简称TMC），TSS当时是TMC的下属部门；为了扩大企业的影响，

Roman在TSS网站上免费发布了那本书的电子版。J2EE程序员要吃下这个香饵，就得在论

坛中注册;注册的同时，多半也会看一眼论坛的内容；一看之下，大部分人都被吸引住，

成了社区的忠实成员。

TSS究竟有什么吸引人的秘诀？首先，它有一支能力过人的运营团队，除了Roman本人之

外，其中还有好几人都是J2EE领域的顶尖专家；第二，TSS和TMC定期会推出专家研讨会/

视频访谈、技术白皮书、评测报告，通读TSS提供的这些内容，基本上就可以把握技术的

当前趋势。但这还不是全部。最可贵的还是TSS的社区风格：他们深谙技术，但不盛气凌

人；思想敏锐，但不因此缺乏审慎和大局感。其中大多数人都已在自己的开发领域颇有

建树，在TSS上的活动既给他们提供了与同行进行深度交流的机会。一个新成员进入社

区，就像参加了一个起点很高的专业俱乐部，这不是一个求解“怎样设置JAVA\_HOME环境

变量”之类问题的地方。事实上，在J2EE技术发展的若干转折点上，TSS都起到了关键的

推动作用。

几经易主之后，J2EE咨询培训公司TMC在2004年关闭；TSS则被IT媒体集团TechTarget收

购。我们期待着它更加繁荣的未来。

### JBoss : 职业开源软件组织

文/刘天北

J2EE的婴儿期，“应用服务器”原本是“昂贵”的代名词。但从1999年起，Marc Fleury

和Rickard Oberg等人就已经着手改变这种状况。他们开发的开源EJB容器当时叫做

“EJBoss”，在Sun公司的干预下（注意，“EJB”是注册商标），JBoss获得了今天的名

字。虽然从问世起就一直受到关注，但JBoss第一个达到产品化标准的版本可能是它的

2.2版。它的易用让人一见难忘：除了标准部署描述符，无需编写专用的xml配置文件。

Oberg自豪地说，“我们的架构并不是按照EJB规范指定的路线设计的，因此也没有走大

多数应用服务器走过的弯路。”

Jboss 3.x版本保持了一贯的创新精神，在用户中间获得了更广泛的认可。但是，文档要

收费下载、在邮件列表上提问常常会遭到Fleury等人的斥责。无疑，JBoss的创始者也意

识到了自己的幼稚：开源软件只能靠服务盈利，卖文档赚钱有限、骂用户当然更损害企

业形象。

虽然以Oberg为首的许多程序员退出了开发队伍（其中很多人成了JBoss的死敌），在开

源软件领域也面临JOnAS Geronimo等新老对手的竞争，但JBoss还是以不断推出的新版本

站稳了脚跟。在技术上，它是策动J2EE演进的重要力量：拟议中的EJB 3也要追随Jboss

4倡导的开发范式，以至于二者的代码样本之间的差别几乎难以分辨；在商业上，JBoss

与Sun公司言和修好，甚至还获得了数量可观的风险投资。JBoss已经像拥护者预期的那

样，成为了应用服务器领域的Linux。

### Borland : 深度介入Java

文/左轻候

除了Sun以外，也许没有一家公司 像Borland这样深层地介入Java。Borland开发了最早

的Java编译器之一，Borland的工程师参与了早期JDK的设计，Borland的JBCL

(JavaBeans Component Library) 技术也成为后来Java Bean规范的基础。但是Borland

对Java世界最大的影响还是JBuilder。

1997年11月，Borland JBuilder 1.0发布。虽然第一个版本相对于竞争对手并没有表现

出明显的优势，但是Borland凭借深厚的技术实力和正确的市场策略，不断地超越了对

手。JBuilder 3.5成为业界第一个100%基于Java架构的开发工具，并且市场份额很快超

过了50%。在随后的版本中，JBuilder持续改进对团队开发、J2EE架构、Mobile技术等方

面的支持，最终成为了Java开发工具市场，特别是大型企业级Java开发市场中的霸主。

JBuilder的成功，很大一个原因来自于Borland坚持的平台中立性，即对不同厂商的解决

方案提供一视同仁的支持。

2005年初，随着Eclipse社区的迅速崛起，Borland进入了Eclipse的董事会，成为战略开

发者(Strategy Developer) ，并宣布将推动Borland的其它产品与Eclipse的集成。在随

后发布的一份文件中，Borland宣称JBuilder的未来版本将放弃原有的PrimeTime架构，

而基于Eclipse架构。这个代号为“Peloton”的版本预计于2006年下半年发布。

Borland对Java的另外两个主要贡献来自Together和BES(Borland Enterprise Server)。

Together是著名的建模工具，能够与包括JBuilder在内的许多开发工具进行集成，全球

市场份额占有率排名第二。BES AppServer是一种J2EE服务器，在全球市场份额占有率上

次于WebLogic和WebSphere，排名第三。

### JCP : Java世界的联合国

文/黄海波

当联合国正在为安理会改革问题 吵得如火如荼时，Java世界的“联合国安理会”已经成

功地运作了七个年头。JCP（Java Community Process）在1998年由Sun发起成立，目标

是通过一个开放、合作和鼓励参与的非盈利组织来发展和推进Java和相关的技术。正是

由于JCP计划的推出可以让所有对Java感兴趣的软硬件厂商，个人和组织都能参与到技术

规范的制定和发展过程中，协调各方的兴趣和利益、集思广益，才可以让Java在短短的

几年内异军突起，成为可以和微软开发平台抗衡的一个主流开发语言。JCP计划既然是一

个组织，自然也有一定的架构。JCP组织架构主要包括PMO(Program Management

Office)、JCP成员、EC、EG。事实上，JCP的架构就好像一个Java世界的联合国。虽然也

有不少人批评JCP成为各派利益的角力场，因而效率低下；但是，它毕竟为Java的顺利发

展很好地掌握了方向。

### 微软与Java : 不得不说的故事

文/孟岩

微软跟Java不对付，地球人都知 道。跟Sun和解了又怎么样？ .NET跟Java就是竞争对

手，没什么说的。但是有点IT掌故的人都知道，微软并非一开始就跟Java过不去。当年

比尔?盖茨盛赞Java是“长期以来最好的程序设计语言”，而且很早就购买了Java许可

证。但是微软作为村里的老大，看着人家的儿子茁壮呈长，不由得生了私心杂念，搞起

了小动作，在Visual J++中加入了一些破坏纯洁性的东西。单独来看，Visual J++是COM

时代微软最棒的开发工具，用WFC写Windows应用程序和COM组件实在是一种享受。但是放在Java大家庭里，这个家伙就显得多少有点不怀好意。一场官司下来，微软被逐出Java

大家庭，Visual J++无疾而终。以后的事情尽人皆知，.NET出笼，利齿直指Java，几年

撕咬下来，没占着便宜也没吃大亏，如今也算是南北朝对峙，二分天下有其一。设想如

果当时微软能够摒弃帝国主义心态，正确对待Java，与其他人一起共建美好的Java“共

产主义社会”，那么今天我们的软件开发世界应该会美好得多。可惜黄粱一梦，终究是

蚂蚁的喜事。2004年，微软与Sun实现了和解，但愿到Java 20周年的时候，我们能更正

面地描述微软对Java发挥的作用。