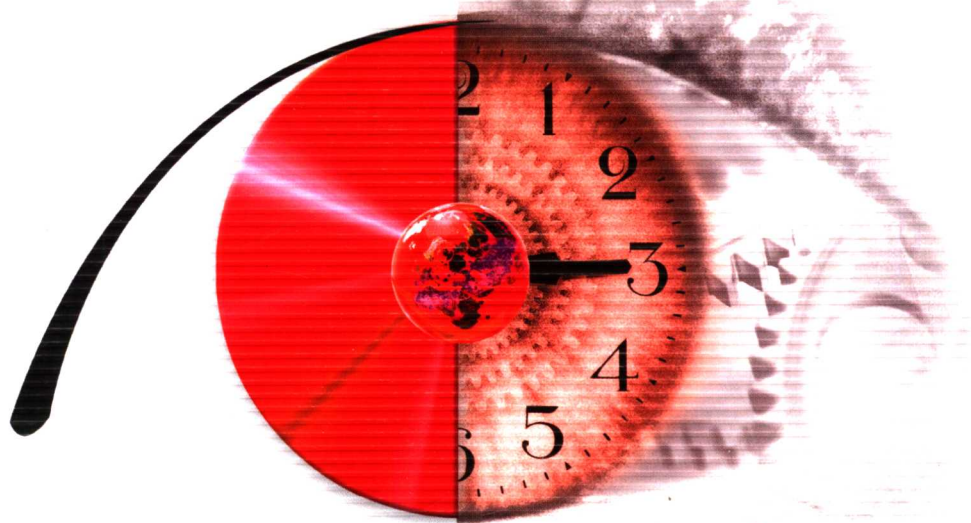


项目开发实践系列丛书

内附范例光盘

Java项目开发实践 ——网络篇

雷之宇 主编 孙颖 雷鸣宇 赵宇阳 编著



◆ 以实用为指导思想，用大家最关心和最常用的功能组织实例：高性能的数据库访问；网络短信、网络文件柜；Web Service 编程；基于MVC的Struts编程。

◆ 所有实例均由组件技术（JavaBean）构架而成，复杂功能轻松实现。

◆ 所使用的组件经过严格测试并在许多大型业务系统中得到了应用，组件代码全部公开，随书奉送。

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

项目开发实践系列

Java 项目开发实践 ——网络篇

雷之宇 主编

孙 颖 雷鸣宇 赵宇阳 编著

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

Java 语言及其体系结构使“编写一次，随处运行”的程序成为现实，也使各种终端设备以同样的方式访问系统成为可能。

本书用一些大家感兴趣的内容组织实例，从实例所涉及的知识点开始讲解，逐步切换到代码原型的介绍，最终呈现给大家一个完整实例。

本书的另一个特点是所有的实例均由组件构架而成。对初学者来说，只要通过简单的组件调用，就可以实现那些复杂的、令人兴奋的功能。而对于那些有经验的程序员，则可以对组件进行改造，使之适用于更多的应用系统。

本书的内容涉及：“高性能数据库访问”、“文件系统访问”、“工作流引擎”、“Web Service 编程”以及全新的“Struts 编程”。

图书在版编目 (CIP) 数据

Java 项目开发实践. 网络篇/雷之宇主编; 孙颖, 雷鸣宇, 赵宇阳编著. —北京: 中国铁道出版社, 2005. 5

(项目开发实践系列)

ISBN 7-113-06525-2

I. J… II. ①雷… ②孙… ③雷… ④赵… III. JAVA 语言-程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2005) 第 047155 号

书 名: Java 项目开发实践——网络篇

作 者: 雷之宇主编 孙 颖 雷鸣宇 赵宇阳编著

出版发行: 中国铁道出版社 (100054, 北京市宣武区右安门西街 8 号)

策划编辑: 严晓舟 魏 春

责任编辑: 苏 茜 林菁菁 刘 颖

封面制作: 白 雪

印 刷: 河北省遵化市胶印厂印刷

开 本: 787×1092 1/16 印张: 21.75 字数: 521 千

版 本: 2005 年 5 月第 1 版 2005 年 5 月第 1 次印刷

印 数: 1~5000 册

书 号: ISBN 7-113-06525-2/TP·1496

定 价: 37.00 元

版权所有 侵权必究

凡购买铁道版的图书, 如有缺页、倒页、脱页者, 请与本社计算机图书批销部调换。

前 言

Java 从 1995 年正式发布，几乎是满载着赞誉一路走来，受到了数以千万计的程序员的推崇，逐渐发展成为最广泛使用的编程语言之一。

Java 所带来的革命，每一个 IT 从业人员都有自己亲身的体会。很多软件开发工程师都投入很多时间和精力来学习 Java 语言及其构架，对于他们而言，掌握 Java 可能意味着更新的技术、更多的就业机会和更高的薪酬。

Java 究竟有着什么样的魔力，能拥有这么多的追随者？我们在招聘 Java 高级开发人员时，也常常提出这样的疑问。我们要求应聘者用一句话描述出 Java 的特点和魅力所在。

“Write once, run anywhere. (编写一次，随处运行)”就是最好的答案。

的确如此，Java 拥有良好的可移植性。用 Java 编写的程序可以轻松地在 Windows、Unix、Linux 甚至 PDA 和移动电话上运行。因此，软件开发商对 Java 也很青睐，对他们而言，基于 Java 构架的产品，意味着可以适应各种用户的系统环境，并为更多的用户提供服务。

本书与其他同类图书不同，并没有对 Java 进行全面的介绍，而从实用的角度入手，每个章节都是围绕着 Java 的一个特点（如网络通信、工作流引擎、Web Service、Struts 等），以实例的方式进行介绍。

本书是一本实例教程。和其他实例教程不同的是，本书中所有的实例都用组件技术 (JavaBean) 构架而成。对于 Java 的初学者，只要按照书中所描述的步骤，对这些组件进行调用，就可以快速实现那些复杂的、令人兴奋的功能；而对于那些有过 Java 编程经验的读者来说，依据书中对那些组件的原理介绍和代码解析，可以对这些组件进行改造，使之适用于更多的应用系统。

本书中所使用的组件均选自“杰软科技 (www.jrsoft.com.cn)”的 Smart 9000 平台。这些组件已经经过严格的测试，并已经在许多大型业务系统中得到了应用。在这里，对“北京杰软科技公司”给予的大力支持表示感谢。这些组件的最新版本及 Smart 9000 平台的详细介绍文档，可以从 www.jrsoft.com.cn 下载。书附光盘内含书中各章实例代码、开发 Java 应用所需的各种工具和环境、Smart 9000 组件库及一个关于“用户管理模块”实例的代码。

在本书出版之际，我想将它献给我可爱的女儿“小米”。在本书初稿提交的时候，她才刚出生几天，我甚至还没有给她取好名字。当我对本书的终稿作最后的修订、确认时，她已经学会用笑容表达自己的情绪了。她的出生，为我们简单的生活添加了生命的活力和快乐的元素。特别要感谢的是我的妻子孙皓，她克服了怀孕期间身体的种种不适，尽量不打扰我，使我终于在我们的宝贝出生前完成了本书的写作。

本书的其他几位作者，他们的敬业精神和责任心很让我钦佩（每次收到他们的邮件都是在凌晨发出的，这很让我感动），他们是：孙颖、雷鸣宇、赵宇阳、王宏斌、韩勇、马泽峰、饶志磊、邱林、张天乐、关新新、王子尤、周烨、商伟娜、黄健、孙皓，没有他们的全力配合，本书不可能完成。

最后，虽然编者在本书的编著过程中力求完美，但由于能力所限，疏漏和错误在所难免，请广大读者不吝赐教。我的邮件地址是：gark@263.net

编者

2005 年 5 月

勾勒 Java 学习地图

项目是软件开发的基本形式

若干年前，第一次用 Borland C++ V3.1 对其中 Project 菜单项的功能颇为迷惑，还特地用词典查到 Project 是“项目，工程”的意思。后来发现这是一种非常好的代码管理和开发的好方法，当你把上万行的代码分在十几个 cpp、hpp 文件中，并通过 Project 来管理、编译、运行……当通过一次次焦头烂额地调试程序，你的程序完全能够成功运行后，你的 C/C++ 程序设计能力完全能提高到另外一个层次，因为你做过一个 Project。

后来又用到 Delphi，Visual Studio 等开发工具，不过我发现以 Project 为中心的代码管理和开发形式一直没有变化，且被每一个程序员所接受。甚至我们在用电脑的时候都有这样的习惯，把某项工作的所有相关文件都放在一个文件夹里面，用一个文本文件来记录相关文件的作用，这个文件常常被命名为项目记录。

书籍的目的不就是记录前人的经验，让后人少走弯路吗？既然亲身体验一个项目开发的全过程，能把一个大学生变成一个程序员，为什么不把这些项目开发过程的记录下来，供后来者参考，使他们蜕变得更快，走得更稳，经验更丰富呢？这就是项目开发实践系列策划的初衷。

实践是最好的教学方法

程序员的成长各不相同，相同之处可能是都在一些地方摔过跟头：不能编译某个程序，不能理解某个宏的用法……“好了伤疤忘了痛”是我们中国民间的一句俗语，用在计算机技术的学习上，也颇有相同之处。当我们学会了 C 语言，变成老手的时候，很快就忘了学习它的时候所爬过的几个坎，往往忘记了很多过去的苦难：C 其实很简单嘛，VB 简直不用学习……之所以一览众山小，是因为你爬到了山顶。当你还在 C 语言的山脚下时候，抬头望去，到处都是困难。

我们没有跟踪摄影机，无法完整地记录一个程序员的成长历程。幸好，我们还有网络，可以交流学习的体会，成长的痛苦和快乐。还有图书，先行者可以发表自己成长的经历和成功的案例，新手可以从中得到启发，少走弯路。

勾勒 Java 学习地图

当“项目开发实践”系列 Java 分支的图书出版完备过后，我们社的 Java 学习图书可以说为读者提供了一个基本的学习框架。总的来说分为两类：

一个是 Java 的基础学习与教学类。这个领域，根据不同的学时要求，我们出版了两本《Java 程序设计》，1 本《Java 程序设计学习辅导》供您学习参考和考试辅导。1 本《Java 2 程序设计技能百练》是提供给众多强调实训课程的学校的教材。当然，这些图书也都适合您自学参考。很多读者可能和我一样早期学习过一点 C/C++，毕竟它们相同之处太多，所以我们有《Java 面向对象程序设计》提供给非零起点的 Java 学习者。

另一大类是项目实践类：《Java 项目开发实践》已经出版到第 2 版，这次我们修订了很多内容，期望能奠定所有 Java 实践类图书的基础。而其他图书，也都在它的基础上衍生出来。

《Java 项目开发实践——网络篇》便是我们的一个尝试，希望读者能够在 Java 网络应用领域获得更多的经验。本书我们请有多年开发经验的雷之宇先生主编，在书中，作者把他们公司使用的成熟的代码抽取出来，让读者体验一下真实运行的商业代码是什么模样。

《J2ME 项目开发实践》（第 2 版）也是我们过去一个成功产品的改版，删除了大量基本语法讲述的内容，重点放在了开发过程的描述。作者差不多是中国第一代 Java 嵌入式应用的开发者，某著名 IT 企业的领军人物，第 2 版中添加了他这几年更多的成功经验。

为了填补第一类和第二类 Java 图书之间的空隙，我们还策划有一本《Java 课程设计》。大家知道，课程设计或毕业设计起到串起所学知识，提升学生综合能力、动手能力、将所学知识应用于实践能力的作用。很多专家教授都推崇课程设计、毕业设计的重要作用，他们认为，通过一个大作业，就能带动学生对系统（程序）设计的技巧、系统分析和数据结构、数据库等许多知识的综合应用，可以大大提高学生的思维能力，为毕业设计和以后的工作打下一个良好的基础。为此，我们项目开发实践大系列中，特别的开辟出了“课程设计·毕业设计”系列，满足这方面的要求。

你的意见，我们乐见

“收罢低声问，深入入时无”。第一版本的“项目开发实践”系列出版后，我们收到很多积极的反馈，谢谢读者，也希望你们继续对我们的图书提出中肯的意见。

一个系列图书的完成有时候就像一个软件项目的开发过程，也是集体劳动的结果。我们首先确定本书的读者目标群后，讨论了写作内容和写作风格。作者每写好一章后就传给我们的秦绪好和林菁菁两位技术编辑，他们基本作为本书的第一个读者，以交互写作的方式向作者提出整改的意见；然后是我们的责任编辑们，他们逐字逐句地阅读了每一本书，期望避免任何一个错别字和错误的格式；在编辑们的身后，还有设计人员，后勤管理人员和销售人员，这是一个庞大的团队。所有的工作都期望达到一个目的，使读者学习时能少跨越陡坡，能节约时间，使读者在 IT 技术上的学习投入能得到更多的回报。

但不能说本书白璧无瑕，我感觉在如何把软件工程中的理念和技术具体应用在项目开发实践中，作者们还有很多经验，可惜在本系列中表现得还不够。

我们亦发现软件开发工具有整合设计与开发的趋势，甚至要整合项目的需求（功能）定义，模式设计，软件开发，代码测试和软件发布等所有功能。原本想让项目开发实践系列的图书都包含设计工具的内容和设计方法，不过本书已经包含太多内容，如果读者迫切需求，也是我们下一版本的发展方向。

感谢各位作者，把自己的开发项目精简后提供给我们。交流产生思想，期待大家通过 beone2000@163.com 对我们的后续工作提出中肯的建议。谢谢！

魏春
2005 年 5 月

目 录

第 1 章 Java 概述.....	1
1-1 Java 基础.....	1
1-1-1 Java 的发展过程.....	1
1-1-2 JSP、Java、JavaBean 与 JavaScript.....	7
1-1-3 JSP 与 CGI、ASP、PHP.....	7
1-2 Java 的困惑.....	9
1-2-1 Java 程序是否存在先天的运行效率问题.....	9
1-2-2 Java 怎样和其他系统沟通.....	11
1-2-3 Java 高性能数据库访问, 程序员心中永远的痛.....	17
1-2-4 Java 不是 JavaScript.....	19
1-2-5 Java 也流行——XML 与 Web Service.....	20
1-3 几种常见的 Java Application Server 介绍.....	21
1-3-1 JSWDK.....	21
1-3-2 Tomcat.....	24
1-3-3 Macromedia JRun Server.....	24
1-3-4 BEA WebLogic.....	25
第 2 章 实例一: 数据连接池的实现.....	27
2-1 Java 访问数据库.....	27
2-1-1 JDBC 技术.....	28
2-1-2 各种驱动程序介绍.....	28
2-1-3 java.sql 包中的方法和类.....	30
2-2 为什么使用连接池.....	33
2-3 连接池的原理介绍.....	34
2-4 设计思路.....	35
2-5 实例代码解析.....	36
2-5-1 连接池组件的实现.....	37
2-5-2 连接池组件的调用.....	44
2-6 本章小结.....	46
第 3 章 实例二: 网络文件柜.....	47
3-1 网络文件柜功能介绍.....	47
3-2 File 类.....	47
3-2-1 File 对象介绍.....	48
3-2-2 文件操作的实现.....	48
3-2-3 目录操作的实现.....	50

3-3	设计思路	54
3-4	实例代码解析	55
3-4-1	FileManage.jsp	55
3-4-2	FileDetail.jsp	56
3-4-3	newFolder.jsp	60
3-4-4	updateFolder.jsp	64
3-4-5	updateFile.jsp	68
3-4-6	deleteFile.jsp	72
3-5	本章小结	77
第 4 章	实例三：网络信使服务	78
4-1	信使服务的功能介绍	78
4-2	TCP/IP 基础	78
4-2-1	什么是 TCP/IP	79
4-2-2	TCP/IP 协议介绍	80
4-3	Socket 编程基础	82
4-3-1	TCP 与 UDP 协议的比较与介绍	83
4-3-2	ServerSocket 和 Socket 对象介绍	85
4-3-3	Network Stream 对象介绍	87
4-3-4	Thread 对象和多线程编程	89
4-3-5	最简单的 Socket 通信程序	90
4-4	设计思路	93
4-5	实例代码解析	97
4-6	本章小结	104
第 5 章	实例四：WebMail 的实现	105
5-1	电子邮箱的功能介绍	105
5-2	JavaMail 基础	105
5-2-1	Session 类	105
5-2-2	Message 类	106
5-2-3	Transport、Store 类	107
5-3	收邮件操作的实现	107
5-3-1	连接 POP3 服务器	107
5-3-2	得到邮件对象数组	108
5-3-3	查看指定邮件的信息	108
5-3-4	删除指定邮件	109
5-4	发邮件操作的实现	109
5-4-1	连接 SMTP 服务器	109
5-4-2	设置邮件的相关信息	110
5-4-3	发送邮件	110

5-5	设计思路	110
5-5-1	电子邮局的用户验证	110
5-5-2	接收、维护邮件列表	111
5-5-3	邮件的发送	111
5-6	实例代码解析	112
5-6-1	ReceiveMail.java	112
5-6-2	SendMail.java	123
5-6-3	login.jsp	127
5-6-4	ReceiveMail.jsp	128
5-6-5	seeContent.jsp	132
5-6-6	DeleteEmail.jsp	134
5-6-7	SendMail.jsp	136
5-7	本章小结	140
第 6 章	实例五: Java 与 Web Service	141
6-1	Web Service 基础	141
6-1-1	Web Service 特性	143
6-1-2	Web Service 的体系结构	145
6-2	如何开发 Web Service 组件	149
6-2-1	使用 JBuilder 创建 Web Service	149
6-2-2	使用 JAX-RPC 创建 Web Service	153
6-3	设计思路	156
6-4	实例代码解析	156
6-4-1	编写个人所得税计算服务 (Web Service)	156
6-4-2	调用 Web Service	162
6-5	本章小结	165
第 7 章	实例六: 智能公文管理系统 (工作流引擎)	166
7-1	工作流基础	166
7-1-1	工作流的概念	167
7-1-2	为什么要用工作流	168
7-2	工作流的实现原理	170
7-3	设计思路	172
7-3-1	WorkFlow 引擎设计综述	172
7-3-2	WorkFlow 引擎所需的数据结构	173
7-4	实例代码解析	175
7-4-1	WorkFlow 定义	175
7-4-2	WorkFlow 实例化	186
7-4-3	工作流执行情况的监控——HistoryRecord 类	210
7-4-4	WorkFlow 执行	212

7-5 本章小结	216
第 8 章 实例七：基于树型结构的用户及权限管理	217
8-1 TreeView 基础.....	218
8-1-1 TreeView 的特征	218
8-1-2 关联关系介绍.....	219
8-1-3 TreeView 的实现方法和表现形式.....	227
8-1-4 各种 TreeView 的实现方式的对比.....	241
8-2 设计思路	242
8-3 实例代码解析	244
8-4 本章小结	247
第 9 章 实例八：基于 XML 的用户管理系统.....	248
9-1 XML 简介	248
9-1-1 XML 的起源.....	248
9-1-2 XML 的用途.....	249
9-1-3 认识 XML 文档.....	250
9-1-4 DTD 和 XML Schema	251
9-2 用 Java 访问 XML 文档.....	251
9-2-1 DOM 简介	252
9-2-2 SAX 简介.....	256
9-2-3 JDOM.....	262
9-3 设计思路	264
9-4 实例代码解析	266
9-5 本章小结	274
第 10 章 实例九：模拟 ATM 处理系统	275
10-1 ATM 处理系统的功能介绍	275
10-2 多线程编程基础.....	276
10-2-1 程序、进程与线程	276
10-2-2 多线程状态和优先级	276
10-2-3 创建并启动线程	278
10-2-4 控制线程.....	281
10-2-5 线程同步.....	282
10-2-6 何时使用多线程	286
10-3 设计思路	287
10-4 实例代码解析	287
10-5 本章小结	295
第 11 章 实例十：Struts 编程实例	296
11-1 Struts 基础	297
11-1-1 Struts 的起源.....	297

11-1-2	什么是 MVC.....	298
11-1-3	Model1 和 Model2	300
11-1-4	什么是 Taglib.....	302
11-1-5	Struts 的优、缺点.....	305
11-2	运行第一个 Struts 应用.....	306
11-3	用 Struts 说 Hello World.....	309
11-3-1	设计思路	310
11-3-1	实例代码解析	310
11-4	本章小结	316
附录 A	使用 JBuilder 开发基于 Struts 的 Web 应用.....	317
一、	Struts 简介	317
二、	使用 JBuilder 创建基于 Struts 框架的 Web 应用	318
附录 B	表示层 (View) 的工具简介.....	327
一、	Calendar 组件	327
二、	WordEdit 组件	329
三、	Chart 系列组件	332

第 1 章

Java 概述

1-1 Java 基础

Java 语言（简称 Java）从诞生之日起直到今天，已经成为使用最广泛的编程语言之一。可以说，Java 语言的发展过程就是 IT 技术在一个时间段发展历程的缩影，各种形式的 Java 语言的流行，甚至对未来 IT 产业的技术发展方向也产生一定的影响。

Java 作为一种类似于 C/C++ 的语言，其平台无关性和互联网发展紧密结合，已经成为互联网和计算机应用的首选平台。更因为它的面向对象、平台无关、分布应用等特点给编程人员带来了一种全新的概念，使 Web 编程从最初单纯的静态信息的发布，到现在基于 Web 技术提供各种信息服务。Java 语言及其平台作为一种革命性的软件开发技术的地位已经确立。而且，Java 被纽约时报评为 1995 年的十大科技成果之一，Java 也将作为一项重大发明载入科技史册。

当微软公司整个产品开发战略从 PC 单机时代向着以网络为中心的计算时代转移时，也选择了加入 Java 开发联盟，作为实现其战略转移时的一个重要决策。

另外，Apple、HP、IBM、Microsoft、Novell、SGI、SCO、Tandem 等公司均在各自开发的操作系统中加入了 Java 开发运行环境。Sun 公司也立即成立了专门负责开发并推广 Java 技术的子公司 Java Soft，通过颁发许可证的办法来允许各家公司把 Java 虚拟机和 Java 的标准类库嵌入他们开发的操作系统。

这种作法使得各种操作系统的开发人员都可以在他们所熟悉的环境中使用 Java 语言编写程序。而且大家用 Java 语言编写出的程序可以轻松地移植到其他平台上运行。这对于程序员无疑很具有吸引力，而“Write once run anywhere”也因为准确地描述 Java 语言的特点，而成为 Java 语言的标准宣传口号。

1-1-1 Java 的发展过程

1991 年，美国 Sun Micro System 公司成立了一个开发小组，最初的目的是为电视机的机顶盒、烤面包箱等家用消费电子产品开发一个可编程的嵌入式系统环境。开发小组最后用 C++ 开发了一种新的语言，取名为 Oak（橡树）。此时，美国伊诺斯大学开发的名为 Mosaic 的 WWW 浏览器，启发了项目组成员，他们用 Oak 编写了 WebRunner 浏览器，很快，这个项



目得到了 Sun 公司领导层的支持。1995 年 1 月, Oak 被重新赋予了一个更响亮的名字——Java, WebRunner 则被命名为 HotJava。随后, Sun 公司将 Java 和 HotJava 在国际互联网上免费发布。

Java 的取名也有一个趣闻: 有一天, 几位开发组的成员正在讨论给这个新的语言取什么名字, 当时他们正在咖啡馆喝着咖啡, 一个人灵机一动说, 就叫 Java (爪哇——咖啡的原产地) 怎样, 他的提议得到了其他人的赞同。于是, Java 这个名字就诞生了。

Java 语言发布后, 发展异常顺利。1995 年春季, Sun 公司公布了完整的 Java 技术规范, 这立刻得到了软件厂商的大力支持, 它们纷纷在自己的浏览器产品内部嵌入了 Java 引擎, 从而支持用 Java 编写的 Applet (小应用程序) 在浏览器中运行。当 Sun 和 Netscape 联合推出了开放式、平台无关的对象脚本语言 JavaScript 后, 更使得 Web 页面变得丰富多彩, 动画、图像等多媒体元素终于可以在浏览器中使用。这些技术直到现在仍然是增强页面交互能力的主要手段被广泛地应用。

从 1995 年到现在 10 年左右的时间里, Java 已经成为 Internet/Intranet 系统开发的首选语言, Java 技术已经被应用到计算机世界的各个领域, 逐渐改变着人们的工作和生活方式。

1. Java 语言的特点

Java 语言的特点主要表现在: 使用简单、完全的面向对象 (Object Oriented Programming, 简称 OOP)、自动内存管理、分布计算、稳定、安全、解释执行、结构中立、跨平台移植、多线程、以及异常处理等方面。

(1) 使用简单

Java 的结构类似于 C 和 C++, 所以一般熟悉 C 与 C++ 语言的开发人员稍加学习就可以掌握 Java 的编程技术了。同时, 由于 Java 不支持指针等复杂而且容易出错的操作, 大大简化了 Java 程序设计开发的难度。而 Java 所特有的垃圾内存 (不再被使用, 而又被程序员所遗忘的内存空间) 自动回收功能, 使得系统对开发人员的要求大大降低, 初级程序员也可以轻松上手。

(2) 完全的面向对象支持

所谓面向对象设计 (OOP), 是一种以数据 (对象) 以及其接口为中心的程序设计技术。通过采用面向对象的设计方法, 可以使程序的结构化程度大大提高, 从而提高了程序的可重用性。

Java 的面向对象机制是 C++ 面向对象机制的延伸。Java 提供了简单的“类”机制和动态的构架模型, 由属性 (状态变量)、方法 (函数) 过程和事件 (触发机制) 封装而成。

对于使用 Java 语言访问系统资源, Java 运行环境中提供了大量的“基类” (以类的方式提供一系列可以访问系统资源的组件) 供开发人员调用。开发人员在编写程序时, 可以通过对基类的引用、继承和重载, 以达到访问、重用和改造基类, 实现代码复用的目的。

(3) 自动内存管理

Java 的自动垃圾 (无用的) 内存回收机制 (Auto Garbage Collection) 实现了内存的自动管理, 因此简化了 Java 程序开发的工作。早期的 GC (Garbage Collection) 对系统资源抢占太多而影响整个系统的运行效率, Java 2 对 GC 功能进行的改良, 使 Java 程序的运行效率有了很大提高。

GC 的工作机制是周期性的自动回收无用存储单元。Java 的自动内存回收机制在简化程序开发的同时, 也提高了程序的稳定性和可靠性。

(4) 分布计算

Java 为程序开发者提供了 `java.net` 包, 该包提供了一组使程序开发者可以轻易实现基于 TCP/IP 的分布式应用系统。此外, Java 还提供了专门针对互联网应用的类库, 如 URL、Java mail 等。

(5) 稳定性

人们最常见的应用程序错误就是“非法访问 xxx 内存”, 其实质是程序指针使用出错。Java 语言中不提供指针的概念, 从而杜绝了由于开发人员对指针的操作失误而造成系统崩溃的可能性。Java 提供真正意义上的数组 (Array), 对运行程序进行下标检查。Java 的自动内存管理在减少编程工作的同时, 大大减少了运行时 (Runtime) 错误。

(6) 安全性

Java 的设计目的是提供一个用于网络/分布式的计算环境。因此, Java 强调安全性, 如确保无病毒、小应用程序运行安全控制等。Java 的验证技术是以公钥 (Public-key) 加密算法为基础, 而且从环境变量、类加载器、文件系统、网络资源等方面实施安全策略。

(7) 解释执行

Java 解释器 (Interpreter) 可以直接在任何已安装解释器的计算机 (或其他设备) 上解释、执行 Java 字节码, 不需重新编译。当然, 其版本向上兼容, 因此如果是高版本环境下编译的 Java 字节码到低版本环境下运行也许会有部分问题。

(8) 跨异构环境

Java 是网络空间的“世界语”, 编译后的 Java 字节码是一种结构中立性 (Architecture Neutral) 的目标文件格式, 可以在所有提供 Java 虚拟机 (JVM) 的多种不同主机、不同处理器的上运行。

而且 Java 的各种形态 (例如: Applet、Application、Servlet、JavaBean) 最终都是以字节码 (.class) 的形式存在, 在 Java 虚拟机中运行。

(9) 平滑移植

“Write once run any where!” 也许是 Java 最诱人的特点。用 Java 开发而成的系统其移植工作几乎为零, 一般情况下只需对配置文件、批处理文件作相应修改即可实现平滑移植。

(10) 多线程

Java 的多线程 (Multithreading) 机制使程序可以并行运行。Java 还有一组同步化基本单元, 同步机制保证了对共享数据的正确操作。多线程使程序设计者可以用不同的线程分别实现各种不同的行为, 而不需要采用全局的事件循环机制, 因此, 使用 Java 语言可以非常轻松地实现网络上的实时交互行为。

(11) 异常处理

早期的程序员大都有使用 goto 语句来实现条件跳转, Java 编程中不支持 goto 语句。Java 采用异常模型使程序的主流逻辑变得更加清晰明了, 并且能够简化错误处理任务。

(12) 可扩充

Java 目前发布的 J2EE 标准主要是为采用 Java 技术为企业级应用提供全面解决方案而提供的技术规范框架, 该构架包含了可分布式部署的 Java 应用规范, 该技术构架的应用将使得 Java 技术可以用于构架大型企业级应用的解决方案。

综上所述, Java 语言由于独立于系统平台运行、完全的面向对象设计、支持多线程编程



以及安全、稳定性等诸多优点，为 Internet 程序员所青睐，并逐渐成为网络开发语言的代名词。

2. Java 语言对软件工程的影响

曾有人预言：“Java 语言的出现，将会引起一场软件革命”，这是因为 Java 语言能在执行码（字节码）上兼容，这使得用 Java 开发的系统运行在不同的机器上成为可能——只要所用的机器能提供 Java 的解释器即可。

时至今日，Java 的优势已经不再仅限于跨平台和动态页面显示，而更多的体现在对 CORBA、Web Service 的支持，以及在企业应用整合等方面所体现出的不可替代性。

要了解 Java 语言对现在以及未来软件工程和软件发展可能产生的影响，可以从以下几个方面进行着手：

（1）软件工程

Java 适用于 Web 的许多特征，如可移植性、安全性、健壮性和可扩展性为面向对象的软件工程设计提供了实现工具和想像的空间。遵循面向对象思想，软件生命周期的各个部分都将从面向对象的角度出发实现。比如用户需求分析将被划分成具有不同状态（数据）和行为的模块，不同模块之间的联系通过消息实现等。

（2）软件的开发方法

由于 Java 语言的面向对象的特性，越来越多的人在用 OO 开发时都在选择使用 Java 技术与方法来开发，这与 Java 的面向对象特性、模块化特性、安全性以及适应 IT 技术发展密不可分。

（3）Web 技术

Java 提供的 CGI、ASP 替代方案 Servlet、JSP 以及 EJB 技术都是与互联网发展需求为导向，因而采用 Java 技术作为基于 Web 应用的企业解决方案是非常理想的选择。

（4）分布式计算

Java 对 CORBA、RMI（远程方法调用）、EJB、XML 和网络处理的支持和应用，也将使得越来越多的企业分布式计算采用 Java 技术实现。

（5）图形处理与多媒体效果

Java 2 以及以后版本提供了 Swing、2D、3D、Sound 等图形、图像和多媒体支持类库，弥补了早期 Java 版本在此方面的不足。Java 的媒体处理能力使程序开发的动画效果远比 GUI 技术更加逼真，尤其是利用 WWW 提供的巨大动画资源空间，可以共享全世界的动态画面资源。

（6）产品开发

目前 Java 技术被广泛地应用在：基于 Web 的实时信息发布系统、Web 邮件系统、用于企业分布式计算的消息传输系统、基于 Servlet/JSP/EJB 开发的电子商务平台、Servlet/JSP/EJB 的运行环境 and 应用服务器、结合 CORBA/XML 等技术开发的分布计算支撑平台等。另外，使用 Java 技术开发多媒体应用也有着广泛的用户基础。

3. JSP 基础知识

JSP (Java Server Pages) 是由 Sun 公司在 Java 语言上开发出来的一种动态网页制作技术，它可使您将网页中的动态部分和静态的 HTML 相分离。您可以使用平常得心应手的工具并按照平常的方式来书写 HTML 语句。然后，将动态部分用特殊的标记嵌入即可，这些标记常常

以“<%”开始并以“%>”结束。

Internet 上曾一度充斥着各式各样的静态网页（或是一些简单的表），而当前各种各样的交互环境下的应用，导致对动态网页的需求越来越大了。以前不少公司都使用 CGI 技术进行动态网页的开发，但由于 CGI 本身存在的性能低下、兼容性差、可维护性较差等多种原因，使得 CGI 技术逐渐被 Web 世界所淘汰。

而 JSP (Java Server Page) 作为一种全新的网页编程技术，不仅提供了一种方便有效的动态网页制作方法，而且作为 Java 家族的一分子，它还具有一般 Java 应用的良好可移植性，这使得 JSP 页面可以在任何支持 Java 的应用服务器中运行。

JSP 是一种应用性很强的技术，很难用抽象的语言给其一个定义。但可以这样理解：JSP 是一种使用服务器端的处理程序来动态生成网页的技术或编程语言。

JSP 的特点是：将应用逻辑 (Business Logic) 从页面设计中分开，同时将这种逻辑放在方便的、可重用的 Java 部件中（例如 JavaBean）。

JSP 页面一般是由如下三部分组成的：

- 静态的 HTML 或 XML 的组件。
- 特殊的 JSP 标签，包括指令 (Directive) 标签和动作 (Action) 标签。
- 脚本元素 (Scripting Element)，即一些直接嵌入的 Java 程序（如：Java Applet）。

另外，作为对 JSP 概念的补充，下面简要介绍一下，基于 Web 的客户端其实就是在传统的 Client/Server 结构中增加了一个 Web 服务器，使其成为多层结构。这种结构的优点在于：对于表现层的用户界面的请求，是直接由 Web Server 进行响应的，所以反馈速度较快。同时，所有的 JSP 页面都被保存在 Web Server 上，当其内容发生变化时，Web Server 自动对其进行编译（编译为字节码）从而保证字节码的版本总是最新的。这样使得 JSP 的程序员不需要了解 Java 的标准编写格式，同时也使得用 JSP 编写的网页易于维护和修改。

现在我们对 JSP 有了一个简单的认识，下面将详细介绍使用 JSP 开发 Web 的应用。

4. 为什么要使用 JSP 开发 Web 应用

(1) JSP 的效率和安全性更高

JSP 技术和微软的 ASP (Active Server Page) 技术相比，后者以源码形式存放，解释执行。浏览器端的用户每次对 ASP 页面的请求都需要对所有源码进行顺序解释执行，运行效率显然不会太高。另外，ASP 和 IIS (Internet Information Server，微软发布的 Web 服务器) 的漏洞频频曝光，也使得系统安全存在严重隐患。

JSP 页面在执行前先被编译成字节码 (Byte Code)，字节码由 Java 虚拟机 (Java Virtual Machine) 解释执行，比源码直接解释执行的效率高。而且，多数支持 JSP 的 Web 服务器上还提供字节码的缓存 (Cache) 机制，也使得字节码的响应效率大大提高。

需要注意的是，第一次访问（请求）JSP 网页的速度可能稍慢，因为该页面在第一次请求时被动态地编译成 class 文件，并放置缓存中，这样以后再次请求同一页面时速度就快得多了。同时，JSP 源程序不大可能被下载，特别是用 JavaBean 编写的系统组件可以被放置在该网站的主目录之外。

(2) JSP 的组件 (Component) 更加方便

ASP 通过 COM 来扩充复杂的功能，如文件上传、发送 E-mail 以及将业务处理或者复杂计算分离出来成为独立可重复利用的模块。JSP 通过 JavaBean 完成同样的功能扩充。



在开发方面,COM 的开发远比 JavaBean 复杂和繁琐,学会 ASP 不难,但学会开发 COM 却不容易,相比较而言,使用 JavaBean 对系统功能进行扩充则较容易实现。而且在后期维护的时候,COM 组件必须在服务器上注册后方可使用,如果对 COM 组件进行了更新,就必须在注册表中对组件进行重新注册,更新 COM 组件的过程,对很多程序员来说都是一段不愉快的经历。

所幸的是 JavaBean 并不需要注册,只要将其放置在 CLASSPATH 所包含的目录中就行了。但若 JavaBean 进行了修改,大多数应用服务器(如:JSWDK 和 Tomcat)都需要关闭和重新运行,以载入最新版的组件。

另外 JavaBean 是完全的 OOP,可以针对不同的业务处理功能方便地建立一整套可重复利用的对象库,例如用户权限控制、E-mail 自动回复等。

(3) JSP 的适应平台更广

ASP 目前仅适用于 Windows 的各种版本。经管我们可以用 ChiliSoft 的插件在 Unix 平台上部署支持 ASP 的 Web 服务器,但是由于 ASP 本身的功能有限,大多数基于 ASP 的应用必须通过 COM 组件来对其进行扩充,而 Unix 平台无法对 COM 进行很好的支持,这也使得在 Unix 平台下运行 ASP 程序无法被多数人所接受。

JSP 则不同,几乎所有平台都支持 Java, JSP+JavaBean 可以在所有平台下通行无阻。我们可以在微软的 Windows 操作系统中的 IIS (Internet Information Server) 中安装插件,来支持 JSP 程序。著名的 Web 服务器 Apache 已经能够支持 JSP。由于 Apache 广泛应用在 Windows、Unix 和 Linux 上,因此 JSP 有更广泛的运行平台。

从一个平台移植到另外一个平台, JSP 和 JavaBean 甚至不用重新编译,因为 Java 字节码都是与平台无关的。

5. JSP 的设计目标

JSP 的设计目标主要是提供一种更为简便、有效的动态网页编写手段,并且增强网页程序的独立性、兼容性和可重用性。

它是通过以下几个方面加以实现的:

(1) 简便性和有效性

通过前面对 JSP 的组成的介绍,可以知道 JSP 动态网页的编写与一般的静态 HTML 的网页的编写是十分相似的。只是在原来的 HTML 网页中加入一些 JSP 专有的标签或是一些脚本程序。这样,使得一个熟悉 HTML 网页编写的设计人员可以比较容易地转变成为一个 JSP 网页的开发人员。

(2) 程序的独立性

JSP 是 Java API 家族的一部分,它拥有一般的 Java 程序的跨平台的特性,也就是说, JSP 拥有 Java 程序的对平台的独立性。

(3) 程序的兼容性

因为 JSP 中的动态内容可以各种形式进行显示,所以它可以为各种客户提供服务:从使用 HTML/DHTML 的浏览器到使用 WML 的各种手提无线设备(如:移动电话和个人数字设备 PDA),再到使用 XML 的 B2B 应用,都可以使用 JSP 的动态页面。

(4) 程序的可重用性

前面已经提到,在 JSP 页面中可以不直接将脚本程序嵌入,而只是将动态的交互部分作