

DVD

含视频讲解
案例源码

深入理解

Java Web 开发技术：

探索基于主流框架的最佳组合

全面介绍Java Web技术的核心知识，内容深入、语言通俗
剖析Struts+Spring+Hibernate框架在综合项目中的使用
采用理论加实践的教学方法，兼顾理论、案例的完美展现

谭贞军 编著

清华大学出版社

深入理解 Java Web 开发技术： 探索基于主流框架的最佳组合

谭贞军 编 著

清华大学出版社
北京

内 容 简 介

本书内容新颖、知识全面、讲解详细，可以帮助读者深入理解 Java Web 开发技术的基本知识，并能挖掘出基于主流框架的最佳组合。

全书共分 16 章，内容都采用了理论加实践的教学方法，每个实例先提出制作思路及列举包含的知识点，然后力求用最通俗的语言将高深的知识阐述出来。

本书适合 Java 各个级别的程序员、研发人员及在职程序员阅读，也可以作为相关培训学校和大专院校相关专业的教学用书。

本书封面贴有清华大学出版社防伪标签，无标签者不得销售。

版权所有，侵权必究。侵权举报电话：010-62782989 13701121933

图书在版编目(CIP)数据

深入理解 Java Web 开发技术：探索基于主流框架的最佳组合/谭贞军编著. --北京：清华大学出版社，2013

ISBN 978-7-302-31010-5

I. ①深… II. ①谭… III. ①JAVA 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字(2012)第 304164 号

责任编辑：魏 莹 桑任松

装帧设计：杨玉兰

责任校对：周剑云

责任印制：何 芊

出版发行：清华大学出版社

网 址：<http://www.tup.com.cn>, <http://www.wqbook.com>

地 址：北京清华大学学研大厦 A 座 邮 编：100084

社 总 机：010-62770175 邮 购：010-62786544

投稿与读者服务：010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈：010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载：<http://www.tup.com.cn>, 010-62791865

印 刷 者：北京鑫丰华彩印有限公司

装 订 者：三河市李旗庄少明印装厂

经 销：全国新华书店

开 本：185mm×260mm 印 张：40 字 数：970 千字

附 DVD1 张

版 次：2013 年 3 月第 1 版

印 次：2013 年 3 月第 1 次印刷

印 数：1~4000

定 价：76.00 元

产品编号：046897-01

目 录

第 1 章 Java Web 初体验	1	2.1.5 插入图片	32
1.1 学习 Java 的巨大优势	2	2.1.6 列表处理	35
1.1.1 排名第一的编程语言	2	2.2 HTML 页面布局	36
1.1.2 良好的就业前景	2	2.2.1 使用表格标记	36
1.2 学习 Java 需要了解的那些事	3	2.2.2 使用框架标记	40
1.2.1 Java 语言的发展历史	3	2.2.3 使用层标记	43
1.2.2 Java 的特点	4	2.3 处理表单	44
1.2.3 Java 的平台无关性	5	2.3.1 表单标记介绍	44
1.3 Web 技术介绍和工作原理	6	2.3.2 使用 form 标记	45
1.3.1 本地计算机和远程服务器	7	2.3.3 使用文本域	45
1.3.2 Web 应用程序的工作原理	7	2.3.4 使用文本区域和按钮	46
1.3.3 常用的 Web 开发技术	8	2.3.5 使用单选按钮和复选框	46
1.4 开发前的准备	9	2.3.6 使用列表菜单	47
1.4.1 安装 JDK	9	2.3.7 使用文件域、图像域和 隐藏域	47
1.4.2 设置 PATH 环境变量	13	2.4 特效和多媒体处理	48
1.4.3 JRE 和 JDK 是有区别的	14	2.4.1 实现滚动效果	48
1.4.4 困扰初学者的环境变量问题	14	2.4.2 设置背景音乐	50
1.5 安装并配置 Tomcat 服务器	15	2.4.3 插入 Flash	50
1.5.1 获取 Tomcat	15	2.4.4 使用 ActiveX 控件	51
1.5.2 配置 Tomcat 的服务端口	18	2.5 谈 XML 与 HTML 的区别	52
1.5.3 列出 Web 应用根路径下的 所有页面	18	2.6 初学者应该明白的几个问题	54
1.5.4 登录控制台	19	2.6.1 用 Dreamweaver 提高效率	54
1.5.5 设置虚拟目录	21	2.6.2 在 HTML 网页中实现空格	56
1.6 初学者应该明白的几个问题	22	2.6.3 网站内部链接优化	56
1.6.1 独特的垃圾回收机制	22	2.6.4 几种增加外部链接的方法	58
1.6.2 对学习者的三条建议	23	第 3 章 CSS 技术基础	59
第 2 章 HTML 技术介绍	25	3.1 CSS 技术介绍	60
2.1 创建基本静态页面	26	3.1.1 CSS 概述	60
2.1.1 设置网页头部和标题	26	3.1.2 CSS 的特点和意义	60
2.1.2 设置页面正文和注释	26	3.2 CSS 的语法结构	61
2.1.3 文字和段落处理	28	3.3 CSS 选择符	62
2.1.4 超级链接处理	30	3.4 调用方式	63



3.5	定位布局.....	65	4.4.1	if 条件语句.....	101
3.5.1	元素排列.....	65	4.4.2	for 循环语句.....	102
3.5.2	浮动属性定位.....	66	4.4.3	while 循环语句.....	103
3.6	使用 CSS 属性.....	67	4.4.4	do...while 循环语句.....	104
3.6.1	属性类别.....	67	4.4.5	break 控制.....	105
3.6.2	定位属性.....	69	4.4.6	switch 循环语句.....	106
3.6.3	内容控制属性.....	70	4.5	JavaScript 的函数.....	106
3.6.4	浮动属性.....	73	4.5.1	JavaScript 函数的构成.....	106
3.6.5	列表属性.....	75	4.5.2	JavaScript 常用函数.....	107
3.6.6	补白属性.....	77	4.6	JavaScript 对象.....	109
3.6.7	边框属性.....	78	4.6.1	对象基础知识.....	109
3.6.8	边界属性.....	81	4.6.2	JavaScript 常用对象和方法... ..	111
3.7	CSS 修饰.....	83	4.7	JavaScript 事件.....	114
3.7.1	文本修饰.....	84	4.7.1	JavaScript 常用事件.....	114
3.7.2	图片修饰.....	85	4.7.2	事件处理程序.....	115
3.7.3	表单修饰.....	85	4.8	JavaScript 的窗口对象.....	116
3.7.4	修饰水平线.....	89	4.8.1	窗口对象.....	116
3.8	初学者应该明白的几个问题.....	90	4.8.2	窗口对象的事件驱动.....	116
3.8.1	体会 ID 选择符的奥妙.....	90	4.8.3	窗口对象中的属性.....	116
3.8.2	每种颜色对应十六进制 颜色值.....	90	4.8.4	窗口对象的方法.....	117
3.8.3	分析相对路径和绝对路径的 优缺点.....	90	4.8.5	JavaScript 窗口对象的应用... ..	117
3.8.4	总结调用 CSS 的方式.....	91	4.9	JavaScript 框架对象.....	118
第 4 章	引入 JavaScript.....	93	4.10	初学者应该明白的几个问题.....	120
4.1	JavaScript 简介.....	94	4.10.1	JavaScript 与 Java 的区别... ..	120
4.1.1	运行环境.....	94	4.10.2	错误处理机制.....	121
4.1.2	JavaScript 格式.....	94	4.10.3	设计师们的几块大蛋糕.....	121
4.1.3	一个典型的 JavaScript 文件.....	94	第 5 章	JSP 基础.....	123
4.2	数据类型.....	95	5.1	JSP 概述.....	124
4.2.1	数据类型概述.....	95	5.2	构建 Web 应用.....	124
4.2.2	JavaScript 常量.....	96	5.3	配置描述文件 web.xml.....	126
4.2.3	JavaScript 变量.....	96	5.4	深入理解 JSP 机制.....	127
4.3	表达式和运算符.....	97	5.5	基本语法.....	131
4.3.1	JavaScript 的表达式.....	97	5.5.1	基本语法介绍.....	131
4.3.2	JavaScript 的运算符.....	98	5.5.2	注释.....	132
4.4	JavaScript 转向控制语句.....	100	5.5.3	JSP 的声明.....	134
			5.5.4	输出表达式.....	135
			5.5.5	脚本.....	136



5.6 编译指令.....	139	6.2.5 修改 web.xml.....	189
5.6.1 page 指令.....	139	6.2.6 应用自定义标签.....	189
5.6.2 include 指令.....	141	6.2.7 将页面片段作为属性的 标签.....	195
5.7 JSP 的动作指令.....	142	6.2.8 动态属性的标签.....	196
5.7.1 七个动作指令.....	142	6.3 JSP 2 的新特性.....	198
5.7.2 响应转发指令 forward.....	143	6.3.1 JSP 2 新特性概述.....	198
5.7.3 动态 include 指令.....	144	6.3.2 配置 JSP 属性.....	199
5.7.4 下载指令 plugin.....	145	6.3.3 JSP 的表达式.....	201
5.7.5 设置参数值指令 param.....	146	6.3.4 Tag File.....	210
5.7.6 JavaBean 指令: useBean、 setProperty 和 getProperty.....	146	6.4 初学者应该明白的几个问题.....	211
5.7.7 将 JavaBean 实例放在 指定的 scope.....	148	6.4.1 JSP 内置对象的实质.....	211
5.8 初学者应该明白的几个问题.....	149	6.4.2 分析 exception 对象的原理... ..	211
5.8.1 了解 JSP 的强势和弱势.....	149	6.4.3 并不是每个表单域都会生成 请求参数.....	212
5.8.2 探索 JSP.....	149	6.4.4 总结 request 中的方法.....	212
5.8.3 体会属性 errorPage 的好处....	150	6.4.5 比较 forward 和 redirect.....	213
5.8.4 静态导入和动态导入的 区别.....	152	6.4.6 Cookie 中的中文字符问题....	214
5.8.5 比较 forward 动作指令和 include 动作指令.....	152	6.4.7 了解 JSP 的 session 机制.....	214
第 6 章 JSP 核心.....	153	第 7 章 Servlet 详解.....	215
6.1 JSP 的内置对象.....	154	7.1 什么是 Servlet.....	216
6.1.1 内置对象简介.....	154	7.1.1 Servlet 技术的优越性.....	216
6.1.2 application 对象.....	155	7.1.2 Servlet 的持久性.....	216
6.1.3 使用 config 对象.....	164	7.1.3 Applet 和 Servlet 的异同.....	217
6.1.4 exception 对象.....	166	7.2 Servlet 开发初体验.....	217
6.1.5 使用 pageContext 对象.....	167	7.3 配置 Servlet.....	219
6.1.6 使用 out 对象.....	169	7.4 Servlet 的功能.....	227
6.1.7 使用 request 对象.....	170	7.4.1 与 Servlet 配置相关的接口....	227
6.1.8 response 对象.....	176	7.4.2 Servlet 编程接口.....	228
6.1.9 session 对象.....	180	7.4.3 Servlet 应用接口.....	228
6.2 自定义 JSP 标签.....	184	7.5 使用 HttpServlet 类封装请求.....	230
6.2.1 推出自定义标签的好处.....	184	7.5.1 HttpServlet 类.....	230
6.2.2 开发自定义标签类.....	185	7.5.2 请求与响应.....	233
6.2.3 编写 TLD 文件.....	185	7.5.3 会话.....	234
6.2.4 使用标签库里的标签.....	188	7.5.4 封装请求.....	237
		7.6 Servlet 过滤器.....	241
		7.6.1 配置过滤器.....	241



7.6.2 使用过滤器.....	241	8.1.3 体验 JavaBean.....	273
7.7 load-on-startup Servlet.....	243	8.1.4 总结 JavaBean 的卓越性能....	275
7.8 访问 Servlet 的配置参数.....	244	8.2 JSP 和 JavaBean 之间的关系.....	275
7.9 Servlet 的“加强版”——Filter.....	247	8.2.1 JSP 访问 JavaBean.....	275
7.9.1 Filter 基础.....	247	8.2.2 在 JSP 中调用 JavaBean.....	277
7.9.2 创建 Filter.....	247	8.3 设置 JavaBean 属性.....	278
7.9.3 配置 Filter.....	249	8.3.1 Simple 属性.....	278
7.10 使用 Listener 类.....	252	8.3.2 Indexed 属性.....	279
7.10.1 Listener 类基础.....	252	8.3.3 Bound 属性.....	280
7.10.2 实现 Listener 类.....	253	8.3.4 Constrained 属性.....	281
7.10.3 使用 ServletContext AttributeListener.....	255	8.3.5 JSP 设置属性.....	282
7.10.4 使用 ServletRequestListener 和 ServletRequestAttribute Listener.....	256	8.4 JavaBean 方法.....	286
7.10.5 使用 HttpSessionListener 和 HttpSessionAttribute Listener.....	257	8.5 JavaBean 的作用范围.....	291
7.11 全新的 Servlet 3.0.....	259	8.5.1 page 作用域.....	292
7.11.1 新的 Annotation.....	260	8.5.2 request 作用域.....	293
7.11.2 对 Web 模块的支持.....	260	8.5.3 session 作用域.....	294
7.11.3 新引入的异步处理.....	262	8.5.4 application 作用域.....	295
7.11.4 改进的 Servlet API.....	264	8.6 JavaBean 的移除.....	297
7.11.5 动态地注册 Servlet 和 Filter 的方法.....	266	8.7 初学者应该明白的几个问题.....	301
7.12 初学者应该明白的几个问题.....	267	8.7.1 在 Web 页面中调用 JavaBean 组件.....	301
7.12.1 Servlet 和 JSP 的比较.....	267	8.7.2 在 JSP 中调用 JavaBean 的优点.....	303
7.12.2 使用 Annotation 配置 Servlet.....	267	8.7.3 获得 JavaBean 的属性.....	304
7.12.3 Filter 和 Servlet 的关系.....	268	第 9 章 JSTL 标签库.....	305
7.12.4 HttpServletRequest 的用法...268		9.1 JSTL 基础.....	306
7.12.5 使用异步监听器.....	268	9.1.1 使用第三方提供的标签库.....	306
第 8 章 深入详解 JavaBean.....	271	9.1.2 JSTL 标签的构成.....	307
8.1 JavaBean 基础.....	272	9.2 JSTL Core 标签库.....	308
8.1.1 JavaBean 介绍.....	272	9.2.1 一般用途的标签.....	308
8.1.2 Java EE 中 JavaBean 处于 哪一层.....	273	9.2.2 条件标签.....	312
		9.2.3 迭代标签.....	314
		9.2.4 与 URL 相关的标签.....	318
		9.3 I18N 标签库.....	321
		9.3.1 <fmt:formatNumber>标签.....	322
		9.3.2 <fmt:parseNumber>标签.....	325
		9.3.3 <fmt:formatDate>标签.....	327

9.3.4	<fmt:parseDate>标签.....	330	10.1.1	Ajax 技术介绍	358
9.3.5	<fmt:setTimeZone>标签	331	10.1.2	Ajax 开发模式与传统开发 模式的比较	359
9.3.6	<fmt:timeZone>标签	332	10.1.3	Ajax 是一门提高用户体验 的技术	360
9.3.7	<fmt:setLocale>标签	334	10.1.4	Ajax 需要注意的几个问题....	361
9.3.8	<fmt:requestEncoding>标签.....	336	10.2	XMLHttpRequest 对象	362
9.3.9	<fmt:setBundle>标签.....	336	10.2.1	创建 XMLHttpRequest 对象	363
9.3.10	<fmt:message>标签	336	10.2.2	XMLHttpRequest 对象 的方法	364
9.3.11	<fmt:param>标签.....	339	10.2.3	XMLHttpRequest 对象 的属性	366
9.3.12	<fmt:bundle>标签.....	339	10.2.4	XMLHttpRequest 对象的 事件句柄函数	367
9.4	使用 SQL 标签库	340	10.3	与服务器通信——发送请求与 处理响应	368
9.4.1	<sql:setDataSource>标签	340	10.3.1	发送请求	368
9.4.2	<sql:query>标签和<sql:param> 标签	341	10.3.2	处理服务器响应	369
9.4.3	<sql:update>标签.....	343	10.4	解决中文乱码问题	371
9.4.4	<sql:dateParam>标签.....	344	10.4.1	发送请求时出现中文乱码 ...	371
9.4.5	<sql:transaction>标签.....	344	10.4.2	获取服务器的响应结果时 出现中文乱码	371
9.5	函数标签.....	345	10.5	Ajax 重构	372
9.5.1	求长度函数.....	345	10.6	使用 Ajax 技术	373
9.5.2	大小写转换函数.....	346	10.6.1	一个简单的 Ajax 程序	374
9.5.3	求子串函数.....	346	10.6.2	分析 Ajax 的步骤	375
9.5.4	去空白函数.....	347	10.7	表单验证实例	379
9.5.5	替换函数.....	347	10.8	无刷新上传进程实例	385
9.5.6	查找函数.....	347	10.9	初学者应该明白的几个问题	389
9.5.7	拆分与组合函数.....	348	10.9.1	异步和同步的问题	389
9.5.8	函数 escapeXml()	349	10.9.2	GET 方式和 POST 方式的 选择	389
9.6	XML 标签库	349	10.9.3	体会 innerText 属性的特殊 作用	389
9.7	初学者应该明白的几个问题	352	10.9.4	充分利用 Ajax 的 开源框架	390
9.7.1	总结 Core 标签库.....	352			
9.7.2	<fmt:timeZone>和 <fmt:setTimeZone>的区别	353			
9.7.3	<fmt:setBundle>与<fmt:bundle> 的区别	353			
9.7.4	体会属性文件在国际化中的 作用	353			
9.7.5	关于 SQL 标签的一些争论	354			
9.7.6	在 JSP 代码中穿插 XML 文件.....	354			
第 10 章 Ajax 技术.....		357			
10.1	Ajax 基础	358			



10.9.5 探讨验证码的无刷新验证的实现思路.....	390
第 11 章 数据库编程	397
11.1 与数据库相关的概念.....	398
11.2 SQL 语言.....	400
11.2.1 数据定义.....	400
11.2.2 数据操纵.....	402
11.2.3 视图.....	404
11.2.4 SQL 高级操作.....	405
11.3 初识 JDBC.....	407
11.3.1 JDBC 驱动类型.....	407
11.3.2 JDBC 的常用接口和类.....	408
11.3.3 JDBC 编程步骤.....	410
11.4 常用的几种数据库.....	413
11.4.1 Access 数据库.....	413
11.4.2 SQL Server 数据库.....	413
11.4.3 MySQL 数据库.....	417
11.5 执行 SQL 语句的方式.....	417
11.5.1 使用 executeUpdate.....	417
11.5.2 使用 execute 方法.....	418
11.6 事务处理.....	420
11.6.1 JDBC 中的事务控制.....	420
11.6.2 检查数据库是否支持事务处理.....	422
11.6.3 JDBC 事务控制的流程.....	422
11.7 存储过程.....	425
11.7.1 创建存储过程.....	426
11.7.2 调用创建的存储过程.....	428
11.8 初学者应该明白的几个问题.....	429
11.8.1 通过课外学习掌握数据库技术.....	430
11.8.2 没有最完美的产品.....	430
11.8.3 合适的就是最好的.....	432
第 12 章 Struts 2 基础	435
12.1 MVC 思想.....	436
12.1.1 什么是 MVC 思想.....	436
12.1.2 MVC 思想及其优势.....	437
12.1.3 Struts MVC 思想.....	437
12.2 下载并安装 Struts 2.....	438
12.2.1 获得并设置 Struts 2.....	438
12.2.2 在 Eclipse 中使用 Struts 2.....	440
12.2.3 使用 Struts 2 的流程.....	441
12.2.4 Struts 2 的工作流程.....	443
12.3 配置 Struts 2.....	443
12.3.1 配置常量.....	443
12.3.2 包含其他配置文件.....	448
12.4 实现 Action.....	448
12.4.1 基础知识.....	448
12.4.2 标准 Action 类.....	450
12.4.3 ActionSupport 基类.....	450
12.4.4 访问 Servlet API.....	455
12.4.5 直接访问 Servlet API.....	458
12.4.6 用 ServletActionContext 访问 Servlet API.....	459
12.5 配置 Action.....	462
12.5.1 命名空间和包.....	462
12.5.2 基本配置.....	466
12.5.3 使用 Action 的动态方法调用.....	466
12.5.4 为 Action 配置 method 属性.....	469
12.5.5 使用通配符映射方式.....	470
12.6 初学者应该明白的几个问题.....	470
12.6.1 MVC 思想和观察者模式.....	471
12.6.2 使用 Action 处理用户请求.....	471
12.6.3 一份完整 struts.xml 文件的常量配置骨架.....	471
第 13 章 Hibernate 基础	475
13.1 Hibernate 基础.....	476
13.1.1 Hibernate 概述.....	476
13.1.2 Hibernate API 简介.....	476
13.1.3 Hibernate 的核心接口.....	477
13.1.4 Hibernate 的体系结构.....	478
13.2 Hibernate 的下载和安装.....	478

13.2.1 下载 Hibernate	478	13.11.7 其他属性	517
13.2.2 为 Eclipse 安装插件	480	13.12 初学者应该明白的几个问题	518
13.3 认识几种简单配置 Hibernate 的 方式	481	13.12.1 应该尽量使用 数据源方式	518
13.3.1 配置数据源	481	13.12.2 何时使用 Hibernate 作为 持久化框架	518
13.3.2 配置 c3p0 连接池	482	13.12.3 分析集合性能	518
13.3.3 配置 proxool 连接池	483	第 14 章 Spring 技术	521
13.3.4 MySQL 连接配置	484	14.1 Spring 技术介绍	522
13.3.5 SQL Server 连接配置	485	14.1.1 Spring 的优点	522
13.3.6 Oracle 连接配置	486	14.1.2 Spring 将改变 Java EE	522
13.4 Hibernate 配置	486	14.2 下载并安装 Spring	522
13.4.1 持久化操作实例	486	14.2.1 获取 Spring 压缩包	522
13.4.2 创建 Configuration 对象	490	14.2.2 在 Eclipse 中使用 Spring	524
13.4.3 Hibernate 的 JDBC 连接	493	14.2.3 在 Web 项目中 使用 Spring	527
13.4.4 数据库方言	493	14.3 依赖注入	530
13.4.5 与 Hibernate 相关的 常用属性	494	14.3.1 依赖注入基础	530
13.5 SessionFactory 接口	496	14.3.2 设置注入	531
13.6 Session 接口	497	14.3.3 代码升级游刃有余	533
13.6.1 save()方法	498	14.3.4 构造注入	534
13.6.2 get()方法	498	14.4 Spring 容器	535
13.6.3 load()方法	499	14.4.1 Spring 的 IoC 容器	537
13.6.4 update()方法	499	14.4.2 BeanFactory 对 IoC 容器的 功能定义	538
13.6.5 delete()方法	500	14.4.3 XmlBeanFactory 的 工作原理	538
13.7 Transaction 接口	501	14.4.4 ApplicationContext 的 国际化	541
13.8 Query 接口	501	14.4.5 详解 ApplicationContext 的 事件机制	542
13.9 Criteria 接口	502	14.4.6 Bean 获取 Spring 容器	546
13.10 持久化对象	503	14.5 Spring 中的 Bean	549
13.10.1 持久化类的规则	503	14.5.1 定义 Bean	549
13.10.2 持久化的状态	503	14.5.2 Spring 中 Bean 的作用域	550
13.10.3 改变持久化对象状态	505	14.5.3 配置依赖	553
13.11 Hibernate 映射文件	505	14.5.4 属性值注入	554
13.11.1 映射文件结构	506	14.5.5 注入 field 值	556
13.11.2 主键映射	507	14.5.6 注入方法返回值	557
13.11.3 普通属性映射	508		
13.11.4 映射引用属性	510		
13.11.5 映射 Set 集合属性	515		
13.11.6 映射 List 集合属性	517		



14.6	Java 配置管理.....	560	15.7.3	分解 Struts + Spring + Hibernate 的层次.....	592
14.7	初学者应该明白的几个问题.....	563	15.7.4	Spring 整合 Struts 的三个 窍门.....	593
14.7.1	使用 HttpInvoke 提高效率.....	563	15.7.5	拦截 Struts.....	595
14.7.2	Hibernate 与 Spring 的 JDBC Template 效率比较.....	567	15.7.6	Spring 整合 Hibernate.....	596
14.7.3	Spring + Hibernate 模式下 数据操作的效率问题.....	568	15.8	Struts 与 Spring 两种 MVC 框架的 比较.....	599
14.7.4	Spring 效率的测试报告.....	568	15.8.1	Struts.....	600
第 15 章	深入挖掘最优框架组合.....	573	15.8.2	Spring.....	601
15.1	纵览各种各样的框架.....	574	第 16 章	OA 办公自动化系统.....	603
15.2	Struts 2 + iBatis + JSON 实现快速 Java Web 2.0 程序开发.....	575	16.1	需求分析.....	604
15.3	JSF 和 Web 的完美组合, 提升 开发效率.....	576	16.2	项目规划.....	604
15.3.1	传统 JSP 开发存在的问题.....	576	16.3	编写项目计划书.....	605
15.3.2	开发基于 JSF 的 Web 应用程序步骤.....	576	16.4	构建开发环境.....	607
15.4	分析 Java Web 的 MVC 实现模型.....	578	16.5	规划系统目录结构.....	607
15.4.1	MVC 介绍.....	578	16.6	搭建数据库.....	607
15.4.2	举例说明 MVC 的用法.....	579	16.6.1	数据库 E-R 图分析.....	608
15.4.3	基本 MVC 的实例.....	581	16.6.2	数据结构表.....	611
15.5	分析常规 Java Web 的开发模式.....	585	16.7	最后的准备工作.....	618
15.5.1	三个标准动作.....	585	16.7.1	数据表的创建及测试数据的 插入.....	618
15.5.2	JSP 的设计模式.....	586	16.7.2	系统文件组织结构.....	619
15.6	总结 Java Web 开发中的几种经典 模式.....	587	16.7.3	数据源及框架的配置.....	619
15.7	SSH 模式.....	589	16.8	具体编码.....	620
15.7.1	SSH 的整合理念.....	589	16.8.1	登录模块.....	620
15.7.2	组装 Struts + Spring + Hibernate 的背景.....	591	16.8.2	管理首页模块.....	623
			16.8.3	紧急报警模块.....	624
			16.8.4	文件管理模块.....	624
			16.9	调试与发布.....	625
			参考文献.....	627	



第 1 章



Java Web 初体验

Java 语言是当今使用率最高的一门编程语言，从它诞生的那一天起，便被广大程序员所接受，并运用到世界各地的项目中。Java 语言在 Web 领域的造诣颇深，已经成为开发动态 Web 站点的核心技术之一。为什么 Java 语言的使用率如此之高？Java Web 究竟具有什么优势？希望读者仔细阅读本章的内容，相信读完之后会对 Java Web 有一个深层次的认识，使相关的疑问获得解答。





1.1 学习 Java 的巨大优势

Java 语言在编程语言中占据了一个重要地位，用 Java 语言开发的项目分布在世界各地的各个领域。为了让读者更加有信心地学好 Java 语言，本节我们首先来介绍学习 Java 语言的优势。

1.1.1 排名第一的编程语言

都说 Java 语言是使用率最高的一门编程语言。下面的表 1-1 是某权威机构做出的编程语言排行榜，信息截止于 2012 年 2 月。

表 1-1 编程语言排行榜

排 名	语 言	使用率(%)
1	Java	17.050
2	C	16.523
3	C#	8.653
4	C++	7.853
5	Object-C	7.062
6	PHP	5.641

1.1.2 良好的就业前景

由表 1-1 的统计数据可以看出，Java 语言是当今使用率最高的编程语言。使用率如此之高，也决定了 Java 程序员的就业机会要大于其他开发者。

Java 的功能比较强大，在服务器领域、移动设备、桌面应用和 Web 领域都占据了重要的地位。

- 服务器领域：Java 在服务器编程方面很强悍，拥有很多其他语言所没有的优势。
- 移动设备：Java 在手机领域的应用比较广泛，手机 Java 游戏随处可见，当前比较知名的移动开发平台 Android 上面的应用项目基本上都是用 Java 开发的。
- 桌面应用：Java 和 C++、C# 一样重要，影响着桌面程序的发展。
- Web 领域：Java Web 有着巨大的优势，无论是开发工具还是开发框架，都是开源的，并且安全性更强。

正是因为 Java 语言可以在多个领域开发项目，所以市面中需要多个领域的 Java 程序员。据统计，当前全球有 30 亿 Java 器件正在运行着 Java 程序，500 多万 Java 开发者活跃在地球上的每个角落，数以千万计的 Web 用户每次上网都亲历 Java 的威力。今天，Java 运行在 9 亿部手机、10 亿张智能卡和 10 亿台 PC 机上，并为 28 款可兼容的应用服务器提供了功能强大的平台。这么多应用，彻底改变了用户的生活。

越来越多的企业，因为使用了 Java 而提高了生产效率。在中国，越来越多的用户，因为 Java 而降低了成本，享受了生活。

作为唯一在互联网上开发的语言，Java 平台以其移动性、安全性和开放性受到追捧。据 IDC 预计，在自 2010 年起的其后 5 年内，采用 Java 的 IT 产品的价值将翻番，在 2015 年将达到 45.53 亿美元，年增长率为 14.9%。截止到 2010 年 5 月，Java 注册开发商超过 1300 万人，对 JRE(Java 运行环境)的下载达 9200 万次。詹姆斯·戈士林博士预计在 3~5 年内 Java 技术开发商将发展到 2000 万，无线 Java 也在迅速攀升。

上述发展趋势在国内更是如此，当前我国对软件人才的新增需求已达很高的数额，并且以每年 20%左右的速度增长。在未来 5 年内，合格软件人才的需求将远大于供给。在过去的 2011 年，我国软件人才的缺口已达 52.5 万人，其中尤以 Java 人才最为缺乏。

根据 IDC 的统计数字，在所有软件开发类人才的需求中，对 Java 工程师的需求达到全部需求量的 60%~70%。这也就不难解释在智联招聘和 51Job 等主流招聘站点中，为何随处可见 Java 工程师的招聘广告了。

1.2 学习 Java 需要了解的那些事

了解了学习 Java 语言的优势之后，接下来需要了解与这门神奇语言相关的几件大事，我们分别介绍该语言的发展历史和特点等知识，为读者步入本书后面知识的学习打下牢固的理论基础。

1.2.1 Java 语言的发展历史

Java 是由 Sun Microsystems 公司于 1995 年 5 月推出的 Java 程序设计语言(以下简称 Java 语言)和 Java 平台的总称。用 Java 实现的 HotJava 浏览器(支持 Java Applet)向我们展示了 Java 语言的魅力——跨平台、动态的 Web、Internet 计算。Java 自从推出之日起，便被广大程序员和企业用户广泛接纳，成为最受欢迎的编程语言之一。

Java 平台由 Java 虚拟机(Java Virtual Machine)和 Java 应用编程接口(Application Programming Interface, API)构成。Java 应用编程接口为 Java 应用提供了一个独立于操作系统的标准接口，可分为基本部分和扩展部分。在硬件或操作系统平台上安装了 Java 环境之后，Java 应用程序就可运行。现在 Java 平台已经嵌入了几乎所有的操作系统。这样 Java 程序只编译一次，就可以在各种系统中运行。Java 应用编程接口已经从 1.1x 版发展到 1.2 版。目前常用的 Java 平台是基于 Java 1.6 的，最新的版本为 Java 1.7。

当 1995 年 Sun 公司推出 Java 语言之后，全世界的目光都被这个神奇的语言所吸引。那么 Java 到底有何神奇之处呢？Java 语言其实最早诞生于 1991 年，起初被称为 OAK 语言，是 Sun 公司为一些消费性电子产品而设计的一个通用环境。Sun 公司的最初目的是为了开发一种独立于平台的软件技术，而且在网络出现之前 OAK 是默默无闻的，甚至差一点夭折。但是随着网络的出现，彻底改变了 OAK 的命运。在 Java 出现以前，Internet 上的信息内容都是一些乏味死板的 HTML 文档。这对于那些迷恋于 Web 浏览的人们来说简直



不可容忍。他们迫切希望能在 Web 中看到一些交互式的内容,开发人员也非常希望能够在 Web 上创建一类无需考虑软硬件平台就可以执行的应用程序,当然这些程序还要有极大的安全保障。对于用户的这种需求,传统的编程语言显得无能为力。Sun 公司的工程师敏锐地察觉到了这一点,从 1994 年起,他们开始将 OAK 技术应用于 Web 上,并且开发出了 HotJava 的第一个版本。并最终在 1995 年,将 Java 技术展现在了世人的面前。

Java 分为如下三个体系。

- Java SE: 是 Java 2 Platform Standard Edition 的缩写,即 Java 平台标准版。
- Java EE: 是 Java 2 Platform Enterprise Edition 的缩写,即 Java 平台企业版。
- Java ME: 是 Java 2 Platform Micro Edition 的缩写,即 Java 平台微型版。

2009 年 4 月 20 日,Oracle(甲骨文)宣布成功收购 Sun 公司,从那以后,Java 成为了软件巨头甲骨文旗下的一款产品,并与 Oracle 数据库一起推动着世界科技继续向前发展。

1.2.2 Java 的特点

Java 具备下列特点。

- 面向对象: Java 语言提供了类、接口和继承等特性。为了简单起见,Java 只支持类之间的单继承和接口之间的多继承,并且也支持类与接口之间的实现机制。总之,Java 语言是一门纯粹面向对象的程序设计语言。
- 简单: Java 语言的语法与 C 语言和 C++ 语言十分接近,这样大多数程序员可以很容易学习和使用 Java。并且 Java 抛弃了 C++ 中的操作符重载、多继承、自动强制类型和指针等做法,这将更有利于我们学习和掌握。另外,Java 还提供了自动垃圾收集机制,使得程序员不必再为内存管理而担忧。
- 分布式: Java 语言支持 Internet 应用开发,在基本的 Java 应用编程接口中有一个网络应用编程接口(java.net),这个接口提供了网络应用编程的类库,包括 URL、URLConnection、Socket、ServerSocket 等。Java 的 RMI(远程方法激活)机制也是开发分布式应用的重要手段。
- 稳定: Java 的强类型机制、异常处理、垃圾的自动收集等是 Java 程序稳定性的重要保证。Java 通过安全检查机制,使 Java 程序更具稳定性。
- 可移植: 移植性是指能够在不同的开发平台和服务器平台上使用。因为 Java 的运行环境是用 ANSI C 实现的,所以 Java 系统本身具有很强的可移植性,可以在很多平台上运行。无论是微软的产品还是 IBM 的产品,都可以运行 Java 程序。
- 高性能: 与解释型的高级脚本语言相比,Java 的确是高性能的。随着 JIT(Just-In-Time)编译器技术的发展,Java 的运行速度已经越来越接近于 C++。
- 多线程: 当程序需要同时处理多项任务时,就需要多线程开发。一个程序在同一时间只能做一件事情的功能过于简单,肯定无法满足现实的需求。在实际的应用中,多线程开发是必不可少的,多线程的目的是在同一时间可以做多件事情。并且可以开启多个线程同时做一件事情,这样可以提高效率。不管是 C 语言、C++ 语言还是其他的程序设计语言,线程都是一个十分重要的知识点,多线程是现代开发软件系统的发展方向,Java 作为当今的主流程序设计语言,当然是支持多线程

程的，具有并发性，其执行的效率很高。

- 动态：Java 语言的设计目标之一是适应于动态变化的环境。Java 程序中的类需要能够动态地被载入到运行环境中，也可以通过网络来载入所需要的类。动态语言的好处是有利于软件升级，另外，Java 中的类有一个运行时刻的表示，能进行运行时刻的类型检查。

1.2.3 Java 的平台无关性

Java 体系结构通过很多方面提供对平台无关性的支持。其中 Java 平台是最主要的方面。Java 平台扮演了一个运行时 Java 程序与其下操作系统(OS)和硬件之间的缓冲角色。换个角度看，Java 平台实际上充当了一个标准的 OS，只是它目前看起来还过于笨重。Java 编程语言本身对平台无关性也提供了相应的支持，但是这是建立在 Java 平台基础上的。对于 C/C++程序员来说，最明显的一个例子就是对基本数据类型的值域和行为的定义上。以前一直强调不同的平台、不同的机器、不同的 CPU 都有不同的字节长度定义，但在 Java 中，已经不存在这种差别了，因为 Java 语言统一了这些。至于究竟实际的硬件是怎么工作的，对于 Java 程序员来说，已经不用去关心了。Java 平台已经自动地完成了这种转换。

另外一个对平台无关性提供支持的方面是 Java 的 Class 文件格式，它对于所有的 Java 虚拟机都是标准统一的，这也为跨平台传递程序代码提供了实现。所以，从总体上来看，Java 体系结构对平台无关性的支持，主要来自于 Java 平台的设计，相当于为不同的硬件和操作系统提供了一层标准的外壳，Java 体系结构将操作建立在一个虚拟的平台上，也就达到了平台无关的目的，虽然同时也带来了很多的缺点。但是这种趋势是不可阻挡的。

说到平台无关性，不可忽略的是 Java 不仅在计算机领域实现了跨平台，而且在智能设备领域也实现了跨平台。这是通过定义不同的虚拟机标准来实现的。也就是我们耳熟能详的 J2EE、J2SE、J2ME。实际上，这三个 Java 版本就是在不同的智能设备上的虚拟机规范。而在 J2ME 领域，平台无关性体现得就更明显了，因为移动/嵌入式设备的厂商和标准都比计算机领域多得多，统一的难度也就大得多了。所以在 J2ME 领域，除了有 J2ME 这个大框架，还针对各个更细小的领域定义了各自的虚拟机标准，这些标准被称作 profile。

虽然平台无关性看起来是那么的诱人，但是在编写 Java 程序的时候，平台无关性只是一个选项，并不是一个必选项。从根本上来说，任何 Java 程序的平台无关程度都依赖于作者怎么编写它。有如下几个因素可以影响平台无关性。

(1) Java 平台的部署

Java 平台的平台无关特性决定了要运行 Java 程序就必须在客户机上安装符合客户操作系统的 Java 平台，也就是 Java 虚拟机。这并不是每一台计算机或每一种操作系统都能满足的条件。但幸运的是，Java 已经得到了广泛的推广，所以这个问题在大部分客户那里已经得到了解决。

(2) Java 平台的版本

这个问题主要集中在 Java 不同版本的 Java API 上。虽然 Java 平台比较稳定，但是 Java API 的改动是比较频繁的，这种改动会导致新版本的 Java 程序不能在旧平台上运行，这也是开发语言的通病。



(3) 本地方法

当 Java 要用到一些操作系统平台的特性, 需要使用本地方法的时候, 就会破坏平台无关性。Sun 公司一直在大力推广纯 Java 程序, 就是想通过程序员的努力, 尽量避免使用本地方法。

(4) 非标准运行时库

虽然本地方法会破坏平台无关性, 但是 Java API 提供的标准方法中使用的本地方法却不在其列。在标准 API 之外, 还有一些厂商和组织提供的非标准方法, 这些方法有可能使用了本地方法, 却不是任何 Java 平台都拥有的, 所以对于这种非标准方法, 一定要谨慎使用, 如果它们使用了本地方法来实现, 那么平台无关性就被破坏了。

(5) 对虚拟机的依赖

在分析 Java 虚拟机的时候, 需要牢记的一点是, 虚拟机不过是 Sun 提供的一个文字标准, 并不是具体的实现, 我们平时用的虚拟机是 Sun 提供的, 但这并不是标准的虚拟机实现, 实际上虚拟机并没有标准的实现, 唯一的标准就是 Sun 给出的那份标准。所以对于虚拟机标准提供的一些特性, 千万不能作为程序逻辑的一部分。因为标准只是指出一定要实现这些特性, 但是具体怎么实现, 是由具体的虚拟机实现决定的。一些可以举例的特性是垃圾回收机制、线程优先级等。它们被实现了, 但并没有一个确定的实现方式, 比如垃圾回收的时机, 线程优先级高低是否一定决定了 CPU 时间分配的多少等。

(6) 对用户界面的依赖

在不同的 Java 平台上, 要保证每个平台的 UI 都让用户满意是很困难的事情。所以在跨平台设计时, 要谨慎地设计 UI 的接口, 以保证在每个平台上都有令人满意的表现。

1.3 Web 技术介绍和工作原理

网站的原理很简单, 只需要看明白图 1-1 即可。

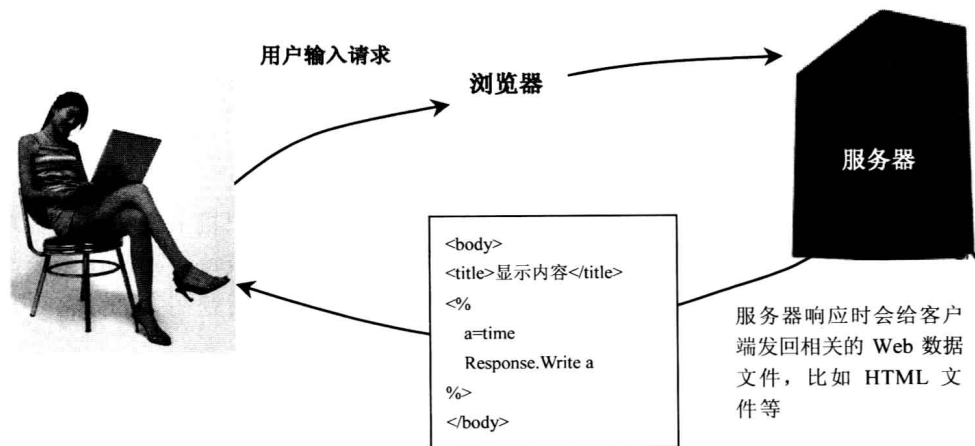


图 1-1 本地计算机和远程服务器的工作流程