

21世纪高等教育计算机规划教材



Java Web 程序设计教程

Java Web Programming Tutorial

■ 陈沛强 主编

■ 谷灵康 金京犬 副主编

- 符合大学生学习特点，由浅入深，通俗易懂
- 覆盖 Java Web 开发内容，全面广泛，案例丰富
- 项目实例贯穿全书，突出实战，实现教、学、练一体



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

21世纪高等教育

COMPUTER

Java Web 程序设计教程

Java Web Programming Tutorial

■ 陈沛强 主编

■ 谷灵康 金京犬 副主编



人民邮电出版社

北京

图书在版编目 (CIP) 数据

Java Web程序设计教程 / 陈沛强主编. — 北京 :
人民邮电出版社, 2016.7
21世纪高等教育计算机规划教材
ISBN 978-7-115-42322-1

I. ①J… II. ①陈… III. ①JAVA语言—程序设计—
高等学校—教材 IV. ①TP312

中国版本图书馆CIP数据核字(2016)第089946号

内 容 提 要

本书作为 Java Web 程序设计的大学教程, 对 Java EE 编程技术进行了系统全面的介绍。全书共分 14 章。首先对 Java 高级编程知识做了详细的介绍, 包括 Java 常用工具类、集合框架、JDBC 编程技术以及 Java 对 XML 编程技术。然后开始对 JSP 网页编程技术进行较详细的介绍, 包括 HTML 基础、CSS 样式表的应用、JavaScript 脚本语言以及 JSP 相关的编程技术(如 JSP 基础、JavaBean 编程、Servlet 编程以及 Filter 等), 还介绍了 EL 和 JSTL 编程技术。最后介绍了本书的高级部分, 也就是目前流行的开源框架, 包括 Struts2 框架、Hibernate 技术、Hibernate 高级应用、Spring 框架、Spring 与 Struts2、Hibernate 框架的整合和 JQuery 等。本书以一个完整的综合实例项目——网上购书系统作为结束, 有助于学生理解知识、应用知识, 达到学以致用目的。

全书每章内容都与开发技巧、经验和实例紧密结合, 并配有习题, 习题中的上机实践题可作为上机编程实验课教学内容。本书源代码全部经过精心测试, 能够在 Windows XP、Windows 7、Windows 8 和 Windows 10 系统下编译和运行。

本书结构合理, 通俗易懂, 内容深入浅出, 由于编写时考虑了大学生和 Java EE 编程初学者的特点, 所以本书适合作为应用型本科院校、软件学院、独立学院软件专业及相关专业的教材, 同时也适合 Java Web 爱好者学习使用。

-
- ◆ 主 编 陈沛强
 - 副 主 编 谷灵康 金京犬
 - 责任编辑 吴 婷
 - 责任印制 沈 蓉 彭志环
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京昌平百善印刷厂印刷
 - ◆ 开本: 787×1092 1/16
印张: 19.75
字数: 520 千字

2016 年 7 月第 1 版

2016 年 7 月北京第 1 次印刷

定价: 48.00 元

读者服务热线: (010)81055256 印装质量热线: (010)81055316

反盗版热线: (010)81055315

前言

Java EE 编程技术是目前流行的开发技术。Java EE 是开放的框架。随着对 Java EE 平台企业版第三方支持的增多, Java EE 成为开发企业级服务器端解决方案的首选平台之一。Java EE 包含 JDBC、JSP、Servlet、JavaBean、EJB 以及基于 Java 的开源技术等。作者结合多年 Java EE 编程与教学实践, 认为一个很好的学好 Java EE 的途径应该是先熟悉 Java 高级编程, 接着学习 JSP 网页编程知识, 再进一步学习基于 Java 的一些重要的开源框架。本书编排正是基于这一思路。

在当前的教育体系下, 实例教学是计算机语言教学的最有效的方法之一, 本书将 Java 知识和实用的实例有机结合起来。一方面, 跟踪 Java 语言的发展, 适应市场需求, 精心选择内容, 突出重点、强调实用, 使知识讲解全面、系统; 另一方面, 全书通过“案例贯穿”的形式, 尽可能围绕一个统一的案例设计实例, 但又不仅仅拘泥于一个案例, 这样可以丰富案例的内容和形式。如此将实例融入到知识讲解中, 使知识与案例相辅相成, 即有利于学生学习知识, 又有利于指导学生实践。另外, 本书在每一章的后面还提供了上机指导和习题, 方便读者及时验证自己的学习效果(包括动手实践能力和理论知识)。

目前在高校 Java EE 教学中没有很合适的教材。很多教材只是针对 Java EE 的某个领域, 不是综合的, 或者技术描述过于抽象复杂, 语言晦涩难懂, 学生不好理解。本书内容丰富, 综合了常见的 Java EE 知识并穿插相应的开发技巧、经验, 结构合理, 通俗易懂, 内容深入浅出, 适合高校教学; 同时本书对想从事 Java EE 编程的人也是一本很好的入门书、参考书。

读者使用本书, 首先要弄清楚书上介绍的基本知识, 理解基本原理, 然后按照书上的例题进行独立调试(书上所有示例都已调试通过)。另外, 建议读者还需要多做实验, 编程离不开动手实践, 实验时肯定会出现各种各样的问题, 多调试, 只有如此你才能快速领会 Java EE 的编程精要所在。最后, 还需要对基本知识进行扩展, 学习本书之后, 就可以顺利阅读市面上那些著名的 Java EE 专业技术书了。同时, 再多参考学习互联网上的最新知识, 你的 Java EE 编程技能必将得到大步提高。

本书作为教材使用时, 课堂教学建议 40~48 学时, 上机指导教学建议 16~24 学时。各章主要内容和学时分配如下表所示, 老师可以根据实际教学情况进行调整。

学时分配表

章	主要内容	课堂学时	上机指导
第 1 章	Java EE 框架概述	1	
第 2 章	常用工具类的使用	2	1
第 3 章	Java 集合框架以及泛型编程	2	1
第 4 章	JDBC 编程技术	4	2
第 5 章	Java 对 XML 编程技术	2	1
第 6 章	网页编程基础知识 html、CSS、JavaScript	2	1
第 7 章	JSP 编程技术, 包括 JSP、Servlet、JavaBean 以及过滤器 Filter 编程技术。 这是 JSP 网页编程的重点	6	4
第 8 章	EL 表达式语言与 JSTL 标签库的使用	2	1

续表

章	主要内容	课堂学时	上机指导
第 9 章	Hibernate 技术, 包括 Hibernate 简介、Hibernate 数据持久化、Hibernate 的缓存、Hibernate 高级应用, 包括关联关系映射、HQL 检索方式	4	2
第 10 章	Struts2 框架, 包括 MVC 设计模式、Struts2 框架概述、Action 对象、Struts2 的配置、Struts2 的标签库、Struts2 的开发模式、Struts2 的拦截器	4	2
第 11 章	Spring 框架, 包括 Spring 概述、Spring Ioc、AOP 概述、Aspect、Spring 持久化	4	2
第 12 章	Spring 与 Struts2、Hibernate 框架的整合, 包括框架整合的优势分析、SSH2 架构分析、如何构建 SSH2 框架、SSH2 实例程序部署	4	2
第 13 章	JQuery 编程及 AJAX 编程技术介绍	2	2
第 14 章	综合案例——网上购书系统, 包括需求分析、总体设计、数据库设计、公共类设计、系统主要模块开发、运行项目、小结	4	

本书由编者在多年的课程教学经验和素材的积累基础上编写。

编者

2016 年 2 月

目 录

第 1 章 Java EE 框架概述.....1

1.1 C/S 架构与 B/S 架构.....1	
1.1.1 C/S 架构.....1	
1.1.2 B/S 架构.....1	
1.2 什么是 Java EE.....2	
1.2.1 Java EE 规范简介.....2	
1.2.2 Java EE 平台主要内容.....3	
1.2.3 Java EE 应用服务器软件及 相关角色.....4	
1.2.4 Java EE 应用软件的体系结构.....5	
1.3 Java EE 能做什么.....6	
习题.....6	

第 2 章 常用工具类的使用.....7

2.1 String 与 StringBuffer 类的使用.....7	
2.1.1 String 类.....7	
2.1.2 StringBuffer 类.....9	
2.2 Java 日历类的使用.....10	
2.2.1 Date 与 DateFormat 的使用.....10	
2.2.2 Calendar 日历类使用.....12	
2.3 Java 定时器 Timer 类使用.....14	
本章小结.....15	
习题.....15	

第 3 章 Java 集合框架及泛型编程.. 16

3.1 Java 集合概念.....16	
3.2 Java 集合的使用.....17	
3.2.1 HashSet 使用.....17	
3.2.2 TreeSet 使用.....19	
3.2.3 ArrayList 使用.....23	
3.2.4 Map 的使用.....24	
3.2.5 中文排序问题.....27	

3.3 Java 泛型编程.....29	
本章小结.....30	
习题.....30	

第 4 章 JDBC 编程技术..... 32

4.1 MySQL 数据库.....32	
4.1.1 MySQL 服务器的安装.....32	
4.1.2 MySQL 的环境和命令.....33	
4.2 JDBC 编程基本概念.....34	
4.2.1 JDBC 基本概念.....34	
4.2.2 JDBC 数据库编程基本步骤.....34	
4.2.3 完整示例.....38	
4.3 JDBC 编程进阶.....40	
4.3.1 PreparedStatement 研究.....40	
4.3.2 如何获得元数据 MetaData.....42	
4.3.3 事务处理.....44	
4.4 数据库分层设计.....46	
4.4.1 O/R 映射.....46	
4.4.2 实战——客户信息系统分层设计...46	
本章小结.....50	
习题.....51	

第 5 章 Java 对 XML 编程..... 53

5.1 XML 基本概念.....53	
5.1.1 XML 文档结构.....53	
5.1.2 XML 基本元素.....54	
5.1.3 使用属性.....55	
5.1.4 XML 解析.....56	
5.2 利用开源 JDOM 对 XML 编程.....57	
5.2.1 基本构成.....58	
5.2.2 入门示例.....58	
5.2.3 实战——读写 XML 文档.....60	
本章小结.....61	

习题	62	7.2.6 pageContext 对象	109
第 6 章 网页编程技术	64	7.2.7 其他内置对象 (page、config、exception)	110
6.1 Web 基础	64	7.2.8 Cookie 对象的使用	110
6.1.1 浏览器	64	7.3 JavaBean 编程	111
6.1.2 Web 服务器	64	7.3.1 JavaBean 概述	111
6.1.3 HTTP	65	7.3.2 实战——JavaBean 数据库综合编程	113
6.2 HTML 基本概念和基本标签	67	7.4 Servlet 编程技术	117
6.3 CSS 使用	69	7.4.1 Servlet 概述	117
6.3.1 CSS 概念	69	7.4.2 Servlet 生命周期	118
6.3.2 CSS 层叠样式表用法	70	7.4.3 实战——Servlet 编程	118
6.3.3 CSS 的常用选择器	71	7.4.4 Servlet 初始化函数	120
6.3.4 CSS 属性	75	7.4.5 Servlet 3.0 的新特性	121
6.4 利用 CSS 与 DIV 网页布局	76	7.5 过滤器 Filter 编程	121
6.5 JavaScript 编程基础	80	7.5.1 Filter 概述	121
6.5.1 面向对象和 JavaScript	81	7.5.2 实战——中文乱码过滤器编程	122
6.5.2 基本形式	81	7.5.3 Filter 配置	123
6.5.3 在什么地方插入 JavaScript	81	7.6 JSP 编程常见技巧	124
6.5.4 JavaScript 基本语法	81	7.6.1 实战——JSP 验证码实现	124
6.5.5 JavaScript 对象	85	7.6.2 实战——JSPSmartUpload 实现文件上传下载	127
6.5.6 JavaScript 与 XHTML 文档	88	本章小结	129
6.5.7 JavaScript 的事件与事件处理	89	习题	129
6.5.8 实战——验证表单输入	92		
本章小结	93	第 8 章 EL 表达式与 JSTL 库	131
习题	93	8.1 EL 表达式	131
第 7 章 JSP 编程技术	95	8.1.1 JSP 中 EL 表达式	131
7.1 JSP 编程基础	95	8.1.2 JSP 中 EL 表达式输出某个范围变量值	132
7.1.1 JSP 运行环境配置	95	8.1.3 EL 运算符	134
7.1.2 JSP 的执行过程	98	8.1.4 EL 输出 JavaBean 中属性值	134
7.1.3 JSP 脚本元素的组成	98	8.2 JSTL 标签库使用	136
7.1.4 JSP 注释	99	8.2.1 JSTL 基本概念	136
7.1.5 JSP 常见指令	100	8.2.2 JSTL 编程入门	137
7.1.6 JSP 常见动作元素	102	8.2.3 JSTL 核心标签库	138
7.2 JSP 常见内置对象	103	8.3 实战——客户信息系统客户页面编辑	142
7.2.1 out 对象	103	本章小结	145
7.2.2 request 对象	104	习题	145
7.2.3 response 对象	106		
7.2.4 application 对象	107		
7.2.5 session 对象	108		

第 9 章 Hibernate 编程技术..... 147

9.1	Hibernate 架构与入门	147
9.1.1	O/R Mapping.....	147
9.1.2	Hibernate 架构.....	148
9.1.3	Hibernate 的工作原理	148
9.1.4	实战——Hibernate 实现客户信息 系统操作.....	149
9.1.5	Hibernate 核心接口	153
9.2	Hibernate 常见操作	155
9.2.1	利用 Hibernate 增、删、改记录 ...	155
9.2.2	Hibernate 主键 ID 生成方式	156
9.2.3	Hibernate 查询方式	159
9.3	Hibernate 多表操作	162
9.3.1	表之间关系.....	162
9.3.2	实战——Hibernate 一对多关系操作 ...	162
9.3.3	级联操作与延迟加载	165
9.3.4	实战——Hibernate 多对多关系 操作.....	169
	本章小结	171
	习题	172

第 10 章 Struts2 编程..... 173

10.1	B/S 设计模式	173
10.1.1	MVC 模式.....	173
10.1.2	MVC 分层架构.....	174
10.2	Struts 概念.....	176
10.2.1	Struts 2 体系结构.....	177
10.2.2	Struts2 框架的处理流程.....	178
10.2.3	实战——Struts2 实录登录操作 ...	178
10.3	深入理解 Struts 2 的配置文件	181
10.4	Action	184
10.4.1	Action 类文件	184
10.4.2	Action 动态处理函数	185
10.4.3	Action 访问 Servlet API	186
10.5	Struts 2 校验框架.....	187
10.5.1	Struts2 校验流程.....	187
10.5.2	Struts 常见校验规则.....	190
10.5.3	实战——Struts 2 中应用客户端输 入校验.....	191

10.6	Struts 2 拦截器	192
10.6.1	什么是拦截器	192
10.6.2	Struts 2 拦截器入门.....	194
10.6.3	在 Struts 2 中配置自定义的 拦截器	196
10.7	Struts 2 转换器	197
10.7.1	在 Struts 2 中配置类型转换器	197
10.7.2	实战——类型转换器应用	199
10.8	Struts 2 国际化	201
10.9	实战——Struts 2 上传下载	203
10.9.1	上传文件.....	204
10.9.2	文件下载.....	206
10.10	Struts 2 标签	207
10.10.1	模板和主题.....	207
10.10.2	Struts 2 常用 UI 标签使用.....	208
10.10.3	实战——Struts 2 常用 UI 标签综 合示例	210
10.10.4	Struts 2 常用非 UI 标签使用.....	211
	本章小结	214
	习题	214

第 11 章 Spring 编程.....215

11.1	Spring 开源框架.....	215
11.2	Spring 入门示例.....	217
11.3	Spring IoC 控制反转	219
11.3.1	Spring 依赖注入	220
11.3.2	Spring Bean 的作用域	224
11.3.3	Spring 自动装配	225
11.4	Spring AOP 编程	226
11.4.1	AOP 概念.....	226
11.4.2	Aspect 对 AOP 的支持	228
11.4.3	实战——AOP Spring	228
	本章小结	231
	习题	231

**第 12 章 Spring、Struts2、Hibernate
整合.....232**

12.1	实战——Spring 与 Hibernate 整合（客 户信息系统查询）	232
12.2	事务处理	236

12.2.1 实战——通过注释实现事务	236
12.2.2 声明式事务	237
12.3 Spring 与 Struts 整合	240
12.4 实战——SS2H 整合（客户信息系统 用户注册）	243
本章小结	249
习题	249

第 13 章 基于 JQuery 编程技术 .. 250

13.1 JQuery 简介	250
13.2 JQuery 配置与使用	251
13.3 JQuery 选择器	252
13.4 JQuery 对 HTML 操作	256
13.4.1 节点标签操作	256
13.4.2 CSS 样式操作	259
13.4.3 读写 HTML 文本	259
13.5 JQuery 事件	260
13.5.1 绑定事件	260
13.5.2 事件冒泡	261

13.6 基于 JQuery 的 AJAX 编程	263
13.6.1 什么是 AJAX	263
13.6.2 实战—AJAX 实现获取音乐 列表	264
13.6.3 JQuery 的 AJAX 编程技术	265
13.6.4 JQuery 中使用 JSON	269
本章小结	270
习题	271

第 14 章 综合实例——网上购书 系统

14.1 系统分析和设计	272
14.2 注册、登录和注销	280
14.3 图书类别及新书展示	291
14.4 显示指定类别图书	295
14.5 图书查询	300
14.6 添加图书到购物车	302
14.7 显示购物车	305
14.8 结账	307

第 1 章

Java EE 框架概述

【学习目标】

- 什么是 Java EE
- Java EE 能做什么

1.1 C/S 架构与 B/S 架构

1.1.1 C/S 架构

C/S (Client/Server) 结构, 即大家熟知的客户机和服务器结构, 如图 1.1 左图所示。它是软件系统体系结构, 通过它可以充分利用两端硬件环境的优势, 将任务合理分配到 Client 端和 Server 端来实现, 降低了系统的通信开销。

C/S 架构具有以下两个优点。

- (1) 可以充分发挥客户端 PC 的处理能力, 即所谓的“胖”客户端;
- (2) 响应速度快。

其缺点如下。

- (1) 只适用于局域网;
- (2) 客户端需要安装专用的客户端软件, 维护成本高;
- (3) 对客户端的操作系统一般也会有限制。

1.1.2 B/S 架构

B/S (Browser/Server) 架构即浏览器和服务器架构, 如图 1.1 右图所示。它是随着 Internet 技术的兴起, 对 C/S 结构的一种变化或者改进的结构。

在这种结构下, 用户工作界面是通过浏览器来实现, 极少部分事务逻辑在前端 (Browser) 实现, 主要事务逻辑在服务器端 (Server) 实现。

B/S 架构正是针对 C/S 架构缺点的改进架构, 因此具有以下 3 个优点。

- (1) 多数业务逻辑存在于服务器端, 减轻了客户端的负荷, 属于一种“瘦”客户端;
- (2) 客户端无须安装专门软件, 更新维护均在服务器端;
- (3) 适于移动、分布式处理。

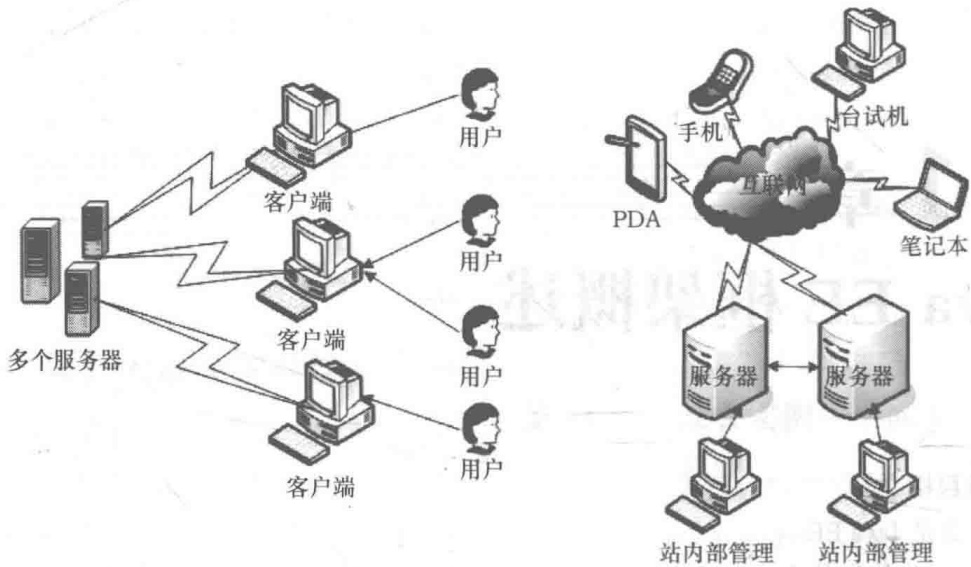


图 1.1 C/S 和 B/S 架构

1.2 什么是 Java EE

1.2.1 Java EE 规范简介

Java EE (Java Platform Enterprise Edition) 是 Sun 公司推出的企业级应用程序版本, 现已被 Oracle 公司收购接管, 目前 Java EE 的版本已经推出 Java EE8, 本书采用的是使用较为广泛的 Java EE 5 版本。

Java EE 既不是一门编程语言, 也不是一个现成的产品, 而是一个标准, 是一个为企业分布式应用的开发提供的一套规范和标准的平台, 帮助企业开发和部署可移植、健壮、可伸缩且安全的服务器端 Java 应用程序。

Java EE 建立在 Java SE 基础上, 用来实现企业级的面向服务体系结构 (SOA) 和 Web 2.0 应用程序。SOA 是面向服务的体系结构, 是一个组件模型, 它将应用程序的不同功能单元 (称为服务) 通过这些服务之间定义良好的接口和契约联系起来。接口是采用中立的方式进行定义的, 它应该独立于实现服务的硬件平台、操作系统和编程语言。这使得构建在各种各样的系统中的服务可以使用一种统一和通用的方式进行交互。Web 2.0 是相对 Web 1.0 的新的时代, 指的是一个利用 Web 的平台, 由用户主导而生成内容互联网产品模式, 为了区别传统由网站雇员主导生成的内容而定义为第二代互联网, 即 Web 2.0, 如个人网站和博客等。

Java EE 包括 JDBC、JNDI (Java 命名和目录接口)、JMS (Java Message Service)、EJB、JSP、JavaBean 和 Servlet 等技术。

推出 Java EE 是为了克服传统 C/S 模式的弊病, 迎合 B/S 架构的潮流, 从而简化企业应用的开发、管理和部署。

作为一个平台, Java EE 指的是使用 Java 编程语言编写的应用程序的运行环境。Java 平台包括以下 3 类。

- (1) Java SE (Java Platform, Standard Edition): Java 标准版。

(2) Java EE (Java Platform, Enterprise Edition): Java 企业版。

(3) Java ME (Java Platform, Micro Edition): Java 微型版。

Java EE 不仅仅是指一种标准平台，更表达了一种软件架构和设计思想。

1.2.2 Java EE 平台主要内容

Java EE 平台是由一系列容器、应用组件和 API 服务所组成, 如图 1.2 所示。

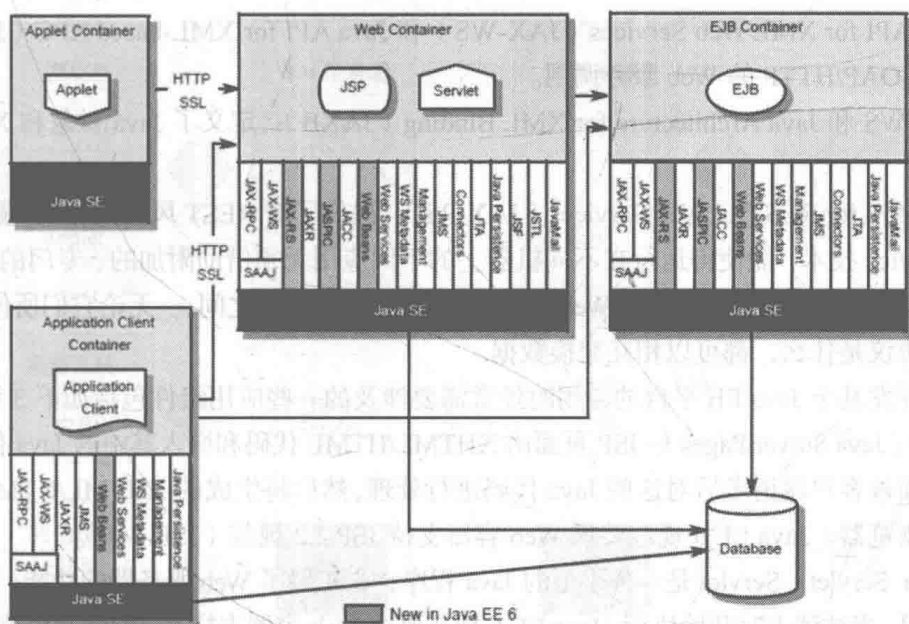


图 1.2 Java EE 平台架构

容器是指为各种应用组件提供 API 服务的 Java EE 运行时的环境,可提供诸如目录服务、事务管理、安全性、资源缓冲池以及容错性等各种公共服务,包括应用客户端容器、Applet 容器、Web 容器和 EJB 容器 4 种。

首先，开发 Java EE 应用涉及的 API 服务包括以下几种。

(1) JDBC (Java Database Connectivity, Java 数据库连接): 一种用于执行 SQL 语句的 Java API, 可为访问不同的关系型数据库提供一种统一的途径。

(2) JNDI (Java Name and Directory Interface, Java 命名和目录接口): JNDI 被用于执行名字和目录服务。它提供了一致的模型来存取和操作企业级的资源, 如 DNS、LDAP、本地文件系统或应用服务器中的对象。

(3) RMI (Remote Method Invoke, 远程方法调用): RMI 定义了调用远程对象上的方法的标准接口。作为一种被 EJB 使用的更底层的协议,它通过使用序列化方式在客户端和服务端传递数据。

(4) Java IDL/CORBA: Java IDL 使得 Java EE 应用组件可通过 IIOP 协议调用外部的可用各种编程语言开发的 CORBA 对象, 从而实现不同应用系统之间的集成。

(5) JMS (Java Message Service, Java 消息服务): JMS 是用于与消息中间件相互通信的应用程序接口。它既支持点对点的消息模型, 也支持发布/订阅的消息模型。Java EE 6 规范要求支持 JMS 1.1 规范 (JSR 914)。

(6) JTA (Java Transaction Architecture, Java 事务架构) 定义了面向分布式事务服务的标准 API, 可支持事务范围的界定、事务的提交和回滚。

(7) JavaMail: JavaMail 用于存取邮件服务器的 API, 提供了一套可访问邮件服务器的抽象类。

(8) JAF (JavaBeans Activation Framework, JavaBeans 激活框架): JavaMail 利用 JAF 来处理 MIME 编码的邮件附件。通过 JAF, MIME 的字节流可以被转换成 Java 对象, 或者转换自 Java 对象。

(9) Web 服务: Java EE 平台通过多种技术提供了对 Web 服务的支持, 包括如下 3 种。

- Java API for XML Web Services (JAX-WS) 和 Java API for XML-based RPC (JAX-RPC): 可支持基于 SOAP/HTTP 的 Web 服务调用。

- JAX-WS 和 Java Architecture for XML Binding (JAXB): 定义了 Java 对象和 XML 数据之间的映射。

- Java API for RESTful Web Services (JAX-RS): 提供了对 REST 风格的 Web 服务的支持。

Web Service 技术, 能使得运行在不同机器上的不同应用无须借助附加的、专门的第三方软件或硬件, 就可相互交换数据或集成。Web Service 规范实施的应用之间, 无论它们所使用的语言、平台或内部协议是什么, 都可以相互交换数据。

其次, 开发基于 Java EE 平台的应用时经常需要涉及的一些应用组件包括如下 3 种。

(1) JSP (Java Server Pages): JSP 页面由 XHTML/HTML 代码和嵌入其中的 Java 代码所组成。服务器在页面被客户端请求后对这些 Java 代码进行处理, 然后将生成的 XHTML/HTML 页面返回给客户端的浏览器。Java EE 6 规范要求 Web 容器支持 JSP 2.2 规范 (JSR 245)。

(2) Java Servlet: Servlet 是一种小型的 Java 程序, 它扩展了 Web 服务器的功能。作为一种服务器端的应用, 当被请求时开始执行。Java EE 6 规范要求 Web 容器支持 Servlet 3.0 规范 (JSR 315)。

(3) EJB (Enterprise JavaBean: 企业 JavaBean): EJB 定义了一个用于开发基于组件的、企业级的、分布式多层应用系统的标准。基于该标准开发的企业 Java Bean 封装了应用系统中的核心业务逻辑。Java EE 6 规范要求 EJB 容器支持 EJB 3.1 规范 (JSR 318)。

1.2.3 Java EE 应用服务器软件及相关角色

我们把实现了 Java EE 规范的服务器软件称为 Java EE 应用服务器软件, 把运行于 Java EE 应用服务器软件之上的应用软件称为 Java EE 应用软件, 它的特点是: 一次开发、到处运行。

目前主流的 Java EE 应用服务器软件有以下 7 种。

(1) IBM WebSphere Application Server (WAS)。

(2) JBOSS。

(3) WebLogic。

(4) Apusic。

(5) Tomcat。

(6) Oracle GlassFish Server。

(7) Apache Geronimo 等。

我们把使用 Java EE 软件的相关人员分为 3 类。

(1) Java EE 应用服务器开发者: 开发符合 Java EE 规范的应用服务器软件的人员, 这些软件包括了组件容器、Java EE API 的实现等。

(2) Java EE 应用软件开发者: 开发、组装和部署基于 Java EE 应用服务器软件的应用软件的

人员。

(3) Java EE 应用系统管理员：配置、监控和管理 Java EE 应用系统的技术人员。

1.2.4 Java EE 应用软件的体系结构

Java EE 应用软件分为典型的 4 层结构，如图 1.3 所示。

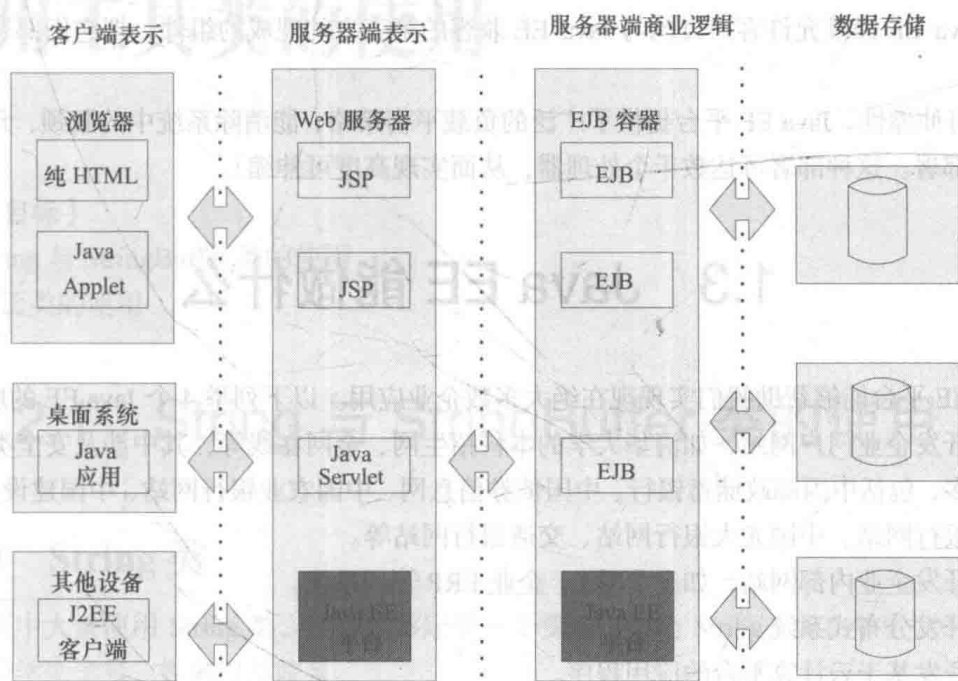


图 1.3 Java EE 应用软件的体系结构

(1) 运行在客户端机器上的客户层：负责与用户直接交互。Java EE 支持多种客户端，可以是 Web 浏览器，也可以是专用的 Java 客户端。

(2) 运行在 Java EE 服务器上的表示层：该层可以是基于 Web 的应用服务，利用 Java EE 中的 JSP 与 Servlet 技术，响应客户端的请求，并可向后访问业务逻辑组件。

(3) 运行在 Java EE 服务器上的业务逻辑层组件：主要封装了业务逻辑，完成复杂计算，提供事务处理、负载均衡、安全、资源连接等各种基本服务。

(4) 运行在 EIS (Enterprise Information System) 层服务器上的企业信息系统：该层包括企业现有系统（数据库系统、文件系统等）。Java EE 提供了多种技术以访问这些系统。

Java EE 应用软件的这种体系结构的优点如下。

(1) 部署代价廉价。它提供了中间层集成框架以满足无需太多费用而又需要高可用性、高可靠性和可扩展性的应用的需求，降低了开发多层应用的费用和复杂性，同时提供对现有应用程序集成强有力的支持。

(2) 保留现存的 IT 资产。它可以充分利用用户原有的投资，如一些公司使用的 BEA Tuxedo、IBM CICS、IBM Encina、Inprise VisiBroker 以及 Netscape Application Server 等。

(3) 高效的开发。它允许公司把一些通用的、很烦琐的服务端任务交给中间件供应商去完成。这样开发人员可以把精力集中在如何创建商业逻辑上，从而可大大缩短开发时间。中间件供应商一般提供以下中间件服务。

- 状态管理服务。
- 持续性服务。
- 分布式共享数据对象 CACHE 服务。

(4) 支持异构环境, 包括如下两种情况。

- 基于 Java EE 的应用程序不依赖任何特定操作系统、中间件、硬件, 只需开发一次就可部署到各种平台。

- Java EE 标准允许客户订购与 Java EE 兼容的第三方的现成的组件, 把它们部署到异构环境中。

(5) 可伸缩性。Java EE 平台提供了广泛的负载平衡策略, 能消除系统中的瓶颈, 允许多台服务器集成部署。这种部署可达数千个处理器, 从而实现高度可伸缩。

1.3 Java EE 能做什么

Java EE 平台能够帮助我们实现现在绝大多数企业应用。以下列举 4 个 Java EE 的应用。

(1) 开发企业门户网站, 如清华大学的本科招生网、金网在线等, 其中涉及安全类和银行类的网站较多, 包括中国邮政储蓄银行、中国债券信息网、中国农业银行网站、中国建设银行网站、中国工商银行网站、中国光大银行网站、交通银行网站等。

(2) 开发企业内部网站, 如企业 OA, 企业 ERP 管理系统。

(3) 开发分布式系统。

(4) 开发基于云计算平台的应用程序。

习 题

1. 进行 Java Web 开发时常用的客户端应用技术有哪些?

2. 简述什么是 C/S 结构、B/S 结构, 以及两者的区别。

第 2 章

常用工具类的使用

【学习目标】

- String 与 StringBuffer 类的使用
- 日历类的使用

2.1 String 与 StringBuffer 类的使用

2.1.1 String 类

在编程中大量使用 String 类，因为经常定义一个变量类型是 String，如在学生类中定义 3 个属性分别为学生学号、姓名以及院系。

```
public class Student{  
    private String sno,sname,sdept;  
    /*构造函数以及 set-get 方法代码省略*/  
}
```

下面介绍 String 类中的几个常用函数用法。

1. 字符串比较

我们从 3 个方面研究串比较，一个是串的内存引用是否相等；另一个是研究内容，也就是值是否相等；最后比较串值 ASCII 码的大小。

(1) 使用 “==” 比较：比较对象是否引用同一块内存。是则返回 true，否则返回 false。

(2) 使用 equals() 函数比较：比较 String 值是否相同，equals(), equalsIgnoreCase() 用于对两个字符串的内容进行等价性检查。其中，equalsIgnoreCase() 方法用于忽略字符串大小写进行相等的比较。

【例 2.1】字符串值忽略大小写比较示例。

```
public static void main(String[] args) {  
    String s1 = "Java";  
    String s2 = new String("JAVA");  
    if (s1.equalsIgnoreCase(s2))  
        System.out.println("s1 与 s2 值相同");  
    else  
        System.out.println("s1 与 s2 值不相同。");  
}
```

读者可以自己上机看看运行的结果。

(3) 使用 `compareTo()` 函数比较: 返回整型数字, 比较串值大小。 `compareTo()` 用于比较两个字符串的大小, 其结果为负、零或正具体取决于两个字符串的字典顺序。注意大写和小写不是相等的。

【例 2.2】 字符串大小比较示例 (`compareTo` 方法)。

```
public static void main(String[] args) {
    String s1=new String("aaa");
    String s2="bbb";
    int x=s1.compareTo(s2);
    if(x>0)
        System.out.println("s1>s2");
    else if(x<0)
        System.out.println("s1<s2");
    else
        System.out.println("s1=s2");
}
```

很显然运行结果应该是 “`s1<s2`”。实际上这个方法是按照字符串的首字母到尾字母一一比较的原则, 直到能比较出大小为止, 然后输出两个字母的 ASCII 码大小的差值。如果所有字母都相等则结果为 0。

2. 分割字符串函数 `split()`

我们经常需要对字符串以某个字符或串 (称为模式) 进行分割, 如将文章中的内容读到字符串中, 然后根据标点或空格对文章中的单词进行分割读出。例如:

```
String s1="aa,bb,cc,dd,ee fff";
String s[]=s1.split(",");//表示以逗号对 s1 分割, 注意返回一个串数组。
String s[]=s1.split(",|");//以逗号或空格分割串 | 表示或。
```

当出现 “.” 或 “-” 或其他相当于 Java 中的关键字时, 要使用转义符 “\”。

```
String s1 = "aa.bb.cc,dd,ee fff";
String s[]=s1.split("\\.|,| ");
```

【例 2.3】 字符串分割示例。

```
public static void main(String[] args) {
    String s1=new String("五莲路归昌路-五莲路凌河路-五莲路东陆路" +
        "-五莲路荷泽路-五莲路兰城路-五莲路莱阳路"+
        "-浦东大道五莲路-浦东大道金桥路-浦东大道居家桥路" +
        "-浦东大道德平路-浦东大道歇浦路-浦东大道北洋泾路"+
        "-浦东大道民生路-浦东大道桃林路-浦东大道源深路" +
        "-浦东大道钱仓路-陆家嘴东路东方医院-浙江南路延安东路"+
        "-金陵中路龙门路-普安路延安东路(人民广场) ");
    String []zd=s1.split("-");
    for(String t:zd)
        System.out.println(t);
    String s2 = "aa.bb.cc,dd,ee fff";
    String s[]=s2.split("\\.|,| ");
    for(String ss:s){
        System.out.println(ss);
    }
}
```

输出结果如图 2.1 所示。