

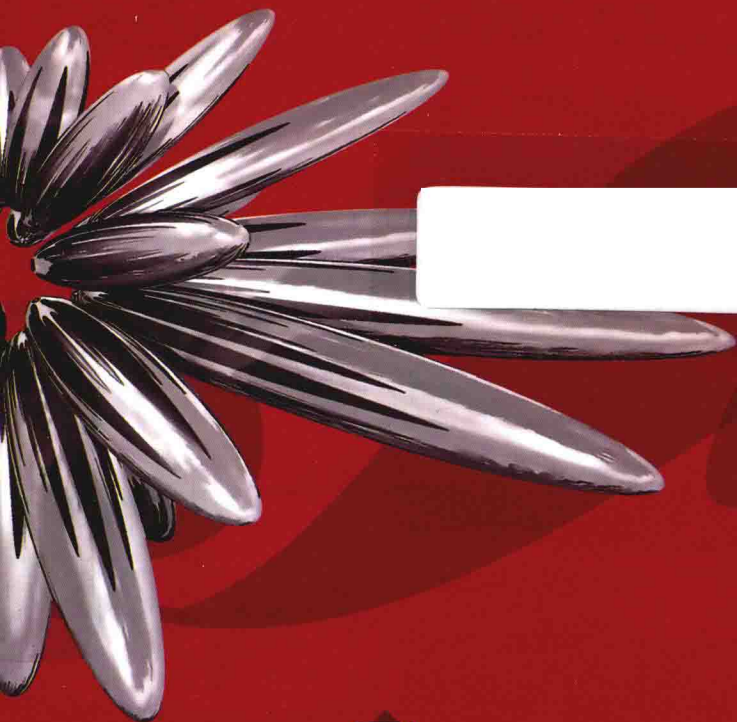


[普通高等教育“十三五”规划教材]

卓越工程师培养计划创新系列教材

Java EE 技术与应用

◆ 张军朝 主编 ◆ 赵荣香 王 浩 副主编



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

普通高等教育“十三五”规划教材
卓越工程师培养计划创新系列教材

Java EE 技术与应用

张军朝 主编 赵荣香 王 浩 副主编

吕丰德 张江华 薛彦龙 参编



电子工业出版社
Publishing House of Electronics Industry
北京·BEIJING

内 容 简 介

本书基于最新的 Java EE 7 规范对 Java EE 的基础知识和应用开发技术进行系统讲解。书中主要内容分为五篇：入门篇对 Java EE 做了基本的介绍。第一篇介绍了 Java EE 的概念和 Java 语言基础以及实际开发中涉及的各种基础技术，包括 SQL 语言与 MySQL、XML、HTML、CSS、JavaScript、Servlet、JSP 等，可使读者全面认识 Java EE 以及各种基础技术。第二篇以 Java EE 企业应用的表现层、数据持久化层和业务逻辑层的开发为主线，重点讲解实际开发中涉及的各种框架技术，包括 Struts 2、Hibernate、Spring 等，使读者掌握开发完整 Java EE 企业应用所用到的框架技能。第三篇讲解了快速开发平台的概念，并以 TDFY 快速开发平台为例，讲解了该平台的功能、设计思想、技术选型、安装部署以及使用步骤，同时还讲解了该平台中常用组件的使用以及手机端的基础接口知识，使读者掌握 TDFY 平台开发技巧，从而能够快速开发出企业级应用。第四篇以一个基于 TDFY 快速开发平台的公共资源交易系统为例，讲解了该项目从需求分析、系统设计、功能设计、详细设计到最后代码编程的整个过程，并以其中一个场地安排模块为例，详细讲解了 TDFY 快速开发平台在实际项目中的使用。本书配套资源有：PPT、源代码、习题解答等。

本书可作为高等学校计算机专业教材，也可作为相关人员的参考书。本书每一章都是一个完整独立的部分，因此教师在授课时可根据授课重点及课时数量进行灵活调整。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有，侵权必究。

图书在版编目(CIP)数据

Java EE 技术与应用/张军朝主编. —北京：电子工业出版社，2016.2

ISBN 978-7-121-27717-7

I. ①J… II. ①张… 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(2015)第 287149 号

策划编辑：任欢欢

责任编辑：郝黎明

印 刷：三河市鑫金马印装有限公司

装 订：三河市鑫金马印装有限公司

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱 邮编：100036

开 本：787×1092 1/16 印张：25.75 字数：659.2 千字

版 次：2016 年 2 月第 1 版

印 次：2016 年 2 月第 1 次印刷

定 价：58.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：(010)88254888。

质量投诉请发邮件至 zltz@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：(010)88258888。

前 言

Java EE 是一套全然不同于传统应用开发的技术架构, 包含许多组件, 主要可简化且规范应用系统的开发与部署, 进而提高可移植性、安全性与再用价值。

Java EE 的核心是一组技术规范与指南, 其中所包含的各类组件、服务架构及技术层次, 均有共同的标准及规则, 让各种依循 Java EE 架构的不同平台之间, 存在良好的兼容性, 解决了过去企业后端使用的信息产品彼此之间无法兼容, 导致企业内部或外部难以互通的问题。

Java EE 典型有四层结构组件: 客户层组件 (HTML, 脚本语言, 各种标签); WEB 层组件 (JSP 页面, Servlets, 使用 JavaBean 处理用户输入); 业务层组件 (EJB 进行处理); EIS 层组件。企业级 JavaBean 有三种: 会话 Bean (与客户端程序的临时交互), 实体 Bean (数据库表中的永久记录), 消息驱动 (允许业务层组件亦不接受 JMS 消息)。

为了帮助众多的软件开发人员尽快地掌握 Java EE 平台的相关知识, 尽快地步入实际项目的开发中, 作者根据多年教学和项目开发经验编写了此书。

本书既介绍了 Java EE 的基本知识, 也对项目开发中流行的几个框架进行了讲解, 还通过一个真实案例向读者介绍了 Java EE 项目完整的开发步骤。读者通过本书可以尽快地掌握在 Java EE 平台下进行项目开发的技术。

本书具有以下特点:

1. 内容饱满、由浅入深

本书内容既包括 Java EE 平台下开发的基础知识, 也有项目编程的实用技巧, 还提供了多个真实案例供读者学习。本书在知识的层次上由浅入深, 使读者可以从 Java EE 的门外汉平稳、快速地步入 Java EE 开发的殿堂。

2. 结构清晰、语言简洁

本书中所有案例都是按照笔者的真实项目开发过程进行介绍的, 结构清晰, 语言简洁, 便于实际练习。为了帮助读者更好地理解相关知识点, 全书穿插了很多实用技巧及温馨提示。

3. 实际商业案例

本书的案例均具有实际商业价值, 如果进行开发, 价格要数万元, 笔者将其完整地展现给了读者。

本书共分 17 章: 第 0 章是 Java EE 概述; 第 1 章全面讲述了 Java 的发展、开发环境、Java 语法及基础知识, 通过学习这些基础知识, 读者可以对 Java 有更深入的了解, 在理解后续框架时会更容易些; 第 2 章主要讲述了 jdk1.5、jdk1.6、jdk7、jdk8 的一些新特性, 了解 jdk 版本更新带给开发者的一些最新技术; 第 3 章全面讲述了 SQL 和 MySQL 的相关概念与使用, 通过本章内容的学习, 读者可以对数据库的常见概念有所了解, 并学会使用常用的 SQL 语句对数据库的数据记录进行操作; 第 4 章主要介绍了标记语言 XML, 并对 XML 的语法规则进行详细介绍, 最后介绍了如何使用 JAXP 解析 XML, 通过学习本章内容读者可以对 XML 有更深入的了解; 第 5 章全面讲述了 HTML、CSS 的概念以及使用, 帮助开发者更灵活地使用 HTML 和 CSS, 本章还对 Bootstrap 这个近期比较流行的响应式框架做了介绍, 利用 Bootstrap 可以简单迅速地做出漂亮的前端页面; 第 6 章主要介绍了前端开发中需要用到的 JavaScript、AJAX、Json、JQuery 等技术的概念、语法以及使用等内容, 使开发者对这些技术有基本的了解; 第 7 章主要讲述了 Servlet 技术, 通过 4 节内容分别对 Servlet

原理、生命周期、服务器内部调整和外部跳转、Session、Cookie、URL 重写等内容进行了介绍；第 8 章主要讲述了 JSP 的概念以及如何使用 JSP，通过本章的学习读者可以知道如何在前端页面中显示 Java 服务器端的数据；第 9 章通过 MVC 模式简介、Struts 2 框架来历简介、Struts 2 概述、Struts 2 原理的介绍、构件基于 Struts 2 的应用等内容全面讲述了 Struts 2 这个框架，通过学习本章内容，读者可以很轻松地运用 Struts 2 来实现 Java 与前端页面的数据交互；第 10 章讲述了 JDBC、Hibernate、MyBatis 等持久层技术的相关概念以及使用的详细讲解，通过学习持久层框架技术，读者可以很方便地通过配置 XML 文件属性的方法简单地实现 Java 与数据库的交互；第 11 章讲述了 Spring 的概念以及 Spring 的 IoC 注入、AOP、代理模式以及 Spring 事务管理机制等内容，通过学习本章内容可以对 Spring 框架整体有所了解，通过使用 Spring 的控制反转和面向切面的特性来编程可以大大提高开发效率；第 12 章通过快速开发平台内置功能、设计思想、技术选型、安装部署、文件结构、系统配置文件等方面内容的介绍，对快速开发平台做了全面的讲述；第 13 章全面介绍了快速开发平台的各种组件，包括布局组件、用户工具、全局缓存、字典工具、功能权限控制、数据权限等常用组件，通过认识这些组件，在开发过程中遇到类似的开发需求就可以直接调用组件而不用重新开发，所以使用组件开发可以提升开发效率；第 14 章详细讲述了快速开发平台中代码生成器的详细使用步骤，利用代码生成器组件可以根据数据库表的信息快速生成 Java 代码，提高开发效率；第 15 章全面讲述了快速开发平台对手机端应用程序提供的基础接口，介绍了传输格式、账号登录、登录成功、登录失败、请求页面、获取基础信息等方面的内容；第 16 章通过讲述公共资源交易平台系统这样一个真实的项目案例，来带领读者了解真正的软件是怎样的开发流程，都包括哪些内容，如何去开发一个实用的应用系统。这里主要从项目概述、需求分析、系统设计、功能设计、场地安排模块等方面详细讲解公共资源交易平台系统。

本书的内容通俗易懂，涵盖了 Java EE 相关的所有基础技术，并向读者介绍了真实项目的开发流程，特别适合做为软件工程、计算机科学与技术、物联网工程、计算机应用、电子商务等专业的高年级本科生和研究生的教材，也适合相关软件开发技术人员参考，同时也是职业技术类学院和各种软件开发培训机构的首选教材。

作者从事工程应用软件开发 15 年，主持开发的工程应用系统有：建设工程招投标信息处理系统、建设工程（土建、装饰、安装、市政、园林绿化、抗震加固、水利水电、电力、公路、邮电通信、煤炭）造价信息处理系统、建筑工程三维可视化算量软件、建设工程招投标企业信用信息系统、建筑工程监管信息系统、公共资源交易系统、重点项目（重点企业）动态监察系统、混凝土质量动态监管系统、大型建筑工地太阳能 3G 无线远程视频监控系统、大型流域和城市防洪预警会商系统、城市火灾预警和消防装备全生命周期管理系统、路灯景观灯照明控制系统等。其中基于 Zigbee 和 GPRS 的路灯照明调光节能控制系统已在太原市滨河东路景观照明系统工程、太原市汾河公园照明工程、长风商务区景观照明工程、汾东商务区路灯照明工程、江苏宜兴团氈公园景观照明工程、山东曹县路灯照明工程、河南中牟县路灯照明工程中推广应用。

本书共 17 章，分为入门篇、基础篇、框架篇、平台篇、应用篇五个部分，总学时为 48 学时，其中授课时间为 40 学时，试验练习时间为 8 个学时。

本书由张军朝担任主编，制定本书大纲、内容安排并指导写作；高保禄负责全书的组织工作；孙靖宇负责全书的统稿工作；赵荣香负责本书所有源代码的调试工作；王浩负责本书应用篇的项目编程规划编写。张军朝编写了第 0、1、2、3 章；赵荣香编写了第 4、5、6 章；高保禄编写了第 7、8、9 章；孙靖宇编写了第 10、11、12 章；吕丰德编写了第 13、14 章，张江华编写了第 15 章，王浩编写了第 16 章。本书由太原理工大学陈俊杰教授主审。

在本书的编写过程中得到了计算机专业教学指导委员会委员、太原理工大学陈俊杰教授，太原

理工大学崔冬华教授，山西太原天地方圆电子科技有限公司赵荣香高工、吕丰德工程师、张江华工程师自始至终的支持和帮助；太原理工大学赵阳硕士、王青文硕士、陶亚男硕士在编写和校对过程中也做了大量的工作。在此一并致以衷心的感谢！

编者力求将实践和理论相结合，科研和教学相结合，工程和教学相结合，硬件和软件相结合，先进和实用相结合，编写出高质量、高水平的教材，但由于编者水平有限，书中错误和不当之处在所难免，敬请读者谅解和指正，联系邮箱：zhangjunchao@tyut.edu.cn。

张军朝

2016年1月1日 于 太原理工大学 国交楼

目 录

第 0 篇 入 门 篇

第 0 章 Java EE 概述.....	1	0.5 Java EE 主要技术.....	4
0.1 Java EE 是什么.....	1	0.5.1 JSP, Servlet.....	4
0.2 Java EE 发展史.....	1	0.5.2 EJB (Enterprise JavaBean)	5
0.3 Java EE 到底要解决什么问题.....	2	0.5.3 Container (容器)	5
0.4 Java EE 组件、容器、服务器.....	2	0.5.4 RMI	6
0.4.1 组件 (Component)	2	0.5.5 JMS.....	7
0.4.2 容器 (Container)	2	0.5.6 JTA	7
0.4.3 服务器 (Server)	3	0.5.7 JAAS.....	7
0.4.4 组件、容器、服务器三者的功能.....	3	0.5.8 Connector.....	7
0.4.5 组件体系结构.....	3		

第 1 篇 基 础 篇

第 1 章 CoreJava.....	8	1.3 表达式和程序控制.....	21
1.1 Java 简介与使用.....	8	1.3.1 局部变量.....	21
1.1.1 JDK	8	1.3.2 实例变量.....	22
1.1.2 设置 Java 开发环境.....	9	1.3.3 操作符.....	24
1.1.3 为什么发明 Java	9	1.3.4 类型转换.....	27
1.1.4 Java 虚拟机的特征.....	9	1.3.5 条件语句.....	28
1.1.5 垃圾回收的概念.....	10	1.3.6 循环语句.....	29
1.1.6 Java 平台代码安全实现策略.....	11	1.3.7 循环语句中流程跳转.....	30
1.1.7 定义类、包、applets 和应用程序.....	11	1.4 数组.....	31
1.2 标识、关键字、类型.....	13	1.4.1 数组变量的声明.....	31
1.2.1 注释.....	13	1.4.2 初始化.....	32
1.2.2 分号、块和空格.....	14	1.4.3 多维数组.....	33
1.2.3 标识符.....	14	1.4.4 数组的边界.....	33
1.2.4 关键字.....	14	1.4.5 数组的复制.....	34
1.2.5 基本类型和引用类型.....	15	1.5 高级语言特性.....	34
1.2.6 boolean 类型.....	15	1.5.1 OOP 中的基本概念.....	34
1.2.7 文本数据类型.....	16	1.5.2 类和对象.....	35
1.2.8 整数类型.....	17	1.5.3 定义方法的形式及参数传递.....	35
1.2.9 浮点类型.....	18	1.5.4 封装.....	36
1.2.10 变量的声明和赋值.....	19	1.5.5 方法重载.....	37
1.2.11 推荐命名规则.....	19	1.5.6 创建和初始化对象.....	37
1.2.12 创建类.....	19	1.5.7 构造方法.....	37
1.2.13 创建实例.....	21	1.5.8 继承.....	39

1.5.9	方法覆盖	39	第 2 章	Java 新特性	93
1.5.10	this 和 super 关键字	40	2.1	JDK1.5 新特性	93
1.5.11	多态及对象类型转换	43	2.1.1	自动装箱/拆箱	93
1.5.12	static 修饰符	44	2.1.2	增强 for 循环	96
1.5.13	final 修饰符	45	2.1.3	类型安全枚举	96
1.5.14	接口	47	2.1.4	可变长参数	99
1.5.15	访问控制	48	2.1.5	静态导入	100
1.6	集合	53	2.1.6	格式化输出	100
1.6.1	Collection 和 Iterator 接口	53	2.1.7	泛型	101
1.6.2	Set、List、Map 集合	54	2.1.8	Annotation	103
1.6.3	反射机制	59	2.2	JDK1.6 新特性	106
1.7	异常	61	2.3	JDK7 新特性	108
1.7.1	异常的基本概念	61	2.4	JDK8 新特性	110
1.7.2	try...catch 语句	62	2.4.1	接口的默认方法	110
1.7.3	finally 语句	62	2.4.2	Lambda 表达式	111
1.7.4	异常调用栈	64	2.4.3	函数式接口	112
1.7.5	异常层级关系	64	2.4.4	方法与构造函数引用	112
1.7.6	异常声明和处理	65	2.4.5	访问局部变量	113
1.7.7	断言	67	2.4.6	访问对象字段与静态变量	113
1.8	线程	68	2.4.7	访问接口的默认方法	114
1.8.1	线程概念	68	2.4.8	Date API	116
1.8.2	线程的组成部分	68	2.4.9	多重 Annotation 注解	118
1.8.3	线程的创建和启动	68	习题		119
1.8.4	线程状态	70	第 3 章	SQL+MySQL	120
1.8.5	线程调度	71	3.1	SQL 概述	120
1.8.6	线程的同步	72	3.2	MySQL	122
1.8.7	线程的通信	73	习题		137
1.8.8	线程死锁、线程让步	75	第 4 章	XML	138
1.8.9	调整线程优先级	76	4.1	标记语言	138
1.9	IO 流和文件流	76	4.2	XML 语法规则	139
1.9.1	流的概念	76	4.3	DTD/Schema	143
1.9.2	字节输入流和输出流	77	4.3.1	DTD/Schema 的由来	143
1.9.3	输入流和输出流层级结构	78	4.3.2	DTD 中的定义规则	143
1.9.4	字符输入/输出流	79	4.3.3	文档类型定义	146
1.9.5	文件流	80	4.3.4	Schema	147
1.9.6	对象的序列化和反序列化	82	4.4	JAXP 解析 XML	148
1.10	网络编程	84	习题		155
1.10.1	网络相关概念	84	第 5 章	HTML/CSS+Bootstrap	156
1.10.2	基于 TCP 的 Socket 编程步骤	85	5.1	HTML	156
1.10.3	基于 UDP 的 Socket 编程步骤	87	5.1.1	HTML 基础	156
习题		89			

5.1.2	几个重要的标记	158
5.2	CSS	168
5.2.1	CSS 定义	168
5.2.2	CSS 选择器	168
5.2.3	样式的继承	172
5.2.4	样式的优先级	172
5.2.5	关键属性	173
5.2.6	块标记和行内标记	174
5.2.7	常见的属性	174
5.3	Bootstrap	179
	习题	184
第 6 章	JavaScript+AJAX	185
6.1	JavaScript	185
6.1.1	JavaScript 相关概念	185
6.1.2	JavaScript 的基本语法	187
6.1.3	JavaScript 中常见内置对象	192
6.1.4	DOM	198
6.1.5	BOM	203
6.2	AJAX	208
6.2.1	AJAX 定义	208
6.2.2	AJAX 编程	209
6.2.3	XmlHttpRequest 对象的重要 属性	211
6.2.4	缓存问题	211
6.2.5	用户注册案例	211
6.2.6	AJAX 的优点和编码问题	216
6.2.7	级联下拉列表案例	217
6.3	JSON	217
6.4	jQuery	218
6.4.1	jQuery 的定义	218
6.4.2	jQuery 使用	218
6.4.3	jQuery DOM 操作	219
6.4.4	jQuery 事件处理	220
6.4.5	jQuery 操作数组的方法	220

6.4.6	jQuery 对 AJAX 的支持	221
	习题	222
第 7 章	Servlet	223
7.1	Servlet 原理	223
7.1.1	Servlet 相关概念	223
7.1.2	Servlet 开发流程	224
7.1.3	Servlet Web 应用的流程	224
7.1.4	Servlet 通信	226
7.2	Servlet 的生命周期	230
7.3	服务器内部和外部跳转	232
7.3.1	服务器内部跳转	232
7.3.2	服务器外部跳转	232
7.4	Session、Cookie、URL 重写	233
7.4.1	Session	233
7.4.2	Cookie	235
7.4.3	URL 重写	236
7.5	ServletConfig 与 ServletContext	239
7.6	Filter (Servlet 过滤器)	242
7.7	Listener (Servlet 监听器)	244
	习题	249
第 8 章	JSP	250
8.1	JSP 相关概念	250
8.2	JSP 的使用	252
8.2.1	第一个 JSP	252
8.2.2	JSP 脚本元素	253
8.2.3	JSP 中的注释	254
8.2.4	动作元素	256
8.2.5	JSP 中默认的内置对象	257
8.2.6	JSP 模型	258
8.2.7	JavaBean	259
8.2.8	EL 表达式	261
8.2.9	JSTL	264
	习题	267

第 2 篇 框 架 篇

第 9 章	Struts2	269
9.1	Struts2 概述	269
9.2	Struts2 工作原理	270
9.3	构件基于 Struts2 的应用	273
9.4	第一个 Struts2 应用程序	276

	习题	278
第 10 章	持久层框架技术	279
10.1	JDBC	279
10.1.1	JDBC 相关概念	279
10.1.2	JDBC 的使用	280

10.2	Hibernate	284	11.5	IoC 中的 annotation 配置	315
10.2.1	Hibernate 相关概念	284	11.5.1	@Autowired	315
10.2.2	Hibernate 工作原理	284	11.5.2	@Resource	316
10.2.3	持久化对象	284	11.5.3	@PostConstruct 和	
10.2.4	延迟加载机制	286		@PreDestroy	316
10.2.5	关系映射	287	11.5.4	@Component	317
10.3	MyBatis	294	11.6	AOP	318
10.3.1	MyBatis 相关概念	294	11.7	Advice	324
10.3.2	MyBatis 的工作流程	295	11.8	AutoProxy 和 aop:config 标签	325
10.3.3	基础配置	296	11.8.1	AutoProxy	325
10.3.4	SQL 映射	298	11.8.2	aop:config 标签	326
10.3.5	SqlSession 接口	299	11.9	Spring+jdbc	327
习题		300	11.10	Spring+Hibernate	329
第 11 章	Spring	302	11.11	Spring 事务管理机制	330
11.1	Spring 概述	302	11.11.1	编程式事务管理	330
11.2	IoC 注入	305	11.11.2	声明式事务管理	331
11.3	PropertyEditor 和创建 Bean	312	11.11.3	事务描述	333
11.3.1	PropertyEditor	312	11.12	HibernateTemplate 类与	
11.3.2	创建 Bean	314		HibernateDaoSupport 类	335
11.4	Bean 的生命周期	315	习题		336

第 3 篇 平 台 篇

第 12 章	快速开发平台	337	12.8	文件结构	341
12.1	平台的概念	337	12.9	系统配置文件	343
12.2	平台的模式	338	第 13 章	常用组件	345
12.3	TDFY 快速开发平台简介	338	第 14 章	代码生成器应用	351
12.4	内置功能	338	第 15 章	手机端基础接口	358
12.5	设计思想	339			
12.6	技术选型	339			
12.7	安装部署	340			

第 4 篇 应 用 篇

第 16 章	公共资源交易平台	362	16.4.3	Java 各层代码修改	372
16.1	项目概述	362	16.4.4	MyBatis 映射文件修改	390
16.2	需求分析	363	16.4.5	WebApp 展示层代码修改	395
16.3	公共资源交易系统功能设计	365	16.4.6	权限控制	397
16.4	场地安排模块快速开发实例	367	16.4.7	程序调试	399
16.4.1	利用 PowerDesigner 设计表	367	习题		400
16.4.2	使用代码生成器生成代码	368	参考文献		401

第0篇 入门篇

第0章 Java EE 概述

0.1 Java EE 是什么

Java 平台有三个版本,这使软件开发人员、服务提供商和设备生产商可以针对特定的市场进行开发:

Java SE (Java Platform, Standard Edition)。Java SE 以前称为 J2SE。它允许开发和部署在桌面、服务器、嵌入式环境和实时环境中使用的 Java 应用程序。Java SE 包含了支持 Java Web 服务开发的类,并为 Java Platform, Enterprise Edition (Java EE) 提供基础。

Java ME (Java Platform, Micro Edition)。这个版本以前称为 J2ME。Java ME 为在移动设备和嵌入式设备(比如手机、PDA、电视机顶盒和打印机)上运行的应用程序提供一个健壮且灵活的环境。Java ME 包括灵活的用户界面、健壮的安全模型、许多内置的网络协议以及对可以动态下载的连接和离线应用程序的丰富支持。基于 Java ME 规范的应用程序只需编写一次,就可以用于许多设备,而且可以利用每个设备的本机功能。

Java EE (Java Platform, Enterprise Edition)。这个版本以前称为 J2EE。企业版帮助开发和部署可移植、健壮、可伸缩且安全的服务器端 Java 应用程序。Java EE 是在 Java SE 的基础上构建的,它提供 Web 服务、组件模型、管理和通信 API,可以用来实现企业级的面向服务体系结构(service-oriented architecture, SOA)和 Web 2.0 应用程序。Java EE 其实是一种企业级应用的软件架构,同时是一种思想,一套规范。

0.2 Java EE 发展史

Java Enterprise Edition 的发展不知不觉已经 17 年了,一开始,Java Enterprise Edition 简称“J2EE”,直到版本 5 才改名为 Java EE,而现在最新的版本则是 Java EE 7。

到这里,或许有人会问,为什么会有这么多套 Java EE 规范?这些版本的差别是什么?

1. J2EE1.2 的出现,主要是将之前各个单独的规范绑定到一起。
2. J2EE1.3,则是继续完善 J2EE 体系结构。
3. J2EE1.4,主要是加入了一个重要主题: Web Service。
4. Java EE 5,主题则是“简化”,简化之前复杂的 J2EE 思想,改善开发体验。
5. Java EE 6,进一步简化开发流程,增加平台的灵活性,从而更好地解决轻量级 Web 应用程序。此外,Java EE 6 开始与开源架构进行无缝集成,并对现有的技术做了精简。

6. Java EE 7, 扩展了 Java EE 6, 利用更加透明的 JCP 和社区参与来引入新的功能, 主要包括加强对 HTML5 动态可伸缩应用程序的支持、提高开发人员的生产力和满足苛刻的企业需求。

Java EE 的发展史示意图如图 0-1 所示。

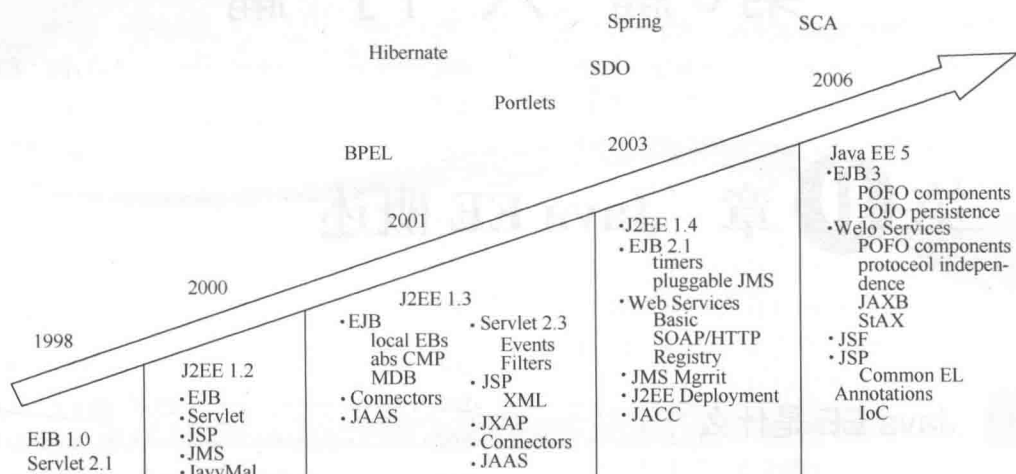


图 0-1 Java EE 发展史示意图

0.3 Java EE 到底要解决什么问题

从 Java EE 发展背景看, 它与“分布式应用”以及“互联网应用”的关系密不可分, 而这两者也正是 Java EE 要解决的问题!

其实, 分布式应用随着 20 世纪 90 年代互联网的兴起逐渐开始普及。在 90 年代中, 各种分布式应用标准逐渐诞生, 如: OMG 的 CORBA, MS 的 DCOM 等, 而 Sun 在推出 Java 的 RMI (Remote Method Invocation) 后, 便以 RMI 作为通信基础构建了 Java EE。笔者认为, Java EE 最核心要解决的问题就是“分布式应用”。而在接下来的竞争中, Java EE 也不负所托, 逐渐取代了 CORBA、DCOM 的地位。

0.4 Java EE 组件、容器、服务器

0.4.1 组件 (Component)

组件可以定义为一种自治的、提供外部公共接口的、动态可用的事物处理过程, 组件可以用来构建其他组件或者应用程序。简言之, 组件就是能完成一定功能的封装体, 小到一个类, 大到一个系统。

组件是满足某种规范的代码, 只要满足这个规范, 你的代码就可以部署到特定的容器 (下边有阐述), 容器就可以运行它了, 我们可以把组件看成容器的增值服务。

举个例子, 我们平常使用的 JVM 也可以认为是一种容器, 你只要把你写的程序交给 jvm 运行就可以了, 我们可以把程序看作一个组件, 但是这个组件有个特征那就是它必定有个 main 方法。

0.4.2 容器 (Container)

Java EE 的组件是不能独立运行的, 必须要为它提供相应的运行环境, 为组件提供运行环境的就是容器。

这里特别讲解 Servlet 容器 (Container), 它是一个 Java 撰写而成的程序, 负责管理 JSP/Servlet 运行过程中所需要的各种资源, 并负责与 Web 服务器进行沟通, 管理 JSP/Servlet 中所有对象的产生与消灭。Servlet 容器的实现必须符合 JSP/Servlet 的规范, 这个规范是由 Sun Microsystems Inc 公司制定的。

例如, 当使用者请求来到 Web 服务器时, Servlet 容器会将请求、响应等信息包装为各种 Java 对象 (如 `HttpRequest`、`HttpResponse`、`Cookies` 等), 对象中包括了客户端的相关信息, 像是请求参数, `session`、`cookie` 等信息, 当您使用 JSP/Servlet 的对象, 例如 `HttpResponse` 发送信息时, Servlet 容器将之转换为 HTTP 信息, 然后由服务器将信息发回客户端。以 JSP 来说, 容器负责将 JSP 转换为 Servlet 程序代码, 然后编译 Servlet 程序代码, 将之加载执行环境并执行, 容器也提供了许多资源, 除了基本的 Servlet 加载与执行之外, 例如 Web 环境设定、使用者认证、`session` 追踪等等, JSP/Servlet 事实上与容器之间的依赖日渐深厚, 要想真正发挥 JSP/Servlet 的功能, 正确设计出良好架构与功能的 Web 应用程序, 了解容器的特性是不可少的。

0.4.3 服务器 (Server)

容器也是不能直接运行的, 容器必须要运行在服务器之上, 一个服务器可以同时运行多个不同的容器。Java EE 容器是底层服务器的组成部分。Java EE 产品供应商通常使用现有的事务处理框架结合 Java SE 技术来实现 Java EE 服务器端功能。Java EE 客户端功能通常构建于 Java SE 技术。

在现有的 Java web 开发中, 关于应用服务器, 大家最熟知的开源有: Tomcat、Jboss、Resin, 目前看来这三个开源应用服务器的使用相当广泛, 主要原因以笔者来看有下面几点:

1. Tomcat 是 Apache 鼎力支持的 Java Web 应用服务器, 由于它优秀的稳定性以及丰富的文档资料, 广泛的使用人群, 从而在开源领域受到最广泛的青睐。
2. Jboss 作为 Java EE 应用服务器, 它不但是 Servlet 容器, 而且是 EJB 容器, 从而受到企业级开发人员的欢迎, 从而弥补了 Tomcat 只是一个 Servlet 容器的缺憾。
3. Resin 也仅仅是一个 Servlet 容器, 然而由于它优秀的运行速度, 使得它在轻量级 Java Web 领域备受喜爱, 特别是在互联网 Web 服务领域, 众多知名公司都采用其作为他们的 Java Web 应用服务器, 譬如 163、ku6 等。

在商用应用服务器里主要有: Weblogic、Websphere, 对于 Weblogic, 有些开发者也只用其当 Servlet 容器, 然而就在同等条件下, 在性能及易用性等方面, 要比 Tomcat 优秀很多。

还有一款由大名鼎鼎的 Sun 公司推出的 Glassfish 的 Java EE 服务器, Glassfish 是一个免费、开放源代码的应用服务, 它实现了 Java EE 5, Java EE 5 平台包括了以下最新技术: EJB 3.0、JSF 1.2、Servlet 2.5、JSP 2.1、JAX-WS 2.0、JAXB 2.0、Java Persistence 1.0、Common Annotations 1.0、StAX 1.0 等。

0.4.4 组件、容器、服务器三者的功能

1. 组件主要由应用开发人员完成, 用来实现应用系统的功能。
2. 容器有两个主要的功能: 一是提供组件运行环境, 二是控制组件生命周期。
3. 服务器也有两个主要的功能: 一是提供容器运行环境, 二是实现 JEE 中的技术对应的规范。

0.4.5 组件体系结构

组件体系结构图如图 0-2 所示。

组件体系结构

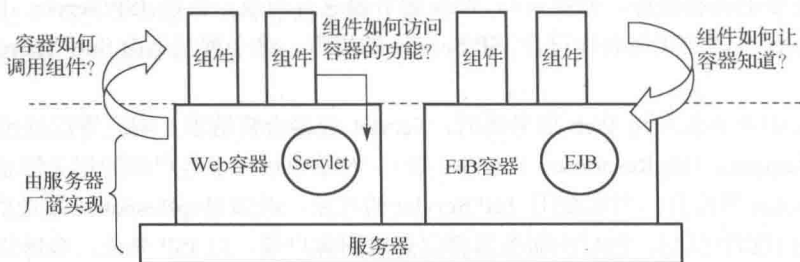


图 0-2 组件体系结构图

0.5 Java EE 主要技术

0.5.1 JSP, Servlet

JSP, Servlet 同属“Web 层”，并都属于“动态网页技术”。所谓“动态网页技术”和传统的“静态网页技术”不一样，传统的“静态网页技术”就是把做好的 html 文件直接上传到服务器并直接供客户浏览，而“动态网页技术”则是每次都根据用户请求，动态生成响应页面并返回。“动态网页技术”的好处不言自明，无论从灵活性，数据保密性等方面说都是“静态网页”所无法媲美的。但“动态网页技术”也是有缺点的，就是相对较慢，现在的解决方案一般是：把“动态网页”中相对固定的部分做缓存，即所谓“静态页面”。

1. Servlet

Servlet 实际上就是按照 Servlet 规范编写的一个 Java 类，与传统的命令行启动的 Java 应用程序不同，Servlet 位于 Web 服务器内部，并由 Web 服务器加载并调用。

2. JSP

JSP 全称是：JavaServer Page。这项技术的推出目的其实很简单，为了弥补 Servlet 一个很重要的缺陷：“麻烦”。

先看看 Servlet 到底什么地方让人觉得麻烦，下面是一个 Servlet 处理 Get 请求例子：

```
public void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException
{
    System.out.println("处理 GET 请求 ing.....");
    response.setContentType("text/html;charset=GB2312");
    PrintWriter out = response.getWriter();
    out.println("<HTML>");           // 静态内容
    out.println("<BODY>");           // 静态内容
    out.println("Hyddd's Servlet Demo " + new Date().toString()); // 动态内容
    out.println("</BODY>");          // 静态内容
    out.println("</HTML>");          // 静态内容
}
```

从上面这个例子,相信大家已经发现问题了,Servlet 主要是把动态内容混合到静态内容中以产生 html,这导致 Servlet 代码中将会输出大量的 html 标识,同时,这也非常不利于程序员和 UI 美工的配合。为了解决这些问题, JSP 诞生了。

JSP 是一种建立在 Servlet 规范之上的动态网页技术,通常做法是:在 html 页面中嵌入 JSP 标记和脚本代码。JSP 把静态内容和动态内容的分离,实现了内容和表示的分离。

3. Servlet 与 JSP 的关系

图 0-3 描述了 Servlet 与 JSP 的关系, JSP 文件先是转换为 Servlet 类,然后编译,并启动 Servlet 实例响应客户端请求。为什么说 JSP 是建立在 Servlet 上的动态网页技术,从这里可以看出来。

Web 层主要就是 JSP 以及 Servlet 这两项技术。

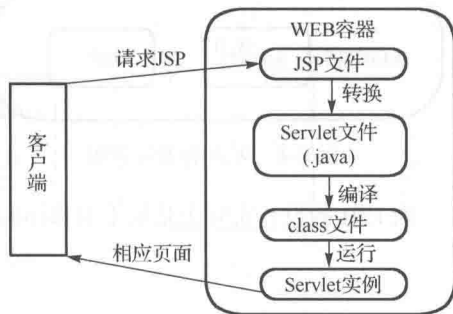


图 0-3 Servlet 与 JSP 的关系

0.5.2 EJB (Enterprise JavaBean)

分布式应用是 Java EE 一个基础的需求,那在不同机器上的“分布式”的应用到底会以一个什么样的形态出现呢?答案就是: EJB。EJB 属于业务逻辑层上的内容。

所谓 Bean,其实是“组件”的意思。EJB 可以让你像搭积木一样,通过本地/分布式调用组装不同应用到大型应用中,使你能集中精力来处理企业的业务逻辑,而像事务、网络、安全等等这些底层服务则统统留给 EJB 服务器开发商来解决。

利用基于组件的开发,可以把代码重用上升到一个新的高度。利用面向对象开发,重用的是类,而基于组件时,重用的则是更大的功能块。

1. EJB vs Java Bean

Java Bean 相当于数据存储类(不涉及具体业务逻辑),专门用来存数数据,提供 getter, setter 方法,并且在 JVM 上可直接运行。EJB 则相当于一个功能模块,提供业务逻辑的服务,而运行时,则需要 EJB 容器的帮助。EJB 是业务逻辑层最重要的技术。

0.5.3 Container (容器)

Container 这个概念经常在 Java EE 中出现,所谓 Container,在 Java EE 5 Tutorial 中有这样一段解释:“Containers are the interface between a component and the low-level platform-specific functionality that supports the component.”,而 Container 的作用,个人的认为是:为“应用程序”提供一个环境,使其可以不必须关注某些问题,如:系统环境变量,事务,生命周期……。通俗地说,Container 就像“秘书”,帮“应用程序”管理着各种杂乱的问题,为其提供运行时支持。

其中,Java EE 里有两个很重要的容器:Web 容器和 EJB 容器。

1. Web 容器

Web 容器是用于托管“Web 应用程序”的 J2EE 容器,主要负责管理“Servlet”和“JSP”运行,如图 0-4 所示。

2. EJB 容器

EJB 容器主要负责管理“EJB”的运行,如图 0-5 所示。

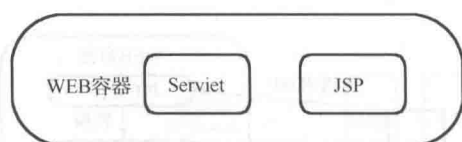


图 0-4 Web 容器示意图

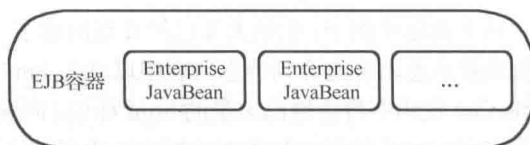


图 0-5 EJB 容器示意图

而 EJB 的设计实际上是基于对象池的思想，你可以认为 EJB=对象池+远程对象池。如图 0-6 所示。

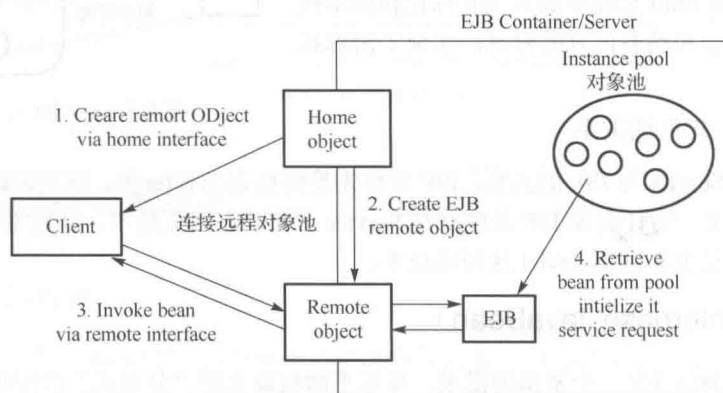


图 0-6 EJB 设计思路示意图

3. Servlet 与 EJB

其实，根据 Servlet 和 EJB 的设计初衷，我们已经可以看出 Java EE 对两者角色的定义了。线程的本质决定了 Servlet 只适合一些比较简单的轻量级应用；一旦问题复杂了，最好的就是使用 EJB。

0.5.4 RMI

RMI 全称：Java Remote Method Invocation，就是利用 Java 对象序列化的机制，实现远程类对象的实例化以及调用的方法。

RMI 在 Java EE 中主要是负责解决通信问题，特别是不同的 EJB 容器之间的通信。大家知道，在分布式应用中，各个功能模块（EJB）之间通信需要有统一的 RPC 协议，否则没法通信，而 RMI 就是负责这方面的工作。

1. RMI 与 CORB

可以说，RMI 就是 CORBA 的 Java 版实现。

2. 远程调用

现在主流的远程调用方式，不管是 com/com+，soap，webservice，rmi，.net remoting，都一样的，就是序列化，网络传输，反序列化。

序列化方式：同种 runtime 的，可以 native 的二进制序列化，序列化的效率高。文本的序列化（xml/json/自定义格式）的方式，可以跨平台和语言，一般基于中间类型。但此序列化方式的效率低，数据量也偏大。

网络传输：可以使 socket/http 或是自定义协议的。socket 数据冗余最小，效率最高。RMI 其实是 socket 上的自定义协议。http 要走 http 的报文，文本的方式最合适，实现最简单，开发和部署方便。

远程调用方式示意图如图 0-7 所示。

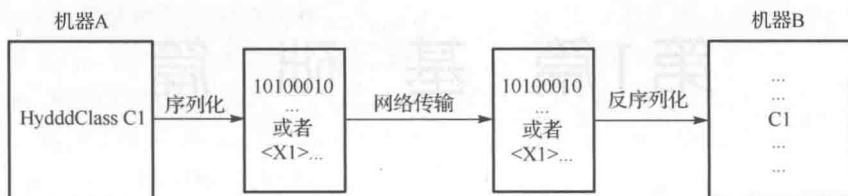


图 0-7 远程调用方式示意图

0.5.5 JMS

JMS: Java Message Service. JMS 提供一种消息机制，主要作用是提供异步通信的支持，是 Java EE 的重要基础模块。值得注意的是，异步通信一般都采用消息机制，这种情况在 Windows 中最常见。

0.5.6 JTA

JTA: Java Transaction API, 主要提供事务服务和分布式事务管理功能, 保证分布式事务的一致性, 是 Java EE 的重要基础模块。

0.5.7 JAAS

JAAS: Java Authentication Authorization Service (Java 认证与授权服务), 提供了对 Java 组件的安全保护, 如哪些 Servlet, JSP 能被哪些用户访问, 哪些 EJB 能被调用等。但需要注意的是, JAAS 只提供了对 Java EE 组件的保护, 对于企业应用业务的权限, 它是做不到的。

0.5.8 Connector

Connector 主要作用就是把其他已有的资源、服务、系统整合到 Java EE 系统中。不同的服务提供商和 Java EE 平台会定义不同的协议，而 Connector 就是指这些协议的实现。