普通高等教育
软件工程 “十二五”规划教材

12th Five-Year Plan Textbooks
of Software Engineering

Java EE 6
企业级应用开发教程

李树秋 ◎主编
王晓燕 姚志林 彭君 李兵 杨馥宁 ◎副主编

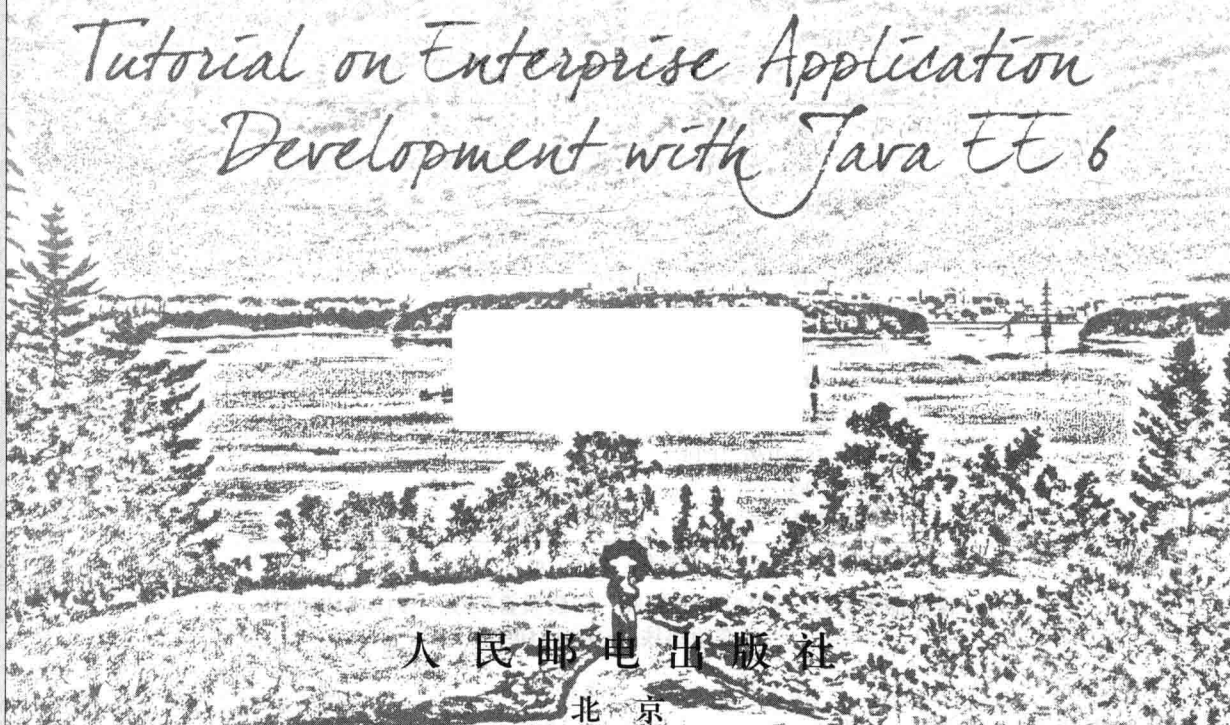
人民邮电出版社
POSTS & TELECOM PRESS

普通高等教育
软件工程“十二五”规划教材

12th Five-Year Plan Textbooks
of Software Engineering

Java EE 6
企业级应用开发教程

李树秋 ◎主编
王晓燕 姚志林 彭君 李兵 杨馥宁 ◎副主编



人民邮电出版社
北京

图书在版编目（CIP）数据

Java EE 6企业级应用开发教程 / 李树秋主编. --
北京 : 人民邮电出版社, 2015.2
普通高等教育软件工程“十二五”规划教材
ISBN 978-7-115-38346-4

Ⅰ. ①J… Ⅱ. ①李… 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015226号

内 容 提 要

本书针对高校计算机应用和软件工程等本科专业中的应用型需求，根据Java EE 6规范，并参考Java EE 6在企业级开发中应用的特点编写而成。全书共分14章，包括Java EE概述，Servlet、JSP和JSF程序开发，JDBC、JNDI和EJB技术，会话Bean、JMS与消息驱动Bean，JPA、JPQL、Web Service，Java EE安全性，以及SSH框架开发。

本书注重知识体系结构的系统性和条理性，注重理论化知识体系结构与开发实践过程相结合，介绍Java EE 6.0中的重要技术，强调技术在实际项目开发中的操作和应用。本书结构紧凑，语言通俗，深入浅出，示例丰富，可读性强，便于教学，可作为高等学校计算机应用和软件工程类专业本、专科学生的教材或教学参考书，也可供计算机专业人士参考使用。

◆ 主　　编　李树秋
副 主 编　王晓燕　姚志林　彭　君　李　兵　杨馥宁
责任编辑　邹文波
执行编辑　税梦玲
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
北京铭成印刷有限公司印刷

◆ 开本：787×1092　1/16
印张：20.5　　2015年2月第1版
字数：582千字　　2015年2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

目　录

第 1 章 Java EE 概述

1.1 Java EE 的产生与发展

Java EE 是 Java Enterprise Edition 的缩写，是建立在 Java 平台上的企业级应用的解决方案。Java EE 基于 Java SE 平台，提供了一组可移植的、健壮的、可伸缩的、可靠的和安全的、可用于开发和运行的服务器端应用程序的 API（Application Programming Interfaue，应用程序编程接口）。

Sun 公司在 1998 年发表 JDK1.2 版本的时候，开始使用名称 Java 2 Platform，即 Java 2 平台，修改后的 JDK 称为 Java 2 Platform Software Developing Kit，即 J2SDK，并分为标准版（Standard Edition, J2SE）、企业版（Enterprise Edition, J2EE）和微型版（Micro Edition，J2ME）。2006 年 5 月，Sun 公司推出 Java SE 5，此时 Java 的各种版本又进行了更名，J2EE 更名为 Java EE，J2SE 更名为 Java SE，J2ME 更名为 Java ME。

1998 年 Sun 发布了 EJB1.0 标准。EJB 为企业级应用中的数据封装、事物处理、交易控制等功能提供良好的技术基础。至此，J2EE 平台的三大核心技术 Servlet、JSP 和 EJB 都已先后问世。1999 年，Sun 正式发布了 J2EE 的第一个版本。紧接着，遵循 J2EE 标准、为企业级应用提供支撑平台的各类应用服务软件相继涌现出来。IBM 的 WebSphere、BEA 的 WebLogic 都是这一领域里成功的商业软件平台。随着开源运动的兴起，JBoss 等开源的应用服务器软件业吸引了许多用户的注意力。2003 年，Sun 的 J2EE 版本已经升级到 1.4 版本，其中 3 个关键组件的版本也升级到了 Servlet 2.4、JSP2.0 和 EJB 2.1。至此，J2EE 体系及相关的软件产品已经成为 Web 服务端开发的一个强有力的支撑环境。

但从 1999 年诞生的第一个 J2EE 版本一直到 J2EE1.4 版本，由于使用不方便而经常被人们抱怨。为了实现一个简单的 J2EE 程序，就需要大量的配置文件。2002 年，J2EE1.4 推出后，J2EE 的复杂程度达到了顶点。尤其是 EJB2.0，开发和调试的难度非常大。Sun 公司一直在试图改变这种状况，终于在 2006 年 5 月正式发布了 J2EE1.5 规范，并改名为 Java EE 5。Java EE 5 大大降低了开发难度。2009 年 12 月 Sun 公司正式发布了 Java EE 6 标准。EJB 3.1 随 Java EE 6 一起发布，进一步简化了使用，并改进了许多常见的使用模式。现如今，Java EE 不仅仅是指一种标准平台，它更多地表达着一种软件架构的设计思想。

1.2 Java EE 6 架构

Java EE 是 J2EE 版本的后续版本，是 J2EE 技术的新生和发展。Java EE 技术具有 J2SE 平台

的所有功能，同时还提供对 EJB、Servlet、JSP、XML 等技术的全面支持。Java EE 的最终目标是成为一个支持企业级应用开发的体系结构，简化企业解决方案的开发、部署和管理等复杂问题。事实上，Java EE 已经成为企业级开发的工业标准和首先平台。

根据 Java EE 规范的定义，Java EE 平台是由一系列容器、应用组件和 API 服务所组成的。这些组件和 API 服务本身也是由 JCP 或其他组织所制定的规范而定义的。因为业务逻辑被组织成了可重用的组件，所以基于组件和平台独立的 Java EE 架构使得 Java EE 应用程序易于编写。另外，Java EE 服务器以容器的形式为每种组件类型都提供了基本的服务。由于不需要自行开发这些服务，使得开发人员能够关注于解决业务问题。Java EE 6 平台包含的元素以及它们之间的关系如图 1-1 所示。

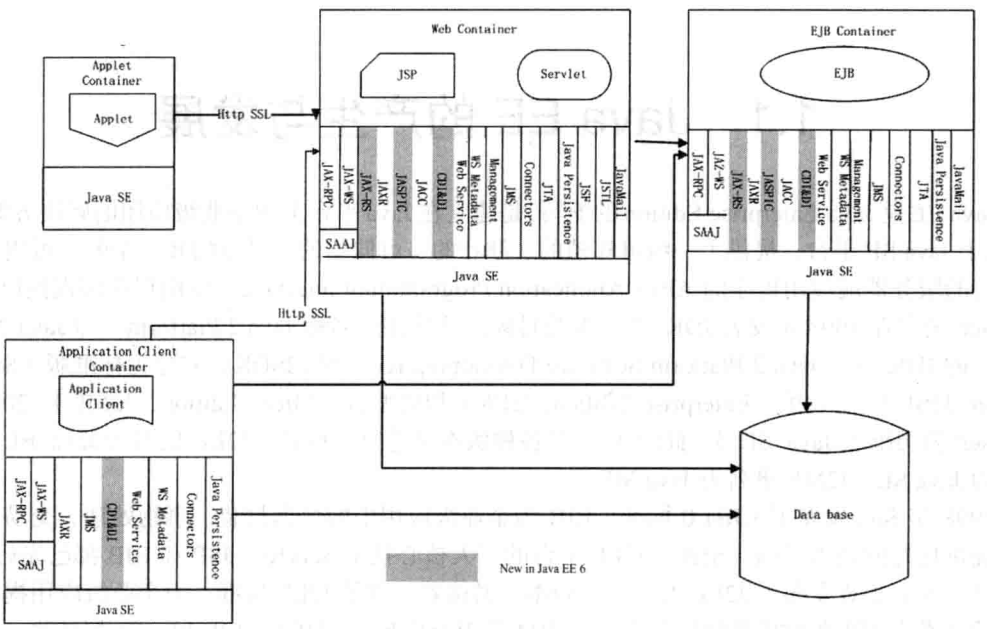


图 1-1　Java EE 6 平台

从图 1-1 中可以看出，Java EE 架构主要包含 4 种容器，分别为应用客户端容器（Application Client Container）、Web 容器（Web Container）、Applet 容器（Applet Container）和 EJB 容器（EJB Container）。容器是指为各种应用组件提供 API 服务的 Java EE 运行时环境，Web 组件（如 JSP、Servlet）、EJB 组件、Applet 组件和应用程序客户组件必须组装成 Java EE 模块并且发布到容器中才能够运行。在应用客户端容器中运行的应用组件主要是指各类桌面 Java 应用程序；在 Applet 容器中运行的应用组件主要是指各种浏览器 Applet。在 Web 容器中运行的应用组件包含可响应 HTTP 请求的 Servlet 和 JSP。EJB 容器中运行的是各种 EJB 组件。

另外容器可提供诸如目录服务、事务管理、安全性、资源缓冲池及容错性等各种可配置的公共服务。Java EE 安全模块允许我们配置一个 Web 组件或 EJB 组件使得只有授权的用户才可以访问系统资源。Java EE 事务模块使得我们可以指定组成一个事务的方法之间的关系，这样一个事务中的所有的方法被作为一个整体处理。JNDI 查找服务为企业中的多个命名和目录服务提供了统一的接口，这样应用程序组件可以很容易地访问这些服务。Java EE 远程连接模块管理客户和 EJB 组件之间的底层通信。当一个 EJB 组件被创建之后，客户端可以像调用同一个 JVM 中的对象一样调用它。

因为 Java EE 架构提供了可配置的服务，同一个 Java EE 应用程序中的不同的应用程序组件可

以有不同的行为方式。例如，可以配置一个 EJB 组件，在一个产品环境中可以拥有某个层次的数据库访问权限，在另外一个产品环境中，可以配置其拥有另外层次的数据库访问权限。

容器同时还管理一些不可配置的服务，如 servlet 和 EJB 的生命周期、数据库连接资源池、数据持久化和访问 Java EE 平台的 API 等。

1.3　Java EE 6 常用技术

1. JDBC

JDBC（Java Database Connectivity，Java 数据库连接）是一种用于执行 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

JNDI（Java Name and Directory Interface, Java 命名和目录接口）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. Servlet

Servlet 技术规范是 Java EE 技术规范中的一个重要组成部分。Servlet 是一种独立于平台和协议的服务器端的 Java 应用程序，可以生成动态的 Web 页面。

4. JSP

JSP（Java Server Pages，Java 服务器页面）是由 Sun 公司倡导、许多公司参与一起建立的一种动态网页技术标准。JSP 技术有点类似 ASP、PHP 等技术，它是在传统的网页 HTML 文件（*.htm,*.html）中插入 Java 程序段（Scriptlet）和 JSP 标记（tag），从而形成 JSP 文件（*.jsp）。运行 JSP，也是需要 Servlet 容器的，原因就是 JSP 在第一次被访问的时候，会被翻译为一个 Servlet。所以，JSP 是在 Servlet 的技术上构建出来的，相比传统的 ASP 脚本的解释方式，JSP 的运行速度快了许多。

5. JSF

JSF （Java Server Faces，Java 构建框架）是一种用于构建 Web 应用程序的 Java 框架，是 Java EE 表示层的技术，其主旨是为了使 Java 开发人员能够快速地开发基于 Java 的 Web 应用程序。它不同于其他 Java 表示层技术的最大优势是其采用的组件模型和事件驱动，确保了应用程序具有更高的可维护性。

6. EJB

EJB（Enterprise JavaBean，Java EE 服务器端组件模型）提供了一个框架来开发和实施分布式商务逻辑，由此显著地简化了具有可伸缩性和高度复杂的企业级应用开发。EJB 规范定义了 EJB 组件在何时如何与它们的容器进行交互作用。

7. JMS

JMS（Java Message Service，Java 消息服务）是用于和面向消息的中间件相互通信的应用程序接口（API）。它既支持点对点的消息模型，也支持发布/订阅的消息模型。

8. RMI

RMI（Remote Method Invoke，远程方法调用）定义了调用远程对象上的方法的标准接口。它是一种被 EJB 使用的更底层的协议，通过使用序列化方式在客户端和服务器端直接传递数据。

9. JTA

JTA（Java Transaction Architecture，Java 事物架构）定义了面向分布式事务服务的标准 API，

可支持事物范围的界定、事物的提交和回滚。

10. JavaMail

许多应用程序需要发送邮件的功能，因此 Java EE 平台包含了 JavaMail API 以及相应的 JavaMail 服务供应商 API，使应用程序组件可以发送邮件。JavaMail API 有两个部分：一个是应用程序组件用于发送邮件的应用程序级接口，另一个是 Java EE SPI 级的服务供应商接口。

11. Web Service

Web Service 是一种通过 WWW 的 HTTP 进行交互和交流的方式，使得运行在不同的平台和框架的软件应用程序之间可以进行互操作。Web Service 可以以松耦合的方式完成复杂的操作，具有强大的互操作能力和可扩展能力。

1.4 Java EE 6 特性

Java EE 6 平台的最主要的目的就是通过提供 Java EE 平台中的各个不同组件的通用的功能来简化开发。通过更多的标注（Anotation）、更少的 XML 配置、更多的 POJO 和简化的打包，使开发人员能够得到更高的生产效率。Java EE 6 平台包含了以下的新特性。

1. JAX-RS

RESTful Web服务是按照REST架构风格构建的Web服务，是比基于SOAP消息的Web Service简单的多的一种轻量级 Web 服务。JAX-RS（RESTful Web Services Java API）为在 Java 中构建 RESTful Web 服务提供了标准化 API，它包括一组标注，以及相关的类和接口。POJO 应用通过使用标注暴露 Web 资源，这个方法使得在 Java 中创建 RESTful Web 服务变得简单。JAX-RS 1.0 技术规范定稿于 2008 年 10 月，包括了一个参考实现 Jersey，Java EE 6 包括了这个技术规范的最新版本 JAX-RS 1.1，这个版本与 Java EE 6 中的新特性保持一致。

2. 托管 Bean

JSF 使用 JavaBean 来达到程序逻辑与视图分离的目的，称为托管 Bean，其作用是在真正的业务逻辑 Bean 及 UI 组件之间搭起桥梁，在托管 Bean 中会调用业务逻辑 Bean 处理使用者的请求，或者是将业务处理结果放置其中，等待 UI 组件取出当中的值并显示结果给使用者。

3. 上下文和依赖注入

上下文和依赖注入（CDI）是新的 Java EE 6 规范，它不仅定义了功能强大、类型安全的依赖注入，而且还引入“上下文”，添加了作用域的概念。CDI 是 Java EE 平台的 Web 层和企业层之间的一座桥梁，企业层通过如 EJB 和 JPA 等技术，对事务性资源提供了强有力的支持。例如，使用 EJB 和 JPA，你可以轻松地构建与数据库交互的应用程序，在数据上提交或回滚事务，以及持久化数据。相比之下，Web 层重点是展示。Web 技术，如 JSF 和 JSP，提供用户界面，显示它的内容，但 Web 技术没有集成处理事务资源的工具。通过 CDI 提供的服务，使 Web 层也支持事务，这样在 Web 应用程序中访问事务资源就更容易了。例如，CDI 使得构建一个用 JPA 提供的持久化访问数据库的 Java EE Web 应用程序变得更容易了。

4. Bean 验证规范

验证数据是应用程序生命周期中一个常见的任务，例如，在应用程序的表示层，你可能想验证用户在文本框中输入的字符数最多不超过 20 个，或者想验证用户在数字字段输入的字符只能是数字。开发人员在应用程序的各层中通常使用相同的验证逻辑，或者将验证逻辑放在数据模型中。Java EE 架构中 Bean 验证（JSR 303）提供了一个标准的验证框架，在框架中相同的验证集可以在应用程序的所有层之间共享，因此使验证变得更简单了，减少了重复、错误和凌乱的现象。

5. JASPIC

Java 容器认证服务提供者接口（Java Authentication Service Provider Interface for Containers，JASPIC）规范定义了服务提供者接口（Service Provider Interface，SPI），通过该接口实现消息认证机制的认证提供者可以集成到客户端或服务器端的容器或运行时刻库中。通过该接口集成的认证提供者对调用它们的容器发出的网络消息进行处理，对发出的消息进行变换以保证接收容器能对该消息通过其认证，同时为了保证接收方返回的回执也能被发送方认证，认证服务提供者除了对进入的消息进行认证以外，还要向发出方返回其身份以建立互信。容器认证服务提供者接口是 Java EE 6 平台新引入的功能，目前的版本为 JASPIC 1.0。

6. EJB 3.1

EJB3.0 本地客户端视图是基于普通旧式 Java 接口（POJI）调用本地业务接口的，本地接口定义了暴露给客户端的业务方法，并要求 Bean 类必须实现此接口。EJB3.1 通过让本地业务接口成为可选组件简化了这个方法，没有本地业务接口的 Bean 暴露的是无接口视图。现在你不用编写独立的业务接口就可以获得相同的企业 Bean 功能，同时添加了单例会话 Bean 以及会话 Bean 的异步调用。

7. Servlet 新特性

Servlet 3.0 作为 Java EE 6 规范体系中一员，随着 Java EE 6 规范一起发布。该版本在前一版本（Servlet 2.5）的基础上提供了若干新特性用于简化 Web 应用的开发和部署。

Servlet 3.0 提供了异步处理模式。在接收到请求之后，Servlet 线程可以将耗时的操作委派给另一个线程来完成，自己在不生成响应的情况下返回至容器。针对业务处理较耗时的情况，这将大大减少服务器资源的占用，并且提高并发处理速度。

Servlet 3.0 新增了若干标注，用于简化 Servlet、过滤器（Filter）和监听器（Listener）的声明，这使得 web.xml 部署描述文件从该版本开始不再是必选的了。另外，开发者可以通过插件的方式很方便地扩充已有 Web 应用的功能，而不需要修改原有的应用。

8. JavaServer Faces 组件新特性

Java EE 6 也使用了新的 JSF 2.0 标准。JavaServer Faces 技术提供了一个服务端组件框架，简化了 Java EE 应用程序用户界面的开发，其中最显著的改进是页面制作，通过使用标准的 JavaServer Faces 视图声明语言（JavaServer Faces View Declaration Language，俗称 Facelets）使得创建一个 JSF 页面更加容易。

1.5　Java EE 6 应用服务器介绍

实现了 Java EE 规范的服务器软件称为 Java EE 应用服务器软件，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。由于所有的厂商开发的 Java EE 应用服务器软件都支持统一的 Java EE 规范，因此在某个 Java EE 应用服务器软件上运行的 Java EE 应用软件可以不加修改地移植到另外一个 Java EE 应用服务器软件上，从而实现“一次开发，到处运行”的目标。

目前，市场上主流的 Java EE 应用服务器软件包括以下几种。

1. WAS

WAS 是 IBM WebSphere Application Server 的简称，它是 IBM WebSphere 软件平台的基础和面向服务的体系结构的关键构件。WebSphere Application Server 提供了一个丰富的应用程序部署环境，其中具有全套的应用程序服务，包括用于事务管理、安全性、群集、性能、可用性、连接性和可伸缩性的功能。它与 Java EE 兼容，并为可与数据库交互并提供动态 Web 内

容的 Java 组件、XML 和 Web 服务提供可移植的 Web 部署平台。目前，IBM 推出的 WAS 版本是 8.5。

2. WebLogic

WebLogic 是美国 BEA 公司（现已被 Oracle 公司收购）出品的一个基于 Java EE 规范的应用服务器软件，后来 BEA 被 Oracle 收购，WebLogic 自然也就归到 Oracle 旗下了。目前最新版本为 Oracle WebLogic Server 12c，它是适用于云环境和传统环境的最佳应用服务器。它通过一个轻型开发平台提供最高的性能和可伸缩性，显著简化了部署和管理，并可加快上市速度。

3. JBoss

JBoss 是一个基于 Java EE 规范的开放源代码的应用服务器软件，它通过 LGPL 许可证进行发布，这使得 JBoss 广为流行。2006 年，JBoss 公司被 Redhat 公司收购。2011 年，JBoss 发布了新版本的 JBoss AS 6 应用服务器，该新版本提供了对 Java EE 6 的完整支持。

4. Tomcat

Tomcat 是 Apache 软件基金会的 Jakarta 项目中的一个核心项目，由 Apache、Sun 和其他一些公司及个人共同开发而成。由于有了 Sun 的参与和支持，最新的 Servlet 和 JSP 规范总是能在 Tomcat 中得到体现。因为 Tomcat 技术先进、性能稳定，而且免费，因而深受 Java 爱好者的喜爱并得到了部分软件开发商的认可，成为目前比较流行的 Web 应用服务器。2010 年 6 月 29 日，Apache 基金会发布了 Tomcat 7。Tomcat 7 最大的改进是其对 Servlet 3.0 和 Java EE 6 的支持。目前 Tomcat 最新版本是 8.0。

5. Apusic

金蝶 Apusic 应用服务器是金蝶中间件有限公司开发的基于 Java EE 规范并获得 Java EE 国际认证的 Java 应用服务器软件，是为数不多的国产 Java EE 应用服务器软件的优秀代表之一。Apusic 应用服务器基于各种现有的被广泛接受的工业标准，为企业应用提供了一个可靠、高效的开发、部署和维护的平台。

6. GlassFish

GlassFish 是用于构建 Java EE 应用服务器的开源开发项目的名称，是 Sun 官方提供的一款开源的应用服务器。在 2005 年 6 月，Sun 将 GlassFish 项目的 Web 站点向公众开放，从而发布了 GlassFish 项目。它基于 Sun Microsystems 提供的 Sun Java System Application Server PE 9 的源代码以及 Oracle 贡献的 TopLink 持久性代码。该项目提供了开发高质量应用服务器的结构化过程，以前所未有的速度提供新的功能。

1.6 Java EE 开发环境的配置

本文中 Java EE 开发环境的配置以 32 位 Windows XP 操作系统为例。

1.6.1 JDK 7 安装与配置

（1）在 Oracle 官方网站下载 JDK 7u45（网址为 http://www.oracle.com /technetwork/java/javase/ downloads/index.html），如图 1-2 所示。选择“Java Platform (JDK) 7u45”上面图标后，从弹出的列表选择 Windows x86 所对应的 jdk 文件 jdk-7u25-windows-i586.exe 下载，如图 1-3 所示。

图 1-2　下载 JDK 步骤 1

Java SE Development Kit 7u45

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

○ **Accept License Agreement** ◉ **Decline License Agreement**

Product / File Description	File Size	Download
Linux ARM v6/v7 Hard Float ABI	67.67 MB	jdk-7u45-linux-arm-vfp-hflt.tar.gz
Linux ARM v6/v7 Soft Float ABI	67.68 MB	jdk-7u45-linux-arm-vfp-sflt.tar.gz
Linux x86	115.62 MB	jdk-7u45-linux-i586.rpm
Linux x86	132.9 MB	jdk-7u45-linux-i586.tar.gz
Linux x64	116.91 MB	jdk-7u45-linux-x64.rpm
Linux x64	131.7 MB	jdk-7u45-linux-x64.tar.gz
Mac OS X x64	183.84 MB	jdk-7u45-macosx-x64.dmg
Solaris x86 (SVR4 package)	139.93 MB	jdk-7u45-solaris-i586.tar.Z
Solaris x86	95.02 MB	jdk-7u45-solaris-i586.tar.gz
Solaris x64 (SVR4 package)	24.6 MB	jdk-7u45-solaris-x64.tar.Z
Solaris x64	16.23 MB	jdk-7u45-solaris-x64.tar.gz
Solaris SPARC (SVR4 package)	139.38 MB	jdk-7u45-solaris-sparc.tar.Z
Solaris SPARC	98.17 MB	jdk-7u45-solaris-sparc.tar.gz
Solaris SPARC 64-bit (SVR4 package)	23.91 MB	jdk-7u45-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	18.26 MB	jdk-7u45-solaris-sparcv9.tar.gz
Windows x86	123.49 MB	jdk-7u45-windows-i586.exe
Windows x64	125.31 MB	jdk-7u45-windows-x64.exe

图 1-3　下载 JDK 步骤 2

（2）双击安装文件 jdk-7u45-windows-i586.exe，JDK 7 安装程序运行，单击“更改”选项转到“更改当前目标文件夹”窗口，把“文件夹名称”改为 D:\Java\jdk7，即把 JDK 7 安装在 D 盘上，如图 1-4 所示。单击“确定”按钮，返回再单击“下一步”开始安装 JDK。

安装 JDK 后，会跳出 “Java SE Runtime Environment 7 自定义安装”窗口，它将安装 JRE（Java 运行环境），和前一步类似，单击“更改”选项转到“更改当前目的地文件夹”窗口，把“文件夹名称”改为 D:\Java\jre7，如图 1-5 所示。单击“确定”按钮，返回再单击“下一步”后开始安装 JRE。安装完成后显示“安装完成”窗口，单击“完成”按钮，这样 JDK 和 JRE 都安装在 D 盘上了。

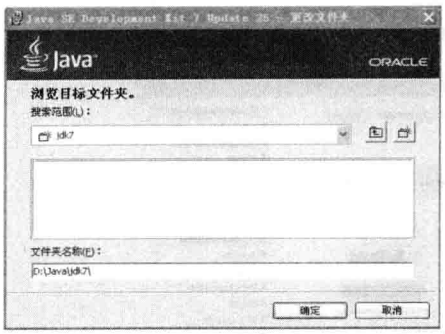


图 1-4　JDK 设置

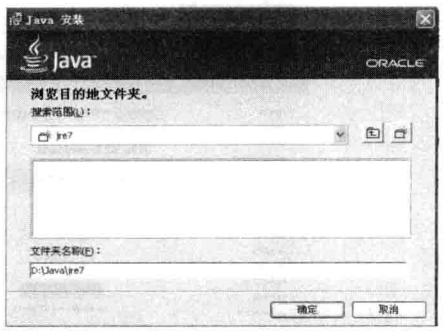


图 1-5　JRE 设置

（3）设置系统变量。右键单击“我的电脑”→“属性”→“高级”→“环境变量”，出现的界面如图 1-6 所示。

① 单击“系统变量”的“新建”，如图 1-7 所示。

图 1-6　WinXP 设置环境变量界面

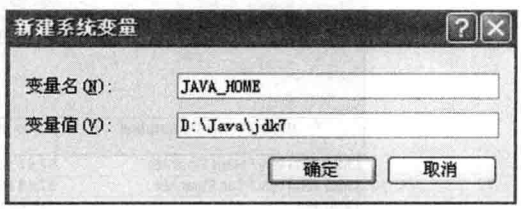


图 1-7　设置 JAVA_HOME 环境变量

在“变量名”中填入 JAVA_HOME，在“变量值”中填入 D:\Java\jdk7（JDK 安装的路径），然后单击“确定”按钮。

② 再单击“系统变量”下面的“新建”，在“变量名”中填入 CLASSPATH，在“变量值”中填入.;%JAVA_HOME%\lib;(注意：在 JAVA_HOME 前面有“.;”)，然后单击“确定”按钮，如图 1-8 所示。

③ 设置 PATH 变量，可以直接在“系统变量”中找到它后双击，没有的话就新建一个。如图 1-9 所示，将 D:\Java\jdk7\bin 目录添加到 PATH 变量中。

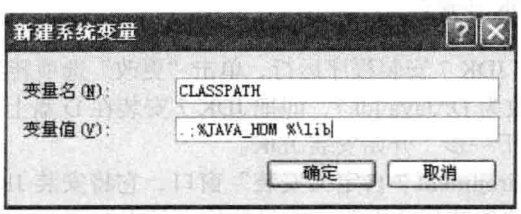


图 1-8　设置 CLASSPATH 环境变量

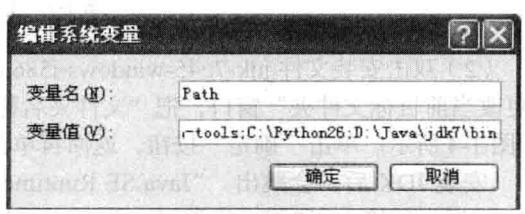


图 1-9　设置 PATH 环境变量

1.6.2　Eclipse IDE 安装

Eclipse 是一个免费的 Java 开发平台，Eclipse 以其代码开源、使用免费、界面美观、功能强大、插件丰富等特性成为 Java 开发中使用最为广泛的开发平台。Eclipse 是一个典型的绿色软件，不需要安装，直接解压到任意文件夹并启动 Eclipse.exe 文件就可以运行 Eclipse。本书中使用的是 Eclipse Jave EE Kepler 版本。下载网址为 http://www.eclipse.org/downloads/，在此页面上选择 Eclipse IDE for Java EE Developers 的 32 位版本下载，下载后文件名为 eclipse-jee-kepler-SR1-win32.zip，将其解压到路径 D:\Eclipse 中。

1.6.3　JBoss AS 7.1.1.Final 安装

JBoss 是目前 Java 市场上应用比较广泛、得到 Sun 认证的 JavaEE 服务器之一，它开源和免费的性质得倒了全球大批专业开发人员的青睐。本书使用的 JBoss 版本为 7.1.1 版本，它支持 Java EE 6 的全部功能。

（1）在 Red Hat 官网上下载 JBoss AS 7.1.1.Final，下载网址为 http://www.jboss.org/jbossas/downloads/，找到 JBoss AS 7.1.1.Final 版本下载，下载后的文件为 jboss-as-7.1.1.Final.zip。

（2）双击下载后的文件 jboss-as-7.1.1.Final.zip，将其解压到目录“D:\jboss7.1.1”，就完成了安装。

（3）设置 JBOSS_HOME 系统变量，如图 1-10 所示。

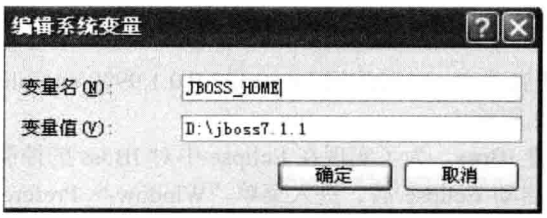


图 1-10　设置 JBOSS_HOME 环境变量

（4）测试 JBoss。运行脚本 D:\jboss7.1.1\bin\standalone.bat 完成启动。访问 http://127.0.0.1:8080/，出现 Welcome to AS 7 访问界面，说明 JBoss 启动成功，如图 1-11 所示。

图 1-11　JBoss 启动界面

（5）设置外网访问。因为 JBoss 安装完成后，默认只能本地访问（即：只有 127.0.0.1/localhost 或 http://localhost:8080 能访问），如果想让其他人也可以访问你的网页，需要修改 JBoss 的配置文件，即修改 standalone.xml，增加本机 Web 地址的内容。

① 打开 D:\jboss7.1.1\standalone\configuration\standalone.xml。

② 在<interfaces>与</interfaces>之间增加一个 interface 节点，内容如下所示。

```
<interface name="any">
```

```
    <any-ipv4-address/>
  </interface>
```

接着，修改以下节点的 default-interface 属性为 any。

```
<socket-binding-group name="standard-sockets" default-interface="any"
                      port-offset="${jboss.socket.binding.port-offset:0}">
    <socket-binding name="management-native" interface="management"
                      port="${jboss.management.native.port:9999}"/>
            ......
</socket-binding-group>
```

修改完后，重启 JBoss7.1 就可以用外网的 IP 来访问了。

（6）添加用户。首次访问 JBoss 服务器时，提示新增用户，可在服务端执行 add-user.bat 来添加管理员用户。add-user.bat 在 JBoss 安装目录的 bin 目录下。

```
add-user.bat
What type of user do you wish to add?
 a) Management User (mgmt-users.properties)
 b) Application User (application-users.properties)
(a):a
回车后
Enter the details of the new user to add.
Realm (ManagementRealm) :                          //回车，选用默认
Username : admin                                   //填写管理员用户名 admin
Password : 123456                                  //填写管理员密码 123456
Re-enter Password :
```

添加管理员后，重新启动 Jboss 后访问 http://127.0.0.1:9990/console，会弹出要求输入用户名和密码的页面，如图 1-12 所示。

（7）在 Eclipse 中配置 JBoss。为了实现在 Eclipse 中对 JBoss 的控制，可以将 JBoss 的启动和停止添加到 Eclipse 中。启动 Eclipse 后，进入菜单“Window-> Preferences-> Server -> Runtime Environments”，如图 1-13 所示。

图 1-12　JBoss 中进入 Console 的用户名和密码页面

图 1-13　进入 JBoss 配置界面

单击“Add”按钮，添加一个新的应用服务器，如图 1-14 所示，选择 JBoss7.1，并将 jre 和 Application Server Directory 进行正确配置，然后单击“finish”按钮。