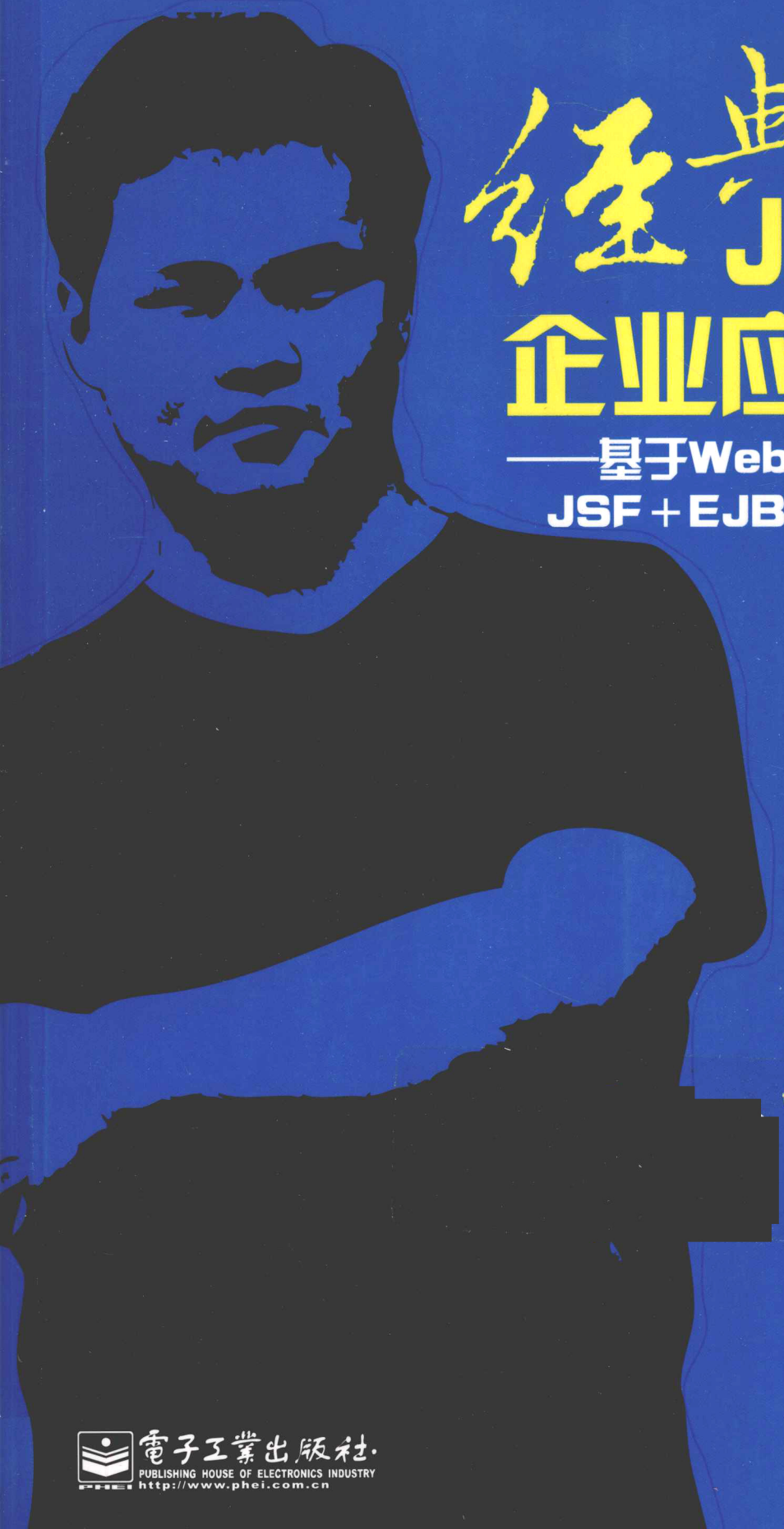
Broadview
www.broadview.com.cn



经典Java EE企业应用实战

——基于WebLogic/JBoss的JSF+EJB 3+JPA整合开发

李刚 编著

疯狂源自梦想

技术成就辉煌

疯狂源自梦想
技术成就辉煌

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

经典Java EE企业应用实战

——基于WebLogic/JBoss的JSF+EJB 3+JPA整合开发

李 刚 编著

電子工業出版社
Publishing House of Electronics Industry
北京·BEIJING

内 容 简 介

本书是《轻量级 Java EE 企业应用实战》的姊妹篇，《轻量级 Java EE 企业应用实战》主要介绍以 Spring+Hibernate 为基础的 Java EE 应用；本书则主要介绍以 EJB 3+JPA 为基础的 Java EE 应用。EJB 3、JPA 规范都属于 Sun 公司所制订的 Java EE 规范，因此把基于 EJB 3+JPA 的应用称为经典 Java EE 架构，目前这种架构在 Java 开发领域也有极大的市场占有率。

本书介绍了 Java EE 规范的三大主要规范 JSF、EJB 3 和 JPA，其中 JSF 是 Sun 公司提供的 JSF RI；EJB 3 部分则包含 Session Bean、Message Driven Bean 的详细介绍。所使用的应用服务器是 JBoss 5.1 和 WebLogic 11g，详细介绍了这两种应用服务器的安装和使用，以及如何在两大主流服务器上安装、部署 Java EE 应用。

本书内容主要包括三部分，第一部分介绍 Java EE 开发的基础知识，以及如何搭建开发环境，包括安装 JBoss、WebLogic 应用服务器，以及如何使用 SVN、NetBeans 等。第二部分详细讲解了 JSF RI、JTA、JNDI、RMI、JMS、JavaMail、EJB 3 的 Session Bean、Message Driven Bean、JPA、JAX-WS 2、JAAS 等 Java EE 知识，这部分知识以 JSF+EJB 3+JPA 整合开发为重点，通过使用 NetBeans IDE 工具上手，带领读者逐步深入 JSF+EJB 3+JPA 整合开发。这部分内容是笔者讲授“疯狂 Java 实训”的培训讲义，也是本书的重点部分。第三部分提供了一个 JSF+EJB 3+ JPA 整合开发的项目：电子拍卖系统。这个项目包括 5 个实体，这 5 个实体之间具有复杂的关联关系，而且业务逻辑也相对复杂，希望让读者理论联系实际，真正将 JSF+EJB 3+JPA 整合真正运用到实际开发中。该案例采用目前最流行、最规范的 Java EE 架构，整个应用分为 JPA 实体层、EAO 层、业务逻辑层、MVC 层和视图层，各层之间分层清晰，层与层之间以松耦合的方法组织在一起。该案例既提供了 IDE 无关的、基于 Ant 管理的项目源码，也提供了基于 NetBeans IDE 的项目源码，最大限度地满足读者的需求。

本书没有介绍 JSP、Servlet 等 Java Web 的相关内容，但这些知识是阅读本书的基础，如果读者还没有相关知识，建议先阅读《轻量级 Java EE 企业应用实战》中相关知识。如果读者在阅读此书时遇到了技术难题，可登录 http://www.crazyit.org 发帖，笔者将会及时予以解答。

阅读本书之前，建议先认真阅读笔者所著的《疯狂 Java 讲义》一书。本书适合有较好的 Java 编程基础，或有较好的 JSP、Servlet 基础的读者阅读。尤其适合于对 JSF、EJB 3、JPA 了解不够深入，或对 JSF+EJB 3+JPA 整合开发不太熟悉的开发人员阅读。

图书在版编目（CIP）数据

经典 Java EE 企业应用实战：基于 WebLogic/JBoss 的 JSF+EJB 3+JPA 整合开发 / 李刚编著. —北京：电子工业出版社，2010.8
ISBN 978-7-121-11534-9

Ⅰ. ①经… Ⅱ. ①李… Ⅲ. ①Java 语言—程序设计 Ⅳ. ①TP312

中国版本图书馆 CIP 数据核字（2010）第 150840 号

责任编辑：张月萍
印　　刷：北京天宇星印刷厂
装　　订：三河市皇庄路通装订厂
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：850×1168　1/16　印张：42.25　字数：1364 千字　彩插：1
印　　次：2010 年 8 月第 1 次印刷
印　　数：4000 册　　定价：79.00 元（含光盘 1 张）

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 zlts@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

目前 Java EE 应用的开发方式大致可分为两种：一种以 Spring、Hibernate 等开源框架为基础，这就是通常所说的轻量级 Java EE 应用；另一种则以 EJB 3+JPA 为基础，也就是本书所介绍的经典 Java EE 应用。在 EJB 3 出现以前，由于 EJB 学习曲线陡峭，使用时也有点困难，因此影响了 EJB 在实际项目中的使用。为此 EJB 3 进行了大刀阔斧的改革，有人说 EJB 3 中的 Session Bean 就像 Spring 容器中的 Bean：只要一个接口和一个实现类即可——其实这句话说反了，应该说 Spring 框架充分借鉴了早期的 EJB 规范，但对 EJB 规范进行了简化，比如它不要求 Bean 继承任何基类，而且 Spring 对 Bean 的要求比较“温柔”：它只是建议面向接口编程；而 EJB 规范则显得很“强硬”：EJB 必须有一个接口和一个实现类。但最终殊途同归：Spring 容器中的 Bean 通常由一个接口和一个实现类组成，EJB 也由一个接口和一个实现类组成。到了 EJB 3 时代，开发 Session Bean 不会比开发 Spring 中的 Bean 更复杂，EJB 3 中的 Session Bean 同样不需要继承任何基类，只要提供一个接口、一个实现类即可，也就是说，EJB 3 规范也吸收了 Spring 框架简单、易用的特性。

Java EE 5 的两个核心规范是 EJB 3 和 JPA，EJB 3 使 Java EE 应用开发变得更加简单；而 JPA 规范则体现了 Sun 公司的良苦用心——Java 开源领域中各种 ORM 框架层出不穷，而 Java EE 开发者则疲于学习各种 ORM 框架：Hibernate 是主流，但下一家公司可能选择其他 ORM 框架，于是开发者不得不重新学习……在这样的背景下，JPA 规范诞生了，JPA 本质上应属于一种 ORM 规范，应用开发者只需要学习 JPA 规范、掌握 JPA API 即可，不需要为使用 Hibernate 学习一套 API，为使用 TopLink 又要重新学习一套 API。开发者面向 JPA 规范编程，而底层则可以在不同 ORM 框架（可理解为 JPA 实现）之间自由切换。通常来说，应用服务器会负责为 JPA 规范提供 ORM 实现；如果开发者希望在 Java SE 应用程序中使用 JPA，这也是允许的，只要开发者自行为 JPA 选择合适的 ORM 实现即可。事实证明，在应用程序中使用 JPA 作为持久化解决方案更方便，而且能在各种 ORM 框架之间自由切换，具有更好的可扩展性。

Java EE 5 规范面世以来，大量开发者重新回归到 EJB 3+JPA 旗下，采用 EJB 3+JPA 开发的企业级应用也越来越多，这也是本书所介绍的企业级应用。除此之外，本书还重点介绍另一个 Java EE 规范：JSF，JSF 作为一个前端 MVC 框架，能与 EJB 3+JPA 完美整合，从而开发出具有高度可扩展性、高度可维护性的企业级应用。

再前言

到现在为止，“疯狂 Java 体系”系列图书已经完成了 5 本，在这几年内，经常收到一些读者邮件，或通过 www.crazyit.org 发帖询问：XXX 图书什么时候可以看到啊？与广大读者期待的心情相比，笔者显得有些不够努力。

不过回过头来看，“疯狂 Java 体系”系列图书的写作其实并不容易，因为这个体系基本覆盖了 Java 学习、工作者的主要相关技术，也覆盖了“疯狂 Java 实训营”的绝大部分课程，因此整个体系非常庞大，而且笔者现在常常有“岁月不饶人”的感觉：以前晚上工作到凌晨 2～3 点，第二天 7 点多起来依然神采奕奕，但现在到了凌晨 1 点就想睡觉了。但只要想到读者殷切的

希望，以及当初的“宏伟”构想：完成疯狂 Java 体系图书，让广大 Java 学习、工作者有一套系统、全面的学习、参考体系，一种巨大的成就感、充实感从心底油然而生，这种感觉鞭策着笔者坚持到底。

今天，疯狂 Java 体系图书已经趋于完成，这套系列图书囊括了 Java 开发领域两种重要的开发方式：①以 Spring+Hibernate 为基础的轻量级 Java EE；②以 EJB 3+JPA 为基础的经典 Java EE。当然，不管学习哪种 Java EE，都应该先打下夯实的 Java 基础，这样学习后面的 Java EE 开发才可以事半功倍。因此笔者并不建议开始就学习 Spring+Hibernate，也不建议开始就学习 EJB 3+JPA，学习还是应该遵循学习规律：先从基础开始，一步一个脚印地学习。

从 2005 年开始创作第一本 Java 技术图书开始，到现在已经过了 5 年多时间，也许很少有人愿意坚持这么长时间来创作 Java 技术图书。认识很多做过多年开发的老程序员，他们往往出版了第一本图书之后，以后再也不写了。因为出版第一本图书可以凭兴趣、凭热情，但出版第一本之后可能会发现：创作一本图书所投入的精力可能远远超出预期，而所获得的金钱回报则远远低于预期，所以大部分都放弃了。但笔者坚持了下来，期间的艰辛只有那些创作过技术图书的人才会懂；那些站在一旁临渊羡鱼、不平妒忌、指手划脚的人是不可能懂的。

这本书同样会让读者感觉到“EJB 3 原来如此简单”、“JPA 原来如此简单”——这也是笔者创作技术图书的一贯原则：用浅显、直白的方式来讲解那些误以为“深奥”的知识，帮助更多有志于软件开发的朋友快速步入实际开发，大大缩短学习周期。

本书有什么特点

本书作为《轻量级 Java EE 企业应用实战》的姊妹篇，两本书在知识体系上互为补充，在知识讲解方式、写作风格上保持一致：两本书同样具有简单、实用的特点，同样坚持为每个知识点都提供配套小实例，以实例为导向，通过实际的实例来介绍各知识点的用法。不仅如此，本书最后还提供了一个基于 JSF+EJB 3+JPA 的电子拍卖项目，方便读者掌握 EJB 3+JPA 在实际项目中的应用，进而在实际开发中熟练运用这种开发架构。

与《轻量级 Java EE 企业应用实战》相似，本书具有如下特点。

1．知识全面，系统性好

本书系统、全面地介绍了 Sun 制订的 Java EE 规范的 JSF、JTA、JMS、JavaMail、Session Bean、Message Driven Bean、JPA、JPQL、JAX-WS 2、JAAS 等规范，而且兼顾 JBoss、WebLogic 两大主流应用服务器。因此学习本书可以全面、深入地掌握 Sun 所制订的经典 Java EE 规范。

2．讲解详细，示范性强

笔者既担任过软件开发的技术经理，也担任过软件公司的培训导师，还从事过职业培训的专职讲师。因此笔者可以对学习、开发中重点及难点进行针对性的详细讲解，并提供配套实例，具有很好的示范性。

3．内容实际，实用性强

本书并不是一本学院派的理论读物，这一点从本书为各知识点所提供的大量实例中即可看出。不仅如此，本书所介绍的 EJB 3+JPA 整合开发采用了严格的分层结构，而不是将各种技术杂乱地糅合在一起号称 Java EE。读者参考本书的架构，完全可以身临其境地感受企业实际开发，并可迅速提升读者对系统架构设计的把握。

本书写给谁看

如果你已经掌握 Java SE 内容，或已经学完了《疯狂 Java 讲义》，而且有一定的 JSP、Servlet 基础，那你比较适合阅读此书。

如果你已经学完《轻量级 Java EE 企业应用实战》一书，阅读本书将非常合适。

如果你对 JSF、EJB 3、JPA 有所了解，但希望掌握它们在实际开发中的应用，本书也将非常适合你。

但如果你对 Java 的掌握还不熟练，或对 JSP、Servlet 一无所知，则建议遵从学习规律，循序渐进，暂时不要购买、阅读此书。

2010-6-20

光盘说明

光盘内容

本光盘是《经典 Java EE 企业应用实战——基于 WebLogic/JBoss 的 JSF+EJB 3+JPA 整合开发》一书的配书光盘，书中的代码按章、按节存放，即第 2 章第 3 节所使用的代码放在 codes 目录的 02\2.3 文件夹下，依此类推。

另：书中每份源代码也给出了与光盘源文件的对应关系，方便读者查找。

本光盘 codes 目录下有 14 个文件夹，其内容和含义说明如下：

（1）02～15 文件夹名对应于《经典 Java EE 企业应用实战——基于 WebLogic/JBoss 的 JSF+EJB 3+JPA 整合开发》中的章名，即第 2 章所使用的代码放在 codes 目录的 02 文件夹下，依此类推。

（2）15 文件夹下有 Auction 和 Auction_NetBeans 两个文件夹，它们是同一个项目的源文件，其中 Auction 是 IDE 平台无关的项目，使用 Ant 来编译即可；而 Auction_NetBeans 是该项目在 NetBeans IDE 工具中的项目文件。

运行环境

本书中的程序在以下环境中调试通过。

（1）安装 jdk-6u18-windows-i586-p.exe，安装完成后，添加 CLASSPATH 环境变量，该环境变量的值为“.;%JAVA_HOME%/lib/tools.jar;%JAVA_HOME%/lib/dt.jar”。如果为了可以编译和运行 Java 程序，还应该在 PATH 环境变量中增加“%JAVA_HOME%/bin”。其中 JAVA_HOME 代表 JDK（不是 JRE）的安装路径。

（2）安装 java_ee_sdk-5_08-windows-ml-nojdk.exe，安装完成后，修改 CLASSPATH 环境变量，修改后该环境变量的值为：.;%JAVA_HOME%\lib\dt.jar;%JAVA_HOME%\lib\tools.jar;D:\Sun\SDK\lib\javaee.jar。

（3）安装 JBoss 5.1.0.GA，直接采用解压缩的安装方式。安装 JBoss 请参看第 1 章。

（4）安装 WebLogic 11g（10.3.2）。安装 WebLogic 请参看第 1 章。

（5）安装 apache-ant-1.7.1。将下载的 Ant 压缩文件解压缩到任意路径，然后增加 ANT_HOME 环境变量，让变量的值为 Ant 的解压缩路径。并在 PATH 环境变量中增加“%ANT_HOME%/bin”环境变量。

（6）安装 MySQL 5.1 或更高版本，安装 MysSQL 时选择 GBK 编码方式。

（7）安装 NetBeans 6.8。

关于如何安装上面工具，请参考本书的第 1 章。

注意事项

（1）独立应用程序的代码中都包括 build.xml 文件，在 Dos 或 Shell 下进入 build.xml 文件所在路径，执行如下命令：

ant compile -- 编译程序

ant run --运行程序

ant build -- 生成 EJB 的 JAR 包或 Java EE 应用的 EAR 包

（2）对于 Web 应用，将该应用复制到“%JBoss_HOME%/server/defaultdeploy”路径下，然后进入 build.xml 所在路径，执行如下命令：

ant compile -- 编译应用

启动 JBoss 服务器，使用浏览器即可访问该应用。

（3）对于 EJB 项目，进入项目中 build.xml 所在的目录下，执行如下命令：

ant build -- 生成 EJB 的 JAR 包

将项目根目录下 build 目录下的 JAR 包复制到应用服务器的自动部署目录下即可。

（4）对于 Java EE 项目，进入项目中 build.xml 所在的目录下，执行如下命令：

ant build -- 生成 Java EE 应用的 EAR 包

将项目根目录下 build 目录下的 EAR 包复制到应用服务器的自动部署目录下即可。

（5）对于 NetBeans 项目文件，导入 NetBeans 开发工具中即可。

（6）第 15 章的案例，请参看项目下的 install.txt 文件。

（7）代码中有大量代码需要连接数据库，读者应修改数据库 URL 以及用户名、密码，让这些代码与读者运行环境一致。如果项目下有 SQL 脚本，导入 SQL 脚本即可；如果没有 SQL 脚本，系统将在运行时自动建表，读者只需创建对应数据库即可。

（8）在使用本光盘中的程序时，请将程序拷贝到硬盘上，并去除文件的只读属性。

四、技术支持

如果您使用本光盘中遇到不懂的技术问题，您可以登录如下网站与作者联系：

http://www.crazyit.org

目录 CONTENTS

第 0 章　学习 Java 的正确方法 …… 1
0.1　我适合不适合编程 …… 2
0.2　走出象牙塔 …… 4
0.3　学习 Java，应该如此疯狂 …… 7

第 1 篇　基础知识

第 1 章　经典 Java EE 应用和开发环境 …… 9
1.1　经典 Java EE 应用概述 …… 10
1.1.1　Java EE 6 相关规范 …… 10
1.1.2　经典 Java EE 应用的分层模型 …… 11
1.1.3　经典 Java EE 应用的组件 …… 13
1.1.4　经典 Java EE 应用架构的优势 …… 13
1.1.5　常用的企业服务器 …… 14
1.2　经典 Java EE 应用相关技术 …… 14
1.2.1　JSP、Servlet 和 JavaBean 及替代技术 …… 14
1.2.2　JSF 及替代技术 …… 15
1.2.3　EJB 组件技术简介 …… 15
1.3　JBoss 的下载和安装 …… 15
1.3.1　下载和安装 JBoss 服务器 …… 16
1.3.2　配置 JBoss 的服务端口 …… 18
1.3.3　进入控制台 …… 19
1.3.4　部署 Web 应用 …… 22
1.4　WebLogic 的下载和安装 …… 24
1.4.1　WebLogic 的下载和安装 …… 24
1.4.2　WebLogic 的基本配置 …… 25
1.4.3　修改 WebLogic 的服务端口 …… 29
1.4.4　部署 Web 应用 …… 30
1.5　NetBeans 的安装和使用 …… 32
1.5.1　NetBeans 的下载和安装 …… 32
1.5.2　使用 NetBeans 开发 Java EE 应用 …… 33
1.5.3　打开 NetBeans 项目 …… 36
1.5.4　导入 Eclipse 项目 …… 37
1.6　使用 SVN 进行协作开发 …… 38
1.6.1　下载和安装 SVN 服务器 …… 39
1.6.2　配置 SVN 资源库 …… 39
1.6.3　下载和安装 SVN 客户端 …… 41
1.6.4　发布项目到服务器 …… 41
1.6.5　从服务器下载项目 …… 42
1.6.6　提交（Commit）修改 …… 42
1.6.7　同步（Update）本地文件 …… 43
1.6.8　添加文件和目录 …… 44
1.6.9　删除文件和目录 …… 45
1.6.10　查看文件或目录的版本变革 …… 45
1.6.11　从以前版本重新开始 …… 46
1.6.12　创建分支 …… 46
1.6.13　沿着分支开发 …… 46
1.6.14　合并分支 …… 47
1.6.15　使用 NetBeans 作为 SVN 客户端 …… 48
1.7　本章小结 …… 50

第 2 篇　整合开发

第 2 章　JSF 的基本用法 …… 51
2.1　MVC 和 JSF …… 52
2.1.1　MVC 和常见 MVC 框架 …… 52
2.1.2　JSF 的优势 …… 56
2.2　下载和安装 JSF …… 57
2.3　JSF 使用入门 …… 60
2.3.1　从输入页面开始 …… 60
2.3.2　开发托管 Bean …… 61
2.3.3　定义导航规则 …… 63
2.4　解读 JSF 配置 …… 65
2.4.1　配置核心控制器（FacesServlet） …… 65
2.4.2　JSF 配置文件结构 …… 68
2.5　托管 Bean 和表达式语言 …… 69
2.5.1　托管 Bean 的属性和表达式语言 …… 69
2.5.2　托管 Bean 的方法 …… 78
2.5.3　托管 Bean 的分类 …… 80
2.5.4　初始化托管 Bean 的属性 …… 85
2.5.5　通过 FacesContext 访问应用环境 …… 89
2.6　导航模型 …… 91
2.6.1　静态导航 …… 93
2.6.2　动态导航 …… 93
2.7　使用 UI 标签创建视图页面 …… 94
2.7.1　UI 标签概述 …… 94
2.7.2　UI 标签的通用属性 …… 97
2.7.3　表单相关标签 …… 98
2.7.4　其他标签 …… 103
2.8　JSF 的运行流程和生命周期 …… 109
2.8.1　恢复视图阶段 …… 110
2.8.2　应用请求值阶段 …… 110

2.8.3 处理输入校验阶段……111
2.8.4 更新模型的值阶段……111
2.8.5 调用应用阶段……111
2.8.6 生成响应阶段……111
2.9 利用 JSF 的消息……112
2.10 本章小结……116
第 3 章 深入使用 JSF……117
3.1 JSF 事件机制……118
3.1.1 Java 事件模型概述……118
3.1.2 Java 事件模型示例……119
3.1.3 JSF 事件模型……120
3.1.4 Action 事件……122
3.1.5 值改变事件……125
3.1.6 生命周期事件……128
3.1.7 将监听器绑定到 Bean 属性……130
3.2 JSF 的国际化支持……132
3.2.1 加载国际化资源文件……132
3.2.2 使用国际化消息……134
3.2.3 动态数据国际化……137
3.2.4 让用户选择语言……142
3.3 使用转换器完成类型转换……143
3.3.1 转换器概述、用途……144
3.3.2 JSF 内建转换器……144
3.3.3 使用转换器……145
3.3.4 转换失败后的错误消息……149
3.4 自定义转换器……154
3.4.1 实现转换器类……154
3.4.2 注册转换器……156
3.4.3 使用自定义转换器……159
3.4.4 绑定到 Bean 属性的转换器……159
3.5 使用验证器进行输入校验……161
3.5.1 输入校验概述……161
3.5.2 JSF 内置校验器……162
3.5.3 校验失败后的错误消息……163
3.5.4 必填校验器……165
3.6 自定义校验器……166
3.6.1 开发自定义校验器……166
3.6.2 注册校验器……167
3.6.3 使用自定义校验器……168
3.6.4 为自定义校验器开发专用标签……169
3.6.5 使用托管 Bean 的方法执行校验……173
3.6.6 绑定到 Bean 属性的校验器……175
3.7 本章小结……177
第 4 章 利用 JDBC 和 JTA 访问数据库和管理全局事务……178
4.1 JDBC 和容器管理的数据源……179
4.1.1 JDBC 概述……179
4.1.2 使用 JDBC 执行数据库访问……180
4.1.3 使用 WebLogic 服务器管理的数据源……182
4.1.4 使用 JBoss 服务器管理的数据源……187
4.2 事务和 JTA……191
4.2.1 事务的基本概念……191
4.2.2 分布式事务处理、XA 规范和 2PC 协议……192
4.2.3 使用 JTA 全局事务保证多数据库的一致性……193
4.3 事务隔离、传播属性的设置……198
4.3.1 并发访问和隔离……198
4.3.2 事务属性……199
4.4 EJB 的事务管理……201
4.4.1 容器管理事务（CMT）……201
4.4.2 Bean 管理事务（BMT）……201
4.5 事务超时设置……201
4.6 本章小结……203
第 5 章 JNDI 和远程方法调用……204
5.1 JNDI 的概念……205
5.1.1 命名服务……205
5.1.2 目录服务……206
5.1.3 JNDI 的优点……206
5.2 JNDI 编程入门……207
5.2.1 文件系统的命名服务……207
5.2.2 JNDI 编程……208
5.3 服务器提供的 JNDI 支持……212
5.3.1 WebLogic 的 JNDI 支持……212
5.3.2 JBoss 的 JNDI 支持……215
5.4 RMI 概述……216
5.4.1 RMI 的相关概念……216
5.4.2 RMI 的作用和意义……217
5.5 RMI 编程……217
5.5.1 开发 RMI 服务器……217
5.5.2 开发 RMI 客户端……220
5.5.3 RMI 的基本原理……220
5.6 同时作为客户端和服务器的 RMI 程序……222
5.6.1 开发客户端程序……222
5.6.2 开发服务器端程序……223

5.7 本章小结 …… 225
第 6 章 利用 JMS 实现企业消息处理 …… 226
6.1 面向消息的架构和 JMS 概述 …… 227
6.1.1 面向消息的应用架构 …… 227
6.1.2 JMS 的基础知识和优势 …… 228
6.1.3 JMS 的两个重要版本 …… 229
6.2 PTP 类型的 JMS …… 230
6.2.1 配置 PTP 的 JMS 服务器 …… 231
6.2.2 PTP 消息的发送 …… 241
6.2.3 PTP 消息的同步接收 …… 244
6.2.4 PTP 消息的异步接收 …… 246
6.3 Pub-Sub 类型的 JMS …… 248
6.3.1 配置 Pub-Sub 模型的 JMS 服务器 …… 248
6.3.2 消息的生产、消费 …… 250
6.3.3 可靠的 JMS 订阅 …… 251
6.4 JMS 消息 …… 253
6.4.1 JMS 消息类型 …… 253
6.4.2 JMS 消息头和消息属性 …… 253
6.4.3 重用消息对象 …… 254
6.4.4 JMS 传递方式和有效时间 …… 255
6.4.5 设置消息的优先级 …… 256
6.4.6 消息的确认方式 …… 256
6.4.7 消息选择器 …… 257
6.4.8 消息的临时目的 …… 261
6.5 使用队列浏览器查看全部消息 …… 264
6.6 JMS 和事务 …… 265
6.6.1 使用事务性 Session …… 265
6.6.2 利用 JTA 全局事务 …… 267
6.7 JMS 服务器的异常监听 …… 268
6.8 JMS 集群 …… 269
6.9 本章小结 …… 270
第 7 章 利用 JavaMail 实现 E-mail …… 271
7.1 E-mail 简介 …… 272
7.1.1 SMTP 协议简介 …… 272
7.1.2 POP3 协议简介 …… 272
7.1.3 IMAP4 协议简介 …… 273
7.1.4 E-mail 的用途 …… 273
7.2 JavaMail 介绍 …… 274
7.2.1 JavaMail 下载和安装 …… 274
7.2.2 JavaMail 的常用 API …… 275
7.2.3 使用 JavaMail 发送邮件 …… 277
7.2.4 使用 JavaMail 接收邮件 …… 281
7.3 应用服务器的 JavaMail 支持 …… 287
7.3.1 在 WebLogic 中配置 JavaMail …… 288
7.3.2 通过 WebLogic 的邮件支持来发送邮件 …… 290
7.3.3 在 JBoss 中配置 JavaMail …… 292
7.4 本章小结 …… 294
第 8 章 会话 EJB …… 295
8.1 EJB 概述 …… 296
8.1.1 EJB 的概念和意义 …… 296
8.1.2 EJB 的发展历史 …… 298
8.1.3 EJB 的优势和使用场景 …… 299
8.2 EJB 的分类 …… 301
8.2.1 Session Bean 的概念和作用 …… 302
8.2.2 Message Driven Bean 的概念和作用 …… 303
8.2.3 实体和 JPA …… 303
8.3 开发无状态的 Session Bean …… 304
8.3.1 开发远程调用的无状态 Session Bean …… 304
8.3.2 开发本地调用的无状态 Session Bean …… 311
8.4 发布 Session Bean …… 314
8.4.1 打包 EJB-JAR …… 315
8.4.2 Annotation 与部署描述文件 …… 315
8.5 开发有状态的 Session Bean …… 318
8.6 Session Bean 的生命周期 …… 321
8.6.1 无状态 Session Bean 的生命周期 …… 321
8.6.2 有状态 Session Bean 的生命周期 …… 322
8.6.3 定制 Session Bean 的生命周期行为 …… 323
8.7 在 Session Bean 中使用事务 …… 327
8.7.1 容器管理事务 …… 327
8.7.2 Bean 管理事务 …… 330
8.8 拦截器 …… 332
8.9 依赖注入 …… 335
8.9.1 EJB 注入 …… 336
8.9.2 资源注入 …… 339
8.10 配置 EJB 引用 …… 340
8.11 使用计时器进行任务调度 …… 342
8.12 本章小结 …… 345
第 9 章 消息驱动 EJB …… 346
9.1 JMS 和 EJB …… 347
9.1.1 为什么使用 MDB …… 347
9.1.2 使用 MDB 的设计原则 …… 348
9.2 使用消息驱动 Bean …… 349

9.2.1 使用@MessageDriven 和 @ActivationConfigProperty …… 350
9.2.2 实现 MessageListener …… 352
9.2.3 MDB 的生命周期 …… 353
9.2.4 MDB 中的依赖注入 …… 356
9.2.5 事务管理和异常处理 …… 359
9.3 使用 NetBeans 开发 EJB …… 359
9.3.1 使用 NetBeans 开发 Session Bean … 359
9.3.2 使用 NetBeans 开发 MDB …… 362
9.4 本章小结 …… 363
第 10 章 Java 持久化 API（JPA） …… 364
10.1 实体简介 …… 365
10.1.1 对象/关系数据库映射（ORM） … 365
10.1.2 JPA 的映射规则 …… 367
10.1.3 JPA 规范简介 …… 368
10.2 实体入门 …… 368
10.2.1 开发实体 …… 369
10.2.2 在 Java SE 环境下使用 Hibernate JPA 实现 …… 370
10.2.3 在 Java SE 环境下使用 TopLink JPA 实现 …… 374
10.2.4 在 Java SE 环境下使用 EntityManager …… 377
10.2.5 使用 orm.xml 管理 O/R 映射 …… 379
10.3 理解实体 …… 382
10.3.1 持久化上下文和持久化单元 …… 382
10.3.2 实体类的要求 …… 382
10.3.3 实体的状态 …… 383
10.3.4 管理实体的方法 …… 384
10.4 实体的基本映射 …… 387
10.4.1 映射实体类的属性 …… 387
10.4.2 将实体映射到多个表 …… 394
10.4.3 映射复合类型的属性 …… 396
10.4.4 映射实体类的主键 …… 398
10.5 关联关系映射 …… 402
10.5.1 单向 $N-1$ 关联 …… 403
10.5.2 单向 1—1 关联 …… 406
10.5.3 单向 $1-N$ 关联 …… 409
10.5.4 单向 $N-N$ 关联 …… 411
10.5.5 双向 1—1 关联 …… 414
10.5.6 双向 $1-N$ 关联 …… 417
10.5.7 双向 $N-N$ 关联 …… 420
10.5.8 使用 Map 集合记录关联实体 …… 423
10.5.9 对关联实体进行排序 …… 424
10.6 继承关系映射 …… 426
10.6.1 整个类层次对应一张表的映射策略 …… 427
10.6.2 连接子类的映射策略 …… 430
10.6.3 每个具体类对应一张表的映射策略 …… 434
10.7 使用抽象实体和非实体父类 …… 436
10.7.1 抽象实体 …… 436
10.7.2 非实体父类 …… 438
10.7.3 重定义子类实体的外键列 …… 440
10.8 实体的生命周期和监听器 …… 444
10.8.1 实体的生命周期与回调事件 …… 444
10.8.2 使用专门的监听器实现回调 …… 448
10.8.3 为全部实体配置默认监听器 …… 450
10.8.4 排除监听器 …… 452
10.9 本章小结 …… 455
第 11 章 JPA 的查询支持 …… 456
11.1 查询 API …… 457
11.1.1 面向对象的 JPQL …… 457
11.1.2 查询 API 简介 …… 457
11.2 执行查询 …… 459
11.2.1 使用 Query 创建查询 …… 459
11.2.2 设置查询参数 …… 459
11.2.3 取得查询结果 …… 460
11.3 JPQL 语法 …… 461
11.3.1 使用 from 子句 …… 462
11.3.2 使用 select 子句 …… 463
11.3.3 查询部分属性 …… 463
11.3.4 查询中使用构造器 …… 464
11.3.5 使用 distinct 排除相同的记录 …… 465
11.3.6 where 子句和条件表达式 …… 466
11.3.7 使用 JPQL 函数 …… 466
11.3.8 多态查询 …… 467
11.3.9 关联和连接 …… 469
11.3.10 使用 order by 进行结果排序 …… 473
11.3.11 JPQL 查询的聚集函数 …… 473
11.3.12 使用 group by 进行分组 …… 474
11.3.13 结果集分页 …… 475
11.3.14 使用子查询 …… 475
11.3.15 命名查询 …… 476
11.4 批量更新和批量删除 …… 478
11.4.1 批量更新 …… 479
11.4.2 批量删除 …… 480
11.5 原生 SQL 查询 …… 481

11.5.1 使用原生 SQL 查询 …… 481
11.5.2 结果集映射和实体查询 …… 482
11.5.3 使用命名的原生 SQL 查询 …… 486
11.5.4 调用存储过程 …… 488
11.6 本章小结 …… 490
第 12 章 Web 层和 EJB 的整合 …… 491
12.1 Java EE 应用的架构 …… 492
12.1.1 SSH 架构的复习和应用架构的思考 …… 492
12.1.2 MVC 层和业务层整合 …… 494
12.1.3 DAO 模式和 EAO 模式 …… 495
12.1.4 使用 Session Facade 模式 …… 499
12.2 从 Web 层访问 Session Bean …… 500
12.2.1 使用依赖注入访问无状态 Session Bean …… 501
12.2.2 通过 EJB 引用访问有状态 Session Bean …… 507
12.2.3 在工具类中调用 Session Bean …… 509
12.3 在 Web 层使用 JPA …… 511
12.3.1 使用容器管理的 EntityManager …… 512
12.3.2 在容器内使用应用程序管理的 EntityManager …… 516
12.3.3 通过 ThreadLocal 在容器外使用安全的 EntityManager …… 518
12.4 基于 JBoss 的 JSF+EJB 3+JPA 整合 …… 522
12.4.1 开发 JPA 实体 …… 522
12.4.2 开发 EAO 对象 …… 524
12.4.3 开发业务逻辑组件 …… 526
12.4.4 定义 JSF 的托管 Bean 来处理请求 …… 527
12.5 基于 WebLogic 的 Struts 2+EJB 3+JPA 整合 …… 531
12.5.1 开发实体并配置持久化单元 …… 531
12.5.2 开发 EAO 组件和业务逻辑组件 …… 532
12.5.3 配置 EJB 引用 …… 532
12.5.4 定义 Action 来处理用户请求 …… 532
12.6 本章小结 …… 536
第 13 章 EJB 和 Web Service …… 537
13.1 Web Service 概述 …… 538
13.1.1 Web Service 概述 …… 538
13.1.2 Web Service 平台概述 …… 539
13.1.3 Web Service 的广泛应用 …… 542
13.2 JAX-WS：Java EE 5 Web Service 平台 …… 543
13.2.1 Java EE 的 Web Service 支持 …… 543
13.2.2 为什么选择 EJB 开发 Web Service …… 544
13.3 使用 JAX-WS 2.0 开发 Web Service …… 545
13.3.1 使用@WebService …… 545
13.3.2 使用@WebMethod …… 551
13.3.3 使用@SOAPBinding 指定 Web Service 风格 …… 553
13.3.4 使用@WebParam …… 555
13.3.5 使用@WebResult …… 556
13.3.6 使用@OneWay …… 558
13.4 客户端调用 Web Service …… 558
13.5 本章小结 …… 559
第 14 章 利用 JAAS 开发安全的应用 …… 560
14.1 JAAS 概述 …… 561
14.1.1 Java EE 应用的安全概述 …… 561
14.1.2 安全域、用户、组和角色概念 …… 561
14.1.3 JAAS 的基本流程 …… 563
14.2 管理服务器的用户和组 …… 565
14.2.1 管理 WebLogic 服务器上的用户和组 …… 565
14.2.2 管理 JBoss 服务器上的用户和角色 …… 573
14.2.3 使用 RDBMS 管理 Jboss 服务器上的用户和角色 …… 574
14.3 开发安全的 Web 应用 …… 576
14.3.1 声明安全性 …… 576
14.3.2 映射安全角色 …… 579
14.3.3 基于 JBoss 服务器的表单登录 …… 580
14.3.4 基于 WebLogic 服务器的安全角色映射 …… 583
14.3.5 基于 WebLogic 服务器的表单登录 …… 583
14.3.6 Web 应用中编程式安全 …… 584
14.4 开发安全的 Java EE 应用 …… 585
14.4.1 为 EJB 声明安全性 …… 585
14.4.2 应用客户端访问被保护的方法 …… 587
14.4.3 使用 Web 组件调用 EJB 被保护的方法 …… 588
14.4.4 EJB 中编程式安全 …… 592
14.5 使用 SSL 建立安全连接 …… 592

14.5.1 SSL 基础知识 …… 593
14.5.2 安装和配置 SSL 支持 …… 593
14.5.3 在配置描述符中指定安全连接 …… 595
14.6 本章小结 …… 597

第 3 篇 应用实践

第 15 章 电子拍卖系统 …… 598
15.1 系统功能简介和架构设计 …… 599
15.1.1 系统功能简介 …… 599
15.1.2 系统架构设计 …… 599
15.2 持久层设计 …… 600
15.2.1 系统实体 …… 601
15.2.2 系统 E-R 图和数据表 …… 601
15.2.3 实现 JPA 实体 …… 603
15.2.4 管理持久化单元 …… 612
15.3 实现系统 EAO 层 …… 614
15.3.1 实现 EAO 基类 …… 615
15.3.2 实现系统 EAO 组件 …… 618
15.4 实现业务逻辑层 …… 622
15.4.1 定义业务逻辑组件接口 …… 622
15.4.2 依赖注入 EAO 组件 …… 624
15.4.3 业务逻辑组件中的异常处理 …… 625
15.4.4 处理用户竞价 …… 627
15.4.5 判断拍卖物品状态 …… 631
15.4.6 事务管理 …… 633
15.5 实现系统 Web 层 …… 633
15.5.1 安装 JSF …… 633
15.5.2 处理用户登录 …… 634
15.5.3 图形验证码 …… 639
15.5.4 登录控制 …… 642
15.5.5 添加物品 …… 643
15.5.6 处理用户竞价 …… 650
15.6 使用 SiteMesh 页面装饰 …… 656
15.6.1 在 Web 应用中安装 SiteMesh …… 656
15.6.2 定义页面装饰 …… 657
15.7 本章小结 …… 659

CHAPTER

0

第0章 学习Java的正确方法

本章要点

- 我适合不适合编程
- 走出象牙塔
- 学习Java，应该如此疯狂

就目前国内软件业的现状来看，有一种非常奇怪的现状：一方面，大量大学应届毕业生非常乐意进入软件开发行业，但往往苦于没有公司愿意要；另一方面，又有很多公司在长期招聘，为了找人花去大量的时间开销、人力开销，却找不到合适的人。这些现象表明：国内软件业的从业人数依然太少，但大量愿意投身软件行业的人却不得门而入，关键在于他们在大学里所使用的学习方法存在不足。

当这里提到“国内软件业的从业人数依然太少”这句话时，可能会有人提出反击，甚至担心：每次我找工作面试时总有其他人和我一起竞争，要是程序员更少，没人和我竞争说不定我就可以直接进公司了。笔者曾经有不少学员也表示了类似担心：老师你还是不要继续做培训了，你培训的学生越来越多，到时候他们都会来和我一起抢饭碗，以后我找工作就更难了。

其实这些“准程序员”没有想明白一个问题：他们面试的对手往往并不是与之一起面试的其他人，而是他们自己，他们应该问一问自己是否能真正满足企业的开发要求？他们的开发工作是否可以真正为企业带来价值？如果这两个问题是肯定的，那就可以胜任这份工作；否则将很难满足公司的要求。

那么国内程序员到底是多了呢？还是少了？关于这个问题我们可以看看国外，美国有多少程序员？印度有多少程序员？他们尚且没有担心程序员过剩，我们担心什么？换一个角度来看问题：如果程序员数量太少，这样国内将很难形成规模化的软件产业，这样就业机会就会更少；只有当程序员数量足够多时，才可以形成真正大规模的软件产业，软件产业才可以更好地盈利，从而也可以提供更多的就业机会。

可能有人会说：我们不是有很多“准程序员”吗？这一点没错，可惜这些准程序员离真正可以产生价值的程序员还有一段距离，如果不掌握正确的学习方法，他们也许就会错过。也许有人会说：我们国内程序员大多是“软件蓝领”，只能从事简单的、重复式开发。其实这并不是问题，当“软件蓝领”的数量多起来之后才会产生更多的软件高手，高手是不可能凭空产生的。当“软件蓝领”足够多之后，基本功足够扎实之后，慢慢就会有一部分人突破自己。

本章内容将会帮助“准 Java 程序员”来分析自己的学习方法，找出多年来传统学习方法中可能存在的诸多弊端。也许正是这些不太正确的学习方法最终导致了许多大学毕业生无法满足企业要求，许多初级程序员无法突破自己。

如果你正处于以下这些阶段，阅读本章内容将会对你有不错的帮助。

- 有志于进入 IT 行业的学子，问得最多的问题就是学什么语言好。
- 正在培训机构参加学习（或者上大二、大三），问得最多的是学习曲线，即怎么学。
- 正在求职途中的应届生，希望了解行业概况。
- 已经在行业中，寻求事业上的提升。

0.1 我适合不适合编程

由于眼下紧张的就业形式，越来越多的学生打算进入软件编程行业，对这些苦读十多年的学生而言，也许他们需要的只是一份工作，而不是一个职业。也许头几天他们还在投市场推广方面的简历，但过几天他们就开始投软件开发招聘的职位了。他们并没有考虑自己是否适合软件编程，更没有为从事软件编程做任何准备，也许只是找不到工作时临时想出的权宜之计而已。由于这样的学生不在少数，因此导致现在有些软件公司听到应届毕业生就直接拒绝。

不要看着自己的其他同学、朋友去编程了，而且薪资似乎也还过得去，你就盲目地随大流去选择编程！编程虽然并不是一个高难度的职业，但这个职业还是有一些基本要求的。

关于大学生就业的一个主流观点，也是笔者一直不太认可的，许多关于大学生就业的“专家”认为：大学生应该先就业，再择业。所谓先就业，就是说不管三七二十一，先找一个容身之所待着，至于这份工作是否适合自己则全然不管，也许这就是造成许多大学生就业时相当盲目的理论依据。

可惜就业市场也是残忍的，也许太多大学生抱着权宜之计的想法去软件公司应聘，这样就大大增加了软件公司的招聘成本，最后许多软件公司干脆直接拒绝所有应届毕业生。换一个角度来看，软件

公司拒绝应届毕业生的恶果未尝不是应届毕业生自己种下的。

软件编程这个职业是一个非常实在的职业，说得不好听点，这其实是一个“民工”职业（笔者一向非常尊重民工，民工总在踏实、认真地做着实事，他们的劳动完善、美化着我们身边的世界），程序员必须能开发出满足市场要求的产品，这样才可以为软件公司带来价值。

在“疯狂Java实训营”的授课过程中，笔者曾经尖锐地告诫那些学生，当你们开发手上这些项目时，你们不能用学生的标准来要求自己，你们应该用程序员的标准来要求自己。如果你们达不到程序员的标准，你们怎么去应聘软件公司的职位？即使让你进了一个软件公司，你应该如何为你所在的公司创造价值？如果你不能为公司创造价值，公司从哪里拿钱来给你支付薪水？说得严重一点，如果你不能满足程序员的标准就去应聘软件开发，那就是一种欺诈行为。

当然，笔者并不是想打击那些有志于进入软件行业的“准程序员”的信心，只是想提醒一下大家，当你选择编程之前，至少应该问自己以下几个问题。

- 我是否喜欢按部就班、有条不紊地完成一件事情？
- 以前上学时数学、物理等学科成绩是否还过得去？
- 我是否有不错的逻辑思维能力？
- 我是否可以长时间静下心来专注于某件事情？

如果上面几个问题的答案是肯定的，那么学习软件编程应该会事半功倍；否则学习软件编程可能需要比别人花费更多的时间和精力。

很多开始打算学习软件编程的人在学习编程之前，总是信心满满地做着详细的学习计划，开始总是怀着十二分的热情投入，毕竟软件开发是一件很有成就感的事情，即使只是做出一个小游戏，也会带给学习者无比的快乐。不过所有学习者需要有心理准备的是，软件开发并不是玩游戏，对于一个刚学习编程的学习者来说一定要有心理准备，即使开发一个非常小的程序都有可能遭遇极大的挫折，这就需要学习者能持之以恒地坚持。

注意：

千万不要相信某些图书、某些广告上的宣传口号：10天、20天学会Java编程，10天、20天学会C编程！编程虽然并不是特别难，却需要学习者持之以恒地坚持，10天、20天能入门，能把基本语法掌握就算不错了！如果你想着10天、20天就从一个门外汉学会编程，笔者建议你还是不要浪费时间的好！

好了，如果上面几个问题你的答案是肯定的，而且你还没有被笔者描述的困难吓着，那你可以问一个所有初学者都要问的问题了：选择一门开发语言重要吗？关于这个问题的答案大致可以分为两类。

- 高手型的答案：所有编程语言都是相通的，选择任何编程语言都一样。
- 专一型的答案：XXX语言最优秀，掌握了XXX语言就够了。

这两种回答都有一定的道理，不过笔者的建议是：“深度优先，广度补足”。所谓深度优先指的是建议初学者先选择一门最主流、市场应用最广泛的编程语言，认真、深入地掌握它。这里所说的掌握并不是说熟悉它的基本语法，而是指能使用该语言进行实际项目开发，并对该语言在计算机底层的运行机制都有较深入的体会。当对这门语言有较好的掌握之后，接下来再去补充学习其他编程语言，此时关键是注意其他编程语言与已掌握编程语言之间的差异。

从另外一个角度来看，每个学习编程的学子都有一个亟待解决的问题摆在眼前：就业，因此推荐选择一门主流、市场占有率高的编程语言，如Java、C都是极好的选择。

根据TIOBE网站（tiobe.com）从2002年到2009年的编程语言排行榜上的数据：排在第一位的一直是Java编程语言（除了2004年到2005年之间略低于C语言）；排在第二位的就是C语言了。详细信息可以参考如下页面：http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html。