



刘中兵Java研究室 编著
飞思科技产品研发中心 监制

Java 开发专家



Java Web系统设计与架构

(UML建模 + 设计模式 + 面向服务架构)

全面规划 Java 开发者的技术体系

实力畅销书作者刘中兵 | 开发者突击 | 携 Java 研究室年度奉献

专业论坛技术交流、答疑

随盘赠送全书实例源代码



随书赠送手册一本《为您规划 Java 技术体系》



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

刘中兵Java研究室 编著
飞思科技产品研发中心 监制

Java 开发专家



精 深 透

Master

Java

高手真经

(系统架构卷):

Java Web系统设计与架构

(UML建模 + 设计模式 + 面向服务架构)

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内容简介

本书讲解 JavaWeb 系统设计与架构,包括 Java 与 UML 建模技术、Java 设计模式、面向服务架构 SOA、面向资源架构 ROA、面向云架构 COA,这 5 部分内容层层递进,涵盖了 Java EE 开发中的各种设计与架构技术。讲解的主要内容包括如下。

Java 与 UML 建模:讲解 UML 建模的 5 类 10 种图,包括用例图、静态图(类图、对象图、包图)、实现图(组件图、部署图)、行为图(状态图、活动图)、交互图(序列图、协作图),并结合 Bug 管理系统、ATM 取款机等实例进行讲解。

Java 设计模式:包括 5 种创建型模式(工厂方法模式、抽象工厂模式、单例模式、建造者模式、原型模式)、7 种结构型模式(适配器模式、装饰器模式、代理模式、外观模式、桥接模式、组合模式、享元模式)、11 种行为型模式(策略模式、模板方法模式、观察者模式、迭代子模式、责任链模式、命令模式、备忘录模式、状态模式、访问者模式、中介者模式、解释器模式),并通过 Java 中的经典案例进行展示。

面向服务架构 SOA:讲解 Web 服务协议(SOAP、WSDL、UDDI)、Web 服务开发技术(使用 JAX-WS、Axis 开发 Web 服务)、SOA 架构技术(Sun 公司的 JBI 和 IBM/BEA 公司的 SCA/SDO、OSGi 架构)、SOA 延伸技术(BPEL 业务流程管理、ESB 企业服务总线与 EAI 企业应用集成)、SOA 扩展应用(Portal 门户组件与 Portlet 窗口组件)。

面向资源架构 ROA:讲解 REST 产生的背景、概念、基本原理、设计方法,以及常用 REST 开源框架、ROA 面向资源架构。

面向云架构 COA:讲解云计算的几个发展阶段的概念,包括网格计算、集群计算、分布式计算和云计算,以及云服务的特点、实现形式和实际应用。随书附赠光盘内容为本书各种原型包、系统源程序。

本书适合作为 Java 相关培训机构的教材,也可作为 Java 自学人员的参考手册。

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 高手真经. 系统架构卷 : Java Web 系统设计与架构 : UML 建模+设计模式+面向服务架构 / 刘中兵编著. -北京: 电子工业出版社, 2009.12

(Java 开发专家)

ISBN 978-7-121-09728-7

I. J… II. 刘… III. Java 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2009)第 189124 号

责任编辑: 杨 鸽 赵树刚

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: 860×1092 1/16 印张: 36.25 字数: 1044 千字

印 次: 2009 年 12 月第 1 次印刷

印 数: 4 000 册 定价: 69.00 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。

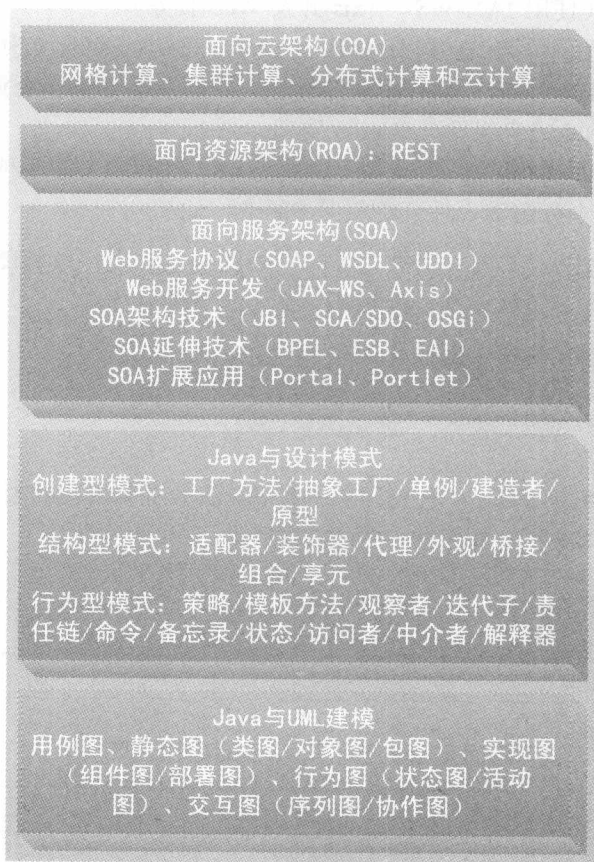
质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

一、本书培训目标

Java Web 即 Java EE，它是面向企业级的网络系统，其中的“企业级”和“网络”（分布式）代表了 Java Web 开发的核心目标。在 Java EE 最新的规范中提供了进行企业级开发的框架组件 EJB，并同时提供了多种分布式服务开发技术方案，以及多种企业应用的业务解决方案。

在进行分布式架构和设计中涉及到如下 5 个层次的技术：



本书将针对这 5 个层次的技术进行详细讲解：

如果说 Java Web 的各种技术与框架是构建一个大厦的砖与梁，那么进行 Java Web 系统的设计与架构，就是使用这些砖和梁来建造各种结构和风格的大厦，这就是软件架构设计的主旨。要进行 Java Web 系统的架构设计，首先需要从抽象的层次对 Java 语言进行各种基本功的训练。

(1) Java 与 UML 建模：建模是进行软件架构设计的表示工具，共 5 类 10 种图，包括用例图、静态图（类图、对象图、包图）、实现图（组件图、部署图）、行为图（状态图、活动图）、交互图（序列图、协作图）。

(2) Java 设计模式：设计模式是进行软件架构设计的代码模式，共包括 3 种类型 21 种模式。

- 创建型模式（5 种）：抽象工厂/建造者/工厂方法/原型/单例。
- 结构型模式（7 种）：适配器/桥接/组合/装饰/外观/享元/代理。

- 行为型模式（11种）：责任链/命令/解释器/迭代子/中介者/备忘录/观察者/状态/策略/模板方法/访问者。

然后训练各种风格的架构形式，除了 Java Web 高级技术中讲解的分布式架构外，还有 3 种架构形式。

(3) 面向服务架构 SOA：以 WebService 为基础，基于 SOAP 协议实现与各种语言系统的分布式调用，讲解的技术包括：

- Web 服务协议（SOAP、WSDL、UDDI）。
- Web 服务开发技术（使用 JAX-WS、Axis 开发 Web 服务）。
- SOA 架构技术（Sun 公司的 JBI 和 IBM/BEA 公司的 SCA/SDO、OSGi 架构）。
- SOA 延伸技术（BPEL 业务流程管理、ESB 企业服务总线与 EAI 企业应用集成）。
- SOA 扩展应用（Portal 门户组件与 Portlet 窗口组件）。

(4) 面向资源架构 ROA：以资源的形式规划网络结构，将网络上的各种增、删、改、查的操作都定义为统一的 URL 资源，保持永久的唯一性，这就是 Restful 风格。

(5) 面向云架构 COA：以云的形式提供广阔的网络服务，让用户能够随时随地利用互联网做各种事情，讲解的技术包括网格计算、集群计算、分布式计算、云计算。

以上内容实际上也是进行分布式架构设计的完美诠释。

二、本书培训课程安排

本书的内容总体上分为 4 个部分，从开发环境的准备，到企业级开发组件 EJB，再到各种分布式技术，最后讲解各种业务开源技术与框架。各部分内容安排如下：

分 类	培 训 技 术	课时（分钟）	上机（分钟）
第 1 部分 Java 与 UML 建模	01.UML 快速入门	60	30
	02.用例图	30	30
	03.静态图——类图、对象图、包图	50	40
	04.实现图——组件图与部署图	30	30
	05.行为图——状态图	30	90
	06.行为图——活动图	30	90
	07.交互图——序列图	30	90
	08.交互图——协作图	30	90
	09.使用 UML 进行系统设计——Bug 管理系统	60	90
第 2 部分 Java 设计模式	10.设计模式概述	60	
	11.创建型模式（5种）	120	
	12.结构型模式（7种）	160	
	13.行为型模式（11种）	220	
第 3 部分 面向服务架构 SOA	14.Web 服务协议——SOAP、WSDL、UDDI 和 WS 协议集	30	
	15.Web 服务开发技术——使用 JAX-WS 开发 Web 服务	30	

(续表)

分 类	培 训 技 术	课时 (分钟)	上机 (分钟)
	16.Web 服务开发框架——使用 Axis 框架开发 Web 服务	60	
	17.SOA 面向服务架构——Sun 的 JBI 和 IBM/BEA 的 SCA/SDO	30	
	18.JBI 业务集成架构——Java 实现 SOA 的标准途径	30	
	19.SCA 服务组件架构——实现 SOA 的最佳方式	180	
	20.SDO 服务数据对象——SCA 数据服务组件	40	
	21.OSGi 体系架构——SCA 的一种实现方式	60	
	22.SOA 延伸技术——BPEL 业务流程管理、ESB 企业服务总线与 EAI 企业应用集成	30	
	23.SOA 扩展应用——Portal 门户组件与 Portlet 窗口组件	30	
	24. SOA 服务架构开发实战——基于 Tuscany+Spring 的企业信息管理系统	120	
第 4 部分 面向资源架构 ROA	25.ROA 面向资源架构——基于 REST 风格的 Web 服务	30	
第 5 部分 面向云架构 COA	26.COA 面向云架构——云计算与云架构	30	
第 6 部分 架构师之路	27.软件架构师职业规划	30	
合计 (小时)		27 小时	10 小时

以上的章节覆盖了 Java Web 高级开发的几乎所有技术内容,并将讲解中的源代码都打包放在光盘中。每一章的正文内容和上机操作都根据内容多少和重要性规定了学习和上机的时间(分钟),总的正文学习时间为 1640 分钟(27.3 小时),上机时间为 580 分钟(9.7 小时)。这是对于讲课培训给出的参考时间,读者自学时需要至少 2 倍的集中时间学习。

三、本书范例程序

本书在讲解 UML、设计模式和 SOA 时,每一个样例包都可以作为你开发时的原型包进行扩展开发,也可以作为公司、学校教学的参考案例。

本书所有的测试代码和实例都打包放在光盘中 source 目录下,共包括 60 个样例文件和程序包:

(1) UML 样例源文件(8 个样例文件)

uml/uml.mdl	//01.UML 快速入门
uml/usecase.mdl	//02.用例图
uml/class.mdl	//03.静态图——类图、对象图、包图
uml/component.mdl	//04.实现图——组件图与部署图
uml/statechart.mdl	//05.行为图——状态图
uml/activity.mdl	//06.行为图——活动图
uml/sequence.mdl	//07.交互图——序列图

uml/collaboration.mdl //08.交互图——协作图

(2) UML 上机演练源文件 (16 个实例文件)

uml/usecase_product.mdl //02.用例图——企业产品生产销售管理系统
 uml/class_login.mdl //03.类图和包图——企业信息管理系统用户登录和注册模块
 uml/component_emis.mdl //04.组件图与部署图——企业信息管理系统
 uml/statechart_bug.mdl //05.状态图——Bug 管理系统
 uml/statechart_atm.mdl //05.状态图——ATM 机存取款
 uml/statechart_thread.mdl //05.状态图——Java 线程类 Thread
 uml/activity_bug.mdl //06.活动图——Bug 管理系统
 uml/activity_atm.mdl //06.活动图——ATM 机存取款
 uml/activity_thread.mdl //06.活动图——Java 线程类 Thread
 uml/sequence_bug.mdl //07.序列图——Bug 管理系统
 uml/sequence_atm.mdl //07.序列图——ATM 机存取款
 uml/sequence_tel.mdl //07.序列图——打电话
 uml/collaboration_bug.mdl //08.协作图——Bug 管理系统
 uml/collaboration_atm.mdl //08.协作图——ATM 机存取款
 uml/collaboration_tel.mdl //08.协作图——打电话
 uml/bug.mdl //09.综合实例——Bug 管理系统

(3) 设计模式样例 (24 个讲解样例程序)

pattern/src/principle/liskovsubstitution //10.3.2 里氏代换原则
 pattern/src/creation/factorymethod //11.1 工厂方法模式
 pattern/src/creation/abstractfactory //11.2 抽象工厂模式
 pattern/src/creation/singleton //11.3 单例模式
 pattern/src/creation/builder //11.4 建造者模式
 pattern/src/creation/prototype //11.5 原型模式
 pattern/src/structure/adaptor //12.1 适配器模式
 pattern/src/structure/decorator //12.2 装饰器模式
 pattern/src/structure/proxy //12.3 代理模式
 pattern/src/structure/facade //12.4 外观模式
 pattern/src/structure/bridge //12.5 桥接模式
 pattern/src/structure/composite //12.6 组合模式
 pattern/src/structure/flyweight //12.7 享元模式
 pattern/src/behavior/strategy //13.1 策略模式
 pattern/src/behavior/templatemethod //13.2 模板方法模式
 pattern/src/behavior/observer //13.3 观察者模式
 pattern/src/behavior/iterator //13.4 迭代子模式
 pattern/src/behavior/chainofresponsibility //13.5 责任链模式
 pattern/src/behavior/command //13.6 命令模式
 pattern/src/behavior/memento //13.7 备忘录模式
 pattern/src/behavior/state //13.8 状态模式
 pattern/src/behavior/visitor //13.9 访问者模式
 pattern/src/behavior/mediator //13.10 中介者模式

pattern/src/behavior/interpreter

//13.11 解释器模式

(4) SOA (12 个程序包)

soa/JAXWSHelloWorldServer.zip

//15.JAX-WS 服务端——HelloWorld 实例

soa/JAXWSHelloWorldClient.zip

//15.JAX-WS 客户端——HelloWorld 实例

soa/AxisHelloWorldServer.zip

//16.Axis 服务端——HelloWorld 实例

soa/AxisHelloWorldClient.zip

//16.Axis 客户端——HelloWorld 实例

soa/SCAJavazip

//19.SCA 实例——计算器实例

soa/SCAWSServer.zip

//19.SCA 服务端实例——HelloWorld 实例

soa/SCAWSClient.zip

//19.SCA 客户端实例——HelloWorld 实例

soa/OSGiHelloWorld.zip.zip

//21.OSGi 实例——HelloWorld 实例

soa/demo.sql

//24.SOA 服务架构实战——企业信息管理系统数据库脚本

soa/ssh2.zip

//24.SOA 服务架构实战——企业信息管理系统 SSH2 代码

soa/EmisWeb.zip

//24.SOA 服务架构实战——企业信息管理系统 SCA 服务端

soa/EmisService.zip

//24.SOA 服务架构实战——企业信息管理系统 Web 端

这些样例中的 Java 程序包都是在 Eclipse 3.4 中开发的，你可以使用 Eclipse 的导入功能直接导入这些程序包。导入的方法：首先将要运行的程序包解压缩到 Eclipse 的工作目录（如 D:\eclipse\workspace），单击 Eclipse 左侧的项目工作区空白处，单击鼠标右键，在弹出的右键菜单中选择“导入”命令，然后在弹出的“导入”窗口中选择“从现有项目到工作空间中”，并选择刚才解压缩后的程序目录，单击“确定”按钮即可导入到 Eclipse 中。

四、光盘使用说明

在光盘 software 目录下仅包含了本书所需要软件安装包的下载存放目录，并在 software\readme.txt 中详细记录了每一个软件的官方网站、所需版本和该版本的下载链接地址，请读者在学习本书前到指定的网址中去下载各个软件。

说明：考虑到版权问题，我们只提供了这些软件的下载链接地址。这些下载地址在本书写作时都已经经过了验证确认，有些链接也许需要您注册登录后才能够下载。如果下载的连接地址不存在，则表明该网站已经移除了该链接，请您在指定的官方网站上找到下载的页面来下载对应版本的文件。

本书所用软件、使用版本、下载地址列表：

(1) JDK

本书使用版本：1.6.0

官方网站：<http://java.sun.com/>

下载页面：<http://java.sun.com/javase/downloads/index.jsp>

下载文件：jdk-6u10-beta-windows-i586-p.exe

(2) Tomcat

本书使用版本：6.0.18

官方网站：<http://tomcat.apache.org/>

下载页面：<http://tomcat.apache.org/download-60.cgi>

下载地址：<http://apache.mirror.phpchina.com/tomcat/tomcat-6/v6.0.18/bin/apache-tomcat-6.0.18.zip>

下载文件: apache-tomcat-6.0.18.zip

(3) Eclipse

本书使用版本: 3.4

官方网站: <http://www.eclipse.org/>

下载页面: <http://www.eclipse.org/downloads/>

下载地址: <http://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/ganymede/SR1/eclipse-je-ganymede-SR1-win32.zip>

下载文件: eclipse-je-ganymede-SR1-win32.zip

(4) Axis2

本书使用版本: 1.4.1

官方网站: <http://ws.apache.org/axis2/index.html>

下载页面: http://ws.apache.org/axis2/download/1_4_1/download.cgi

下载地址: http://labs.xiaonei.com/apache-mirror/ws/axis2/1_4_1/axis2-1.4.1-bin.zip

下载文件: axis2-1.4.1-bin.zip

(5) Tuscany

本书使用版本: 2.0

官方网站: <http://tuscany.apache.org/>

下载页面: <http://tuscany.apache.org/tuscany-downloads-documentations.html>

下载地址: <http://www.apache.org/dyn/closer.cgi/tuscany/java/sca/2.0-M3/apache-tuscany-sca-all-2.0-M3.zip>

下载文件: apache-tuscany-sca-all-2.0-M3.zip

五、致谢

参与本书技术调研和程序调试的有: 李金刚、刘剑、陈景春、周同、刘中敏、陈万珍、李瑞霞、李建海。本书在编写过程中得到了 Apache 软件组织和 Sun 公司在技术上的大力支持, 飞思科技产品研发中心负责监制工作, 在此表示衷心的感谢! 尽管我们尽了最大努力, 但本书难免会有不妥之处, 欢迎各界专家和读者朋友批评指正, 我们的联系方式是:

本书专业论坛交流答疑: www.100tt.net

编 著 者

联系方式

咨询电话: (010) 68134545 88254160

服务网址: <http://www.fecit.com.cn> <http://www.fecit.net>

通用网址: 计算机图书、飞思、飞思教育、飞思科技、FECIT

技术答疑邮箱: support@fecit.com.cn

售后服务 QQ 号: support@fecit.com.cn

绪论 软件设计与架构概述.....	1
一、Java 与 UML 建模.....	1
二、Java 设计模式.....	2
1. 创建型模式 2. 结构型模式 3. 行为型模式	
三、系统架构.....	4
1. 面向服务架构 SOA 2. 面向资源架构 ROA 3. 面向云架构 COA	

第 1 部分 Java 与 UML 建模

第 1 课 UML 快速入门.....	9
1.1 UML 建模概述.....	9
1.1.1 面向对象方法.....	10
1.1.2 面向对象建模 OMT.....	10
1.1.3 统一建模语言 UML.....	11
1.1.4 UML 的起源和发展.....	11
1.1.5 为什么要使用 UML.....	12
1.2 UML 基本构成元素.....	13
1.2.1 4 种事物.....	13
1. 结构事物 (Structural Things) 2. 行为事物 (Behavioral Things)	
3. 分组事物 (Grouping Things) 4. 注释事物 (Annotational Things)	
1.2.2 4 种关系.....	16
1. 依赖 (Dependencies) 2. 关联 (Association)	
3. 泛化 (Generalization) 4. 实现 (Realization)	
1.2.3 10 种图.....	17
1. 用例图 (Usecase Diagram) 2. 类图 (Class Diagram)	
3. 对象图 (Object Diagram) 4. 包图 (Package Diagram)	
5. 组件图 (Component Diagram) 6. 部署图 (Deployment Diagram)	
7. 状态图 (Statechart Diagram) 8. 活动图 (Activity Diagram)	
9. 序列图 (Sequence Diagram) 10. 协作图 (Collaboration Diagram)	
1.2.4 UML 图的分类.....	22
1.3 UML 设计工具.....	24
1.3.1 使用 Rose 进行 UML 建模.....	24
1.3.2 使用 Visio 进行 UML 建模.....	25
1.4 本课小结.....	27
1.4.1 本课图例打包 uml.mdl.....	27
1.4.2 总结本课的知识点.....	27
1.4.3 要掌握的关键点.....	28
1.4.4 课后上机作业.....	29
1.4.5 继续下一课: 用例图.....	29

第 2 课 用例图.....	31	
2.1 用例概述.....	31	
2.1.1 什么是用例.....	31	
2.1.2 为什么要使用用例.....	32	
2.1.3 用例图的基本模型元素.....	32	
2.2 用例图的绘制.....	33	
2.2.1 参与者的表示法.....	33	
2.2.2 用例的表示法.....	34	
2.2.3 系统边界的表示法.....	35	
2.2.4 通信关联的表示法.....	35	
2.3 用例图的建立过程.....	36	
2.3.1 发现参与者及检查标准.....	36	
2.3.2 发现用例及检查标准.....	37	
2.3.3 编写用例规约.....	37	
2.3.4 用例图创建技巧.....	38	
2.4 本课小结.....	39	
2.4.1 本课图例打包 usecase.mdl.....	39	
2.4.2 总结本课的知识点.....	40	
2.4.3 要掌握的关键点.....	40	
2.4.4 课后上机演练——企业产品生产销售管理系统.....	41	
2.4.5 继续下一课：静态图——类图、对象图、包图.....	42	
第 3 课 静态图——类图、对象图、包图.....	43	
3.1 类图的绘制.....	43	
3.1.1 类图的表示法.....	43	
1. 类图的表示法	2. 类图属性的表示法	3. 类图操作的表示法
3.1.2 接口的表示法.....	46	
3.1.3 类图之间的 4 种关系.....	46	
1. 泛化关系 (Generalization) ——表示类的继承	2. 关联关系 (Association) ——表示类的变量	
3. 依赖关系 (Dependency) ——表示局部变量	4. 实现关系 (Realization) ——表示接口的实现	
3.2 对象图的绘制.....	50	
3.2.1 对象图的表示法.....	50	
3.2.2 对象图的使用场景——序列图和协作图.....	50	
3.3 包图的绘制.....	52	
3.3.1 包图的表示法.....	52	
3.3.2 使用包包含类.....	52	
3.3.3 使用包包含子包.....	53	
3.3.4 包的依赖关系.....	53	
3.4 本课小结.....	54	

3.4.1	本课图例打包 class.mdl	54
3.4.2	总结本课的知识点	57
3.4.3	课后上机演练——企业信息管理系统用户登录和注册模块	58
3.4.4	继续下一课：实现图——组件图与部署图	59
第4课	实现图——组件图与部署图	61
4.1	组件图的绘制	61
4.1.1	组件的表示法	61
4.1.2	程序的表示法	62
	1. 子程序规范 2. 子程序体 3. 主程序	
4.1.3	包的表示法	62
	1. 包规范 2. 包体	
4.1.4	任务的表示法	63
	1. 任务规范 2. 任务体	
4.1.5	组件之间的依赖	63
4.2	部署图的绘制	64
4.2.1	处理器节点的表示法	64
4.2.2	设备节点的表示法	64
4.2.3	节点之间的关联	64
4.3	本课小结	65
4.3.1	本课图例打包 component.mdl	65
4.3.2	总结本课的知识点	66
4.3.3	要掌握的关键点	67
4.3.4	课后上机演练——企业信息管理系统部署图	67
4.3.5	继续下一课：行为图——状态图	68
第5课	行为图——状态图	69
5.1	状态图的绘制	69
5.1.1	状态图的基本模型	69
5.1.2	状态的表示法	70
5.1.3	转换的表示法	71
5.2	状态的绘制	72
5.2.1	初始状态的表示法	72
5.2.2	终止状态的表示法	73
5.2.3	子状态的表示法	73
5.2.4	历史状态的表示法	73
5.3	本课小结	74
5.3.1	本课图例打包 statechart.mdl	74
5.3.2	总结本课的知识点	75
5.3.3	要掌握的关键点	75

5.3.4	课后上机演练	75
	1. Bug 管理系统状态图 2. ATM 机存取款状态图 3. Java 线程类 Thread 的状态图	
5.3.5	继续下一课: 行为图——活动图	78
第 6 课	行为图——活动图	81
6.1	活动图的绘制	81
6.1.1	活动图的基本模型	81
6.1.2	活动的表示法	82
6.1.3	转换的表示法	83
6.1.4	分支与合并的表示法	84
6.1.5	分叉与汇合的表示法	84
6.1.6	泳道的表示法	85
6.2	本课小结	87
6.2.1	本课图例打包 activity.mdl	87
6.2.2	总结本课的知识点	88
6.2.3	要掌握的关键点	88
6.2.4	课后上机演练	89
	1. Bug 管理系统活动图 2. ATM 机存取款活动图 3. Java 线程类 Thread 的活动图	
6.2.5	继续下一课: 交互图——序列图	91
第 7 课	交互图——序列图	93
7.1	序列图的绘制	93
7.1.1	序列图的基本模型	93
7.1.2	对象生命线的表示法	94
7.1.3	激活的表示法	95
7.1.4	消息的表示法	95
7.1.5	自调用消息的表示法	97
7.2	本课小结	97
7.2.1	本课图例打包 sequence.mdl	97
7.2.2	总结本课的知识点	98
7.2.3	要掌握的关键点	98
7.2.4	课后上机演练	98
	1. Bug 管理系统序列图 2. ATM 机存取款序列图 3. 打电话序列图	
7.2.5	继续下一课: 交互图——协作图	102
第 8 课	交互图——协作图	103
8.1	协作图的绘制	103
8.1.1	协作图的基本模型	103
8.1.2	对象的表示法	104
8.1.3	连接的表示法	104
8.1.4	消息的表示法	104

8.1.5	自调用消息的表示法	105
8.2	本课小结	105
8.2.1	本课图例打包 collaboration.mdl	105
8.2.2	总结本课的知识点	106
8.2.3	要掌握的关键点	106
8.2.4	课后上机演练	106
	1. Bug 管理系统协作图 2. ATM 机存取款协作图 3. 打电话协作图	
8.2.5	继续下一课: 使用 UML 进行系统设计——Bug 管理系统	109
第 9 课	使用 UML 进行系统设计——Bug 管理系统	111
9.1	UML 系统设计概述	111
9.1.1	软件开发的驱动方法	112
	1. 数据库驱动开发 DDD (Database-Driven Development) 2. 测试驱动开发 TDD (Test-Driven Development)	
	3. 模型驱动开发 MDD (Model-Driven Development)	
9.1.2	正向工程与逆向工程	113
	1. 正向工程 (Forward Engineering) 2. 逆向工程 (Reverse Engineering)	
9.1.3	UML 建模的受众分析	114
9.1.4	UML 建模的过程	114
9.2	Bug 管理系统的模型驱动开发过程	115
9.2.1	需求分析——用例图	116
9.2.2	系统分析: 分析业务规则——状态图	117
9.2.3	系统分析: 分析业务流程——活动图	117
9.2.4	系统设计: 设计静态结构——类图和包图	118
9.2.5	系统设计: Action 类被调用关系——序列图	120
9.2.6	系统设计: 用户调用 Action 类的过程——协作图	121
9.2.7	系统架构——组件图和部署图	121
9.2.8	编码实现	123
9.2.9	UML 图例打包 bug.mdl	123
9.3	本课小结	123
9.3.1	总结本课的知识点	123
9.3.2	要掌握的关键点	124
9.3.3	课后上机演练	125
9.3.4	继续下一课: Java 设计模式	125

第 2 部分 Java 设计模式

第 10 课	设计模式概述	131
10.1	了解设计模式	131
10.1.1	设计模式的概念	131
10.1.2	设计模式的产生——23 种设计模式与 GoF “四人帮”	132
10.1.3	设计模式不是技术而是思想——构建可重用的程序	132

10.1.4	为什么要使用设计模式	132
10.2	设计模式的分类	133
10.2.1	创建型模式	133
1.	工厂方法模式 (Factory Method)	
2.	抽象工厂模式 (Abstract Factory)	
3.	单例模式 (Singleton)	
4.	建造者模式 (Builder)	
5.	原型模式 (Prototype)	
10.2.2	结构型模式	133
1.	适配器模式 (Adapter)	
2.	装饰器模式 (Decorator)	
3.	代理模式 (Proxy)	
4.	外观模式 (Facade)	
5.	桥接模式 (Bridge)	
6.	组合模式 (Composite)	
7.	享元模式 (Flyweight)	
10.2.3	行为型模式	134
1.	策略模式 (Strategy)	
2.	模板方法模式 (Template Method)	
3.	观察者模式 (Observer)	
4.	迭代子模式 (Iterator)	
5.	责任链模式 (Chain of Responsibility)	
6.	命令模式 (Command)	
7.	备忘录模式 (Memento)	
8.	状态模式 (State)	
9.	访问者模式 (Visitor)	
10.	中介者模式 (Mediator)	
11.	解释器模式 (Interpreter)	
10.2.4	并发型模式	135
10.2.5	线程池模式	136
10.3	设计模式六大原则	136
10.3.1	开闭原则 (Open Closed Principle)	136
10.3.2	里氏代换原则 (Liskov Substitution Principle)	138
10.3.3	依赖倒转原则 (Dependence Inversion Principle)	141
10.3.4	接口隔离原则 (Interface Segregation Principle)	141
1.	接口对应的角色	
2.	角色对应的接口	
10.3.5	迪米特法则 (最少知道原则) (Demeter Principle)	143
10.3.6	合成复用原则 (Composite Reuse Principle)	144
10.4	模式的讲解思路	145
1.	模式的核心思想	
2.	何时使用	
3.	Java 中的应用	
10.5	本课小结	146
10.5.1	总结本课的知识点	146
10.5.2	要掌握的关键点	147
10.5.3	继续下一课: 5 种创建型模式	148
第 11 课	5 种创建型模式	149
11.1	工厂方法模式 (Factory Method)	150
11.1.1	工厂方法模式的核心思想及分类	150
11.1.2	第一种——工厂方法模式	152
11.1.3	第二种——多个工厂方法模式	154
11.1.4	第三种——静态工厂方法模式 (简单工厂模式)	156
11.1.5	何时使用工厂方法模式	157

11.1.6	Java 中的应用——Swing 中的静态工厂类 BorderFactory.....	158
11.2	抽象工厂模式 (Abstract Factory)	161
11.2.1	抽象工厂模式的核心思想	161
11.2.2	何时使用抽象工厂模式	164
11.3	单例模式 (Singleton)	164
11.3.1	单例模式的核心思想	164
11.3.2	何时使用单例模式	165
11.3.3	属性文件加载工厂实例	165
11.3.4	Java 中的应用——日历单例类 Calendar.....	167
11.4	建造者模式 (Builder)	168
11.4.1	建造者模式的核心思想	168
11.4.2	何时使用建造者模式	170
11.4.3	Java 中的应用——字符串建造者类 StringBuilder	171
11.4.4	Java 中的应用——进程建造者类 ProcessBuilder.....	173
11.5	原型模式 (Prototype)	174
11.5.1	原型模式的核心思想	174
11.5.2	浅复制与深复制	175
11.5.3	何时使用原型模式	178
11.6	本课小结	179
11.6.1	总结本课的知识点	179
11.6.2	要掌握的关键点	179
11.6.3	继续下一课: 7 种结构型模式	180
第 12 课	7 种结构型模式	181
12.1	适配器模式 (Adapter)	182
12.1.1	适配器模式的核心思想	182
12.1.2	第一种: 类的适配器模式 (对类进行适配)	183
12.1.3	第二种: 对象的适配器模式 (对对象进行包装)	185
12.1.4	第三种: 接口的适配器模式 (对接口抽象化)	187
12.1.5	何时使用适配器模式	190
12.1.6	Java 中的应用——Iterator 适配器 (对象的适配器模式)	190
12.1.7	Java 中的应用——Enumeration 适配器 (对象的适配器模式)	192
12.1.8	Java 中的应用——AWT 事件适配器 (接口的适配器模式)	194
12.1.9	Java 中的应用——I/O 字节流到字符流的适配器 (对象的适配器模式)	197
1.	字节输入流转换为字符输入流	
2.	字节输出流转换为字符输出流	
12.2	装饰器模式 (Decorator)	200
12.2.1	装饰器模式的核心思想	201
12.2.2	何时使用装饰器模式	205
12.2.3	Java 中的应用——I/O 输入/输出流管道的装饰器模式	205

Contents

12.2.4	Java 中的应用——Sitemesh 装饰器	207	
12.3	代理模式 (Proxy)	208	
12.3.1	代理模式的核心思想	208	
12.3.2	何时使用代理模式	211	
12.3.3	Java 中的应用——Java 动态代理机制	212	
12.4	外观模式 (Facade)	214	
12.4.1	外观模式的核心思想	214	
12.4.2	何时使用外观模式	218	
12.4.3	Java 中的应用	218	
12.5	桥接模式 (Bridge)	218	
12.5.1	桥接模式的核心思想	218	
12.5.2	何时使用桥接模式	222	
12.5.3	在 Java 中的应用——JDBC 桥 DriverManager	222	
12.6	组合模式 (Composite)	223	
12.6.1	组合模式的核心思想——树形结构	223	
12.6.2	何时使用组合模式	225	
12.6.3	Java 中的应用——AWT 容器 Container	226	
12.7	享元模式 (Flyweight)	226	
12.7.1	享元模式的核心思想——共享	227	
12.7.2	何时使用享元模式	227	
12.7.3	Java 中的应用——数据库连接池	228	
12.8	本课小结	230	
12.8.1	总结本课的知识点	230	
12.8.2	要掌握的关键点	232	
12.8.3	继续下一课: 11 种行为型模式	232	
第 13 课	11 种行为型模式	233	
13.1	策略模式 (Strategy)	234	
13.1.1	策略模式的核心思想	235	
13.1.2	何时使用策略模式	239	
13.1.3	Java 中的应用——AWT 布局管理器	239	
13.1.4	Java 中的应用——Swing 边界类 Border	240	
13.2	模板方法模式 (Template Method)	241	
13.2.1	模板方法模式的核心思想	241	
13.2.2	何时使用模板方法模式	246	
13.2.3	Java 中的应用——HTTP 请求处理类 HttpServlet	246	
	1. init()方法	2. service()方法	3. destroy()方法
	4. GetServletConfig()方法	5. GetServletInfo()方法	
13.3	观察者模式 (Observer)	249	