

<<Java高手真经（系统架构卷）>>

图书基本信息

书名：<<Java高手真经（系统架构卷）>>

13位ISBN编号：9787121097287

10位ISBN编号：7121097281

出版时间：2009-12

出版时间：电子工业

作者：刘中兵

页数：552

字数：1044000

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

前言

一、本书培训旧标 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JavaWeb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Pottlet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（20A）：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。

<<Java高手真经（系统架构卷）>>

内容概要

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JavaEE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。

<<Java高手真经（系统架构卷）>>

作者简介

刘中兵，Java研究室首席技术专家，应用数学专业，曾任职于清华同方、NEC等企业，长期深入Java/Java EE大型企业应用的架构、设计与开发工作，目前专注于手机无线互联网与网络通信领域的架构设计与研究工作。

<<Java高手真经 (系统架构卷)>>

书籍目录

绪论	软件设计与架构概述	一、Java与UML建模	二、Java设计模式	1.创建型模式	2.结构型模式
3.行为型模式	三、系统架构	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	1.1 UML建模概述	1.1.1 面向对象方法	1.1.2 面向对象建模OMT	1.1.3 统一建模语言UML	1.1.4 UML的起源和发展
1.1.5 为什么要使用UML	1.2 UML基本构成元素	1.2.1 4种事物	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	1.依赖 (Dependencies)	2.关联 (Association)	3.泛化 (Generalization)	4.实现 (Realization)
1.2.3 10种图	1.用例图 (UsecaseDiagram)	2.类图 (ClassDiagram)	3.对象图 (Object Diagram)	4.包图 (Package Diagram)	5.组件图 (Compoment 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
1.3 UML设计工具	1.3.1 使用Rose进行UML建模	1.3.2 使用Visio进行UML建模	1.4 本课小结	1.4.1 本课图例打包UML.mdl	1.4.2 总结本课的知识点
1.4.3 要掌握的关键点	1.4.4 课后上机作业	1.4.5 继续下一课：用例图	第2课 用例图	2.1 用例概述	2.1.1 什么是用例
2.1.2 为什么要使用用例	2.1.3 用例图的基本模型元素	2.2 用例图的绘制	2.2.1 参与者的表示法	2.2.2 用例的表示法	2.2.3 系统边界的表示法
2.2.4 通信关联的表示法	2.3 用例图的建立过程	2.3.1 发现参与者及检查标准	2.3.2 发现用例及检查标准	2.3.3 编写用例规约	2.3.4 用例图创建技巧
2.4 本课小结	2.4.1 本课图例打包usecase.mdl	2.4.2 总结本课的知识点	2.4.3 要掌握的关键点	2.4.4 课后上机演练——企业产品生产销售管理系统	2.4.5 继续下一课：静态图—类图、对象图、包图
第3课 静态图——类图、对象图、包图	3.1 类图的绘制	3.1.1 类图的表示法	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	3.1.3 类图之间的4种关系	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	3.2.1 对象图的表示法	3.2.2 对象图的使用场景—序列图和协作图	3.3 包图的绘制	3.3.1 包图的表示法	3.3.2 使用包包含类
3.3.3 使用包包含子包	3.3.4 包的依赖关系	3.4 本课小结	3.4.1 本课图例打包class.mdl	3.4.2 总结本课的知识点	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
3.4.4 继续下一课：实现图——组件图与部署图	第4课 实现图——组件图与部署图	4.1 组件图的绘制	4.1.1 组件的表示法	4.1.2 程序的表示法	1.子程序规范
2.子程序体	3.主程序	4.1.3 包的表示法	1.包规范	2.包体	4.1.4 任务的表示法
1.任务规范	2.任务体	4.1.5 组件之间的依赖	4.2 部署图的绘制	4.2.1 处理器节点的表示法	4.2.2 设备节点的表示法
4.2.3 节点之间的关联	4.3 本课小结	4.3.1 本课图例打包component.mdl	4.3.2 总结本课的知识点	4.3.3 要掌握的关键点	4.3.4 课后上机演练—企业信息管理系统部署图
4.3.5 继续下一课：行为图—状态图	第5课 行为图——状态图	5.1 状态图的绘制	5.1.1 状态图的基本模型	5.1.2 状态的表示法	5.1.3 转换的表示法
5.2 状态的绘制	5.2.1 初始状态的表示法	5.2.2 终止状态的表示法	5.2.3 子状态的表示法	5.2.4 历史状态的表示法	5.3 本课小结
5.3.1 本课图例打包statechart.mdl	5.3.2 总结本课的知识点	5.3.3 要掌握的关键点	5.3.4 课后上机演练	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	第6课 行为图——活动图	6.1 活动图的绘制	6.1.1 活动图的基本模型	6.1.2 活动的表示法
6.1.3 转换的表示法	6.1.4 分支与合并的表示法	6.1.5 分叉与汇合的表示法	6.1.6 泳道的表示法	6.2 本课小结	6.2.1 本课图例打包activity.mdl
6.2.2 总结本课的知识点	6.2.3 要掌握的关键点	6.2.4 课后上机演练	1.Bug管理系统活动图	2.ATM机	

<<Java高手真经 (系统架构卷)>>

存取款活动图 3.Java线程类Thread的活动图 第7课 交互图——序列图 7.1 序列图的绘制
 7.1.1 序列图的基本模型 7.1.2 对象生命线的表示法 7.1.3 激活的表示法 7.1.4 消息的表示法
 7.1.5 自调用消息的表示法 7.2 本课小结 7.2.1 本课图例打包sequence.mdl 7.2.2 总结
 本课的知识点 7.2.3 要掌握的关键点 7.2.4 课后上机演练 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
 第8课 交互图——协作图 8.1 协作图的绘制 8.1.1 协作图的基本模型 8.1.2 对象的表示
 法 8.1.3 连接的表示法 8.1.4 消息的表示法 8.1.5 自调用消息的表示法 8.2 本课小结
 8.2.1 本课图例打包collaboration.mdl 8.2.2 总结本课的知识点 8.2.3 要掌握的关键点
 8.2.4 课后上机演练 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 第9课 使用UML进行系
 统设计——Bug管理系统 9.1 UML系统设计概述 9.1.1 软件开发的驱动方法 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
 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 9.1.4 UML建模的过程 9.2 Bug管理系统的模型驱动开发过程 9.2.1 需求分析一用
 例图 9.2.2 系统分析：分析业务规则—状态图 9.2.3 系统分析：分析业务流程—活动图 9.2.4
 系统设计：设计静态结构——类图和包图 9.2.5 系统设计：Action类被调用关系——序列图
 9.2.6 系统设计：用户调用Action类的过程—协作图 9.2.7 系统架构—组件图和部署图 9.2.8 编
 码实现 9.2.9 UML图例打包bug.mdl 9.3 本课小结 9.3.1 总结本课的知识点 9.3.2 要掌
 握的关键点 9.3.3 课后上机演练 9.3.4 继续下一课：Java设计模式第2部分 Java设计模式
 第10课 设计模式概述 10.1 了解设计模式 10.1.1 设计模式的概念 10.1.2 设计模式的产生
 ——23种设计模式与GoF“四人帮” 10.1.3 设计模式不是技术而是思想——构建可重用的程序
 第11课 5种创建型模式 第12课 7种结构型模式 第13课 11种行为型模式第3部分 面各服
 务架构SOA 第14课 Web服务协议——SOAP、WSDL、UDDI和WS协议集 第15课 Web服务开发
 技术——使用JAX-WS开发Web服务 第16课 Web服务开发框架——使用Axis框架开发Web服务
 第17课 SOA面向服务架构——Sun的JBI和IBM/BEA的SCA/SDO 第18课 JBI业务集成架构——Java实
 现SOA的标准途径 第19课 SCA服务组件架构——实现SOA的最佳方式 第20课 SDO服务数据对
 象——SCA数据服务组件 第21课 OSGi体系架构——SCA的一种实现方式 第22课 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 第23课 SOA扩展应用——Portal门
 户组件与Portlet窗口组件 第24课 SOA服务架构开发实战——基于Tuscany+Spring企业信息管理系统
 第25课 ROA面向资源架构——基于REST风格的Web服务第5部分 面向云架构COA 第26课
 COA面向架构——云计算与云架构第6部分 架构师之路 第27课 软件架构师职业规划《高手真
 经》系列后记

章节摘录

以前，我们在用FORTRAN、C语言进行编程时，使用了面向过程的程序设计方法，也就是所谓的结构化方法。

这种方法强调对系统功能进行抽象，系统功能的实现是通过对若干个模块的调用来完成的。

这种方法并不能够降低软件开发成本，提高软件生产率。

为了提高生产效率，便产生了面向对象的方法。

在面向对象方法中，把客观世界的实体称为问题对象，对象都有自己的运动状态及运动规律，不同对象之间的相互作用和相互通信就构成了完整的客观世界。

由于采用了数据抽象和封装技术，面向对象的程序设计降低了各模块间的关联程度，这就相对减少了程序员之间的相互影响。

这项技术是在设计初期只有很少的程序员介入的情况下，通过在对象系统中建立一个高层次的通信环境来实现的，它使得今后更改引起的成本大大降低。

使用面向对象的方法详细定义用户的数据类型，将它们封装在一起又能够提高代码利用率。

面向对象的方法中对象是核心，封装、继承和多态是面向对象的三大特性。

因此，面向对象的方法的核心思想就是对象、封装、继承与多态。

对象就是人们所感兴趣的任何事物，它可以是有形的实体、某种作用及性能等。

对象都有其运动状态和运动规律，因此对象具有很强的表达能力和描述功能。

对象中包含数据和对数据操作的代码，它实现了数据和操作的结合。

具有相同结构、操作并遵守相同约束规则的对象集合就是类。

封装是一种组织软件的方法，它的基本思想是把客观世界中联系紧密的元素及相关操作组织在一起，构造具有独立含义的软件实现，使其相互关系隐藏在内部，而对外仅仅表现为与其他封装体间的接口关系。

封装的目的就是信息隐匿。

在面向对象的方法中，是通过对象和类来实现封装的。

继承用于描述类之间的共同性质，它减少了相似类的重复说明。

继承使得程序员对共同的操作及属性只说明一次，并且在具体的情况下可以扩展、细化这些属性及操作。

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>