

<<软件设计与体系结构>>

图书基本信息

书名：<<软件设计与体系结构>>

13位ISBN编号：9787030344298

10位ISBN编号：7030344294

出版时间：2012-5

出版时间：科学出版社

作者：周华 编

页数：424

字数：723250

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>

<<软件设计与体系结构>>

内容概要

软件设计与体系结构从CDIO工程理念出发，围绕构思（Conceive）、设计（Design）、实现（Implement）、运作（Operate）四个方面展开阐述，以一种崭新的视角展现软件设计和体系结构的内容，尽可能做到覆盖面广和内容新颖，在保持经典内容的基础上，力求提供来自业界最新的内容和进展。

构思（C）给出一个软件要做什么的总体思路；设计（D）给出一个软件怎么做的方法和手段；实现（I）给出一个软件实际做出来的实现技术和路线；运作（O）给出一个软件如何成功运作的模式和方法。

CDIO各个部分相对独立但又互有联系，能够让读者以全面的、主动的和实用的方式学习和掌握相关内容，并着重强调工程实践训练和综合能力培养。

软件设计与体系结构可作为计算机相关专业本科生和研究生的教材，同时也是软件工程领域专业人员的优秀参考读物。

<<软件设计与体系结构>>

作者简介

周华

<<软件设计与体系结构>>

书籍目录

第一篇 构思篇(Conceive)第1章 软件环境1.1 软件与软件分类1.1.1 软件的特点1.1.2 软件的分类1.2 程序设计语言与开发环境1.2.1 程序设计语言1.2.2 开发环境1.3 数据库环境1.3.1 关系数据库1.3.2 面向对象数据库1.3.3 非结构化数据库1.4 平台环境1.4.1 集成平台1.4.2 虚拟化平台1.4.3 云平台1.5 软件工程环境1.5.1 软件过程模型1.5.2 软件工程经济学1.5.3 软件过程管理1.5.4 软件维护及演化1.6 习题与思考题第2章 软件需求2.1 软件需求与需求工程概述2.1.1 需求的定义2.1.2 需求工程概述2.2 需求获取2.2.1 引言2.2.2 需求获取的实质2.2.3 常用方法2.3 需求建模与分析2.3.1 面向对象建模2.3.2 过程建模2.3.3 数据建模2.4 需求规约与验证2.4.1 需求规约2.4.2 需求验证2.5 需求管理2.6 习题与思考题第3章 软件体系结构3.1 “4+1”视图模型3.1.1 逻辑视图3.1.2 开发视图3.1.3 进程视图3.1.4 物理视图3.1.5 场景3.2 软件体系结构的定义3.3 软件体系结构的核心模型3.4 软件体系结构描述方法3.5 软件体系结构描述语言(ADL)3.6 软件体系结构设计原则3.7 基于体系结构的软件开发过程3.8 软件体系结构的风格3.8.1 管道-过滤器风格3.8.2 分层风格3.8.3 客户/服务器风格3.8.4 浏览器/服务器风格3.8.5 事件驱动风格3.9 体系结构模式3.9.1 从混沌到结构3.9.2 分布式系统3.9.3 交互系统3.9.4 适应性系统3.10 习题与思考题第二篇 设计篇(Design)第4章 软件设计4.1 设计目标及要素4.2 设计原则4.2.1 抽象化4.2.2 模块化4.2.3 信息隐藏4.2.4 模块的功能独立性4.2.5 降低模块间耦合度的方法4.3 设计规约4.4 设计方法4.4.1 结构化设计方法4.4.2 面向对象的设计4.5 习题与思考题第5章 统一建模语言UML5.1 UML核心概念5.1.1 元素5.1.2 具名元素5.1.3 命名空间5.1.4 可打包元素5.1.5 可重定义元素5.1.6 分类器5.1.7 特性5.1.8 注解5.1.9 关系5.2 UML关键字与版型5.2.1 关键字5.2.2 版型5.3 UML图示5.3.1 类图5.3.2 对象图5.3.3 包图5.3.4 组件图5.3.5 合成结构图5.3.6 部署图5.3.7 用例图5.3.8 活动图5.3.9 状态机图5.3.10 交互图5.3.11 顺序图5.3.12 通信图5.3.13 计时图5.4 习题与思考题第6章 设计模式6.1 模式思维方法6.1.1 模式的概念6.1.2 模式的重要性6.1.3 软件设计模式的分类6.1.4 模式描述模板6.1.5 模式思维的步骤6.2 设计模式概述6.2.1 创建型模式6.2.2 结构型模式6.2.3 行为型模式6.3 习题与思考题第7章 软件复用7.1 概述7.1.1 概念7.1.2 软件复用的发展7.1.3 可复用的软件制品7.1.4 软件复用的分类7.1.5 软件复用的困难与建议7.1.6 软件复用的宗旨7.2 分析复用7.2.1 分析过程复用7.2.2 分析制品复用7.3 设计复用7.3.1 设计过程复用7.3.2 基于构件的设计复用7.4 代码复用7.5 测试复用7.5.1 面向复用的测试用例设计过程7.5.2 复用测试用例描述要素7.6 习题与思考题第8章 软件界面设计8.1 软件界面设计概述8.1.1 软件界面分析8.1.2 软件界面开发过程8.1.3 软件界面设计基本原则8.2 人机界面基础知识8.2.1 认知心理学8.2.2 软件人机工程学8.2.3 艺术设计8.3 界面的定量分析8.3.1 GOMS击键层模型8.3.2 界面效率的测量8.3.3 Fitts律和Hick律8.4 人本界面8.4.1 认知和关注点8.4.2 界面模式与单调性8.4.3 统一性和元动作8.4.4 易用性和帮助机制8.5 移动设备界面设计8.6 习题与思考题第三篇 实现篇(Implement)第9章 Web开发技术9.1 Web开发概述9.1.1 运作原理及概念9.1.2 开发语言及技术概述9.1.3 Web应用的特点9.2 Web服务器9.2.1 Microsoft IIS9.2.2 Apache9.2.3 Tomcat9.2.4 J2EE服务器9.2.5 Nginx9.3 服务器端开发技术及框架9.3.1 Java技术9.3.2 .Net技术9.3.3 PHP技术9.3.4 Ruby技术9.3.5 MVC框架9.3.6 Spring框架9.4 Web前端开发技术9.4.1 HTML9.4.2 CSS层叠样式表9.4.3 JavaScript技术9.4.4 AJAX技术9.4.5 Silverlight技术9.5 习题与思考题第10章 数据库开发技术10.1 SQL语言10.2 数据库和表10.2.1 创建及删除数据库10.2.2 创建表10.2.3 修改表10.2.4 删除表10.3 索引10.3.1 创建索引10.3.2 删除索引10.4 查询10.4.1 SELECT语句10.4.2 多表查询10.4.3 子查询10.4.4 集合运算10.4.5 聚集函数10.5 数据维护10.5.1 插入10.5.2 更新10.5.3 删除10.6 视图10.6.1 创建视图10.6.2 删除视图10.7 游标10.7.1 声明游标10.7.2 打开和关闭游标10.7.3 删除游标10.7.4 应用游标10.8 存储过程10.8.1 存储过程创建与修改10.8.2 执行存储过程10.8.3 删除存储过程10.9 触发器10.9.1 创建触发器10.9.2 删除触发器10.10 习题与思考题第11章 软件成本估算技术11.1 软件成本估算的步骤11.1.1 建立目标11.1.2 计划所需的数据与资源11.1.3 准确说明软件需求11.1.4 尽可能详细地做出估算11.1.5 采用多个独立的方法与资源11.1.6 比较与迭代估算11.1.7 跟踪与变更11.2 软件成本估算的方法11.2.1 算法模型11.2.2 专家判断11.2.3 通过推理来进行估算11.2.4 帕金森估算11.2.5 价格策略估算11.2.6 自顶向下估算11.2.7 自底向上估算11.2.8 各方法的总结比较11.3 基本COCOMO模型软件成本估算11.3.1 模型定义与假设11.3.2 软件开发模式分类11.3.3 组织型基本COCOMO模型11.3.4 其他模式COCOMO模型11.4 习题与思考题第12章 软件架构技术12.1 构件技术12.1.1 软件构件技术基础12.1.2 基于构件的软件工程12.2 软件架构综述12.2.1 企业架构12.2.2 业务架构12.2.3 应用架构12.2.4 信息架构12.3 中间件12.3.1 中间件的目标及地

<<软件设计与体系结构>>

位12.3.2 中间件的基本类型12.3.3 常见中间件技术介绍12.4 习题与思考题第13章 软件集成技术13.1 软件合成与软件集成13.2 软件集成模式13.2.1 集成适配器模式13.2.2 集成消息器模式13.2.3 集成正面模式13.2.4 集成媒介器模式13.3 企业应用集成13.3.1 EAI的产生和意义13.3.2 企业应用系统的分类13.3.3 企业应用集成的基本原则13.4 数据集成13.4.1 数据集成的基本概念13.4.2 参考数据模型13.4.3 数据集成的元数据管理13.5 基于消息服务的集成框架13.5.1 消息传输模型13.5.2 消息代理任务13.5.3 消息代理拓扑结构13.5.4 消息代理产品的选择13.5.5 使用J2EE的企业消息传递13.6 过程集成13.6.1 工作流和工作流管理系统13.6.2 跨组织过程集成13.7 习题与思考题第14章 软件测试技术14.1 软件测试概述14.1.1 软件测试技术的发展14.1.2 软件开发与软件测试14.1.3 软件测试基本原理与原则14.1.4 软件测试模型14.1.5 测试心理学14.1.6 测试覆盖14.2 软件测试管理14.2.1 质量改进模型PDCA14.2.2 测试需求14.2.3 测试计划14.2.4 测试设计及测试用例14.2.5 测试执行14.2.6 测试记录与跟踪14.2.7 回归测试14.2.8 总结与报告14.3 软件测试技术14.3.1 黑盒测试与白盒测试14.3.2 手工测试与自动化测试14.3.3 单元测试14.3.4 数据库性能检查14.3.5 压力测试14.3.6 安全性测试14.3.7 安装测试14.3.8 环境测试14.4 测试工具14.5 习题与思考题第四篇 运作篇(Operate)第15章 软件架构师角色与职责15.1 软件架构师的定义15.2 软件架构师的工作场景15.3 软件架构师的角色15.4 软件架构师与其他角色的关系及区别15.4.1 软件架构师与产品经理的关系及区别15.4.2 软件架构师与项目经理的关系及区别15.4.3 软件架构师与系统分析员的关系及区别15.5 软件架构师的职责15.6 软件架构师的所应具备的能力15.7 软件架构师的工作评价标准15.8 习题与思考题第16章 软件运作模式16.1 软件企业的崛起16.1.1 大型科研项目的研究是软件业的萌芽16.1.2 独立软件产品的出现预示软件业开始步入正轨16.1.3 企业解决方案让软件业开始兴盛16.1.4 个人电脑及互联网的普及让软件业繁荣16.2 传统的软件运作模式16.3 基于Free的运作模式16.3.1 自由软件16.3.2 免费软件16.3.3 共享软件16.3.4 案例16.4 基于开源的运作模式16.4.1 开源软件16.4.2 开源软件盈利模式16.4.3 案例16.5 基于服务的运作模式16.5.1 软件企业职能转变16.5.2 基础设施即服务(IaaS)16.5.3 平台即服务(PaaS)16.5.4 软件即服务(SaaS)16.6 习题与思考题第17章 案例分析17.1 Google搜索引擎17.1.1 营销模式17.1.2 运作分析17.2 Twitter17.2.1 营销模式17.2.2 运作分析17.3 Facebook17.3.1 营销模式17.3.2 运作分析17.4 淘宝17.4.1 营销模式17.4.2 运作分析17.5 习题与思考题参考文献

<<软件设计与体系结构>>

章节摘录

版权页：插图：1.3.2 面向对象数据库 面向对象技术是一种按照人们对现实世界的认识和思维的方式来研究和模拟客观世界的方法学。

它将现实世界不同的具体事物看成是“对象”，将具有同样性质的某一类对象的群体抽象成“类”，每一个对象都有自己内部的属性和运动规律，不同对象的相互作用和联系构成了我们研究的客观世界。

面向对象方法学所提出的类、对象、方法、消息、封装、继承、多态等概念为我们更好地研究和模拟客观世界提供了坚实的理论基础。

面向对象技术虽然源自于程序设计语言，然而由于面向对象方法学本身所具有的良好机制和能力使得该技术被不断地运用于其他领域：如面向对象软件工程、面向对象操作系统、面向对象数据库等，其中面向对象数据库系统（OODBMS：Object Oriented Data Base Management System）就是数据库技术与面向对象技术的有机结合，并由此诞生了第三代数据库系统，并成为当前数据库领域研究和发展的主要方向之一。

1.面向对象数据库的发展 在面向对象技术和数据库技术相结合的过程中，基本上沿着下面几种途径发展的：（1）纯粹的面向对象数据库管理系统（即OODBMS）。

这种途径往往是以一种面向对象语言为基础，增加数据库的功能，主要是支持持久对象和实现数据共享。

面向对象的数据库不仅在处理多媒体等数据类型时可以做到游刃有余，而且在应用系统开发速度和维护等方面有着极大的优越性。

面向对象数据库系统产生于20世纪80年代后期，它利用类来描述复杂对象，利用类中封装的方法来模拟对象的复杂行为，利用继承性来实现对象的结构和方法的重用。

面向对象数据库系统对一些特定应用领域（如CAD等），能较好地满足其应用需求。

但是，这种纯粹的面向对象数据库系统并不支持SQL语言，在通用性方面失去了优势，因而其应用领域受到了很大的局限。

（2）对象关系数据库管理系统（ORDBMS）。

把面向对象技术与关系数据库相结合，建立对象关系数据库管理系统（ORDBMS），这种系统既支持已经被广泛使用的SQL，具有良好的通用性，又具有面向对象特性，支持复杂对象和复杂对象的复杂行为，是对象技术和传统关系数据库技术的最佳融合。

1997年，对象关系数据库的出现和发展应该算是数据库技术的一次革命，对象技术和关系技术之间互补的优点，吸引着全球数据库厂商竞相研究开发。

如何在现有的关系型数据库产品中加入对象技术，继开发面向对象的关系型数据库成为数据库软件厂商竞争的一个焦点。

（3）ORM（对象关系映射）。

该技术的出现是为了在关系型数据库的基础上解决“阻抗失谐（Impedance Mismatch）”矛盾，简单来说，其本质是：对象系统与关系系统在处理实体之间的互动方面所采取的方式是截然不同的。

表面上看，对象系统和关系系统彼此非常合适，但若深入研究，就会发现二者存在本质差异。

ORM技术提出了一种解决该问题的方法，即在对象层和关系层间建立一个映射层（Mapping Layer），通过这种方式，数据源中的关系数据可以透明地进入对象领域，以对象的面貌出现，供上层应用使用。

<<软件设计与体系结构>>

编辑推荐

《软件工程系列规划教材:软件设计与体系结构》是软件工程系列规划教材之一,可作为计算机相关专业本科生和研究生的教材,同时也是软件工程领域专业人员的优秀参考读物。

<<软件设计与体系结构>>

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>