

<<面向对象高级技术教程>>

图书基本信息

书名：<<面向对象高级技术教程>>

13位ISBN编号：9787302258247

10位ISBN编号：7302258244

出版时间：2012-1

出版时间：清华大学出版社

作者：徐宏础 缺啮

页数：365

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>

<<面向对象高级技术教程>>

内容概要

面向对象 (object oriented , oo) 是当前计算机界关心的重点之一 , 它是20世纪90年代以来软件开发方法的主流 , 国内外许多高校都开设了面向对象的课程 , 但是面向对象涉及的概念很多 , 结构复杂 , 内容广泛 , 使不少初学者感到学习难度较大 , 难以融会贯通。

《面向对象高级技术教程》是一本全面介绍面向对象技术的教材 , 分为基础篇、进阶篇、高级篇三部分。

基础篇重点讲解面向对象的基础知识 , 进阶篇主要讲解了24种面向对象设计模式 , 高级篇介绍了软件架构、基于组件和面向服务的软件开发。

《面向对象高级技术教程》作者参阅了国内外多种相关教材 , 认真分析了初学者在学习过程中遇到的困难 , 为了方便读者的理解和使用 , 本书设计了读者易于学习的教材体系 , 用通俗易懂的语言化解了许多复杂的概念 , 用大量应用实例深入浅出地讲解了面向对象的基础知识和基本应用。本书内容全面 , 例题丰富 , 循序渐进 , 易于学习。

《面向对象高级技术教程》可以作为大学计算机专业本科生、研究生学习设计模式的基础教材 , 也可以作为从事软件研究和软件开发工作有关人员的参考书。

<<面向对象高级技术教程>>

书籍目录

第一部分 基础篇

第1章 面向对象基础

1.1 面向对象方法

- 1.1.1 面向对象方法的内涵
- 1.1.2 面向对象方法的基本概念
- 1.1.3 面向对象的产生
- 1.1.4 面向对象的优势

1.2 类和类的成员

- 1.2.1 类和对象
- 1.2.2 类的成员类型和访问控制
- 1.2.3 字段
- 1.2.4 属性
- 1.2.5 方法
- 1.2.6 构造函数和析构函数
- 1.2.7 委托
- 1.2.8 事件
- 1.3 面向对象的特征
- 1.3.1 封装
- 1.3.2 继承
- 1.3.3 多态

本章小结

习题

第2章 面向对象的开发过程

2.1 面向对象分析

- 2.1.1 概论
- 2.1.2 需求陈述
- 2.1.3 建立对象模型
- 2.1.4 建立动态模型
- 2.1.5 建立功能模型

2.2 面向对象设计

- 2.2.1 面向对象设计的准则
- 2.2.2 问题域部分设计
- 2.2.3 人机交互部分设计
- 2.2.4 任务管理部分设计
- 2.2.5 数据管理部分设计

2.3 面向对象的编程实现

- 2.3.1 编程语言的选择
- 2.3.2 面向对象语言的技术支持
- 2.3.3 面向对象程序设计风格

2.4 面向对象的测试

- 2.4.1 面向对象测试概述
- 2.4.2 面向对象测试策略
- 2.4.3 设计测试用例

本章小结

习题

<<面向对象高级技术教程>>

第3章 面向对象软件设计基础

3.1 面向对象设计视点

3.1.1 复用

3.1.2 扩展

3.1.3 分离与封装变化

3.1.4 低耦合高内聚

3.2 面向对象软件设计原则

3.2.1 开放封闭原则

3.2.2 单一职责原则

3.2.3 里氏代换原则

3.2.4 依赖倒转原则

3.2.5 接口隔离原则

3.2.6 迪米特法则

3.2.7 其他原则

3.3 uml类图简介

3.3.1 类

3.3.2 接口

3.3.3 继承

3.3.4 实现

3.3.5 依赖

3.3.6 关联

3.3.7 聚合（联合）

3.3.8 组合（合成）

本章小结

习题

第二部分 进阶篇

第4章 设计模式基础

4.1 概述

4.1.1 什么是设计模式

4.1.2 设计模式的基本要素

4.1.3 怎样使用设计模式

4.2 设计模式的类型

4.2.1 创建型设计模式

4.2.2 结构型设计模式

4.2.3 行为型设计模式

本章小结

习题

第5章 创建型设计模式

5.1 简单工厂模式

5.1.1 意图

5.1.2 角色及其职责

5.1.3 应用示例

5.1.4 效果分析

5.1.5 特点

5.1.6 适用性

5.2 工厂方法模式

5.2.1 意图

<<面向对象高级技术教程>>

5.2.2角色及其职责

5.2.3应用示例

5.2.4效果分析

5.2.5特点

5.2.6适用性

5.3抽象工厂模式

5.3.1意图

5.3.2角色及其职责

5.3.3应用示例

5.3.4效果分析

5.3.5特点

5.3.6适用性

5.4建造者模式

5.4.1意图

5.4.2角色及其职责

5.4.3应用示例

5.4.4效果分析

5.4.5特点

5.4.6适用性

5.5单件模式

5.5.1意图

5.5.2角色及其职责

5.5.3应用示例

5.5.4效果分析

5.5.5特点

5.5.6适用性

5.6原型模式

5.6.1意图

5.6.2角色及其职责

5.6.3应用示例

5.6.4效果分析

5.6.5特点

5.6.6适用性

本章小结

习题

第6章 结构型设计模式

6.1适配器模式

6.1.1意图

6.1.2角色及其职责

6.1.3应用示例

6.1.4效果分析

6.1.5特点

6.1.6适用性

6.2装饰模式

6.2.1意图

6.2.2角色及其职责

6.2.3应用示例

<<面向对象高级技术教程>>

- 6.2.4效果分析
- 6.2.5特点
- 6.2.6适用性
- 6.3桥接模式
 - 6.3.1意图
 - 6.3.2角色及其职责
 - 6.3.3应用示例
 - 6.3.4效果分析
 - 6.3.5特点
 - 6.3.6适用性
- 6.4享元模式
 - 6.4.1意图
 - 6.4.2角色及其职责
 - 6.4.3应用示例
 - 6.4.4效果分析
 - 6.4.5特点
 - 6.4.6适用性
- 6.5外观模式
 - 6.5.1意图
 - 6.5.2角色及其职责
 - 6.5.3应用示例
 - 6.5.4效果分析
 - 6.5.5特点
 - 6.5.6适用性
- 6.6代理模式
 - 6.6.1意图
 - 6.6.2角色及其职责
 - 6.6.3应用示例
 - 6.6.4效果分析
 - 6.6.5特点
 - 6.6.6适用性
- 6.7组合模式
 - 6.7.1意图
 - 6.7.2角色及其职责
 - 6.7.3应用示例
 - 6.7.4效果分析
 - 6.7.5特点
 - 6.7.6适用性
- 本章小结
- 习题
- 第7章 行为型设计模式
 - 7.1模板方法模式
 - 7.1.1意图
 - 7.1.2角色及其职责
 - 7.1.3应用示例
 - 7.1.4效果分析
 - 7.1.5特点

<<面向对象高级技术教程>>

- 7.1.6适用性
- 7.2观察者模式
 - 7.2.1意图
 - 7.2.2角色及其职责
 - 7.2.3应用示例
 - 7.2.4效果分析
 - 7.2.5特点
 - 7.2.6适用性
- 7.3迭代器模式
 - 7.3.1意图
 - 7.3.2角色及其职责
 - 7.3.3应用示例
 - 7.3.4效果分析
 - 7.3.5特点
 - 7.3.6适用性
- 7.4责任链模式
 - 7.4.1意图
 - 7.4.2角色及其职责
 - 7.4.3应用示例
 - 7.4.4效果分析
 - 7.4.5特点
 - 7.4.6适用性
- 7.5备忘录模式
 - 7.5.1意图
 - 7.5.2角色及其职责
 - 7.5.3应用示例
 - 7.5.4效果分析
 - 7.5.5特点
 - 7.5.6适用性
- 7.6命令模式
 - 7.6.1意图
 - 7.6.2角色及其职责
 - 7.6.3应用示例
 - 7.6.4效果分析
 - 7.6.5特点
 - 7.6.6适用性
- 7.7状态模式
 - 7.7.1意图
 - 7.7.2角色及其职责
 - 7.7.3应用示例
 - 7.7.4效果分析
 - 7.7.5特点
 - 7.7.6适用性
- 7.8访问者模式
 - 7.8.1意图
 - 7.8.2角色及其职责
 - 7.8.3应用示例

<<面向对象高级技术教程>>

7.8.4效果分析

7.8.5特点

7.8.6适用性

7.9中介者模式

7.9.1意图

7.9.2角色及其职责

7.9.3应用示例

7.9.4效果分析

7.9.5特点

7.9.6适用性

7.10策略模式

7.10.1意图

7.10.2角色及其职责

7.10.3应用示例

7.10.4效果分析

7.10.5特点

7.10.6适用性

7.11解释器模式

7.11.1意图

7.11.2角色及其职责

7.11.3应用示例

7.11.4效果分析

7.11.5特点

7.11.6适用性

本章小结

习题

第三部分 高级篇

第8章 软件架构与架构建模技术

8.1软件架构概况

8.1.1软件架构的发展史

8.1.2软件架构的定义

8.2经典软件架构模式

8.2.1管道和过滤器模式

8.2.2面向对象模式

8.2.3分层模式

8.2.4知识库模式

8.3客户机/服务器模式

8.3.1客户机/服务器模式的产生

8.3.2传统两层客户机/服务器模式

8.3.3经典三层客户机/服务器模式

8.4浏览器/服务器模式

8.5基于构件的模式

8.6软件架构建模技术

8.6.1软件架构“4+1”视图模型

8.6.2“4+1”视图模型建模方法

8.6.3软件架构建模的迭代过程

本章小结

<<面向对象高级技术教程>>

习题

第9章 基于组件的软件开发

9.1 组件开发概述

9.1.1 组件及其相关概念

9.1.2 组件复用

9.2 组件模型

9.2.1 corba

9.2.2 ejb

9.2.3 com

9.2.4 三种组件模型的技术对比

本章小结

习题

第10章 面向服务的软件架构——soa

10.1 soa简介

10.1.1 soa参考模型

10.1.2 soa的设计原则

10.1.3 soa实现的主要技术规范

10.2 soa的框架

10.2.1 以服务消费者为中心的soa

10.2.2 以用户为中心的soa

10.3 soa实例——基于soa的oa与erp整合应用

10.4 soa的应用分析

本章小结

习题

参考文献

<<面向对象高级技术教程>>

章节摘录

版权页：插图：4.物理视图——物理架构物理视角主要对系统的物理架构进行建模，是软件至硬件的映射。

物理架构主要关注系统非功能性的需求，如可用性、可靠性（容错性）、性能（吞吐量）和可伸缩性。软件在计算机网络或处理结点上运行，被识别的各种元素（网络、过程、任务和对象），需要被映射至不同的结点；我们希望使用不同的物理配置：一些用于开发和测试，另外一些则用于不同地点和不同客户的部署。

因此软件至结点的映射需要高度的灵活性及对源代码产生最小的影响。

5.场景场景综合所有的视图，对系统总体架构进行建模。

4种视图的元素通过数量比较少的一组重要场景（更常见的是用例）进行无缝协同工作，我们为场景描述相应的脚本（对象之间和过程之间的交互序列）。

在某种意义上场景是最重要的需求抽象。

场景是其他视图的冗余（因此才称为“+1”），但它起到了两个作用：一是作为一项驱动因素来发现架构设计过程中的架构元素；二是作为架构设计结束后的一项验证和说明功能，既以视图的角度来说明又作为架构原型测试的出发点。

6.模型的剪裁并不是所有的软件架构都需要“4+1”视图。

无用的视图可以从架构描述中省略，比如：只有一个处理器，则可以省略物理视图；而如果仅有一个进程或程序，则可以省略过程视图。

对于非常小型的系统，甚至可能逻辑视图与开发视图非常相似，而不需要分开描述。

但场景对于所有的情况均适用，一般不能省略。

8.6.3软件架构建模的迭代过程软件架构的建模过程不是一个简单的线性过程，而是一个循环迭代的过程。

软件架构的建模使用一种更具有迭代性质的方法，即架构先被原形化、测试、估量、分析，然后在一系列的迭代过程中被细化。

该方法除了减少与架构相关的风险之外，对于项目而言还有其他优点：团队合作、培训，加深对架构的理解，深入程序和工具等（此处提及的是演进的原形，逐渐发展成为系统，而不是一次性的试验性的原形）。

这种迭代方法还能够使需求被细化、成熟化并能够被更好地理解。

<<面向对象高级技术教程>>

编辑推荐

《面向对象高级技术教程》是作者多年从事软件开发的经验总结。

书中应用各类设计模式实例深入浅出地阐述了面向对象技术的设计思想，介绍给读者最实用的软件开发理念和知识。

书中提供大量的例子和练习，并给出部分答案，有助于加深对设计模式的理解，提升软件设计视野，让设计理念更加宽广。

对任何希望深入理解高级面向对象设计灵魂和精髓的读者来说，《面向对象高级技术教程》都是不容错过的经典书籍。

也适合作为高等院校计算机及相关专业本科生、研究生的教材，以及从事软件研究和开发的工作人员使用。

<<面向对象高级技术教程>>

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>