

目 录

序

第 1 章 引言

- 1.1 软件工程的基本概念
- 1.2 软件需求分析和开发的结构化方法
 - 1.2.1 结构化分析与设计方法
 - 1.2.2 Jackson 系统开发方法
- 1.3 结构化方法的特点与不足
 - 1.3.1 结构化方法的特点
 - 1.3.2 结构化方法存在的不足
- 1.4 面向对象技术
 - 1.4.1 面向对象的分析、设计与编程方法
 - 1.4.2 面向对象软件工程
 - 1.4.3 OMT 方法
 - 1.4.4 一体化建模语言
- 1.5 Booch 方法简介

第 2 章 对象模型

- 2.1 对象模型的基本元素
 - 2.1.1 抽象
 - 2.1.2 封装
 - 2.1.3 模块化
 - 2.1.4 层次性
 - 2.1.5 类型
 - 2.1.6 并发性
 - 2.1.7 持久性
- 2.2 面向对象方法与对象模型的优点

第 3 章 对象和类

- 3.1 对象
 - 3.1.1 对象的状态
 - 3.1.2 对象的行为
 - 3.1.3 对象的标识
 - 3.1.4 对象的生命周期
- 3.2 对象间的关系
 - 3.2.1 连接
 - 3.2.2 聚合
- 3.3 类
- 3.4 对象与类之间的关系及不同作用

第 4 章 对象与类的抽象机制及表示方法

- 4.1 类与对象抽象机制与方法
- 4.2 如何建造高质量的类和对象
 - 4.2.1 检验抽象的质量
 - 4.2.2 选择合适的操作
 - 4.2.3 选择合适的关系
 - 4.2.4 选择合适的实施方法
- 4.3 对象与类的基本表示形式
 - 4.3.1 类图及类范畴图
 - 4.3.2 其它类图表示形式
 - 4.3.3 形式化定义
- 4.4 状态转换图
 - 4.4.1 状态
 - 4.4.2 态转换的深入的概念
- 4.5 对象图
- 4.6 交互图
- 4.7 模块图
- 4.8 处理图

第 5 章 面向对象的设计工具

- 5.1 概述
 - 5.1.1 面向对象的设计工具的特点
 - 5.1.2 面向对象方法的系列设计开发工具
- 5.2 面向对象的设计工具 Rational Rose/C++简介
- 5.3 Rose/C++使用方法
- 5.4 使用 ROSE/C++建立系统模型
 - 5.4.1 Rose 的安装与启动
 - 5.4.2 类图的建立
 - 5.4.3 面向对象设计中其它模型的建立
- 5.5 ROSE/C++对面向对象设计实施提供的其它支持功能
 - 5.5.1 Use-Case 建模工具
 - 5.5.2 UML 建模功能
 - 5.5.3 Rose 对代码生成、DDL 和 IDL 接口的支持
 - 5.5.4 方便的用户联机帮助和自学指导

第 6 章 面向对象设计开发过程

- 6.1 面向对象设计开发过程概述
- 6.2 微过程方法
 - 6.2.1 确定类和对象
 - 6.2.2 确定类和对象语义
 - 6.2.3 确定类和对象之间的关系
 - 6.2.4 实施类和对象
- 6.3 面向对象设计开发主流程
 - 6.3.1 概念建立
 - 6.3.2 系统分析

6.3.3 系统设计

6.3.4 系统实施

6.3.5 运行维护

第7章 面向对象软件开发规范与文档生成工具

7.1 面向对象软件开发规范

7.2 面向对象软件设计开发文档细则

7.2.1 文档组成

7.2.2 文档细则

7.3 面向对象文档生成工具 SoDA for Word

7.3.1 SoDA 简述

7.3.2 SoDA 入门

7.3.3 SoDA 模板定制

7.3.4 模板示例

第8章 应用实例研究

8.1 应用实例背景介绍

8.1.1 集成平台的产生背景

8.1.2 CIMS 应用集成平台的基本概念

8.1.3 制造业 CIMS 应用集成平台的体系结构与功能

8.1.4 运控代理模型

8.1.5 控制代理

8.2 应用实例的面向对象设计分析

8.2.1 概念建立阶段文档

8.2.2 系统分析阶段文档

8.2.3 系统设计阶段文档

8.2.4 系统实施阶段文档

8.2.5 运行维护阶段文档

8.3 结论

第9章 软件设计开发技术展望

9.1 CORBA 规范及其应用简介

9.1.1 CORBA 的产生

9.1.2 对象管理参考模型及 CORBA 体系结构

9.1.3 CORBA 的应用状况

9.2 软件代理技术简介

9.2.1 为什么软件需要代理

9.2.2 什么是软件代理

9.2.3 代理通信语言

9.2.4 代理实现的典型应用

参考文献

附录:BOOCH 表示方法快速参考图例