

第1章 面向对象技术概述

本章学习目标

- 了解两种系统设计方法学
- 了解面向对象的基本概念
- 了解面向对象的分析与设计
- 了解面向对象方法的优缺点
- 了解面向对象的程序设计语言
- 了解典型的面向对象方法

第1章 面向对象技术概述

- 1.1 结构化方法和面向对象方法
- 1.2 面向对象方法的由来
- 1.3 面向对象的基本概念与术语
- 1.4 面向对象的软件开发
- 1.5 面向对象方法的优缺点
- 1.6 面向对象程序设计语言
- 1.7 面向对象的分析与设计
- 1.8 典型的面向对象方法

第1.1节 结构化方法和面向对象方法

- 结构化方法

- 特点：根据系统的功能，来分解系统

- 原则：自顶向下，逐层分解，模块化，模块之间的接口

- 模型：数据流图，数据字典，E-R图

第1.1节 结构化方法和面向对象方法

■ 面向对象方法

- 特点：根据系统的客观对象及事务，来分解系统
- 原则：自顶向下，逐层分解，对象/类，对象之间的消息
- 模型：对象协作图，类图

第1.1节 结构化方法和面向对象方法

■例1：花名册管理系统

- 结构化方法：新建花名册，打开花名册，添加学生，修改学生信息，删除学生

- 面向对象方法：学生，班级，专业，年级，花名册

第1.2节 面向对象方法的由来

- **20世纪60年代**，产生了结构化程序设计思想
- 在**20世纪70年代到80年代**，结构化方法成为了所有软件开发设计领域及每个程序员都采用的方法
- **不足**：把数据和处理数据的过程分离。当数据结构改变时，所有相关的处理过程都要进行相应的修改

第1.2节 面向对象方法的由来

- **20世纪80年代**，产生了面向对象编程语言，之后产生了面向对象设计思想
- **进入21世纪**，大部分（**95%**）的软件系统都采用面向对象方法进行设计
- **长处**：把数据和处理数据的过程封装在一起。当数据结构改变时，只考虑相应的对象本身，以及有消息联系的其他对象

第1.3节 面向对象的基本概念

- 对象
- 是客观现实世界中一个实际存在的事物
- 类
- 具有相同属性和行为的一组对象的集合
- 封装
- 把对象的属性和行为组成一个独立的系统单位，并尽可能隐蔽对象的内部细节

第1.3节 面向对象的基本概念

■ 继承

- 特殊类的对象拥有其一般类的全部属性和行为

■ 多态性

- 多态性是指在一般类中定义的属性或行为，被特殊类继承之后，可以具有不同的数据类型或表现出不同的行为

第1.3节 面向对象的基本概念

- 消息通信
- 对象之间只能通过消息进行通信，不允许在对象之外直接存取对象内部的属性
- Peter Coad和Edward Yourdon的观点
- 面向对象=对象+类+继承+消息通信

第1.4节 面向对象的软件开发

- 面向对象分析**OOA**
- 面向对象设计**OOD**
- 面向对象编程**OOP**
- 面向对象测试**OOT**
- 面向对象的软件维护**OOSM**

第1.4节 面向对象的软件开发

- 面向对象分析**OOA**
- 根据系统的需求，找出所有的对象，对象的属性及方法。根据对象的共性，定义类。
- 面向对象设计**OOD**
- 对象、属性、方法，三者封装在一起。对象之间的协作用消息通信实现

第1.4节 面向对象的软件开发

- 面向对象编程**OOP**
- 选择一门面向对象的编程语言，实现每个对象的属性定义和方法中的代码定义
- 面向对象测试**OOT**
- 以对象/类为基本单位，进行测试。进一步测试对象之间的消息通信

第1.4节 面向对象的软件开发

- 面向对象的软件维护 **OOSM**
- 未来不会改变的功能，封装到固定的对象/类上
- 未来会改变的功能，通过修改相应对象/类的属性/方法来实现
- 未来会新增的功能，通过新增相应对象/类的方法来实现
- 未来会新增的功能，通过定义子类或接口实现来实现

第1.5节 面向对象方法的优缺点

■ 优点

- 对软件质量的支持
- 对模块性的支持
- 对软件开发过程的支持

第1.5节 面向对象方法的优缺点

■缺点

- 开发可重用模块要增加项目的成本
- 缺乏商业上可用的对象库、类库、构件库
- 构件的分类、检索和评价是一个复杂的问题。有独立于领域的构件和依赖于领域的构件
- 如果要全面地采用对象技术，肯定会引起组织和文化上的变动

第1.6节 面向对象编程设计语言

- Simula和Smalltalk
- Objective-C, C++, C#
- Eiffel
- Java
- Ada
- PowerBuilder

第1.7节 面向对象的分析与设计

- 什么是软件的分析与设计
- 分析是解决“做什么”的问题。软件人员根据用户的需求描述，归类出软件应该具备哪些功能
- 在**SA**中，得出哪些模块
- 在**OOA**中，得出哪些对象/类

第1.7节 面向对象的分析与设计

- 什么是软件的分析与设计
- 设计是解决“怎么做”的问题。软件人员根据用户的工作流程描述，归类出软件代码的操作步骤
- 在**SA**中，得出单个模块的程序流程图，多个模块之间的接口
- 在**OOA**中，得出对象/类的单个方法的活动图，多个对象之间的消息通信

第1.7节 面向对象的分析与设计

- 面向对象的分析与设计的一般过程

- 1) 描述需求
- 2) 识别潜在对象
- 3) 筛选对象
- 4) 对象的命名
- 5) 识别对象的属性
- 6) 识别对象的行为
- 7) 识别对象所属的类
- 8) 定义类的结构

第1.8节 典型的面向对象方法

- **Coad & Yourdon方法**
- **OOA和OOD采用完全一致的方法**
- **主要概念：对象，类，属性，服务，整体-部分结构，一般-特殊结构，实例连接，消息连接，主题**
- **OOA模型包含5层：主题层，类和对象层，结构层，属性层，服务层**

第1.8节 典型的面向对象方法

- **Rumbaugh方法**
- **对象建模技术OMT**
- **3个视角表述完整的系统：对象模型，动态模型，功能模型**
- **对象模型是系统的“数据”视图，动态模型是系统的“控制”视图，功能模型是系统的“功能”视图**

第1.8节 典型的面向对象方法

■ Booch方法

- OO开发过程分为：微过程，宏过程。
是一个渐进、反复的过程
- 宏过程对应整个软件生命周期各阶段
- 每个阶段包含若干个微过程
- 微过程有4个活动
- 模型图有：类图，对象图，状态转换图，
时序图，模块图，进程图

第1.8节 典型的面向对象方法

- **Jacobson方法**
- 面向对象软件工程**OOSE**
- **OO**开发过程分为：分析，构造，测试
- 需求模型，分析模型，设计模型，实现模型
- 用例驱动
- 每个用例建立一个交互图

第1.8节 典型的面向对象方法

■ **RDD方法**

- 职责驱动法**RDD**， 又称**CRC**（类-责任-协作）法
- 职责概括了一个对象应做的事
- 使用**CRC**卡片， 设计系统模型
- 寻找“类”的一种利器

第1章 面向对象技术概述

- 作业1——P27
- 1.10.3、简答题：1~7
- 1.10.4、简单分析题：银行取款系统，找出所有的对象，每个对象的属性及方法

第1章 面向对象技术概述

End

Thanks

侯爱民, 13538377208, 748697

zhhouam@163.com