

9面向对象设计 (4*)

9.1面向对象设计概述

面向对象工程是以面向对象方法为基础，用UML模型元素描述需求、设计、实现和测试的系统开发全过程，OOD建立在OOA的基础上。

面向对象设计原则

1.信息隐藏和模块化

2.重用

代码重用

设计模式重用

3.单一原则

4.划和统一接口

5.优先使用聚合

6.开放封闭原则

所谓开放性原则是指对系统功能扩展的完善性设计，应该立足于现有类的基础上提供新的属性和行为，尽量避免类的重新开发。

9.2精化类及类间关系 (5*)

OOA的类图没有详细描述类中包含的属性、提供的方法和类间关系的细节。

OOD的首要任务是精化类的设计，不仅详细定义类的属性、方法、关联，而且结合功能模型和动态模型给出的UML图，分析出抽象类、领域类、边界类等一系列新类，为类图的完整定义和软件系统的修改、扩展维护提供灵活的设计方案。

9.2.1设计类的属性

(1) 复杂属性的分离和描述

(2) 类间重数的属性表示

在类中定义指针，指向另一个关联类的对象列表

如果编程语言不支持指针，通过定义关联类的对象数组来实现

将一对多转化为多个一对一

(3) 对属性的约束

(4) 对属性值的初始化

(5) 导出新的“属性”

9.2.2设计类的方法

(1) 具有公共服务性质的方法，应该放置在继承结构的最高层类中，使得方法重用达到最大化。如果公共方法过多，或涉及核心算法，可定义新类来封装他们。

(2) 尽量在已有类中定义新方法，或重用已有的代码。

(3) 反映类之间的动态关系，即类间的每个消息都要有相应的操作。

9.2.3设计类间泛化关系

类的泛化关系分为单继承和多继承

1.单继承和聚合

单继承：继承该类来访问该类

聚合：在一个类中定义另一个类的子对象来访问该类

如果只是使用类提供的方法，聚合会更好

2.多继承和转换

多继承会导致二义性问题

解决方式

将多继承转换为单继承

将多继承转换为聚合方式