

第三部分 设计篇

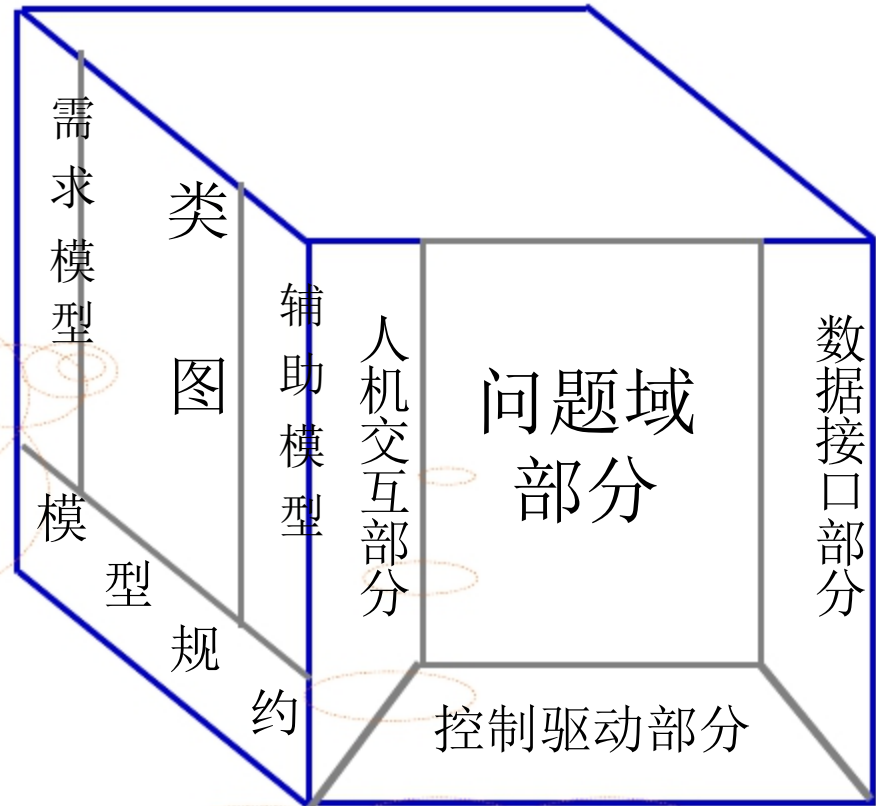
福州大学数学与计算机科学学院
面向对象分析与设计

OOD模型框架

——从两个侧面来描述

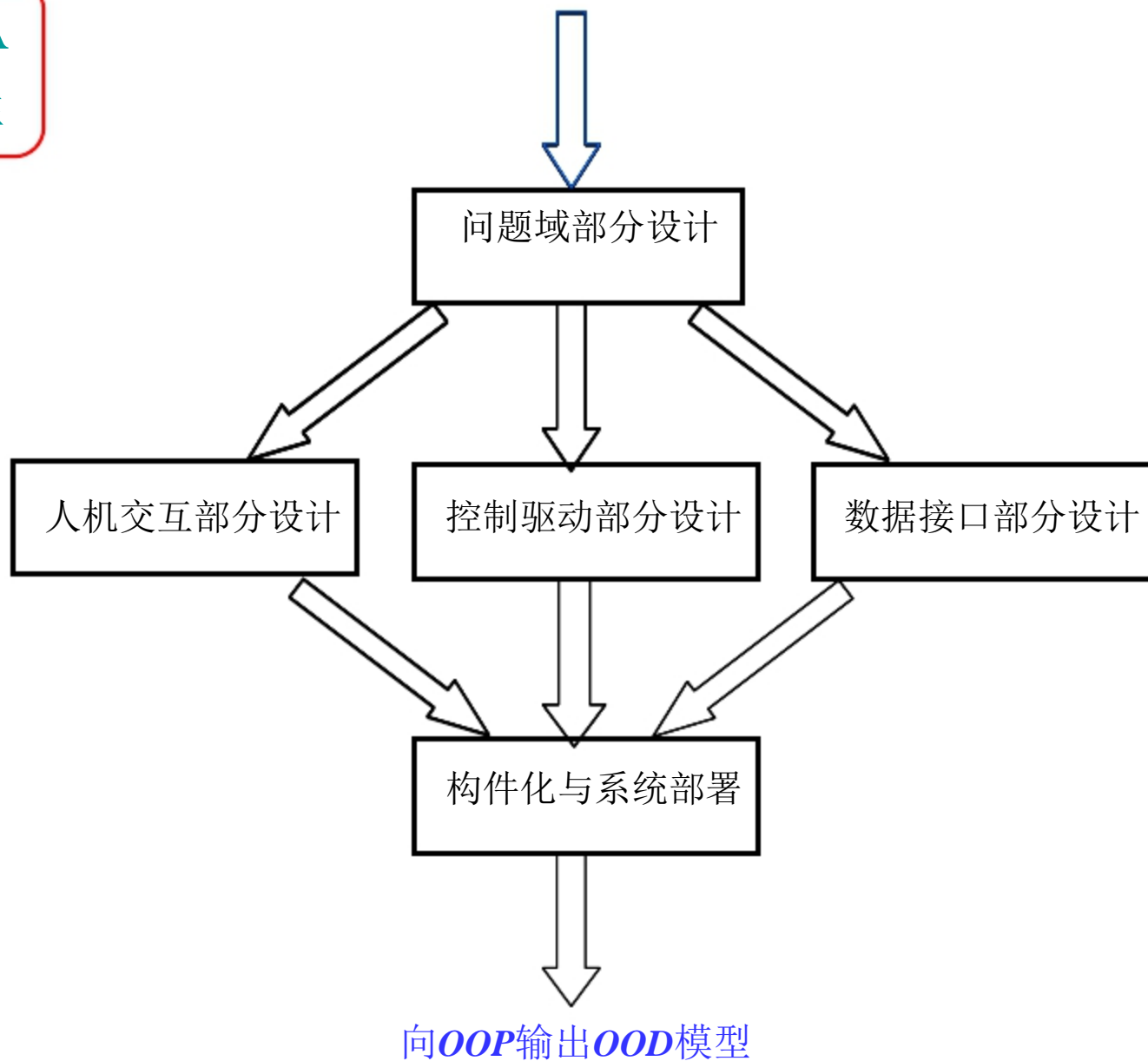
回顾

从另一侧面看：
OOD模型每个部分
如何用OO概念表达？
采用与OOA相同的概念及
模型组织方式



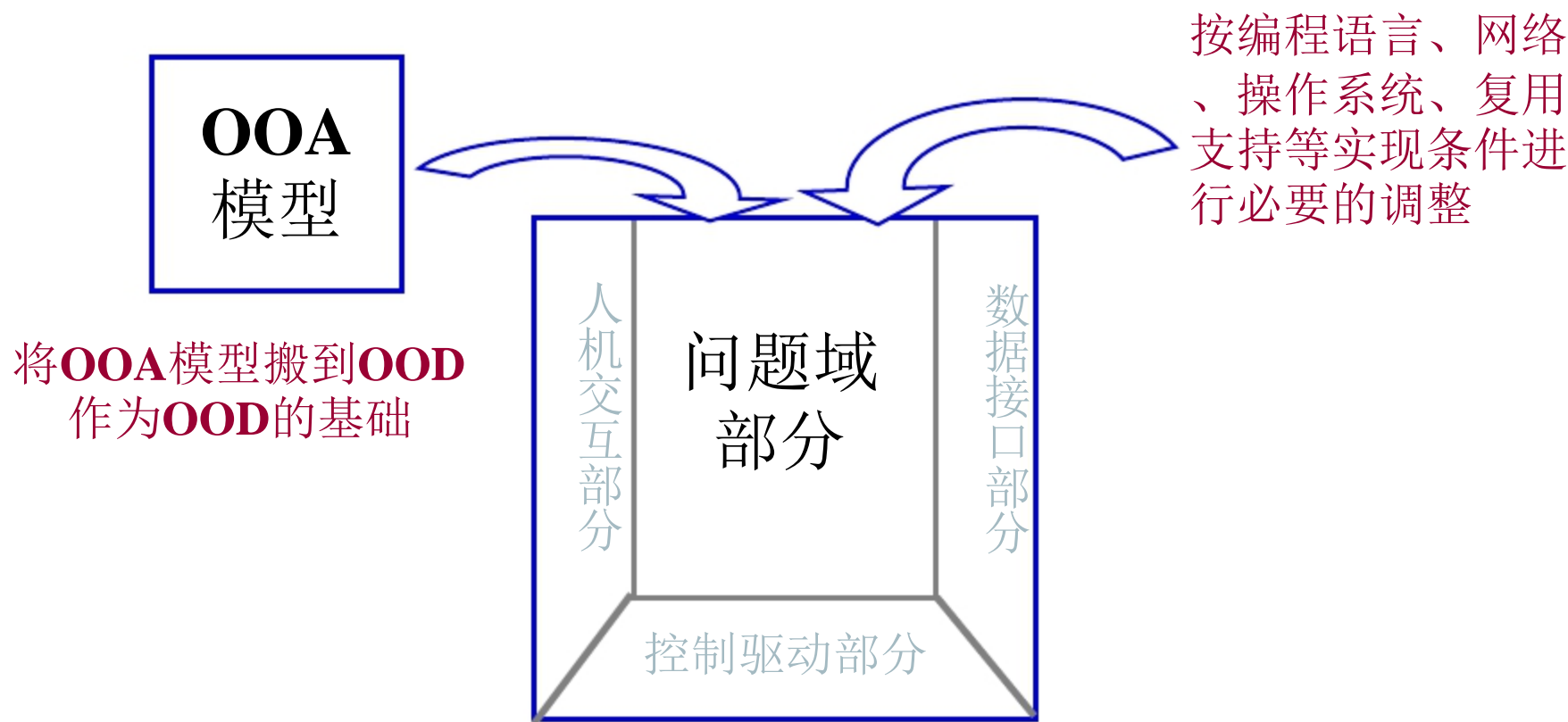
从一个侧面看：
OOD模型包括几个主要部分？
一个核心加三个外围

OOA 过程

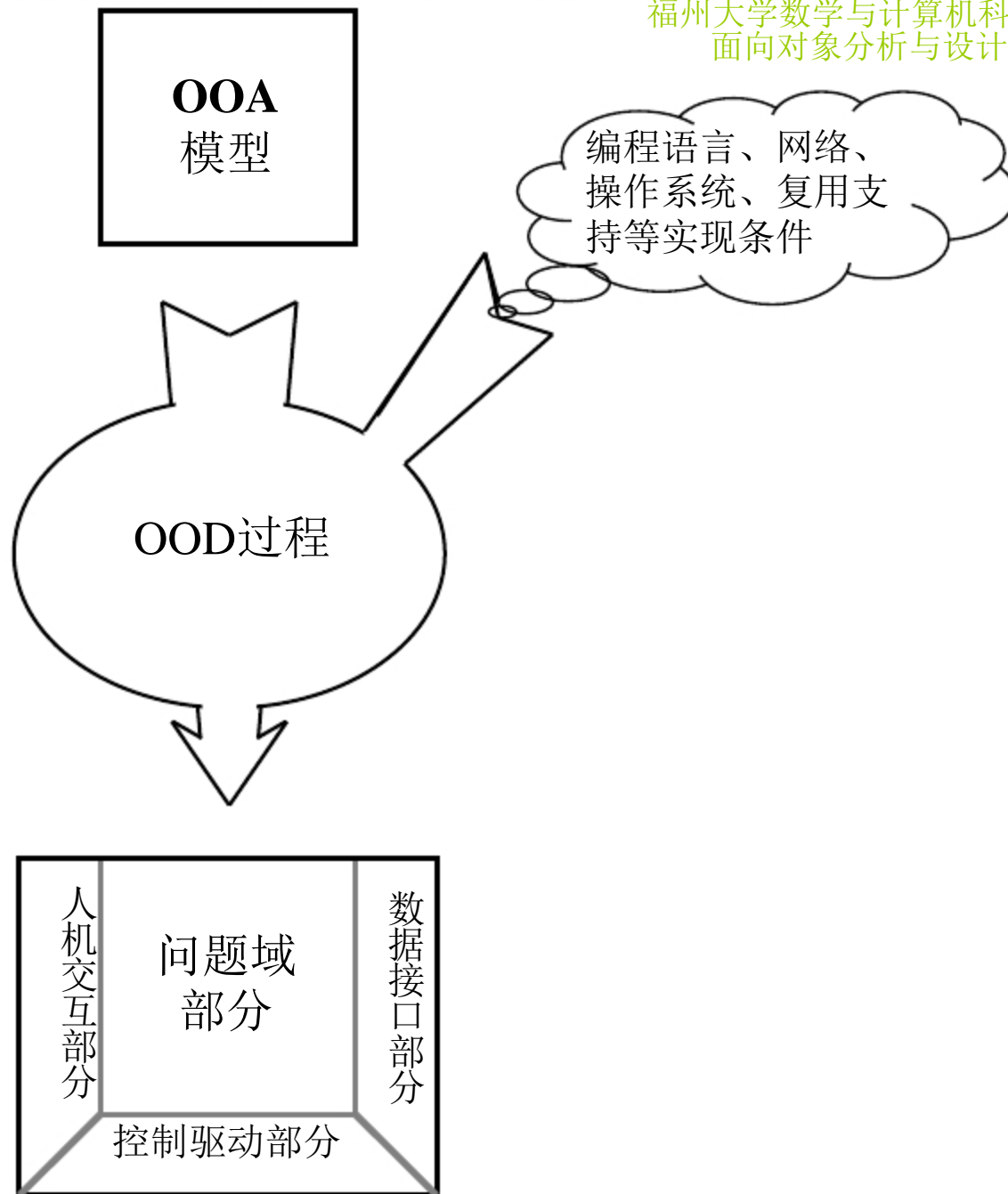


11.1 什么是问题域部分

问题域部分是**OOD**模型的四个组成部分之一，由来自问题域的对象构成，是在**OOA**模型基础上，按照具体的实现条件进行必要的修改、调整和细节补充而得到的。



从MDA的
观点看问
题域部分
的产生



11.2 实现条件对问题域部分的影响

编程语言

语言的实现能力

硬件、操作系统及网络设施

对象分布、并发、通信、性能

复用支持

根据复用支持对模型做适当调整，以实现复用

数据管理系统

对问题域部分做某些修改以实现数据的永久存储

界面支持系统

问题域部分与人机界面之间的消息传输

11.3 设计过程

设计准备

- 保留**OOA**文档

- 复制**OOA**文档，作为**OOD**的输入

- 根据需求的变化和发现的错误进行修改

设计内容与策略（本节的重点）

- 针对编程语言支持能力的调整

- 增加一般类以建立共同协议

- 实现复用

- 提高性能

- 为实现对象永久存储所做的修改

- 完善对象的细节

- 定义对象实例

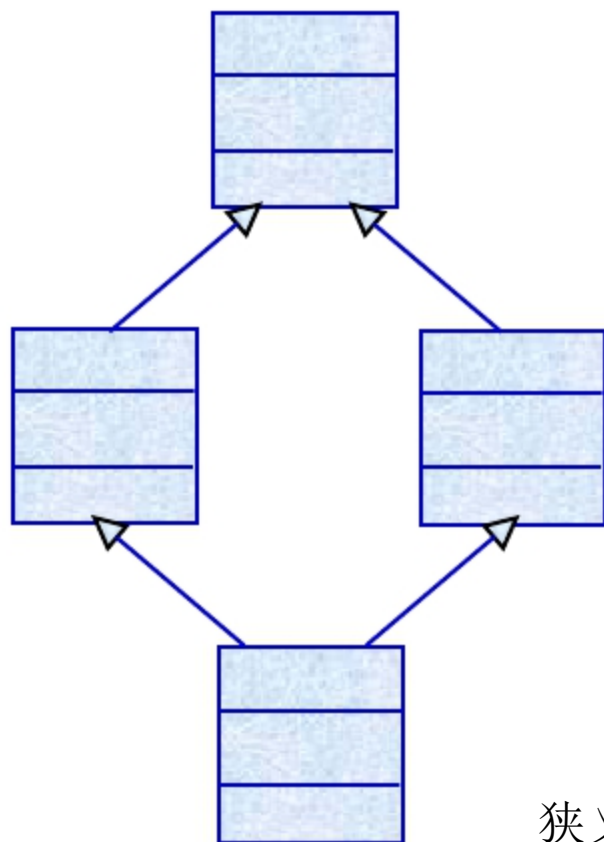
- 对辅助模型、模型规约的修改和补充

建立**OOD**文档与**OOA**文档的映射

1、按编程语言调整继承与多态

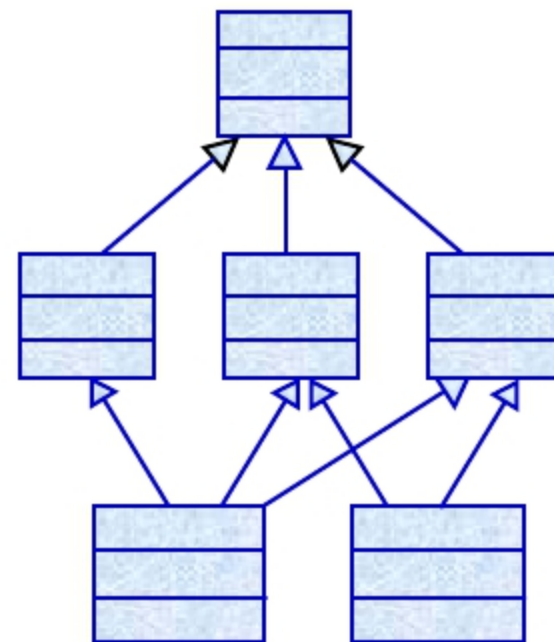
起因：**OOA**强调如实地反映问题域，**OOD**考虑实现问题，
所用语言不支持多继承或者不支持多态

(1) 多继承化为单继承



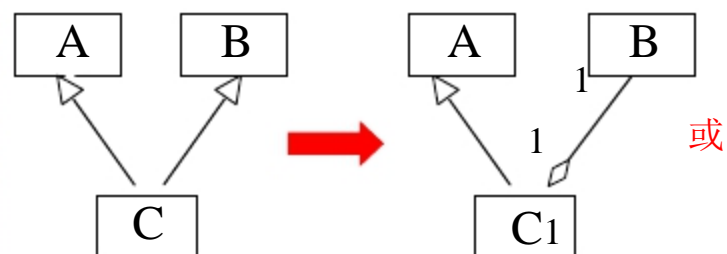
狭义菱形

多继承模式

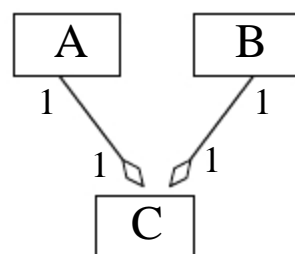


广义菱形

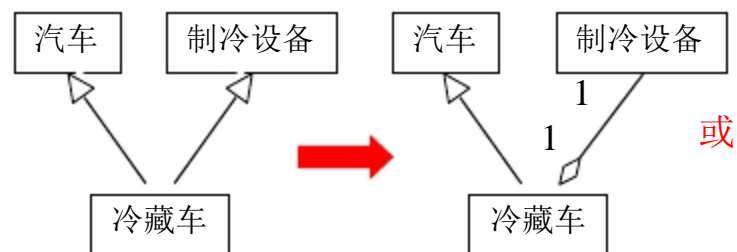
方法1：简单转换



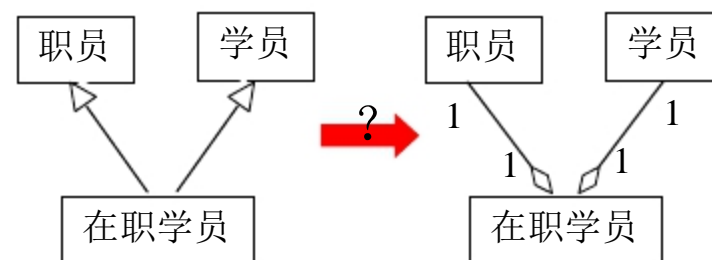
(a) 一般方法



(c) 不合适的例子

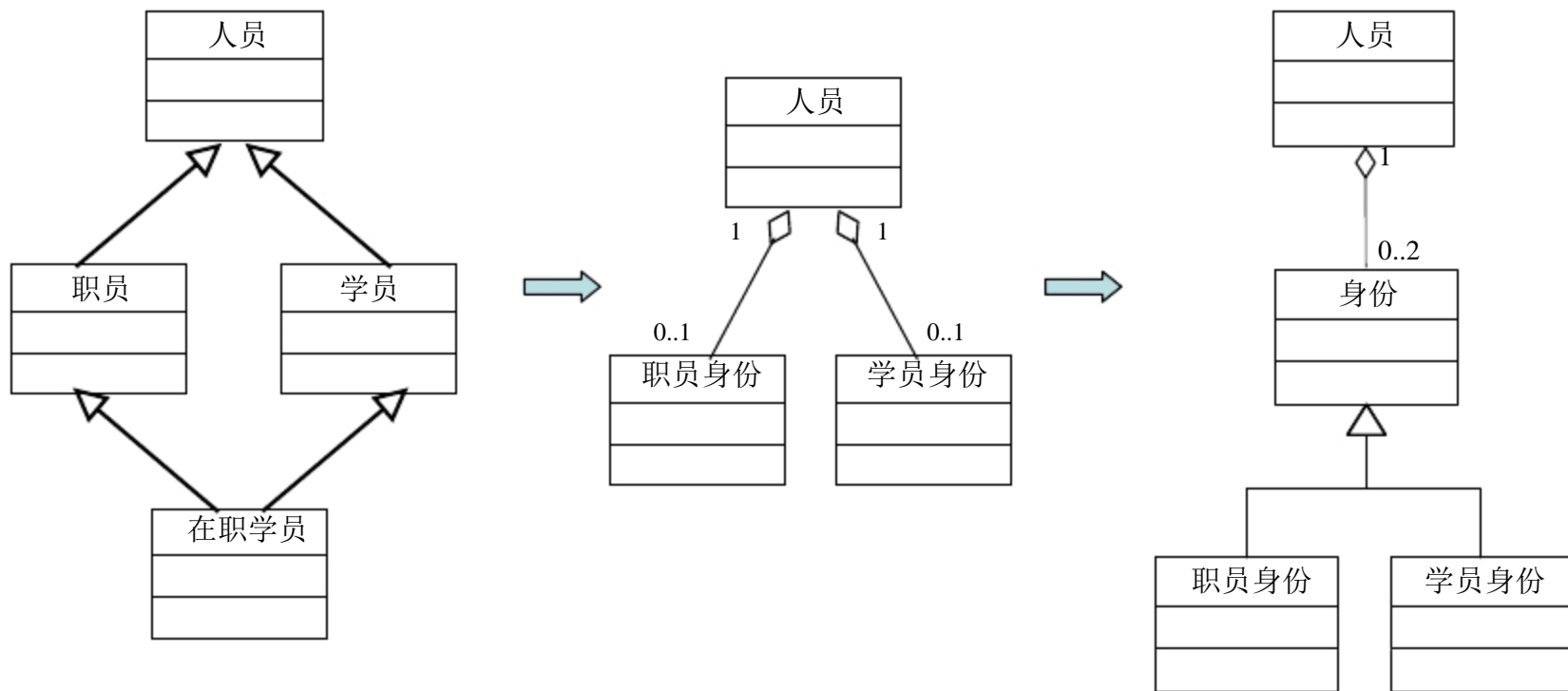


(b) 合适的例子

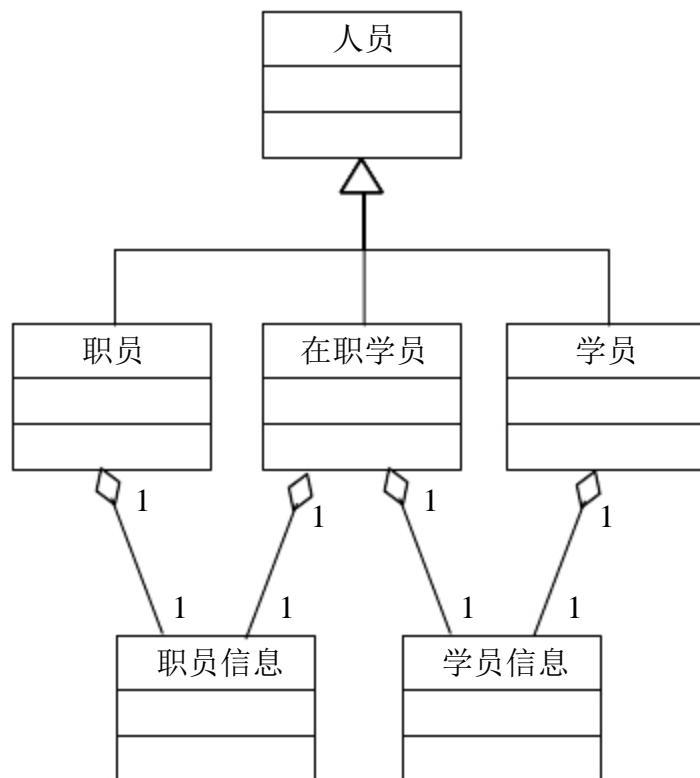


(d) 转换产生信息重复

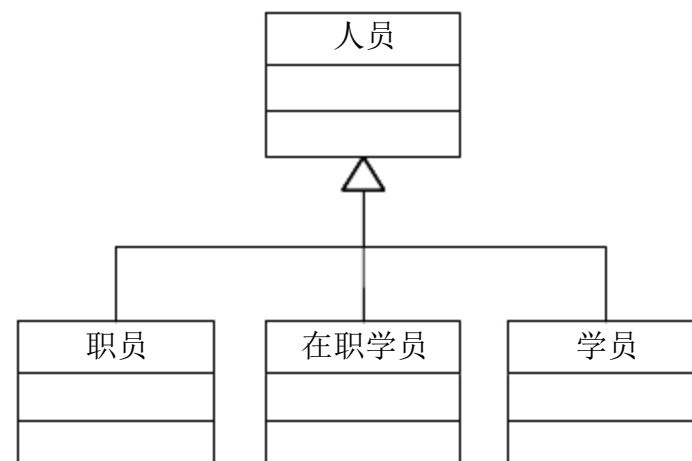
方法2：重新定义对象类，化解多继承



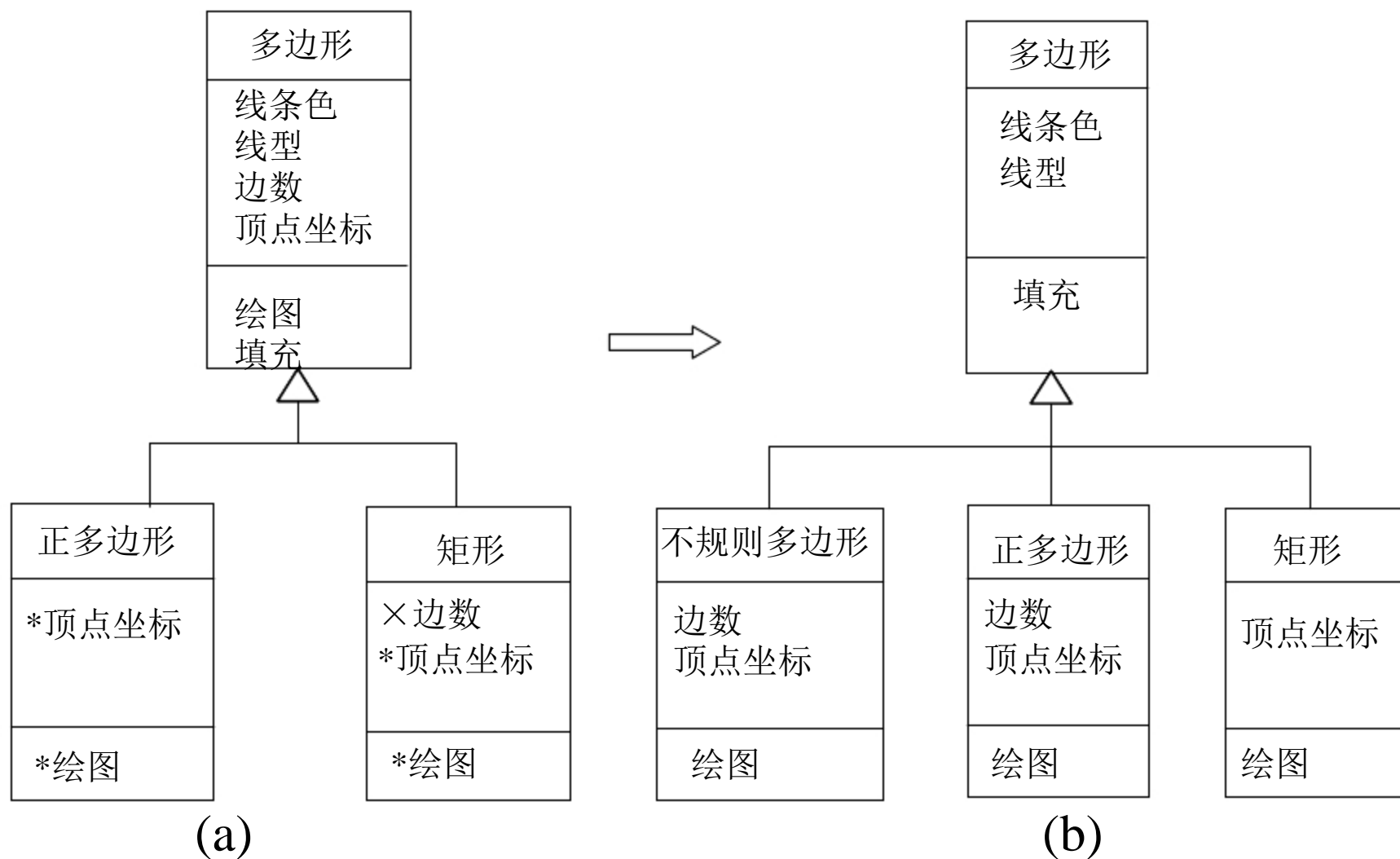
方法3：保持分类，剥离多继承信息



不适当的方法增加程序代码



(2) 取消多态性



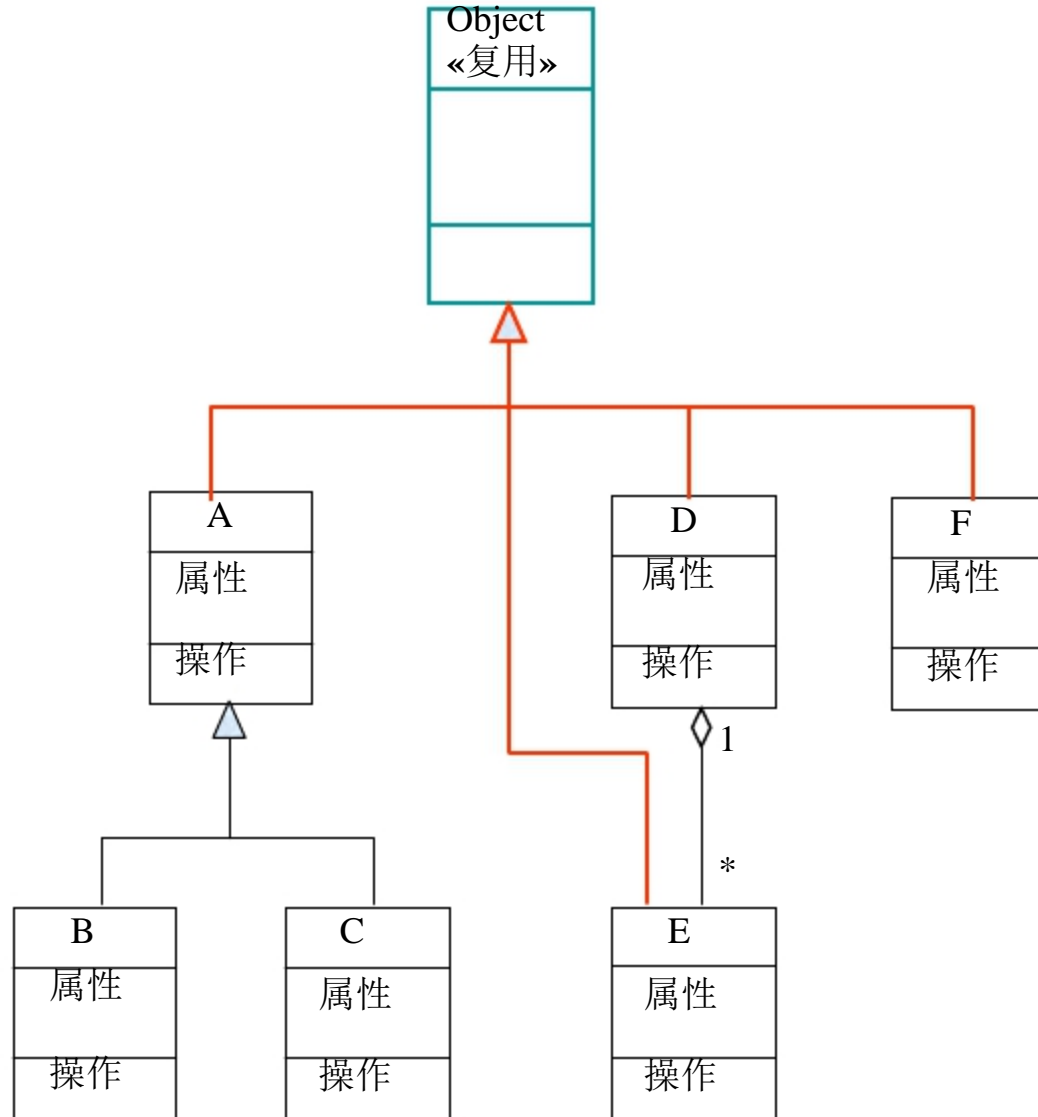
2、增加一般类以建立共同协议

增加根类： 将所有的类组织在一起
提供全系统通用的协议

例： 提供创建、删除、复制等操作

增加其他一般类： 提供局部通用的协议
例： 提供永久存储及恢复功能

例:



3、实现复用的设计策略

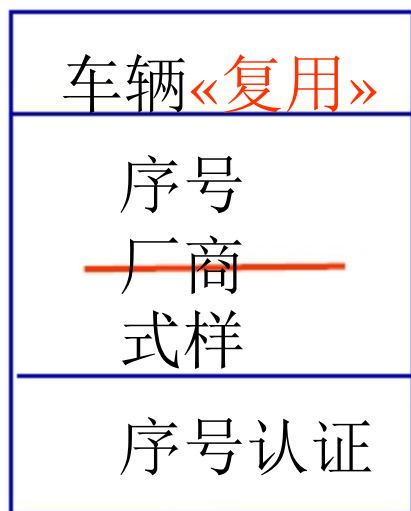
如果已存在一些可复用的类，而且这些类既有分析、设计时的定义，又有源程序，那么，复用这些类即可提高开发效率与质量。

目标：尽可能使复用成分增多，新开发的成分减少

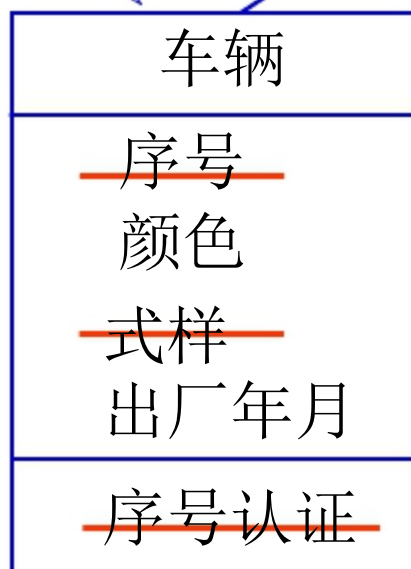
可复用类定义的信息	比	当前所需的类的信息	=	直接复用
			<	通过继承复用
			>	删除可复用类的多余信息
			≈	删除多余信息，通过继承而复用

例：

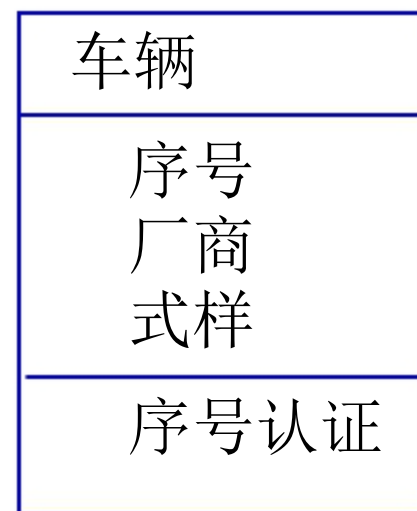
可复用的类



问题域部分的类

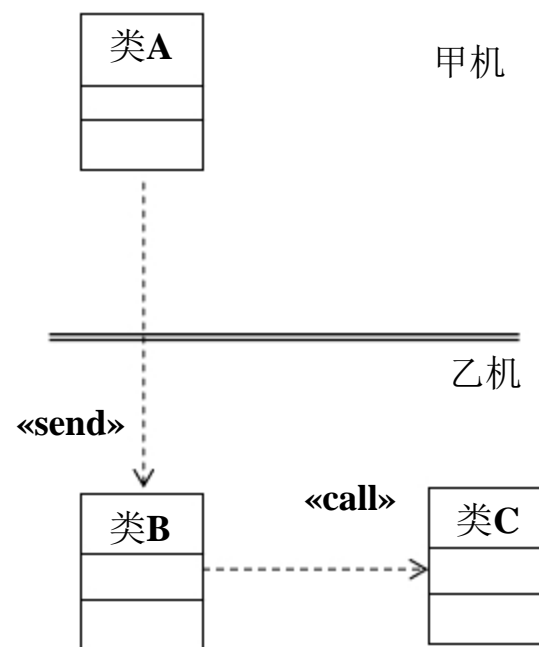
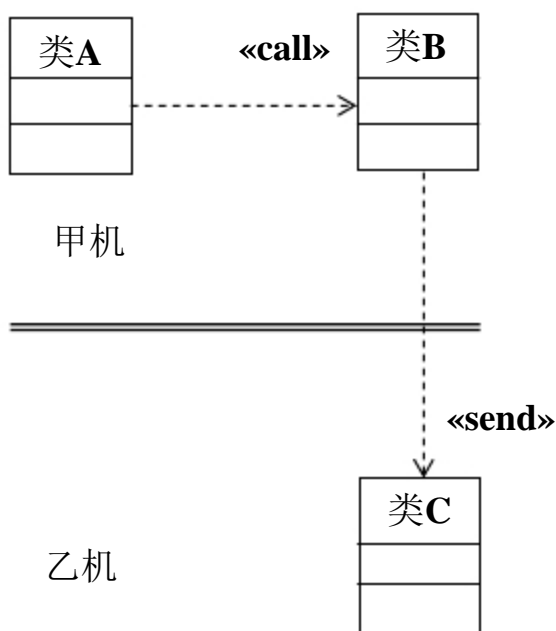


可复用的类



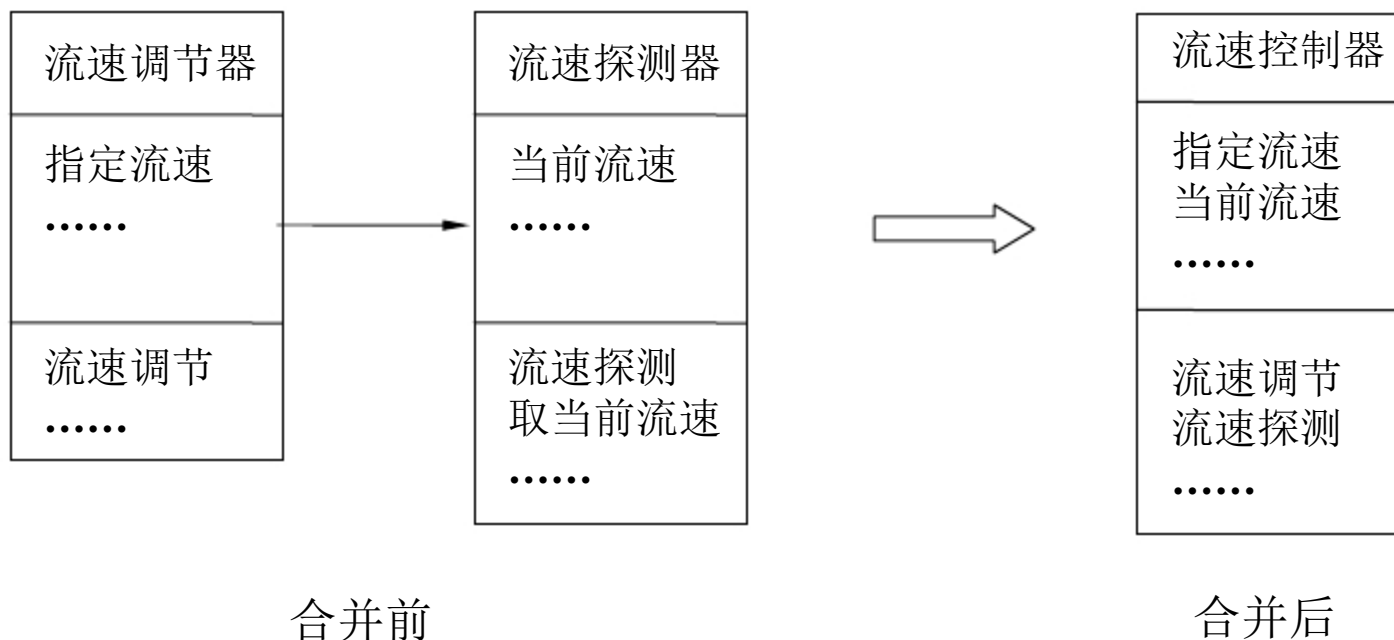
4、提高性能

(1) 调整对象分布



(2) 缩短对象存取时间 设立缓冲区

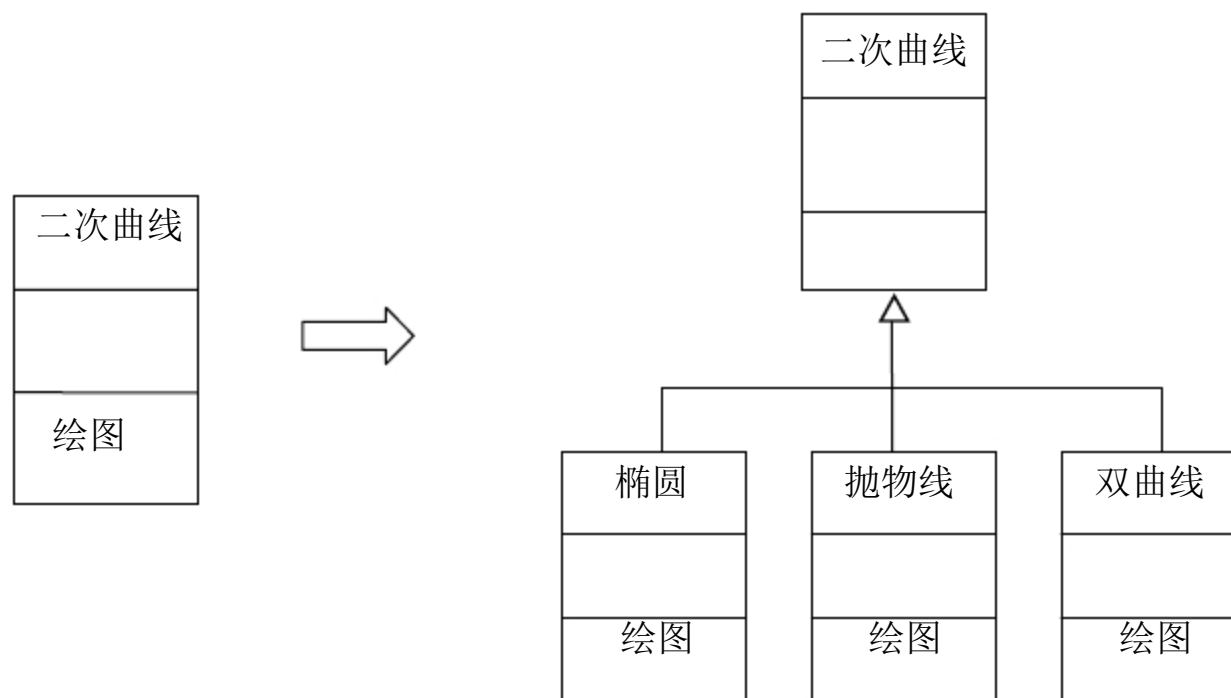
(3) 合并通讯频繁的类



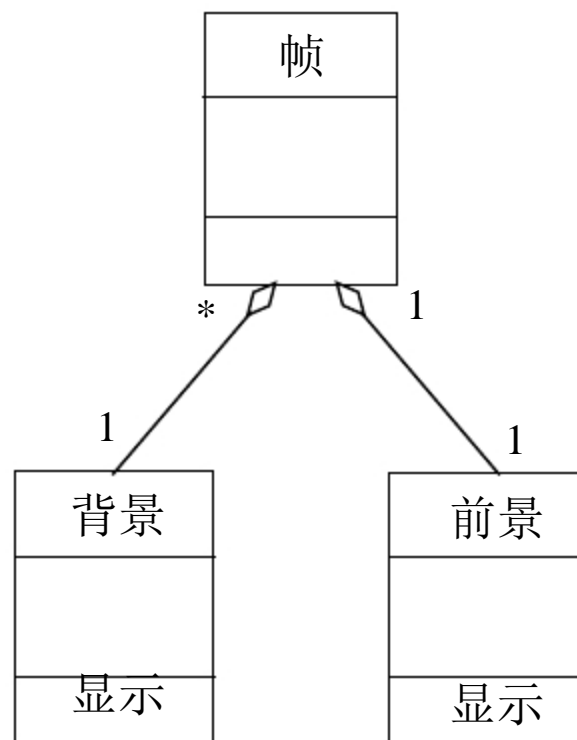
(4) 增加属性以减少重复计算

(5) 降低算法的计算复杂性

(6) 细化对象的分类



(7) 将复杂对象化为整体-部分结构



5、为数据存储管理增补属性与操作
在数据接口部分设计中介绍

6、完善对象的细节

OOD在**OOA**模型基础上所做的主要工作，不能用“细化”二字概括，但细化是不可缺少的

- (1) 完善与问题域有关的属性和操作
在**OOA**阶段允许不详尽，**OOD**必须加以完善
- (2) 解决**OOA**阶段推迟考虑的问题，包括：
因封装原则而设立的对象操作
与**OOD**模型其他部分有关的属性和操作
- (3) 设计类的每个操作
必要时用流程图或者活动图表示
- (4) 设计表示关联的属性
区分多重性的3种情况，决定属性设置在哪一端
- (5) 设计表示聚合的属性
区分组合与松散的聚合
对于组合，用嵌套对象实现
对于松散的聚合，采用与关联相同的策略

7、定义对象实例

在逻辑上，一个类的对象实例是：

问题域中所有可用这个类描述的实际事物

在物理上，一个类的对象实例可以是：

内存中的对象变量

文件的一个记录，或数据库表的一个元组

一个类的对象实例可以分布到不同的处理机上

对每一台处理机

说明在它之上创建的每一个（或组）内存对象

说明在它之上保存的外存对象

类的对象实例说明：

```
{  
  处理机：<结点名>{,<结点名>}  
  内存对象：{<名称>[（n元数组）][<文字描述>]}  
  外存对象：{<名称>[<文字描述>]}  
}
```

8、修改或补充辅助模型及模型规约

包图

类的增减、拆分、合并以及各个类之间关系的变化

顺序图

操作与消息

活动图

操作流程

其他模型图

状态机图、定时图、交互概览图、组合结构图

模型规约

类的属性、操作及其对外关系的修改或细化

建立与OOA文档的映射

OOA 类与 OOD 类映射表

指出**OOA**模型中的哪个（或哪些）类演化为**OOD**模型中的哪个（或哪些）类

映射方式	OOA 类	OOD 类
1 = 1		
1 to 1		
1 to m		
m to 1		
m to m		
0 to 1		

图 3.13 OOA 类与 OOD 类的映射表