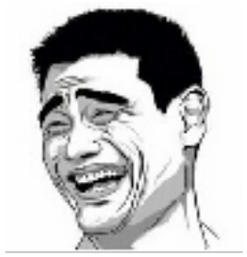
结构模式

Structural Patterns

概述

- 结构性模式主要关注如何构造复杂结构
  - 提升结构的灵活性

案例分析



小明：我正在写一个XML编辑器，并在GitHub上找到了一种新型的编辑控件，看起来很炫！但该编辑控件只能支持显示和操作符合EMF API规范的对象，我该怎么办?

Bob：Hi，我想最简单的办法是把那个控件的源码拿来，然后把它移植到XML API上。据我所知EMF的API和XML的API比较接近。

案例分析

EMF控件

setContent(o:EObject)
refresh()
updateChildren(o:EObject)
…

this.content = o;
this.refresh();

List<EObject> c = o.eContent();
…
for(EObject e in c) {
 …
  updateChildren(e);
}

XML控件

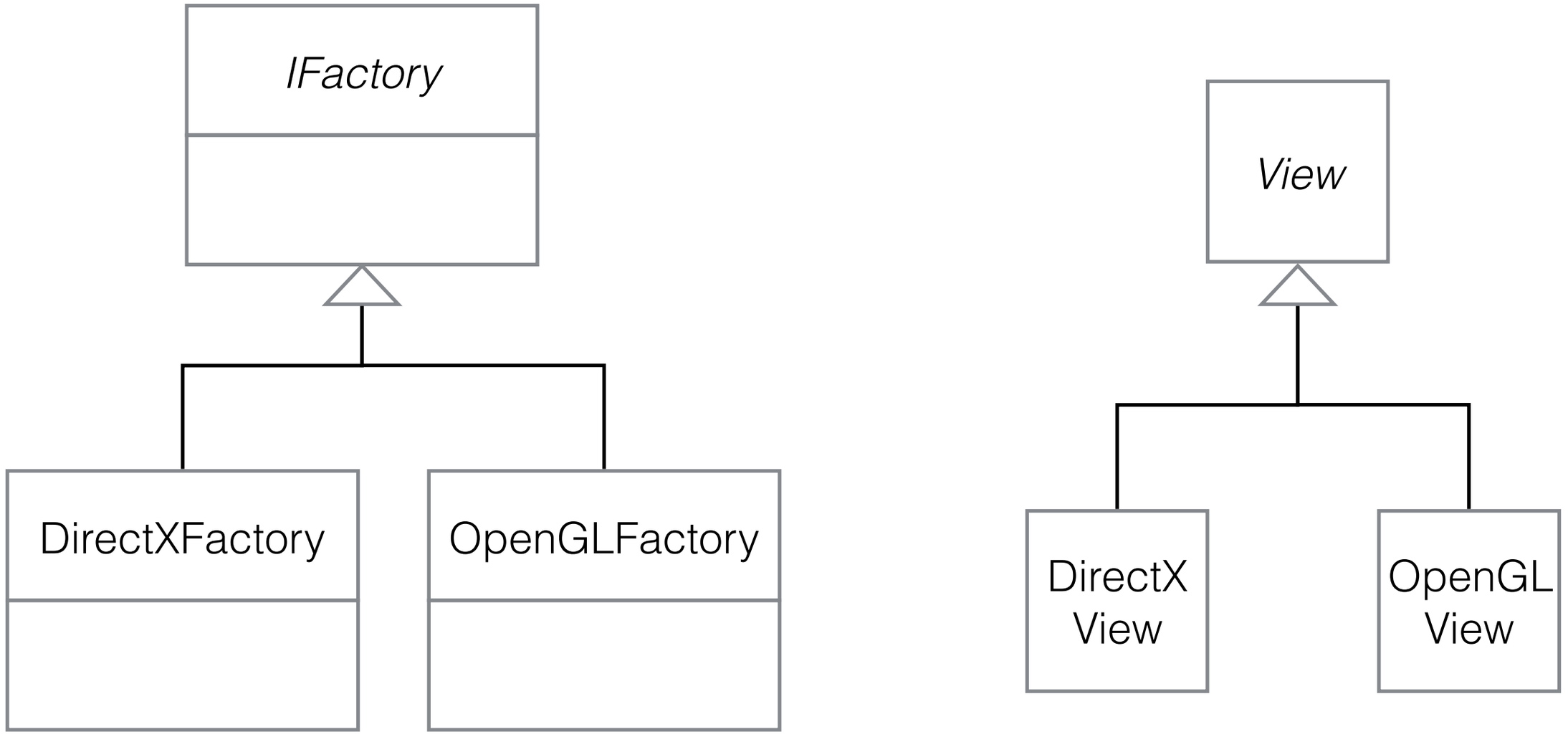
setContent(o:XMLElement)
refresh()
updateChildren(o:XMLElement)
…

案例分析

Bob：我现在在开发一款3D游戏，需要支持OpenGL和DirectX两种接口。假设该游戏使用View表示一个可见的视图，请问如何设计才能支持两种接口?
随着开发的进行，有需要设计FloatView来实现浮动视图，请问该如何进行设计?

银星：上课不是讲过吗，使用抽象工厂模式啊。为OpenGL和DirectX分别写两个工厂，然后创建各自的对象！完美～

案例分析


IFactory
DirectXFactory
OpenGLFactory
View
DirectX
View
OpenGL
View

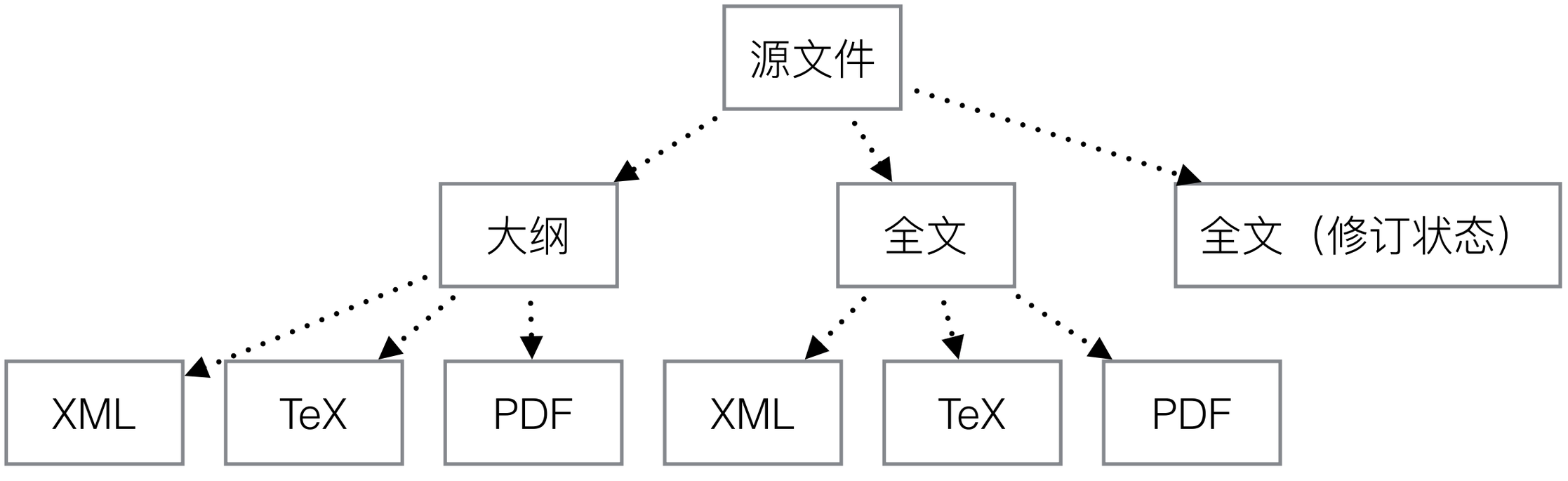
案例分析

银星：老板让我写一个树型控件，要求能够将给定的数据按照树的形式展示出来。考虑到树形控件可能展示不同类型的数据结构（例如目录结构、XML文档等），我该如何设计一个～完美～的控件接口？

月月：设计一个抽象的树形控件类，里面实现共有功能。然后对每种数据结构实现一个具体子类。这样无论有什么样的结构都能够支持！我的天呐！

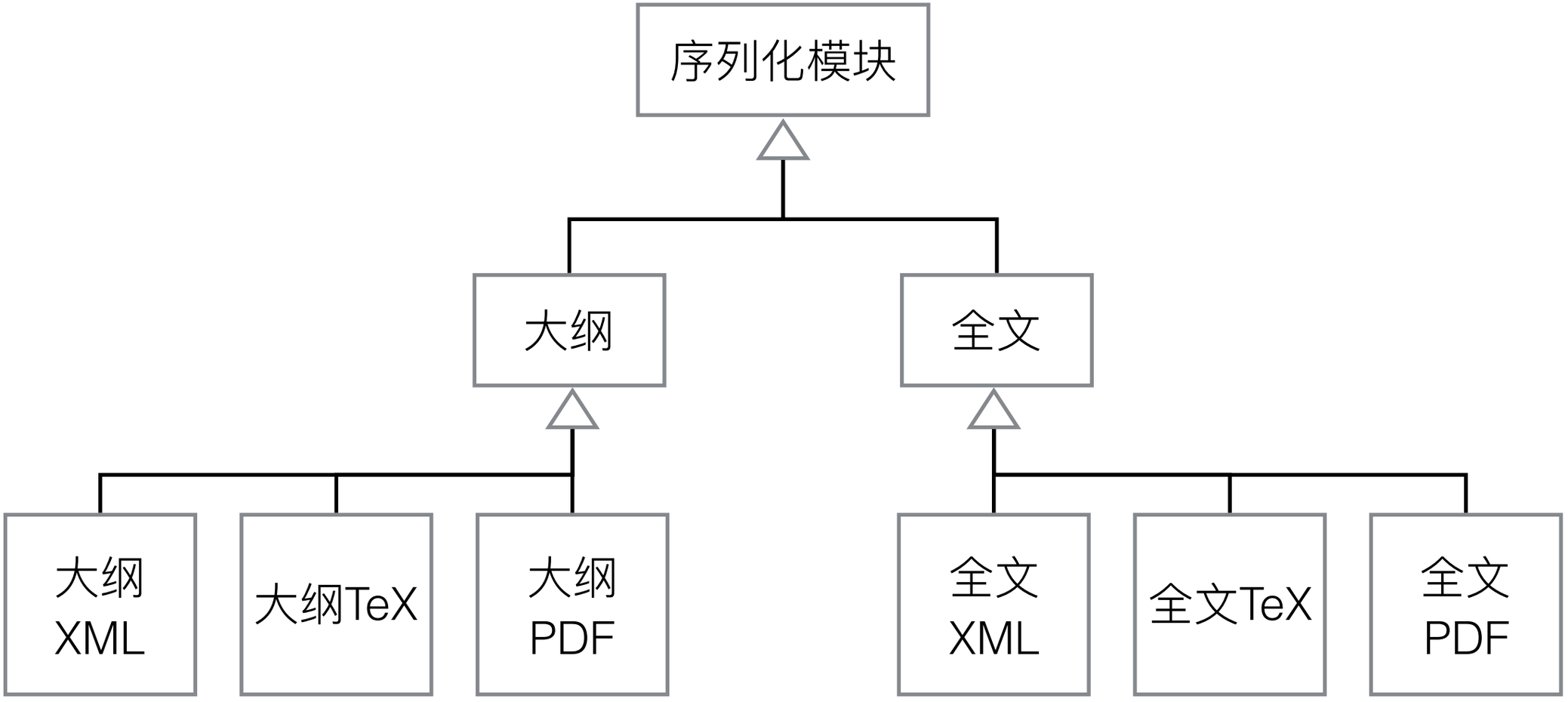
案例分析

月月：一款论文编辑工具能够生成论文的大纲、全文、带修订状态的全文等不同内容的文件，同时可以按照不同格式（XML、TeX、PDF）等格式进行存储。请问为了保持文件内容和存储格式的灵活性，应当如何设计序列化构件?



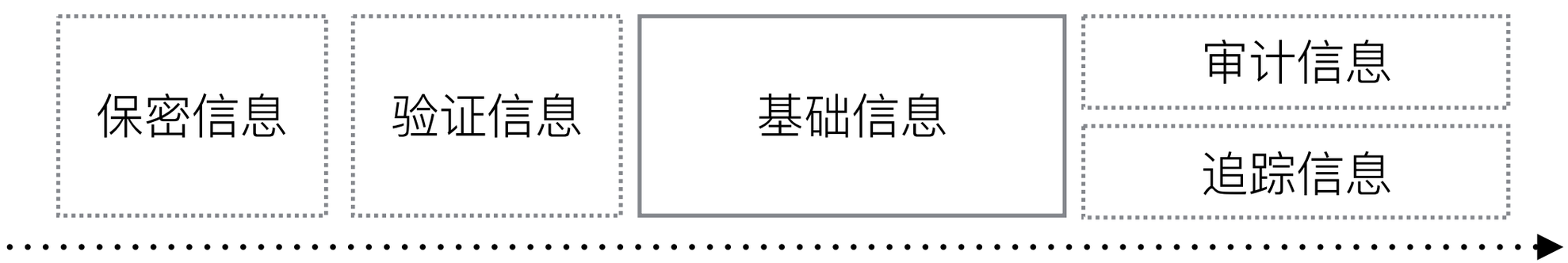
案例分析

三胖：一个功能一个类的思密达



案例分析

三胖：一个负责网络通讯的构件，其主要功能是发送信息到指定地址。由于不同对接系统的要求，除基础信息之外，还需要附加若干信息。与此同时，出于审计、追踪的要求，发送信息时还可能产生一些记录信息，或者审计记录。如果所有附加功能都是可选（可任意组合），如何设计这样的构件思密达?



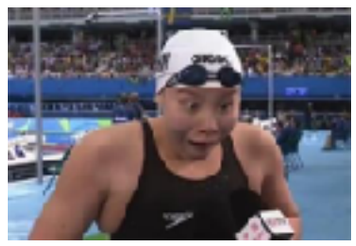
案例分析

Barack：就像上次课说的文件选择对话框一样，你可以用一个Builder去构造你需要的通信构件对象，OK？当然你也可以使用一个复杂的带参数的构造函数，让用户选择通信构件应该具备的功能。

案例分析

Barack：一款在线3D游戏中，可能需要从服务器上实时下载某些人物形象。考虑到网络延迟可能会影响游戏的流畅性，请问如何设计才能兼顾流畅性与画面质量?

圆圆：提前下载不就好了吗



案例分析

圆圆：一款RPG游戏中，人物绘制构件的基本功能是绘制人物形象。当人物使出洪荒之力后，会在人物周边绘制金色光芒；而人物中毒后，则会在人物形象外添加一层具有一定透明度的绿色图层。请问如何设计该构件，才能最大程度地提高灵活性?

Jack：用一个变量控制绘制特效。

案例分析

Jack：一款支持图文编辑的文件编辑器，允许用户读入并修改图片。考虑到一张图片可能在同一个文档中出现多次，但每次出现都可能进行不同的修改。由于图片会占用较多的处理资源，如何设计才能降低图片开销?

宝宝：我觉得最纠结的是，为了减少同一图片出现多次的开销，我似乎应该保证图片数据在程序中只有一个实例，但为了支持每次出现都可以进行不同的修改，似乎有需要保持多个副本

案例分析

一款分布式系统，一个构件可能调用本地函数也可能调用远程函数。由于本地调用和远程调用的实现过程各不相同，请问如何设计才能降低这两种调用之间的接口差异?

案例分析

一款RPG游戏中，地图数据由若干等大小的位图块组成。如何设计地图部分的结构?

案例分析

地图软件中，允许用户对地图进行缩放，并根据缩放等级载入相应的资源。请问如何设计地图结构—支持上述两种功能?

案例分析

一款在线支付软件提供多种付款模式，包括银行卡支付，信用卡支付，余额支付，白条支付，礼品卡支付，虚拟币支付，分享支付（AA）。用户可以使用单一或者组合其中的几种。请问如何设计支付方式的结构?

案例分析

在线购物系统分成多个子系统，包括：商品推荐子系统、商品信息查询子系统、商品订购子系统，支付子系统，商品搜索子系统等。每个子系统委托给一个独立开发团队实现。由于每个开发团队使用不同的技术（Java、C#等），基于不同的操作系统（Linux、Windows等），遵循不同的编程规范。请问如何设计才能处理这些异构性?

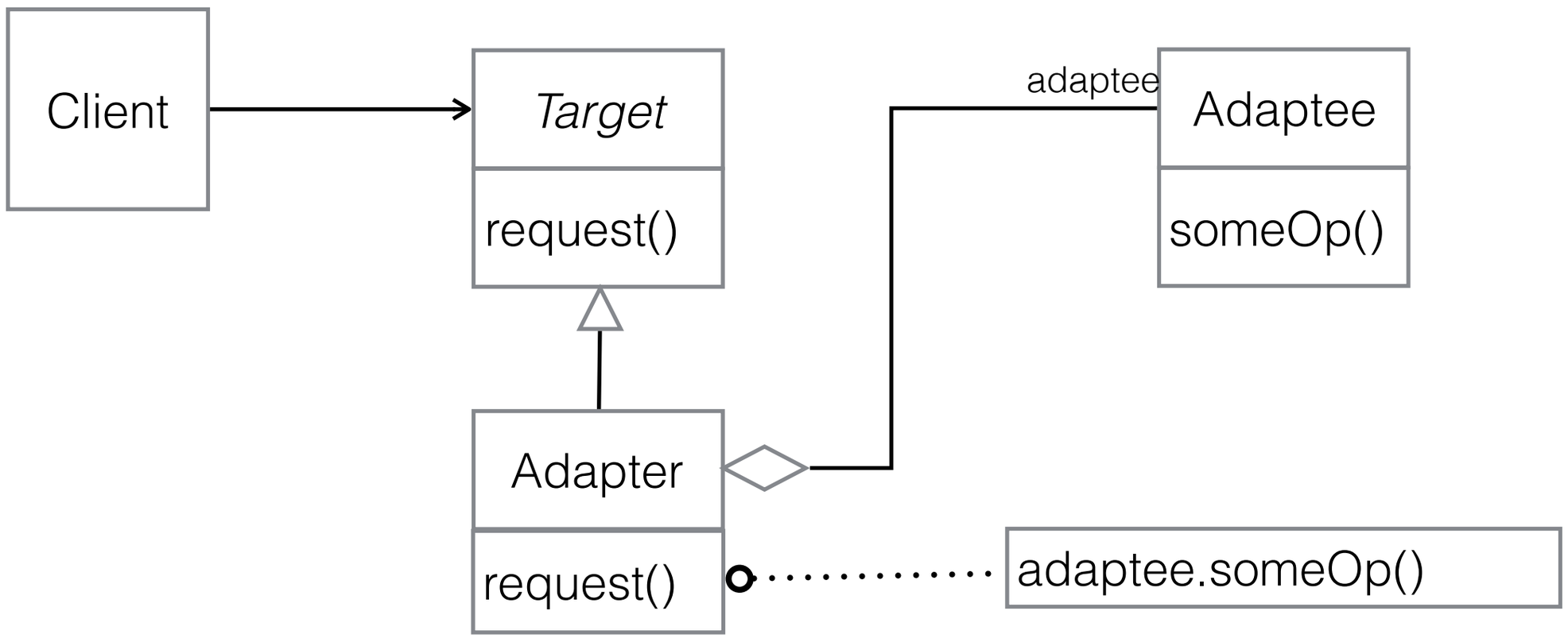
案例分析

一个子系统的核心接口中包含多个操作，可以给具有不同权限的用户使用。如何设计以便支持不同的访问控制策略?

结构性模式

适配器模式
(Adapter)

- 意图:
  - 将一个接口转换成另一个接口，使得由于接口不一致的两个类可以互操作



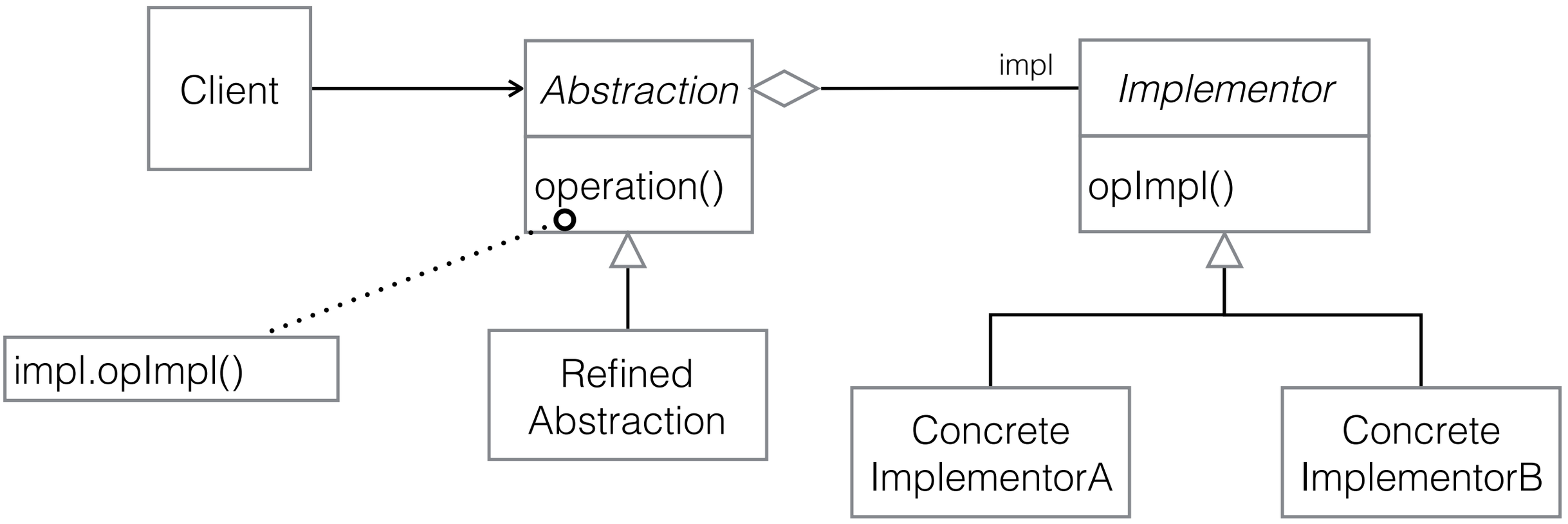
适配器模式
(Adapter)

- java.io.InputStreamReader(InputStream) (returns a Reader)

- java.io.OutputStreamWriter(OutputStream) (returns a Writer)

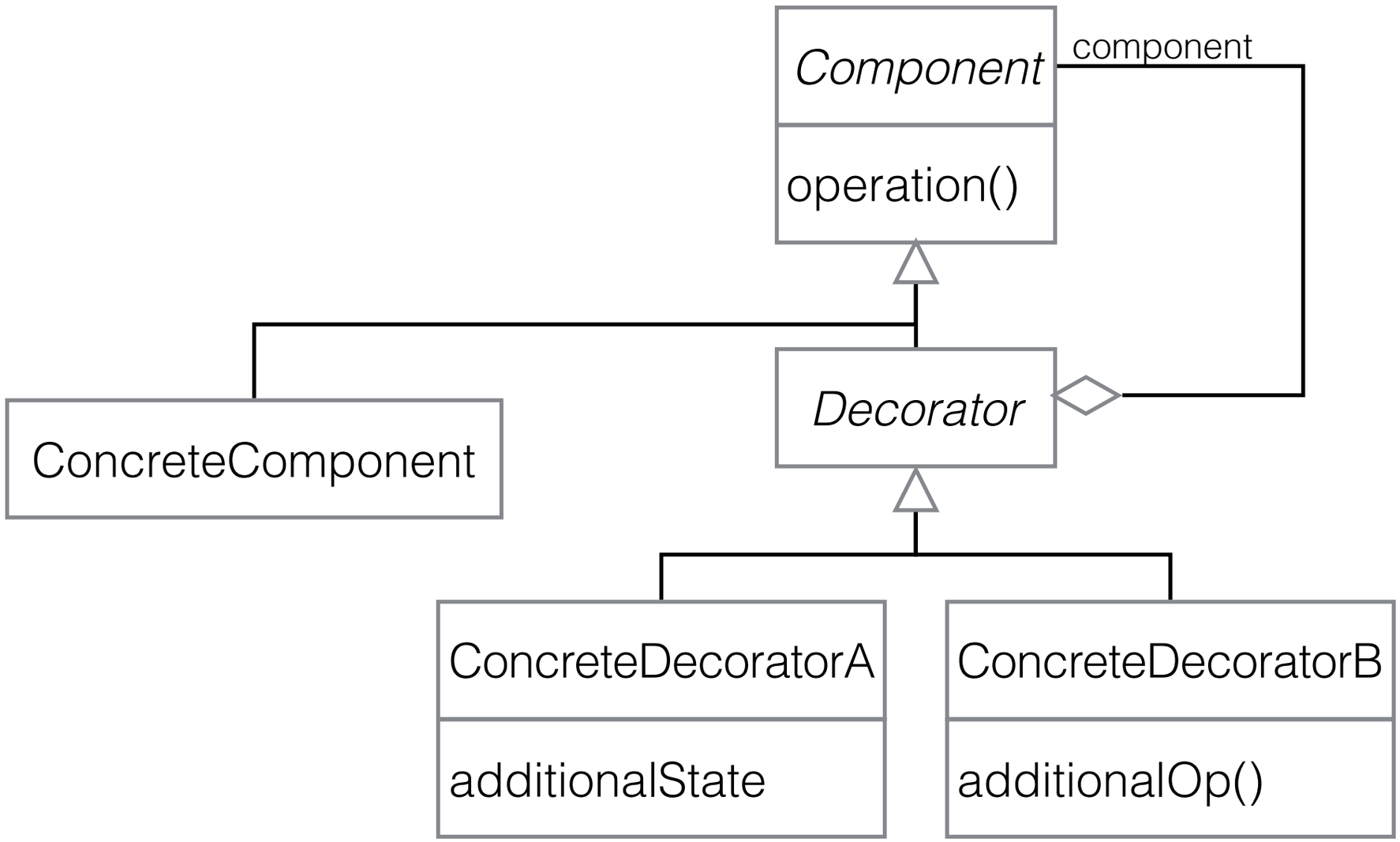
桥接模式
(Bridge)

- 意图:
  - 将一个抽象概念及其实现方式分离，使它们可以分别演化



装饰者模式
Decorator

- 意图
  - 为一个对象动态增加额外的功能

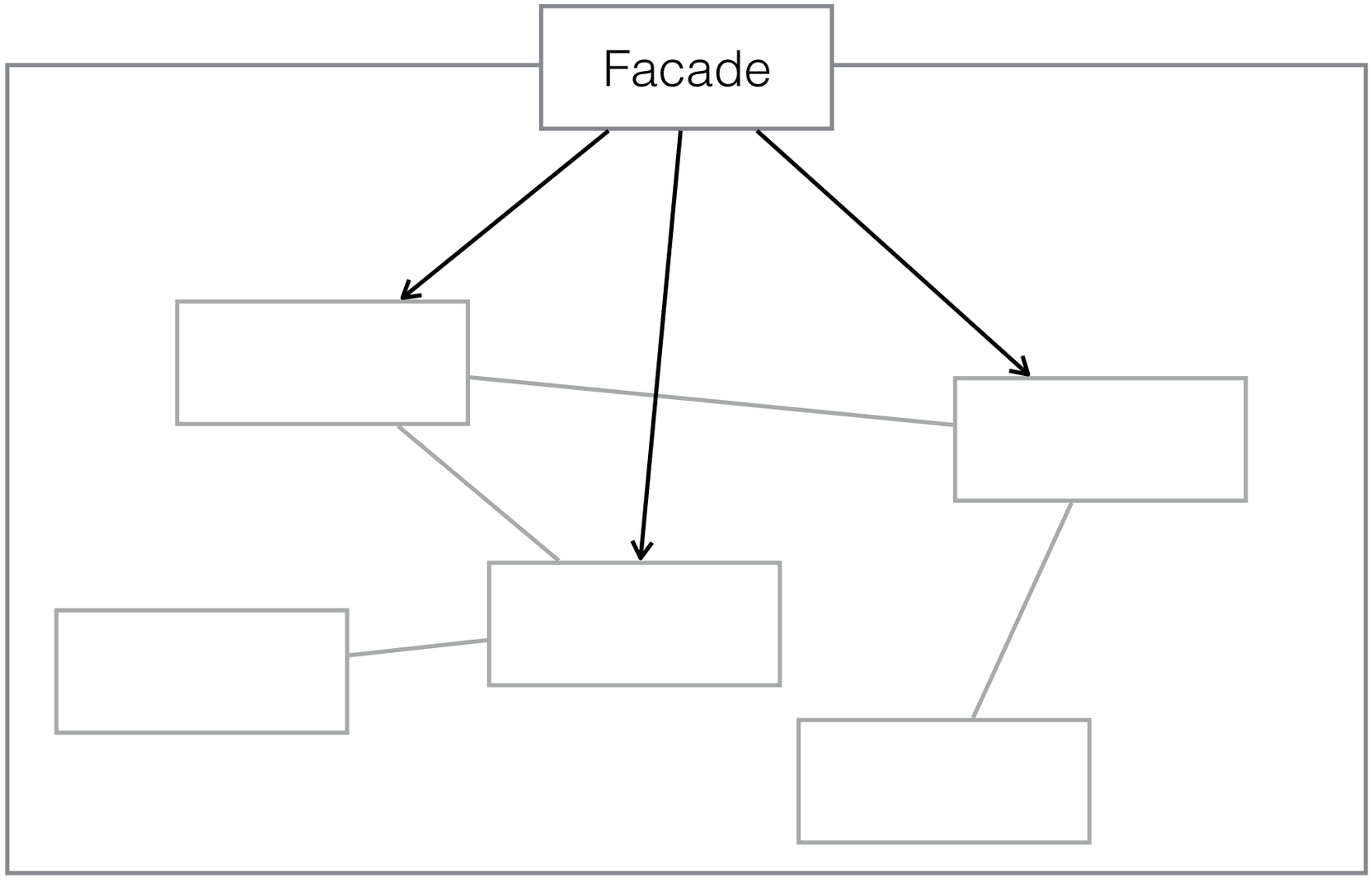


装饰者模式
Decorator

- All subclasses of java.io.InputStream, OutputStream, Reader and Writer have a constructor taking an instance of same type.

外观模式
Facade

- 意图
  - 为子系统中的一组接口提供一个统一的对外接口，方便调用该子系统

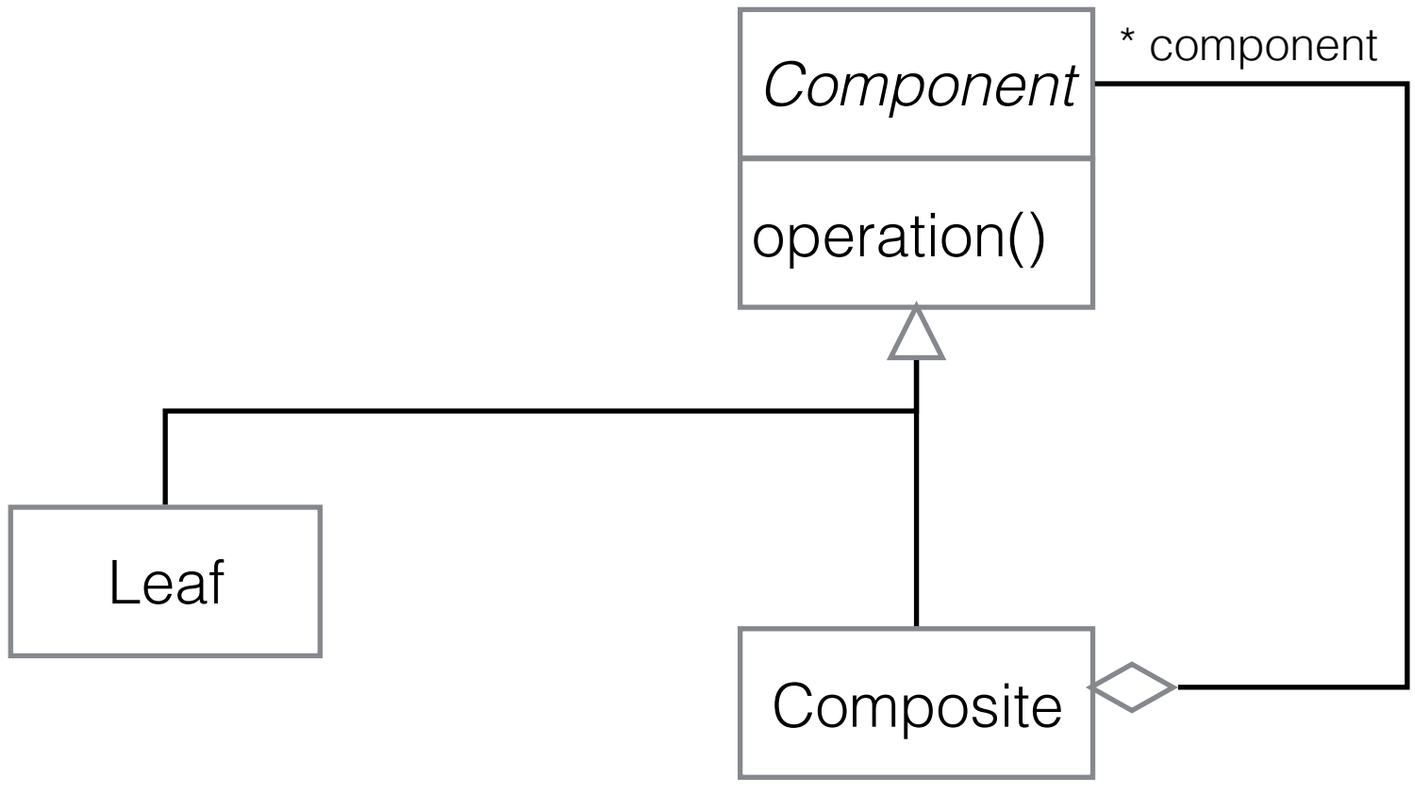


外观模式
Facade

- javax.faces.context.ExternalContext,
- which internally uses ServletContext, HttpSession, HttpServletRequest, HttpServletResponse, etc.

组合模式
Composite

- 意图
  - 将对象组织成树形结构，以表示整体部分关系；组合对象和简单对象具有同样的接口



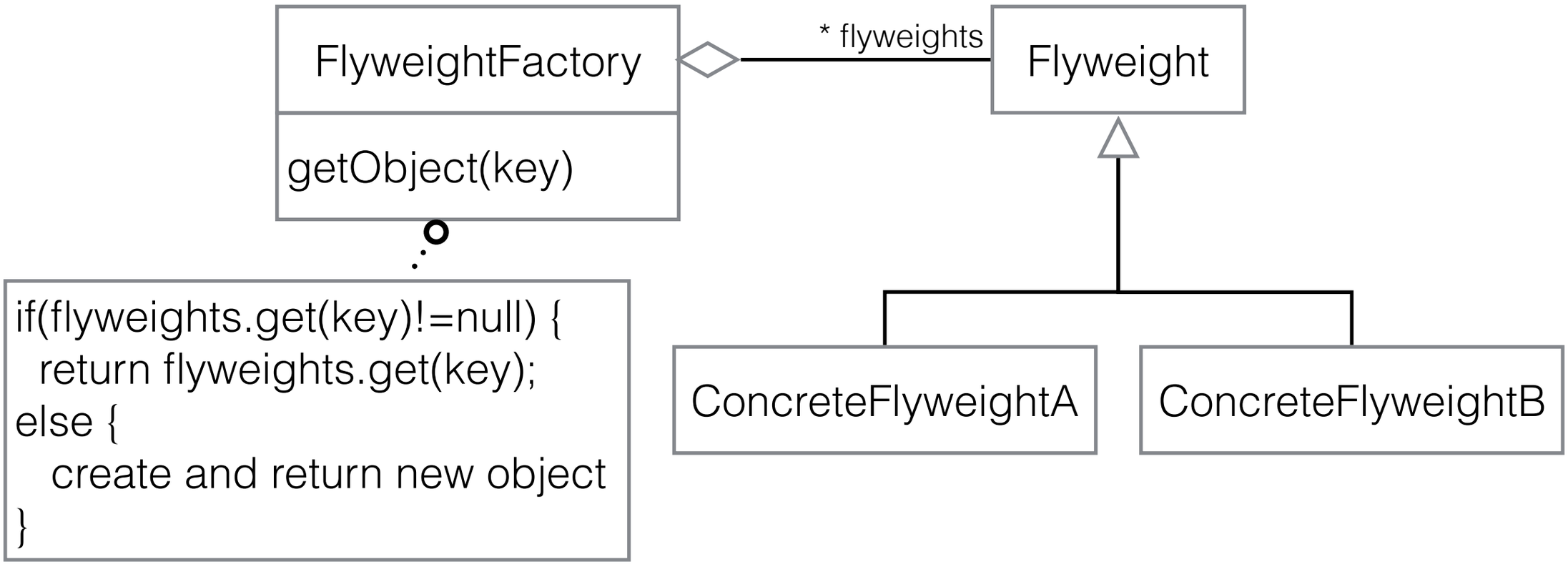
组合模式
Composite

- java.awt.Container#add(Component) (practically all over Swing thus)

- javax.faces.component.UIComponent#getChildren() (practically all over JSF UI thus)

享元模式
Flyweight

- 意图
  - 通过共享的方式来支持大量零碎对象

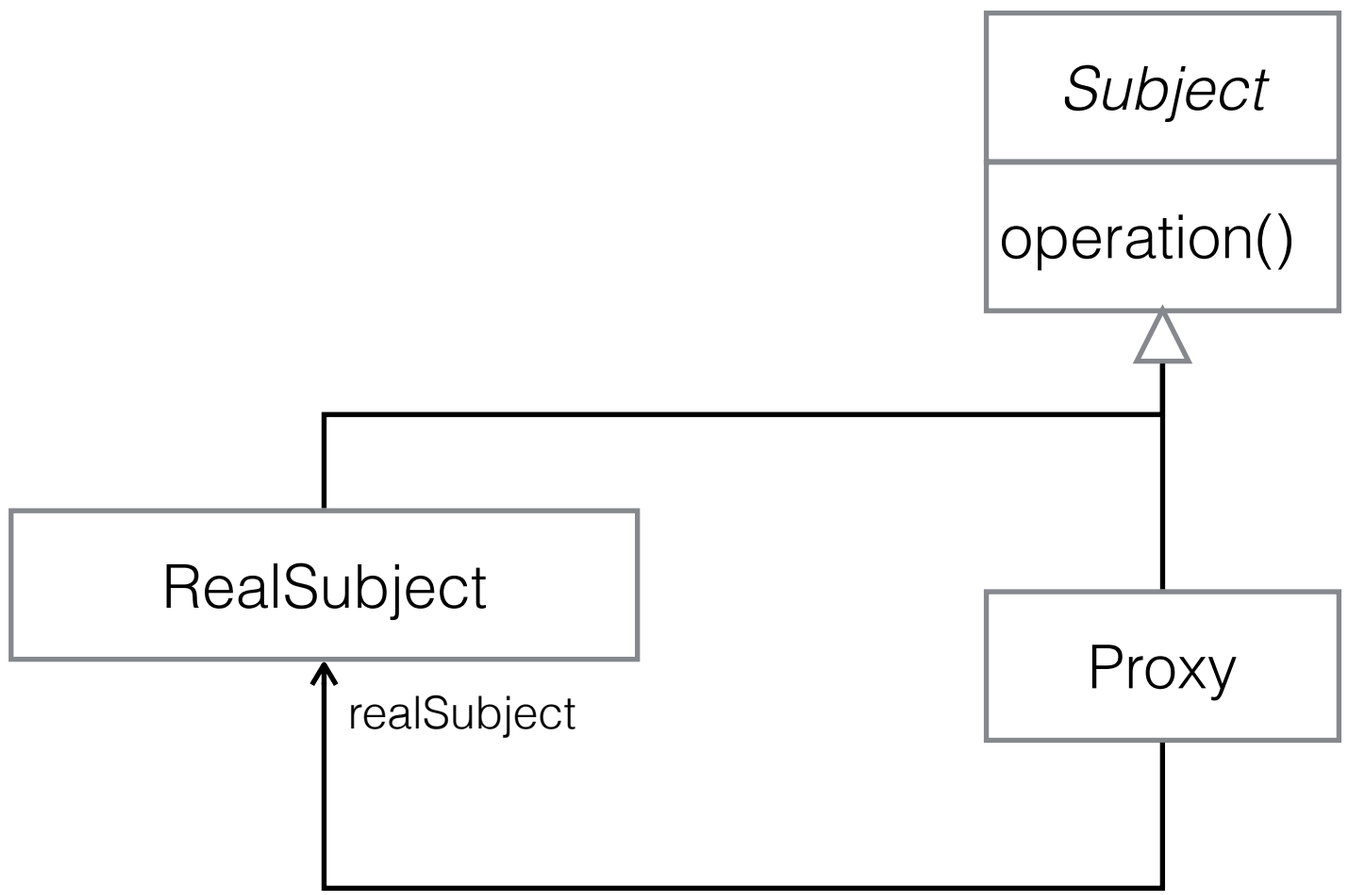


享元模式
Flyweight

- java.lang.Integer#valueOf(int) (also on Boolean, Byte, Character, Short, Long and BigDecimal)

代理模式
Proxy

- 意图
  - 为一个对象动态增加额外的功能



代理模式
Proxy

- java.lang.reflect.Proxy