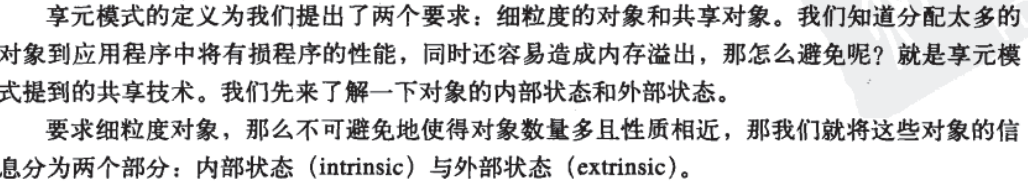
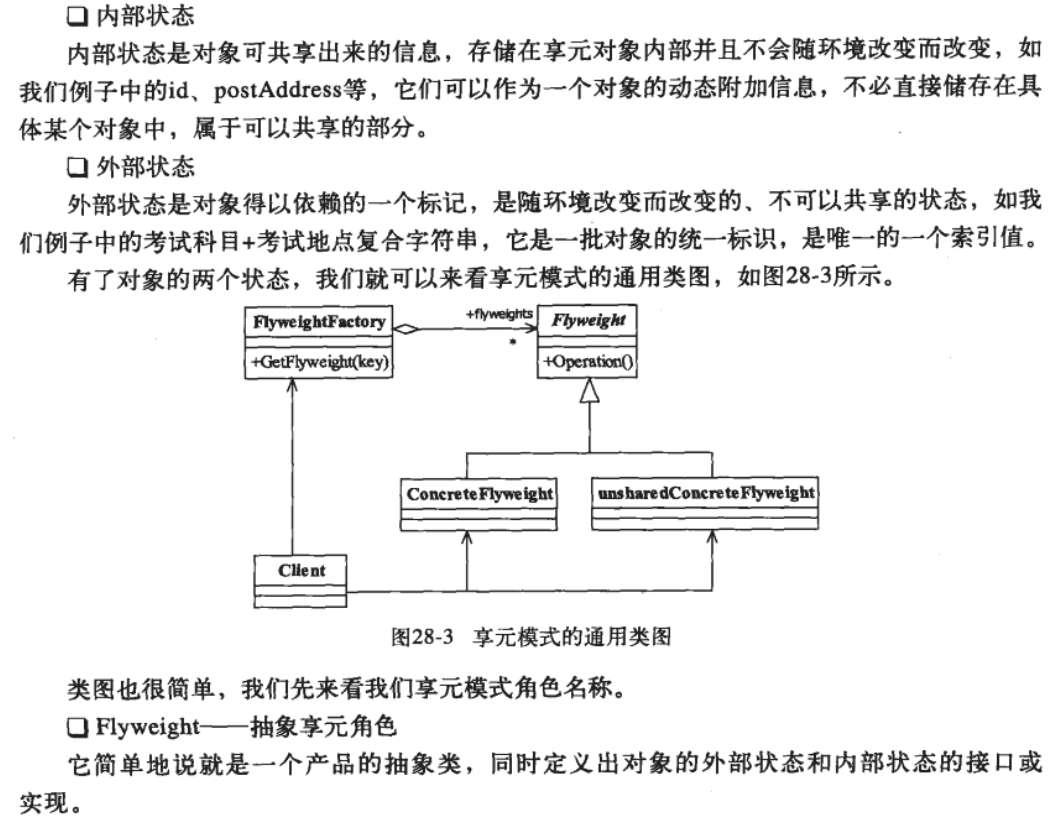
# 享元模式

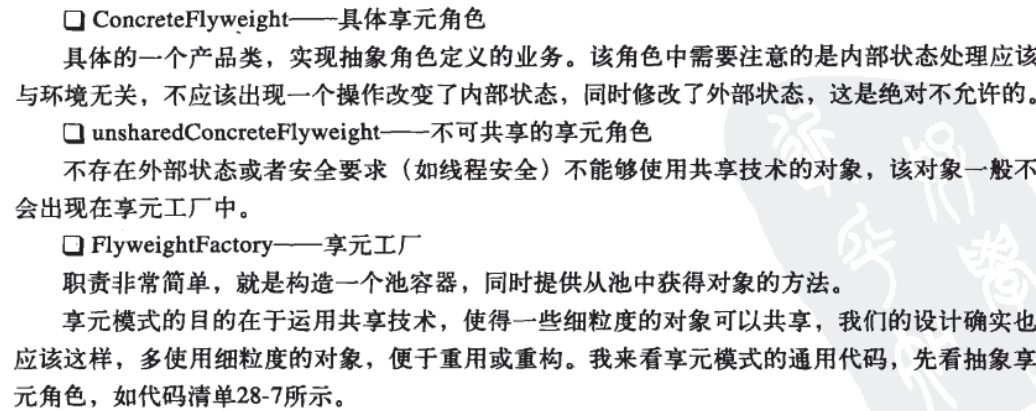
1. 定义

Use sharing to support large of numbers of fine-grained objects efficiently(使用共享对象可以有效地支付大量的细粒度的对象)

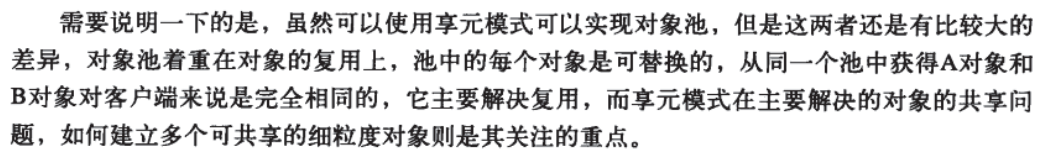
1. 类图







1. 模式的应用
2. 模式的优点
3. 减少了对象的创建；
4. 节省了系统资源；
5. 增加了程序性能
6. 模式的缺点
7. 提高了系统的复杂性；
8. 外部状态具有固化特性，不应该随着内部状态改变而改变
9. 实际应用场景
10. 系统中存在大量相似对象；
11. 细粒度对象都具备相似的外部状态，内部状态与环境无关，也就是说对象没有特定身份，是哪一个都可以；
12. 需要缓冲池的场景。
13. 最佳实践



1. 注意事项

