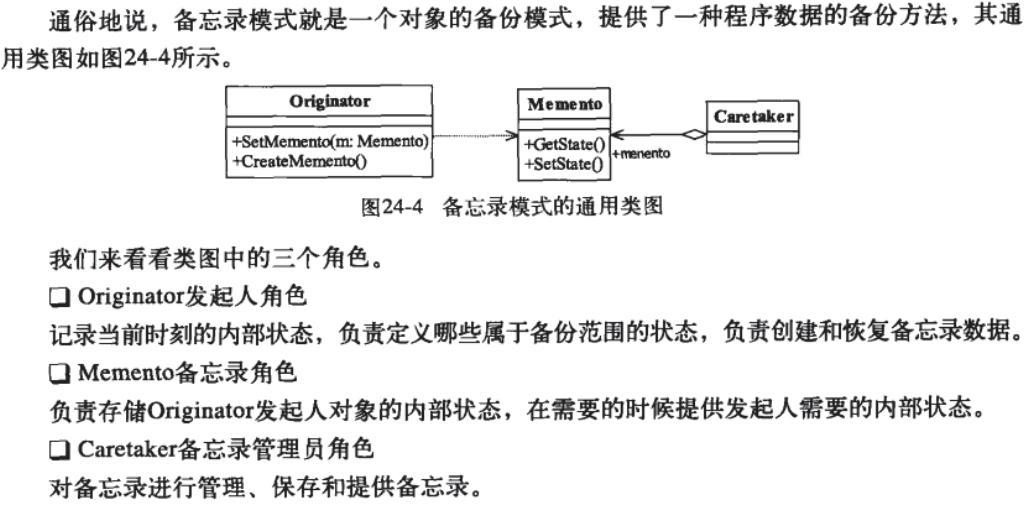
# 备忘录模式

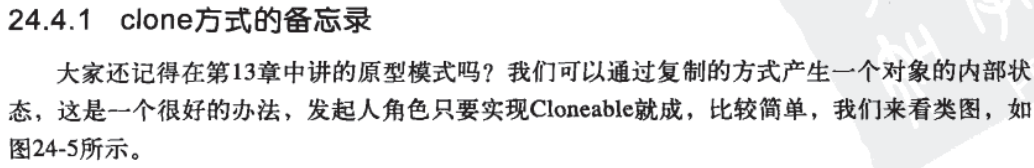
1. 定义

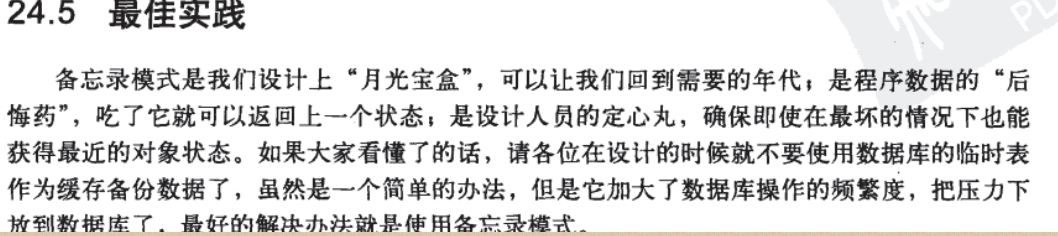
Without violating encapsulation, capture and externalize an object’s internal state so that the object can be restored to this state later.(在不破坏封装性的前提下，捕获一个对象的内部状态，并在该对象之外保存这个状态。这样以后可以将该对象恢复到原先保存的状态)

1. 类图



1. 模式的应用
2. 模式的优点
3. 算法可以自由切换：只要实现抽象策略，成为策略家族的一个成员，通过封装角色进行封装，保证对外提供“可自由切换”的策略。
4. 避免使用多重条件判断。
5. 扩展性良好
6. 模式的缺点
7. 策略类数量增多：每一个策略是一个类，复用性很小，类数量多；
8. 所有策略类都需要对外暴露：上层模块必须知道哪些策略，然后才能决定使用哪一个策略，这与迪米特法则（最小知道原则）是相违背的。可以通过其他模式来修正：工厂、代理、享元。
9. 实际应用场景
10. 需要保存和恢复状态的相关场景；
11. 提供一个可回滚rollback的操作；Word中Ctrl+Z组合键，IE浏览器中的后退按钮，文件管理器上的backspace键；
12. 需要监控的副本场景：例如要监控一个对象的属性，但监控不应该作为系统的主要业务来调用，它只是边缘应用，即使监控不佳、报错告警也影响不大。
13. 数据库连接的事务管理就是用的备忘录模式。
14. 最佳实践





1. 注意事项

