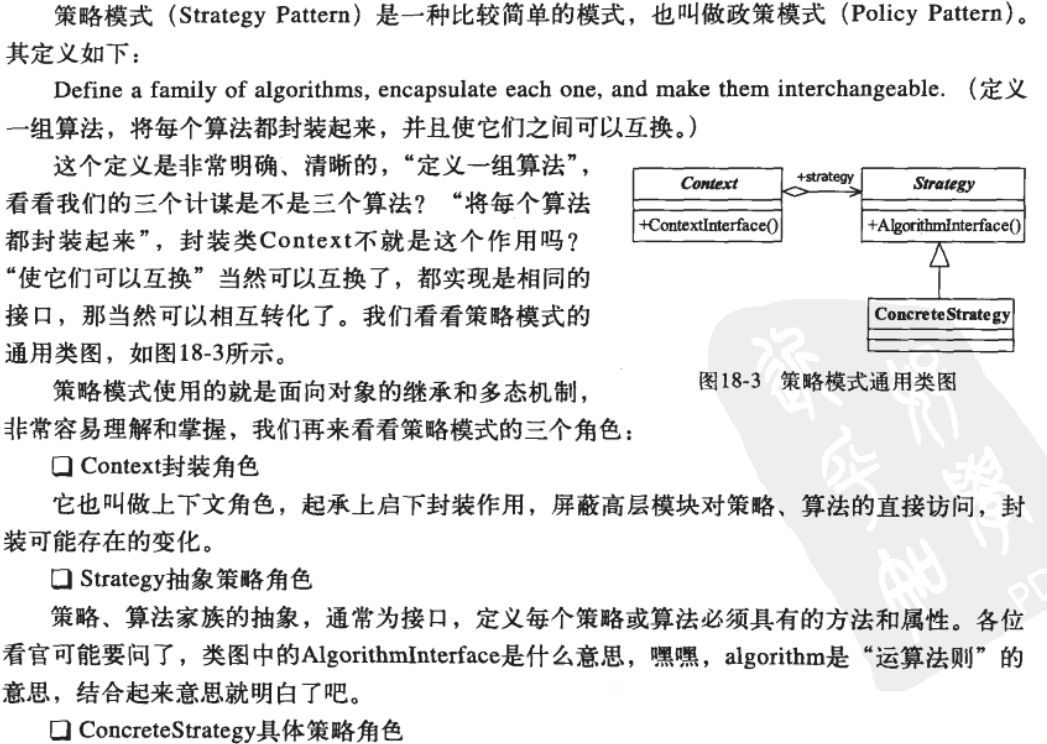
# 策略模式

1. 定义

Define a family of algorithms,encapsulate each one, and make them interchangeable.(定义一组算法，把每个算法封装起来，并且使他们之间可以互换)

1. 类图

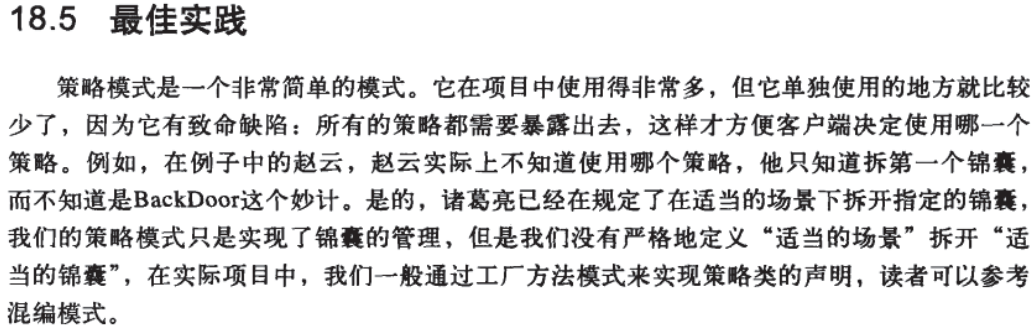




1. 模式的应用
2. 模式的优点
3. 算法可以自由切换：只要实现抽象策略，成为策略家族的一个成员，通过封装角色进行封装，保证对外提供“可自由切换”的策略。
4. 避免使用多重条件判断。
5. 扩展性良好
6. 模式的缺点
7. 策略类数量增多：每一个策略是一个类，复用性很小，类数量多；
8. 所有策略类都需要对外暴露：上层模块必须知道哪些策略，然后才能决定使用哪一个策略，这与迪米特法则（最小知道原则）是相违背的。可以通过其他模式来修正：工厂、代理、享元。
9. 实际应用场景

策略模式又称为：政策模式。

1. 多个类只有在算法和行为上稍有不同的场景；
2. 算法需要自由切换场景：算法是由使用者决定或者算法在进化；
3. 需要屏蔽算法的场景：通过传递一个参数，然后反馈一个结果。
4. 最佳实践



1. 注意事项

