**命令模式**

# 1. 定义

Encapsulate a request as an object, thereby letting you parameterize clients with different requests, queue or log requests, and support undoable operations.

（将一个请求封装成一个对象，从而让你使用不同的请求把客户端参数化，对请求排队或者记录请求日志，可以提供命令的撤销和恢复功能。）

解析：

将一个请求封装为一个对象，使发出请求的责任和执行请求的责任分割开。这样两者之间通过命令对象进行沟通，这样方便将命令对象进行储存、传递、调用、增加与管理。

# 2. 优缺点

1. 降低系统的耦合度。命令模式能将调用操作的对象与实现该操作的对象解耦。
2. 增加或删除命令非常方便。采用命令模式增加与删除命令不会影响其他类，它满足“开闭原则”，对扩展比较灵活。
3. 可以实现宏命令。命令模式可以与组合模式结合，将多个命令装配成一个组合命令，即宏命令。
4. 方便实现 Undo 和 Redo 操作。命令模式可以与后面介绍的备忘录模式结合，实现命令的撤销与恢复。

缺点：

可能产生大量具体命令类。因为针对每一个具体操作都需要设计一个具体命令类，这将增加系统的复杂性。

参考资料：

1. 《设计模式之禅》

2. <https://www.dofactory.com/net/command-design-pattern>

3. <http://c.biancheng.net/view/1380.html>