# 设计模式

一、单例模式

饿汉：类加载时初始化实例使用private static声明

懒汉：第一次调用时实例化，使用volatile + 同步双重校验构造唯一实例

二、模板模式

父类定义操作流程，子类按照模板实现不同的逻辑。

三、代理模式

静态代理：代理类持有委托类（确定类型），两者具有相同的行为（可以实现相同的接口），由代理对象调委托对象。好处对委托类功能增强，符合开闭原则，只修改代理类。

动态代理：代理类持有委托类（任意类型），构造代理对象时传入委托对象，调用时调用proxy.invoke(method, args)，执行委托对象的method方法。动态代理可以代理任何对象。

四、工厂模式

简单工厂：根据传入参数不同构造不同类型的对象（跑车、卡车、轿车）

抽象工厂：todo

1. Facade模式（外观模式）

核心：封装交互，简化调用

如果客户端需要调用5个关联模块，可以使用Facade类封装这些模块的调用，提供统一的调用接口给客户端。dubbo服务提供的服务就是外观模式。