|  |  |  |
| --- | --- | --- |
| **设计模式** | **一句话总结** | **例举** |
| 工厂模式 | 统一创建对象，封装创建过程 | BeanFactory |
| 单例模式 | 保证只会创建一个对象 | ApplicationContext |
| 原型模式 | 利用原有的对象创建新的对象 | ArrayList |
| 代理模式 | 使用新的类代替原来的类，新类持有原来类的引用 | JdkDynamicAopProxy、CglibAopProxy |
| 委派模式 | 交给其他类去实现 | DispatcherServlet |
| 策略模式 | 把繁杂的过程封装，对外提供简单的选择 | InstantiationStrategy |
| 模版模式 | 父类定义一个流程，不确定的环节交给子类去实现 | JdbcTemplate、HttpServlet |
| 适配器模式 | 把一个不匹配的对象转换成一个匹配的对象 | AdvisorAdapter、HandleAdapter |
| 装饰模式 | 包装老类产生新类，新老类都有共同父类 | BufferedReader、InputStream |
| 观察者模式 | 观察者监听被观察者，被观察者发送变化时通知观察者 | ContextLoaderListener |