资料整理

1. ****设计模式****
2. **工厂模式（Factory）：**分为简单工厂和抽象工厂，都是解决实际过程中标准化生产或者创建的问题。比如从三无小作坊---->到标准工厂生产---->国际化流水线。
3. **单例模式（Singleton）：**保证一个类只有一个实例，并提供一个访问它的全局访问点

常见有懒汉式、饿汉式、注册登记式（静态内部类）

1. **原型模式（Prototype）：**以对象为原型进行克隆复制，技术实现有实现cloneable重写clone方法、字节码操作实现、反射机制实现。
2. **代理模式（Proxy）：**分为动态代理和静态代理，在代理对象和被代理对象中间的处理

有jdk动态代理和cglib代理

1. **策略模式（Strategy）：**执行过程中执行逻辑可以选择不同策略来执行，但是最终指向结果是固定的，比如购物结算可以选择不同的支付结算方式
2. **模板方法模式（Template Method）：**执行流程固定，中间步骤有差别比如jdbcTemplate都需要进行访问数据库流程，但是不同数据库可以切换。
3. **委派模式（Delegate）：**是一种特殊的代理模式，关键字全权代理，比如项目经理派发共工作认、ServletDispatcher
4. **适配器模式（Adapter）：**注重兼容、转换适配不同的情况。比如兼容老系统的接口
5. **装饰器模式（Decorator）：**主要实现对原有功能的包装和增强，注重覆盖和扩展
6. **观察者模式（Observer）：**当一个操作结束后通知被观察者，比如监听器listener
7. ****Spring源码****
8. **IOC/DI**
9. **定位 对springBean定义的xml进行定位**
10. **加载 BeanDefinition保存类信息，包括OOP关系**
11. **注册 Factroy、Context 就是把用户所定义的Bean放到IOC容器中（Map）**
12. **SpringMvc**

**配置 在web.xml中配置init-param、servlet-pattern入口参数**

**初始化 通过对servlet的继承调用onRefresh然后对9大组件进行初始化**

**请求处理 通过调用doService在调用doDispatch来进行分发处理请求然后返回字符串写入response**

1. ****Mybatis源码****