

# 一.代理设计模式

1设计模式:前人总结的一套解决特定问题的代码

2代理设计模式优点:

- 2.1保护真实对象
- 2.2让真实对象职责更明确
- 2.3扩展

3代理设计模式

- 3.1真实对象(老总)
- 3.2代理对象(秘书)
- 3.3抽象对象(抽象功能),谈小目标

# 二.静态代理设计模式

1由代理对象代理所有真实对象的功能

- 1.1自己编写代理类
- 1.2每个代理的功能需要单独编写

2静态代理设计模式的缺点

- 2.1当代理功能比较多时,代理类中方法也需要写很多

# 三.动态代理

1为了解决静态代理频繁编写代理功能缺点

2分类:

- 2.1JDK提供的
- 2.2cglib动态代理

# 四.JDK动态代理

1. 和cglib动态代理对比

- 1.1优点:jdk自带,不需要额外导入jar
- 1.2缺点:

- 1.2.1 真实对象必须实现接口
- 1.2.2 利用反射机制, 效率不高

## 五.cglib动态代理

### 1. cglib优点:

#### 1.1 基于字节码, 生成真实对象的子类

##### 1.1.1 运行效率高于JDK动态代理

#### 1.2 不需要实现接口

### 2 cglib缺点

#### 2.1 非JDK功能, 需要额外导入jar (cglib.jar, 以及字节码解析包asm.jar)

### 3. 使用 spring aop 时, 只要出现 Proxy 和真实对象转换异常

#### 3.1 设置为 true 使用 cglib

#### 3.2 设置为 false 使用 jdk (默认值)

`<aop:aspectj-autoproxy proxy-target-class="true"></aop:aspectj-autoproxy>`