静态代理设计模式

# 代理设计模式

代理设计模式是Java中的一种**常用的设计模式**，包括**静态代理设计模式**和**动态代理设计模式**。

# 静态代理设计模式：

## 静态代理设计模式的条件：

### 具备真实角色；

### 具备代理角色：**持有真实角色的引用**；

### 真实角色和代理角色**实现相同的接口**。

## 具体编程中的设计模式理解

不直接利用某个对象的功能，而是将该对象传递给代理对象，通过调用代理对象的功能实现原对象的功能。要实现静态代理设计模式，就必须要求代理对象必须和原对象实现相同的接口。

## 举例说明

### 打比方说明

自己买房子：自己找个中介代理买房子，只需要把要求给中介，中介可以帮你完成，但是真正买房子的是**自己**，而联系房主的是中介。

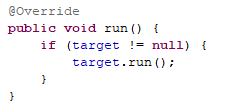
### Thread代理：典型的静态代理设计模式

真实角色(真正的线程任务)：class **Task** implements Runnable

代理角色：class **Thread** implements Runnable

Task 与 Thread都实现了Runnable接口。

通过Thread thread = new Thread(**new Task()**);将线程任务task交给Thread代理thread执行，Thread就是作为Task的代理角色。  
Thread类run()方法的源码：



# 动态代理设计模式

通过**反射机制**进行代理设计。

通常一些框架就是利用反射机制(动态代理设计模式)实现的。