# 代理模式（Proxy）

目录

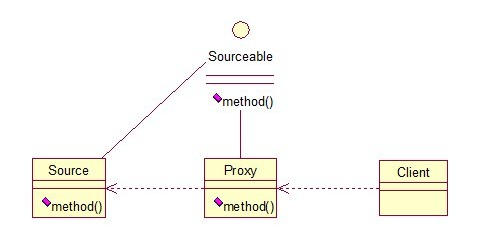
[代理模式（Proxy） 1](#_Toc23233132)

[**一、 综述** 1](#_Toc23233133)

1. **综述**

代理模式就是多一个代理类出来，替原对象进行一些操作，比如我们在租房子的时候回去找中介，为什么呢？因为你对该地区房屋的信息掌握的不够全面，希望找一个更熟悉的人去帮你做，此处的代理就是这个意思。再如我们有的时候打官司，我们需要请律师，因为律师在法律方面有专长，可以替我们进行操作，表达我们的想法。先来看看关系图：

工厂模式可以分为三类：



代码：

public interface Sourceable {

public void method();

}

public class Source implements Sourceable {

@Override

public void method() {

System.out.println("the original method!");

}

}

public class Proxy implements Sourceable {

private Source source;

public Proxy() {

super();

this.source = new Source();

}

@Override

public void method() {

before();

source.method();

atfer();

}

private void atfer() {

System.out.println("after proxy!");

}

private void before() {

System.out.println("before proxy!");

}

}

测试类：

public class ProxyTest {

public static void main(String[] *args*) {

Sourceable source = new Proxy();

source.method();

}

}

输出：

before proxy!  
the original method!  
after proxy!

代理模式的应用场景：

如果已有的方法在使用的时候需要对原有的方法进行改进，此时有两种办法：

1、修改原有的方法来适应。这样违反了“对扩展开放，对修改关闭”的原则。

2、就是采用一个代理类调用原有的方法，且对产生的结果进行控制。这种方法就是代理模式。

使用代理模式，可以将功能划分的更加清晰，有助于后期维护！