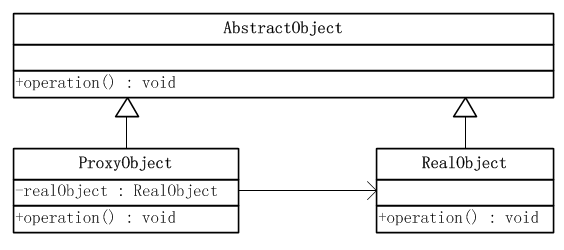
在阎宏博士的《JAVA与模式》一书中开头是这样描述代理（Proxy）模式的：

**代理模式是对象的结构模式。代理模式给某一个对象提供一个代理对象，并由代理对象控制对原对象的引用。**

**代理模式的结构**

　　所谓代理，就是一个人或者机构代表另一个人或者机构采取行动。在一些情况下，一个客户不想或者不能够直接引用一个对象，而代理对象可以在客户端和目标对象之间起到中介的作用。

　　代理模式类图如下：



　　在代理模式中的角色：

　　●　　**抽象对象角色：**声明了目标对象和代理对象的共同接口，这样一来在任何可以使用目标对象的地方都可以使用代理对象。

　　●　　**目标对象角色：**定义了代理对象所代表的目标对象。

　　●　　**代理对象角色：**代理对象内部含有目标对象的引用，从而可以在任何时候操作目标对象；代理对象提供一个与目标对象相同的接口，以便可以在任何时候替代目标对象。代理对象通常在客户端调用传递给目标对象之前或之后，执行某个操作，而不是单纯地将调用传递给目标对象。

**源代码**

　　抽象对象角色

public abstract class AbstractObject {

//操作

public abstract void operation();

}

　　目标对象角色

[复制代码](javascript:void(0);)

public class RealObject extends AbstractObject {

@Override

public void operation() {

//一些操作

System.out.println("一些操作");

}

}

[复制代码](javascript:void(0);)

　　代理对象角色

[复制代码](javascript:void(0);)

public class ProxyObject extends AbstractObject{

RealObject realObject = new RealObject();

@Override

public void operation() {

//调用目标对象之前可以做相关操作

System.out.println("before");

realObject.operation();

//调用目标对象之后可以做相关操作

System.out.println("after");

}

}

[复制代码](javascript:void(0);)

　　客户端

[复制代码](javascript:void(0);)

public class Client {

public static void main(String[] args) {

// TODO Auto-generated method stub

AbstractObject obj = new ProxyObject();

obj.operation();

}

}

[复制代码](javascript:void(0);)

　　从上面的例子可以看出代理对象将客户端的调用委派给目标对象，在调用目标对象的方法之前跟之后都可以执行特定的操作。