

阿拉丁博客

寻梦岁月，不言辛苦几许，不问收获几多！

[登录](#) [注册](#) [博客首页](#)

□□

[空间](#) [博客](#) [好友](#) [相册](#) [留言](#)

用户操作

[\[留言\]](#) [\[发消息\]](#) [\[加为好友\]](#)

订阅我的博客



aladdinty的公告

文章分类

- [C#基础](#)
- [JAVA](#)
- [LINQ](#)
- [泛型](#)
- [集合](#)
- [技术收藏](#)
- [设计模式](#)
- [通信](#)
- [委托与事件](#)

存档

- [2010年03月\(1\)](#)
- [2009年09月\(5\)](#)
- [2009年07月\(3\)](#)
- [2009年05月\(1\)](#)
- [2009年04月\(1\)](#)
- [2009年03月\(3\)](#)
- [2009年02月\(2\)](#)
- [2009年01月\(19\)](#)
- [2008年12月\(33\)](#)
- [2008年11月\(30\)](#)

[java中InvocationHandler 用于实现代理。](#) [收藏](#)

InvocationHandler 用于实现代理。

如果不用InvocationHandler接口实现代理的话，我们写代码是这样的：
定义一个接口：

Java code

```
interface Greet
{
    void sayHello(String name);

    void goodBye();
}
```

实现这个接口：

Java code

```
class GreetImpl implements Greet
{
    public void sayHello(String name)
    {
        System.out.println("Hello " + name);
    }

    public void goodBye()
    {
        System.out.println("Good bye.");
    }
}
```

Java code

```
public class SimpleProxy implements Greet
{
    private Greet greet = null;

    SimpleProxy(Greet greet)
    {
        this.greet = greet;
    }

    public void sayHello(String name)
    {
        System.out.println("--before method sayHello");
        greet.sayHello(name);
        System.out.println("--after method sayHello");
    }

    public void goodBye()
    {
        System.out.println("--before method goodBye");
        greet.goodBye();
        System.out.println("--after method goodBye");
    }

    /**
     * @param args
     */
    public static void main(String[] args)
    {
        Greet greet = new SimpleProxy(new GreetImpl()); //生成代理
        greet.sayHello("walter");
        greet.goodBye();
    }
}
```

代理其实没什么的，再看看如果实现了InvocationHandler接口，我们怎样实现代理。
还是要实现原来的Greet接口。

接口的实现还是GreetImpl。

Java code

```
import java.lang.reflect.InvocationTargetException;
import java.lang.reflect.Method;

public class DebugProxy implements java.lang.reflect.InvocationHandler
{
    private Object obj;

    public static Object newInstance(Object obj)
    {
        return java.lang.reflect.Proxy.newProxyInstance(obj.getClass().getClassLoader(),
            obj.getClass().getInterfaces(), new DebugProxy(obj));
    }

    private DebugProxy(Object obj)
    {
        //Greet接口的實現: GreetImpl
        this.obj = obj;
    }

    //Method m: 調用的方法
    //Object[] args: 方法要傳入的參數
    public Object invoke(Object proxy, Method m, Object[] args) throws Throwable
    {
        Object result;
        try
        {
            //自定義的處理
            System.out.println("--before method " + m.getName());
            //調用GreetImpl中方法
            result = m.invoke(obj, args);
        }
        catch (InvocationTargetException e)
        {
            throw e.getTargetException();
        }
        catch (Exception e)
        {
            throw new RuntimeException("unexpected invocation exception: " + e.getMessage());
        }
    }
}
```

```

    }
    finally
    {
        System.out.println("--after method " + m.getName());
    }
    return result;
}

/**
 * @param args
 */
public static void main(String[] args)
{
    Greet tmp = new GreetImpl();

    Greet greet = (Greet) DebugProxy.newInstance(tmp);
    //生成的greet和tmp有相同的hashCode

    greet.sayHello("walter");
    greet.goodBye();
}
}

```

发表于 @ 2009年03月11日 21:50:00 | [评论\(0\)](#) | [举报](#) | [收藏](#)

[旧一篇: 简单Servlet示例](#) | [新一篇: java实现静态代理](#)

[.Net E-mail Component](#)

Reliable POP3, IMAP, SMTP, SSL Download right now!
www.afterlogic.com

[笔记本电脑 特价680元](#)

出售特价笔记本电脑 手机等 销售电话
 姐
www.kl666888.co.to

发表评论

表情:



评论内容:

用户名: 匿名用户

[登录](#) [注册](#)

验证码:



[重新获得验证码](#)



www.CarSing.com.cn

Google 提供的广告

Copyright © aladdinty

Powered by CSDN Blog