

实验四.txt

实验四, Spring1前通知

PreBeforeAdviceDemo



步骤一,

//业务接口

```
public interface IHello {
```

```
    // 业务方法声明
```

```
    public void hello(String name);
```

```
}
```

步骤二,

//被代理对象

```
public class HelloSpeaker implements IHello {
```

```
    // 业务方法实现
```

```
    public void hello(String name) {
```

```
        System.out.println("你好, " + name);
```

```
    }
```

```
}
```

步骤三,

实验四.txt

```
public class LogBeforeAdvice implements MethodBeforeAdvice {  
  
    private Logger logger =  
Logger.getLogger(this.getClass().getName());  
  
    public void before(Method method, Object[] args, Object  
target)  
        throws Throwable {  
  
        logger.info("日志开始");  
  
    }  
  
}
```

步骤四，

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<beans xmlns="http://www.springframework.org/schema/beans"
```

```
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
xsi:schemaLocation="http://www.springframework.org/schema/beans
```

```
http://www.springframework.org/schema/beans/spring-beans-2.0.xsd">
```

```
<!-- 目标对象 -->
```

```
<bean id="HelloSpeaker" class="tarena.HelloSpeaker" />
```

```
<!-- 通知 -->
```

```
<bean id="LogBeforeAdvice"  
class="tarena.LogBeforeAdvice" />
```

```
<!-- 代理工厂类 -->
```

```
<bean id="ProxyFactoryBean"
```

```
class="org.springframework.aop.framework.ProxyFactoryBean">
```

```
<property name="target">
```

实验四.txt

```
        <ref bean="HelloSpeaker" />
    </property>
    <property name="interceptorNames">
        <list>

<value>LogBeforeAdvice</value>
        </list>
    </property>
</bean>
</beans>
```

步骤五，

```
public class Test {

    public static void main(String[] args) {

        ApplicationContext context = new
ClassPathXmlApplicationContext(
                                "tarena/spring-config.xml");

        IHello proxy = (IHello)
context.getBean("ProxyFactoryBean");

        proxy.hello("赵君");

    }

}
```

步骤六，

信息：日志开始
你好，赵君

实验四.txt