实验五，　Spring2前通知

```
BeforeAdviceDemo
            |
            |-----tarena
                        |
                        |---------spring-config.xml
```

加入Spring2Core包
加入Spring2AOP包

步骤一，

```
public interface IHello {
            public void hello(String name);
}
```

步骤二，

```
public class HelloSpeaker implements IHello {

            public void hello(String name) {
                        System.out.println("你好，" + name);
            }

}
```

步骤三，

```
//日志切面
@Aspect
public class LogAspect {

            private Logger logger =
Logger.getLogger(this.getClass().getName());

            // 日志代码
            @Pointcut("execution(* tarena.IHello.*(..))")
```

```
		private void logger() {
		}

		@Before("logger()")
		public void before(JoinPoint joinPoint) {
			System.out.println("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"

xmlns:aop="http://www.springframework.org/schema/aop"

xsi:schemaLocation="http://www.springframework.org/schema/bea
ns

http://www.springframework.org/schema/beans/spring-beans-2.0.x
sd
		http://www.springframework.org/schema/aop

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

		<!-- 被代理对象 -->
		<bean id="HelloSpeaker" class="tarena.HelloSpeaker"/>

		<!-- 前通知-->
		<bean id="LogAspect" class="tarena.LogAspect"/>

		<!-- 动态代理对象 -->
		<aop:aspectj-autoproxy/>

</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("HelloSpeaker");

			proxy.hello("赵君");
		}

}
```

步骤六，
日志开始
你好，赵君