

## 疑惑

1.比如说一个接口Book，一个实现类BookImpl

那么创建对象时，

是Book book = (Book)applicationContext.getBean("Book");

还是BookImpl bookImpl = (BookImpl)applicationContext.getBean("Book");

结论：创建对象，new接口

2.Object obj = method.invoke(userService, args);

底层是如何运行的？

不用Object obj 行不行？意思就是直接method.invoke(userService, args);

```
enhancer.setCallback(new MethodInterceptor(){

    public Object intercept(Object proxy, Method method, Object[] args,
        MethodProxy methodProxy) throws Throwable {

        //前
        myAspect.before();

        //执行目标类的方法
        method.invoke(userService, args);
        //执行代理类的父类,执行目标类（目标类和代理类 父子关系）
        methodProxy.invokeSuper(proxy, args);

        //后
        myAspect.after();
        System.out.println(obj);
        return null;
    }
});
```

答案：

```
enhancer.setCallback(new MethodInterceptor(){

    public Object intercept(Object proxy, Method method, Object[] args,
        MethodProxy methodProxy) throws Throwable {

        //前
        myAspect.before();

        //执行目标类的方法
        Object obj = method.invoke(userService, args);
        //执行代理类的父类,执行目标类（目标类和代理类 父子关系）
        methodProxy.invokeSuper(proxy, args);

        //后
        myAspect.after();
        return obj;
    }
});
```

要有个object类返回值

