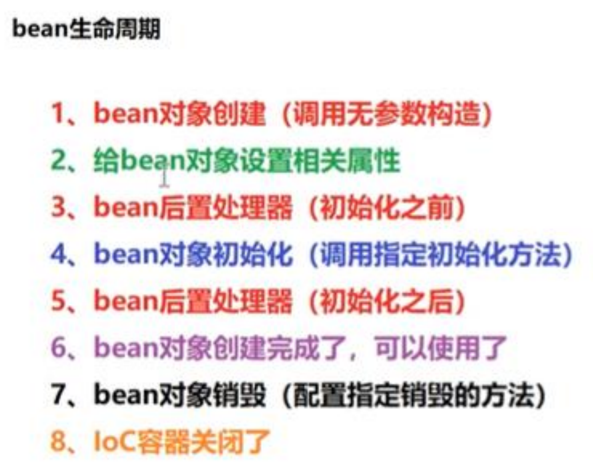
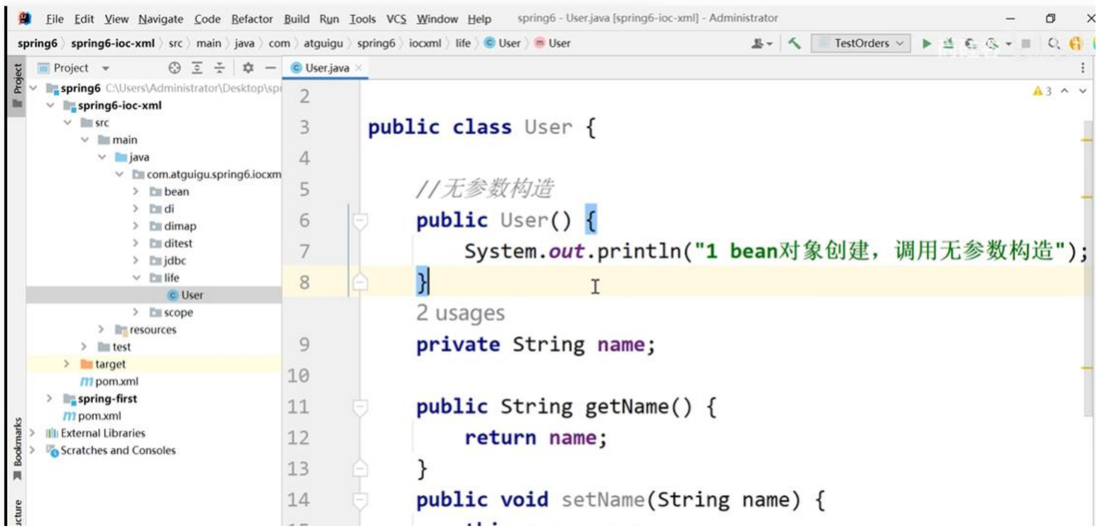
实验 11：bean 的生命周期


bean生命周期

1、bean对象创建（调用无参数构造）
2、给bean对象设置相关属性
3、bean后置处理器（初始化之前）
4、bean对象初始化（调用指定初始化方法）
5、bean后置处理器（初始化之后）
6、bean对象创建完成了，可以使用了
7、bean对象销毁（配置指定销毁的方法）
8、IoC容器关闭了

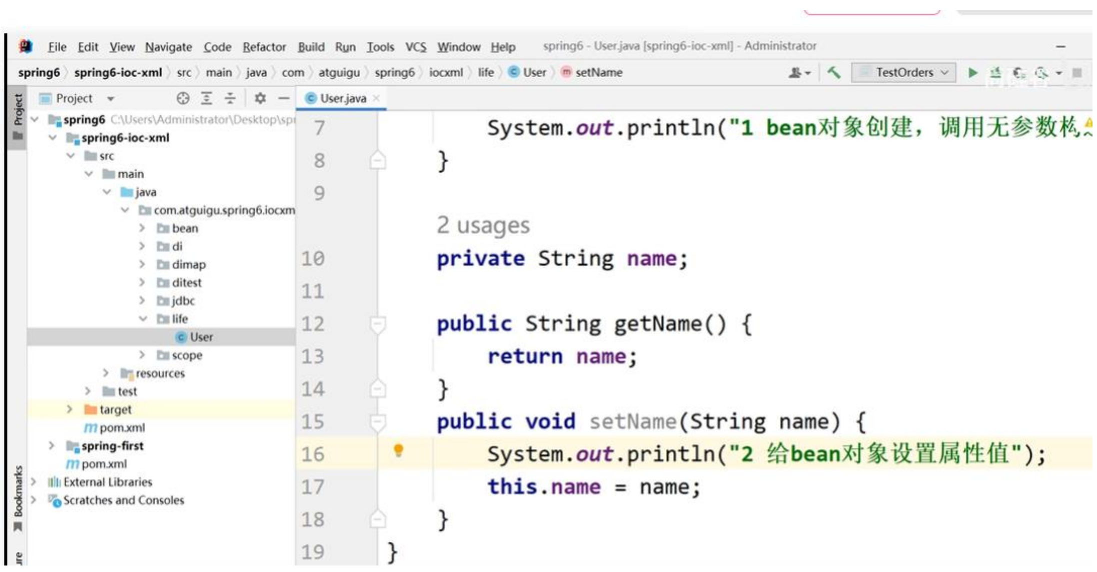
环境准备：

1、



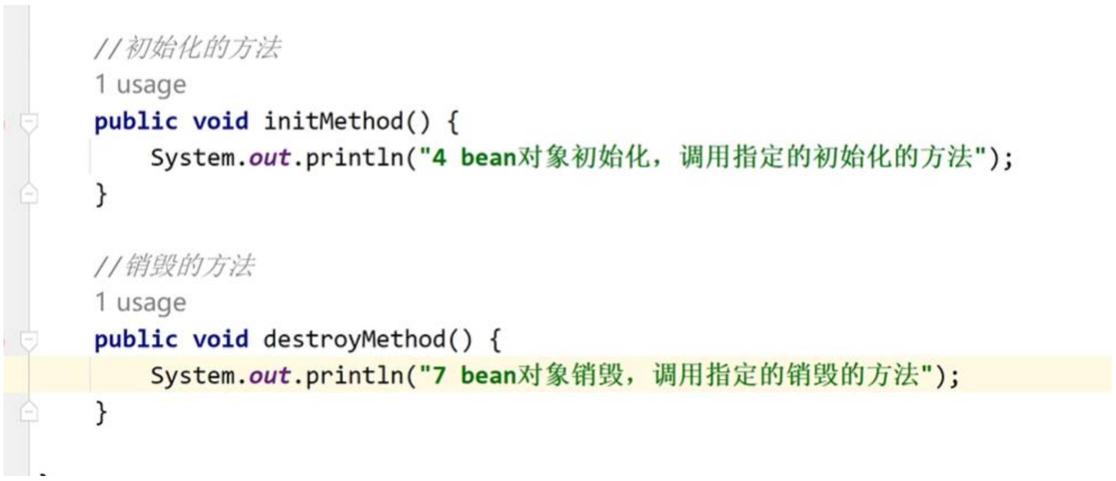
```java
public class User {

    //无参数构造
    public User() {
        System.out.println("1 bean对象创建，调用无参数构造");
    }
    2 usages
    private String name;

    public String getName() {
        return name;
    }
    public void setName(String name) {
```

2



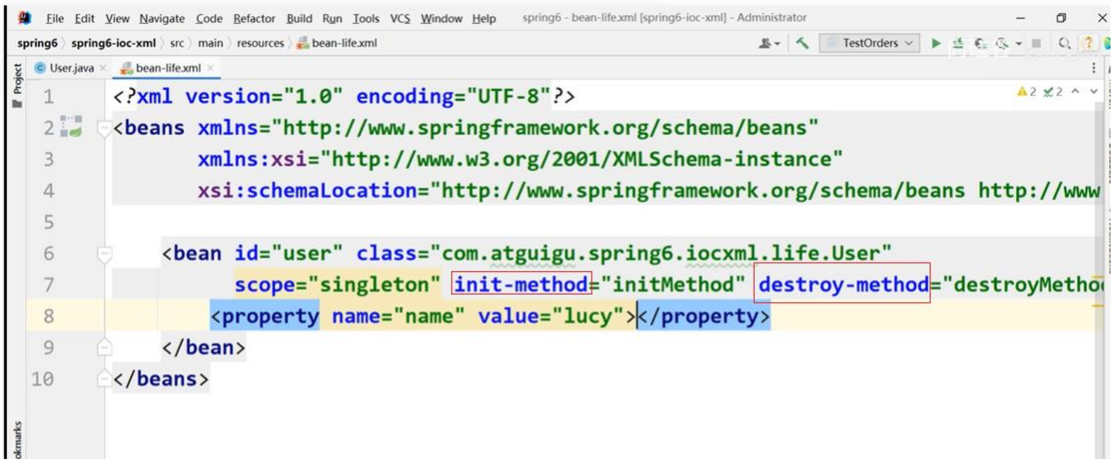
```java
        System.out.println("1 bean对象创建，调用无参数构
    }

    2 usages
    private String name;

    public String getName() {
        return name;
    }
    public void setName(String name) {
        System.out.println("2 给bean对象设置属性值");
        this.name = name;
    }
}
```

3、



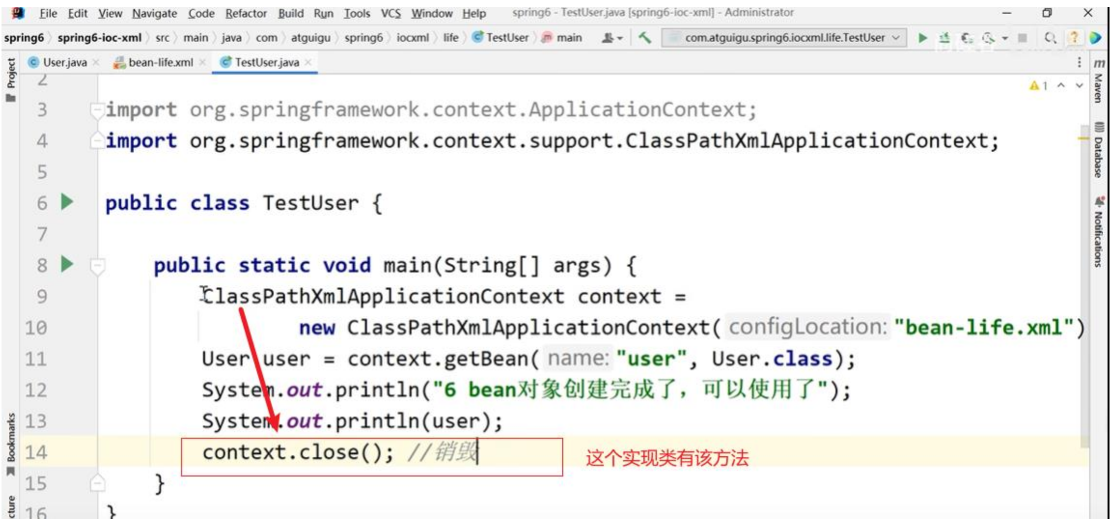
```java
//初始化的方法
1 usage
public void initMethod() {
    System.out.println("4 bean对象初始化，调用指定的初始化的方法");
}

//销毁的方法
1 usage
public void destroyMethod() {
    System.out.println("7 bean对象销毁，调用指定的销毁的方法");
}
```

测试：



File Edit View Navigate Code Refactor Build Run Tools VCS Window Help spring6 - bean-life.xml [spring6-ioc-xml] - Administrator
spring6 〉 spring6-ioc-xml 〉 src 〉 main 〉 resources 〉 bean-life.xml
TestOrders
User.java bean-life.xml

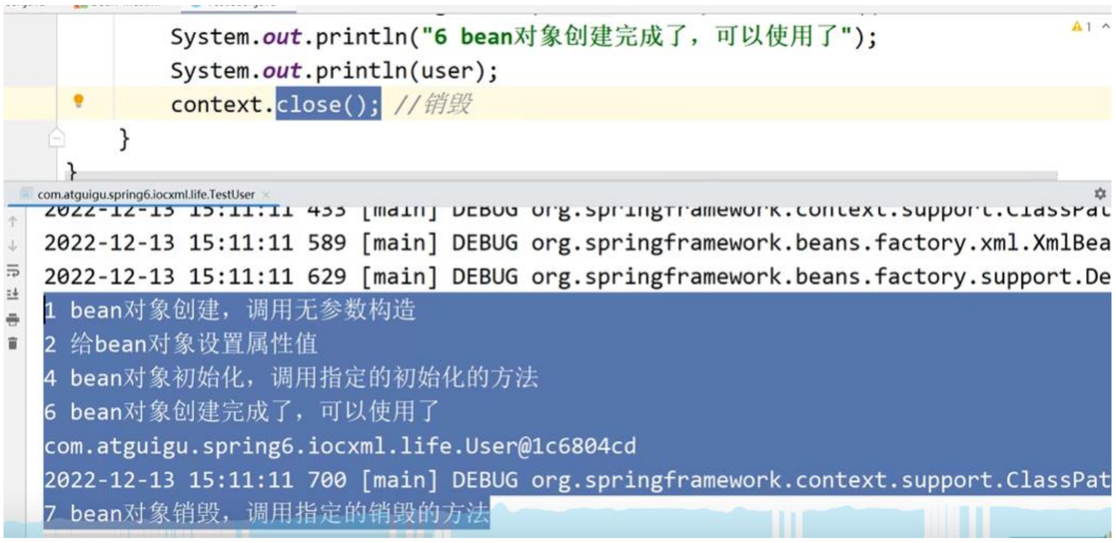
```xml
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="http://www.springframework.org/schema/beans http://www

    <bean id="user" class="com.atguigu.spring6.iocxml.life.User"
          scope="singleton" init-method="initMethod" destroy-method="destroyMetho
        <property name="name" value="lucy"></property>
    </bean>
</beans>
```



File Edit View Navigate Code Refactor Build Run Tools VCS Window Help spring6 - TestUser.java [spring6-ioc-xml] - Administrator
spring6 〉 spring6-ioc-xml 〉 src 〉 main 〉 java 〉 com 〉 atguigu 〉 spring6 〉 iocxml 〉 life 〉 TestUser 〉 main
com.atguigu.spring6.iocxml.life.TestUser
User.java bean-life.xml TestUser.java
这个实现类有该方法

```java
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

public class TestUser {

    public static void main(String[] args) {
        ClassPathXmlApplicationContext context =
                new ClassPathXmlApplicationContext( configLocation: "bean-life.xml");
        User user = context.getBean( name: "user", User.class);
        System.out.println("6 bean对象创建完成了，可以使用了");
        System.out.println(user);
        context.close(); //销毁
    }
}
```

结果：

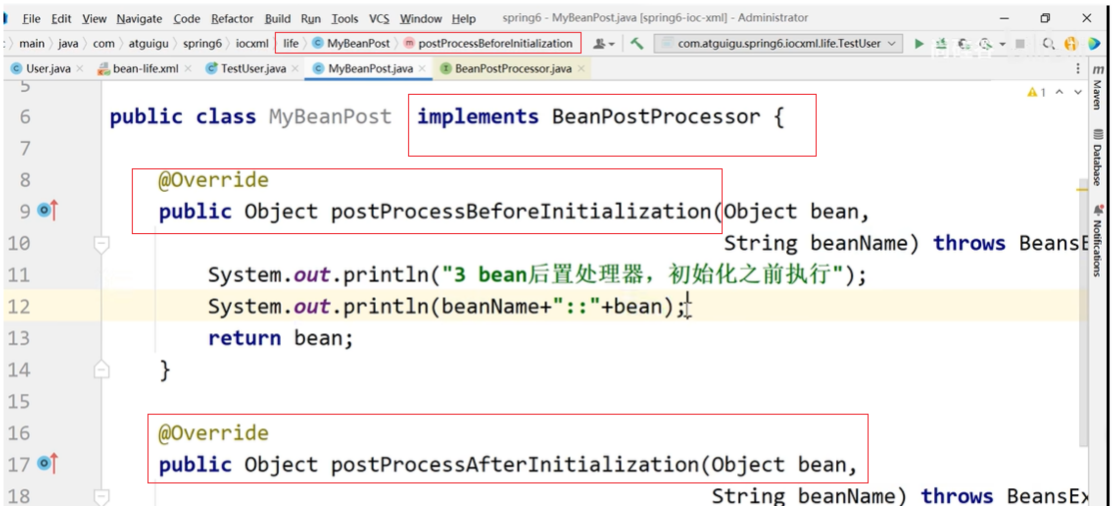


```
        System.out.println("6 bean对象创建完成了，可以使用了");
        System.out.println(user);
        context.close(); //销毁
    }
}
```

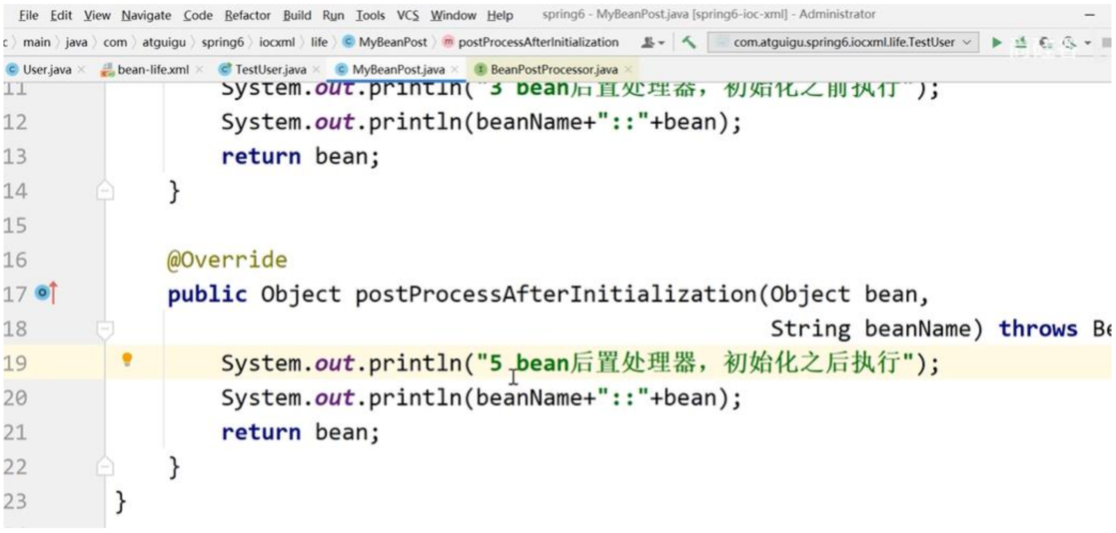
com.atguigu.spring6.iocxml.life.TestUser

```
2022-12-13 15:11:11 433 [main] DEBUG org.springframework.context.support.ClassPat
2022-12-13 15:11:11 589 [main] DEBUG org.springframework.beans.factory.xml.XmlBea
2022-12-13 15:11:11 629 [main] DEBUG org.springframework.beans.factory.support.De
1 bean对象创建，调用无参数构造
2 给bean对象设置属性值
4 bean对象初始化，调用指定的初始化的方法
6 bean对象创建完成了，可以使用了
com.atguigu.spring6.iocxml.life.User@1c6804cd
2022-12-13 15:11:11 700 [main] DEBUG org.springframework.context.support.ClassPat
7 bean对象销毁，调用指定的销毁的方法
```

环境准备：补充第三步和第五步：

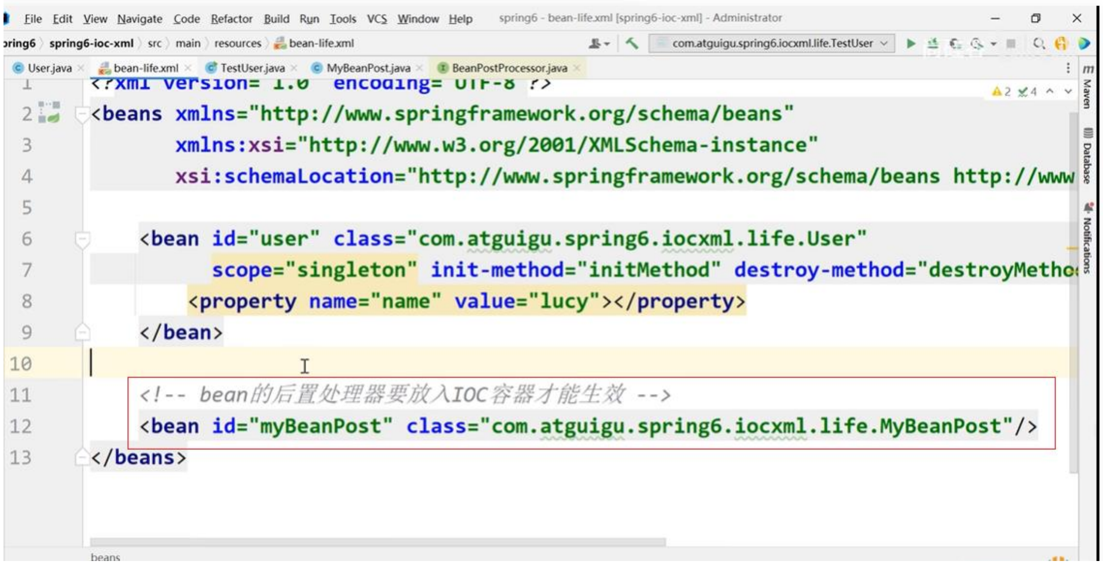


```
public class MyBeanPost implements BeanPostProcessor {

    @Override
    public Object postProcessBeforeInitialization(Object bean,
                                                  String beanName) throws BeansE
        System.out.println("3 bean后置处理器，初始化之前执行");
        System.out.println(beanName+"::"+bean);
        return bean;
    }

    @Override
    public Object postProcessAfterInitialization(Object bean,
                                                 String beanName) throws BeansEx
```



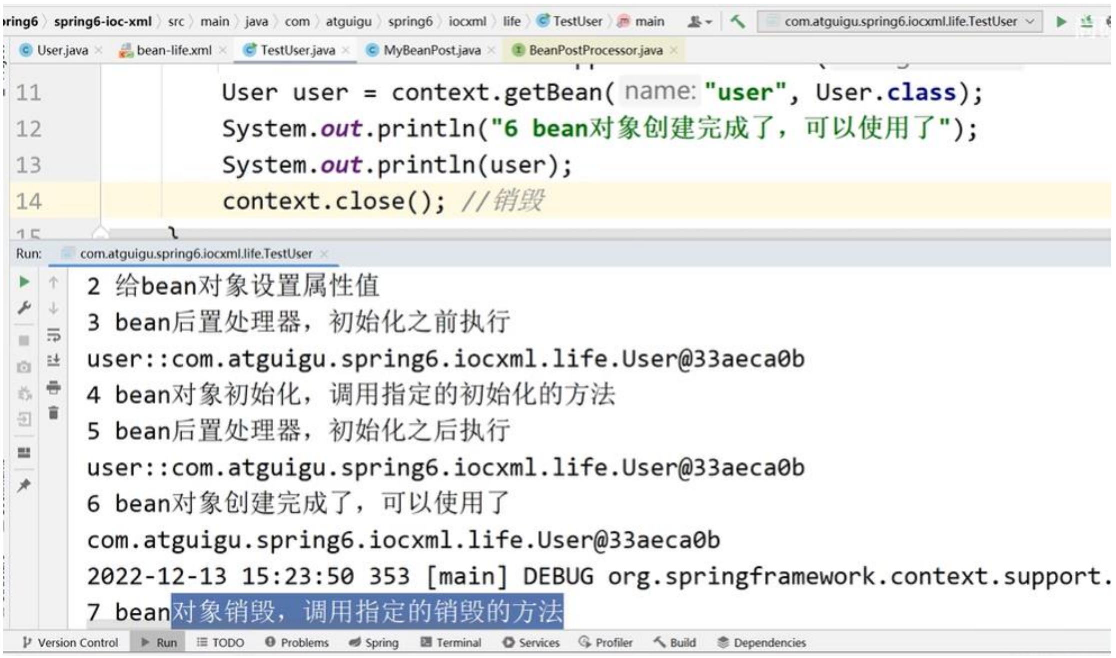
```
        System.out.println("3 bean后置处理器，初始化之前执行");
        System.out.println(beanName+"::"+bean);
        return bean;
    }

    @Override
    public Object postProcessAfterInitialization(Object bean,
                                                 String beanName) throws Be
        System.out.println("5 bean后置处理器，初始化之后执行");
        System.out.println(beanName+"::"+bean);
        return bean;
    }
}
```

测试：



```
<?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

    <bean id="user" class="com.atguigu.spring6.iocxml.life.User"
          scope="singleton" init-method="initMethod" destroy-method="destroyMetho
        <property name="name" value="lucy"></property>
    </bean>

    <!-- bean的后置处理器要放入IOC容器才能生效 -->
    <bean id="myBeanPost" class="com.atguigu.spring6.iocxml.life.MyBeanPost"/>
</beans>
```

结果:



```
        User user = context.getBean( name: "user", User.class);
        System.out.println("6 bean对象创建完成了，可以使用了");
        System.out.println(user);
        context.close(); //销毁
```

```
2 给bean对象设置属性值
3 bean后置处理器，初始化之前执行
user::com.atguigu.spring6.iocxml.life.User@33aeca0b
4 bean对象初始化，调用指定的初始化的方法
5 bean后置处理器，初始化之后执行
user::com.atguigu.spring6.iocxml.life.User@33aeca0b
6 bean对象创建完成了，可以使用了
com.atguigu.spring6.iocxml.life.User@33aeca0b
2022-12-13 15:23:50 353 [main] DEBUG org.springframework.context.support.
7 bean对象销毁，调用指定的销毁的方法
```