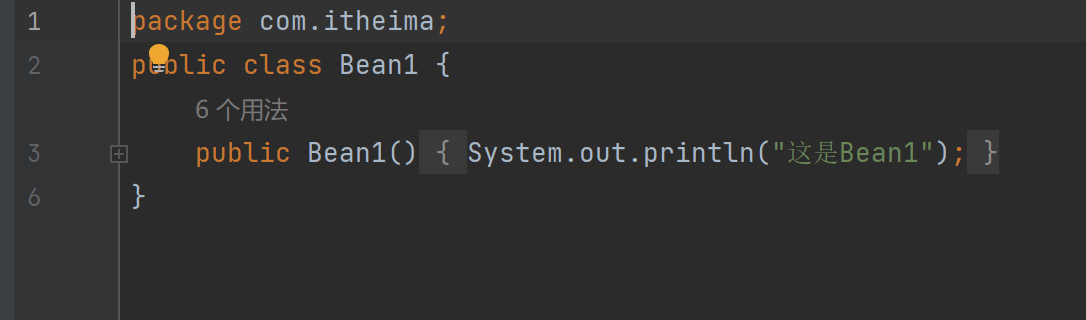
**Spring IoC容器**

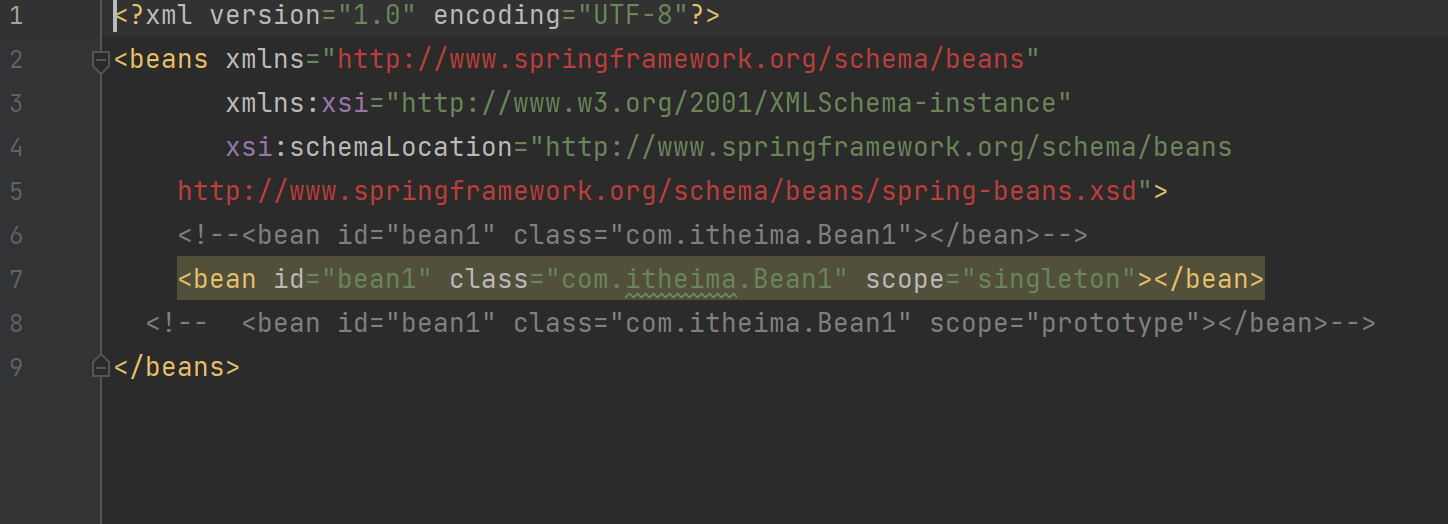
（1）**在IDEA中创建一个名为chapter07的Maven项目，然后在项目的pom.xml文件中配置需使用到的Spring四个基础包和Spring的依赖包。**



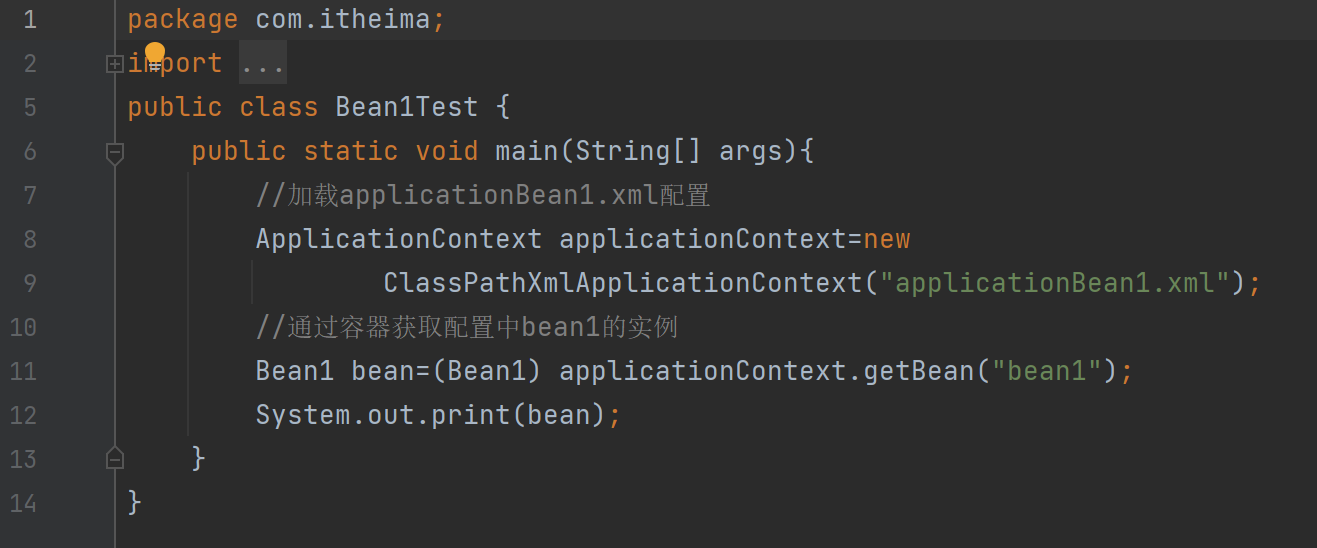
（2）**在chapter07项目的src/main/java目录下，创建一个名称为com.itheima的包，在该包中创建Bean1类。**



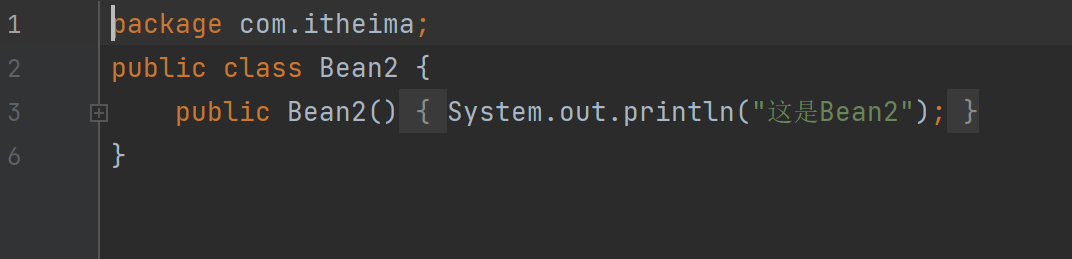
（3）**在chapter07项目的src/main/resources目录下新建applicationBean1.xml作为Bean1类的配置文件，在该配置文件中定义一个id为bean1的Bean，并通过class属性指定其对应的实现类为Bean1。**

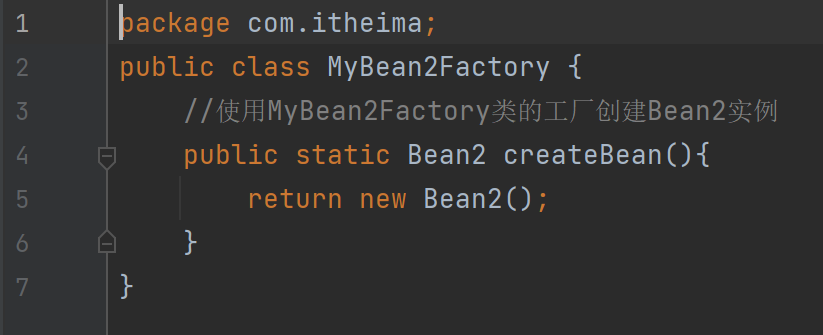


**（4）在chapter07项目的com.itheima包中创建测试类Bean1Test，在main()方法中通过加载applicationBean1.xml配置文件初始化Spring容器，再通过Spring容器生成Bean1类的实例bean1，用来测试构造方法是否能实例化Bean1。**

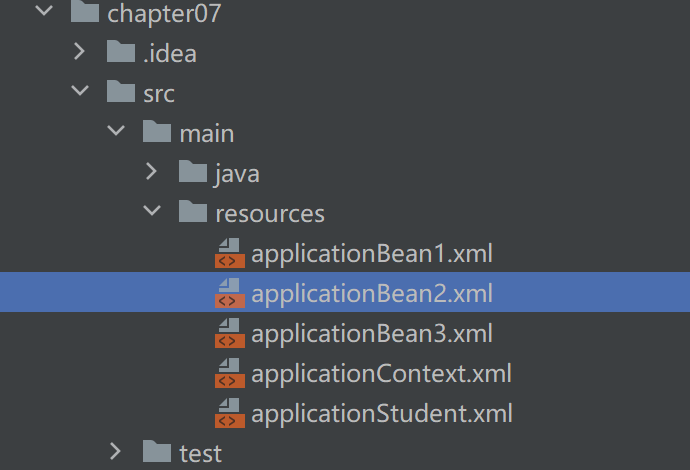


**Spring IoC容器**

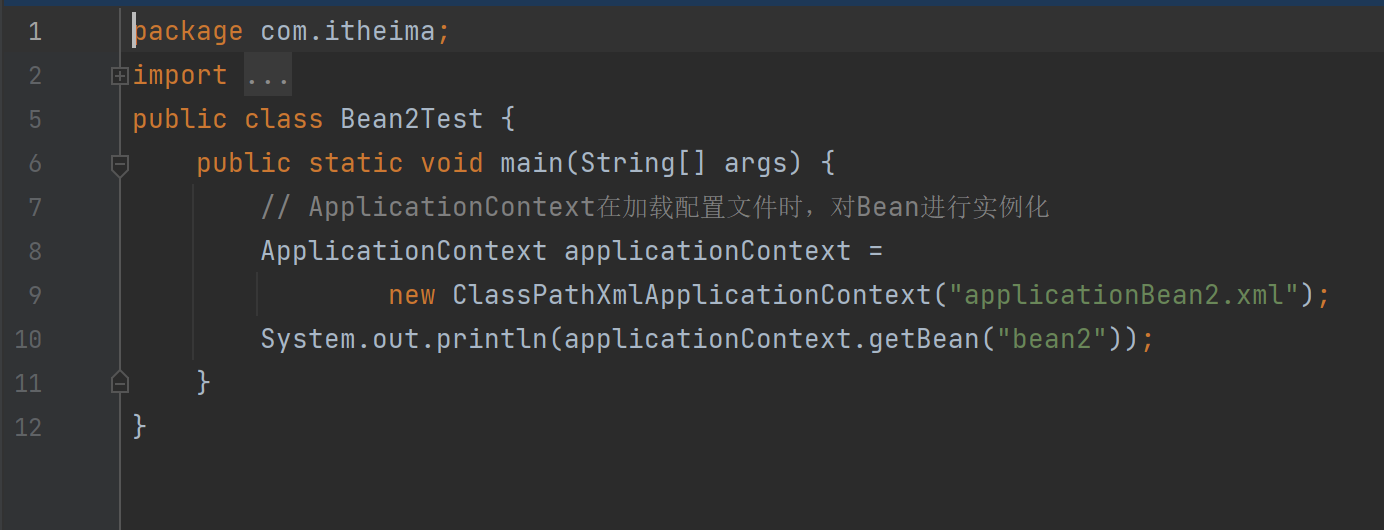
**（1）在chapter07项目的com.itheima包中创建Bean2类，该类与Bean1类一样，只定义一个构造方法，不需额外添加任何方法。**

**（2）在com.itheima包中，创建一个MyBean2Factory类，在该类中定义一个静态方法createBean()，用于创建Bean的实例。createBean()方法返回Bean2实例。**

**（3）在chapter07项目的src/main/resources目录下新建applicationBean2.xml文件，作为MyBean2Factory类的配置文件。**

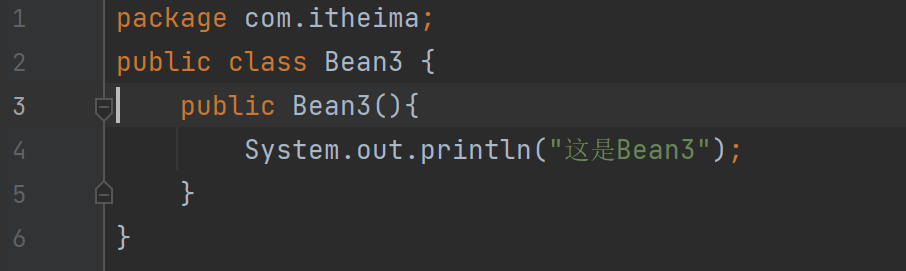


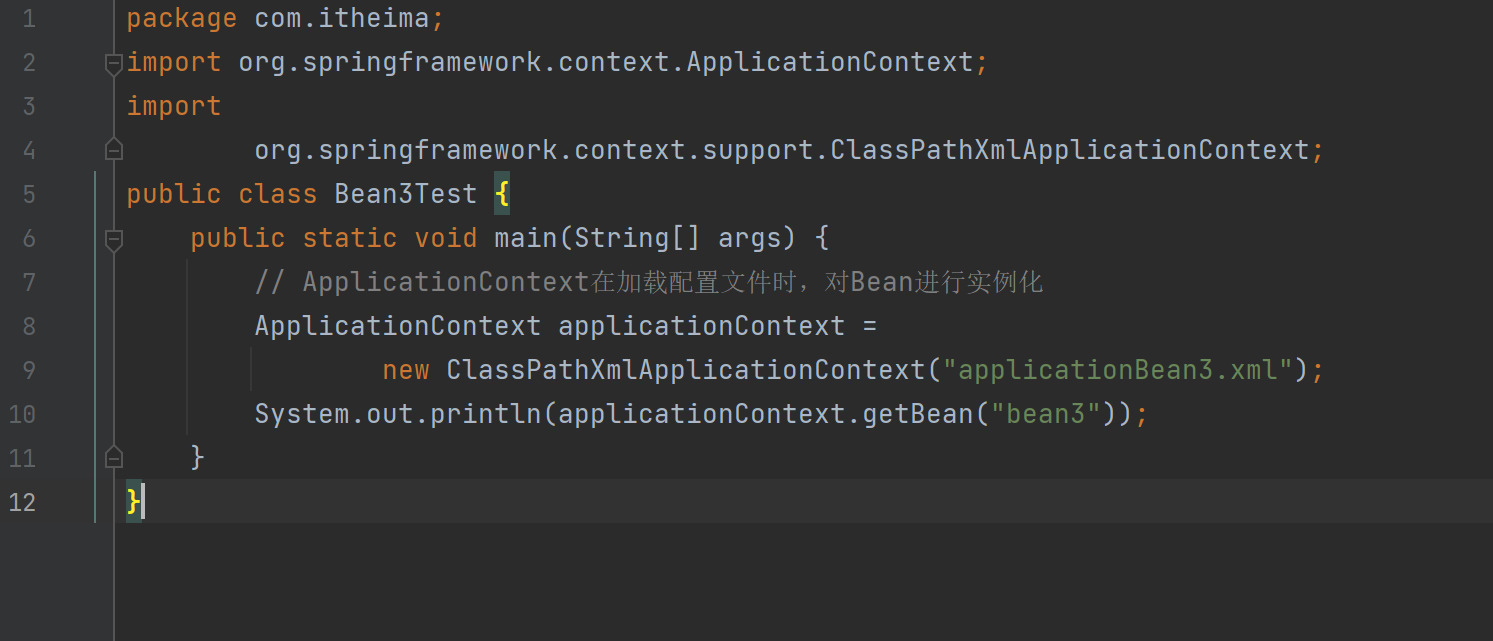
**（4）在com.itheima包中，创建一个测试类Bean2Test，用于测试使用静态工厂方式是否能实例化Bean。**



**静态工厂实例化**

**（1）在chapter07项目的com.itheima包中创建Bean3类，该类与Bean1一样，不需添加任何方法。**

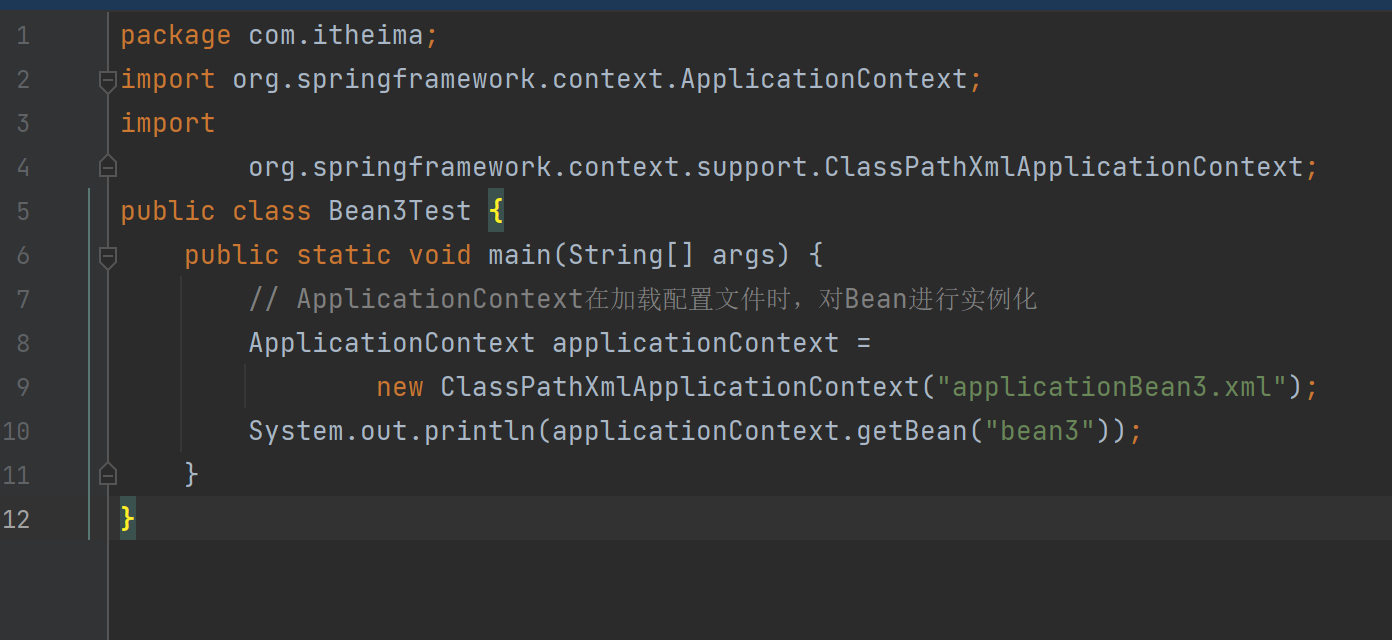


**（2）在com.itheima包中，创建一个MyBean3Factory类，在该类中定义无参构造方法，并定义createBean()方法用于创建Bean3对象。**

**（3） 在chapter07项目的src/main/resources目录下新建applicationBean3.xml文件，作为MyBean3Factory类的配置文件**



**（4） 在com.itheima包中，创建一个测试类BeanTest3，用于测试使用实例化工厂方式是否能实例化Bean。**



**bean的作用域**

**（1） 演示Spring容器中singleton作用域的使用  
：将id为bean1的作用域设置为singleton。**

**（2） 在chapter07项目的com.itheima包中创建测试类scopeTest，在main()方法中通过加载applicationBean1.xml配置文件初始化Spring容器，通过Spring容器获取Bean1类的两个实例，判断两个实例是否为同一个。**

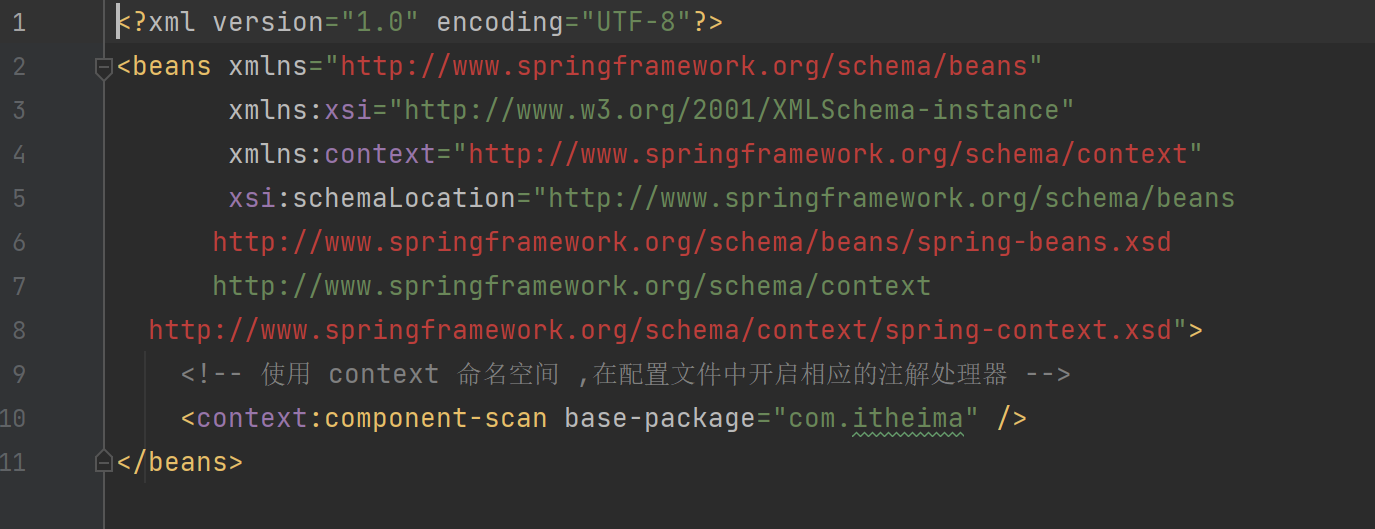


**Bean的装配方式**

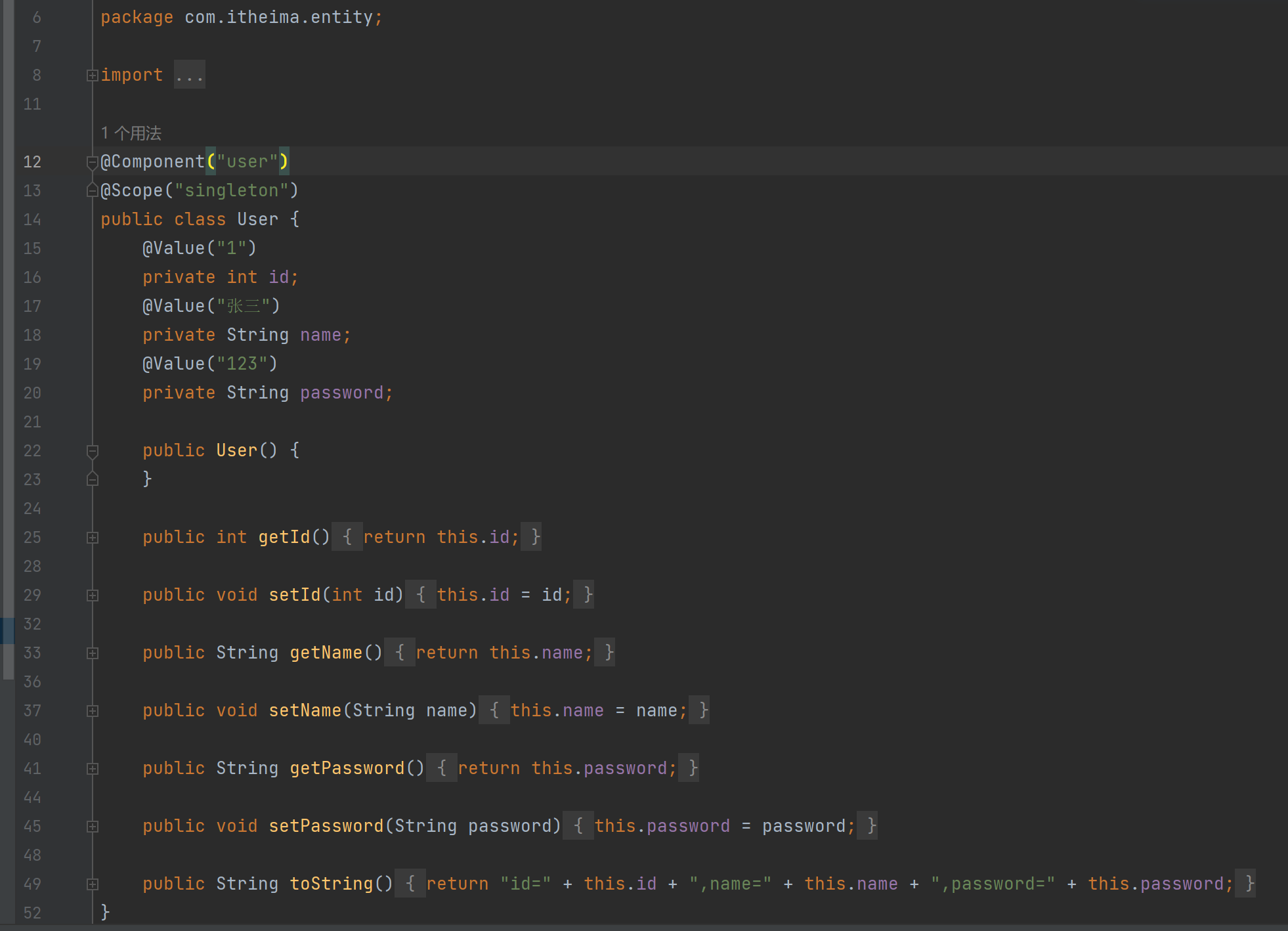
**（1） 在项目chapter07的pom.xml文件中导入spring-aop-5.2.8.RELEASE.jar依赖包。**

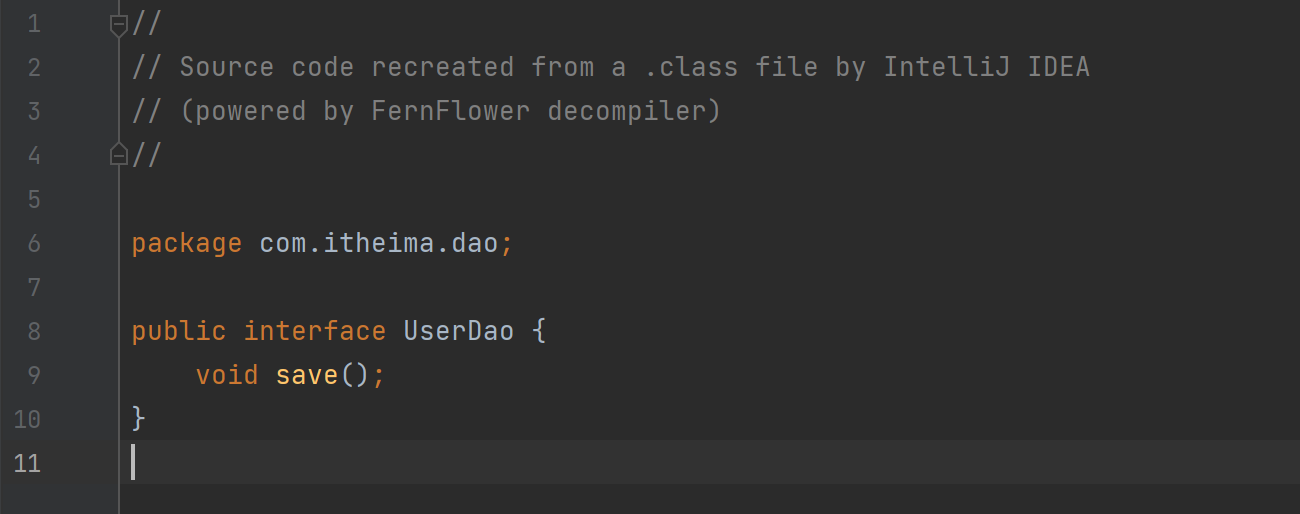


**（2） 在chapter07项目的src\main\resources目录下创建applicationContext.xml，在该文件中引入Context约束并启动Bean的自动扫描功能。**

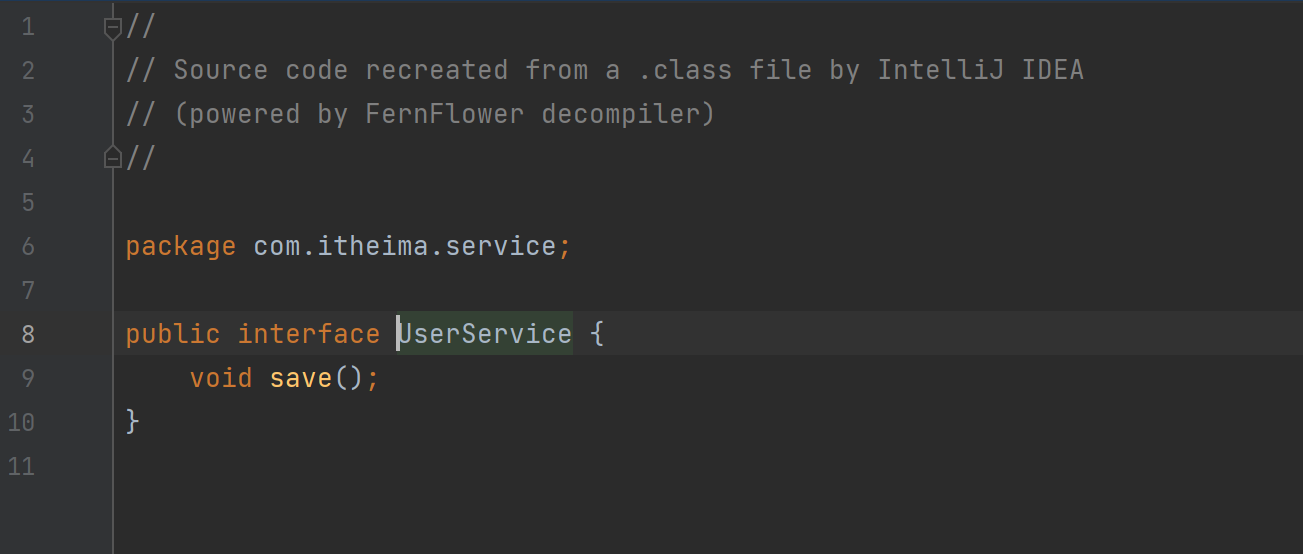


**（3） 在项目chapter07的com.itheima包下新建entity包，在entity包下创建User实体类。**

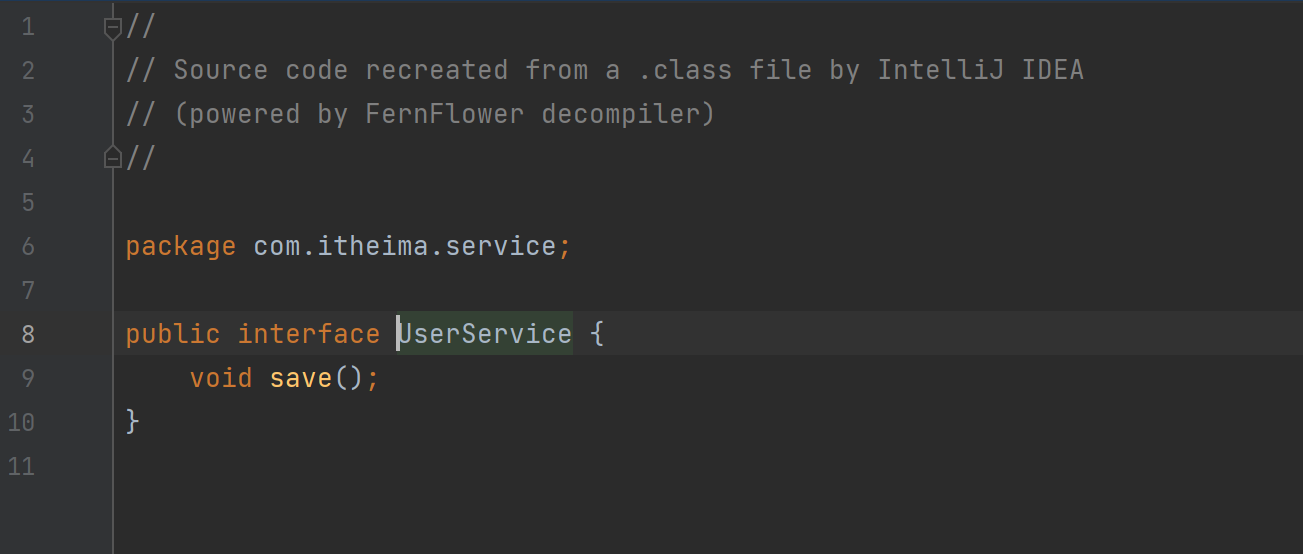


**（4） 在chapter07项目的com.itheima包下创建dao包，在dao包下创建UserDao接口作为数据访问层接口，并在UserDao接口中声明save()方法，用于查询User实体的对象信息。**

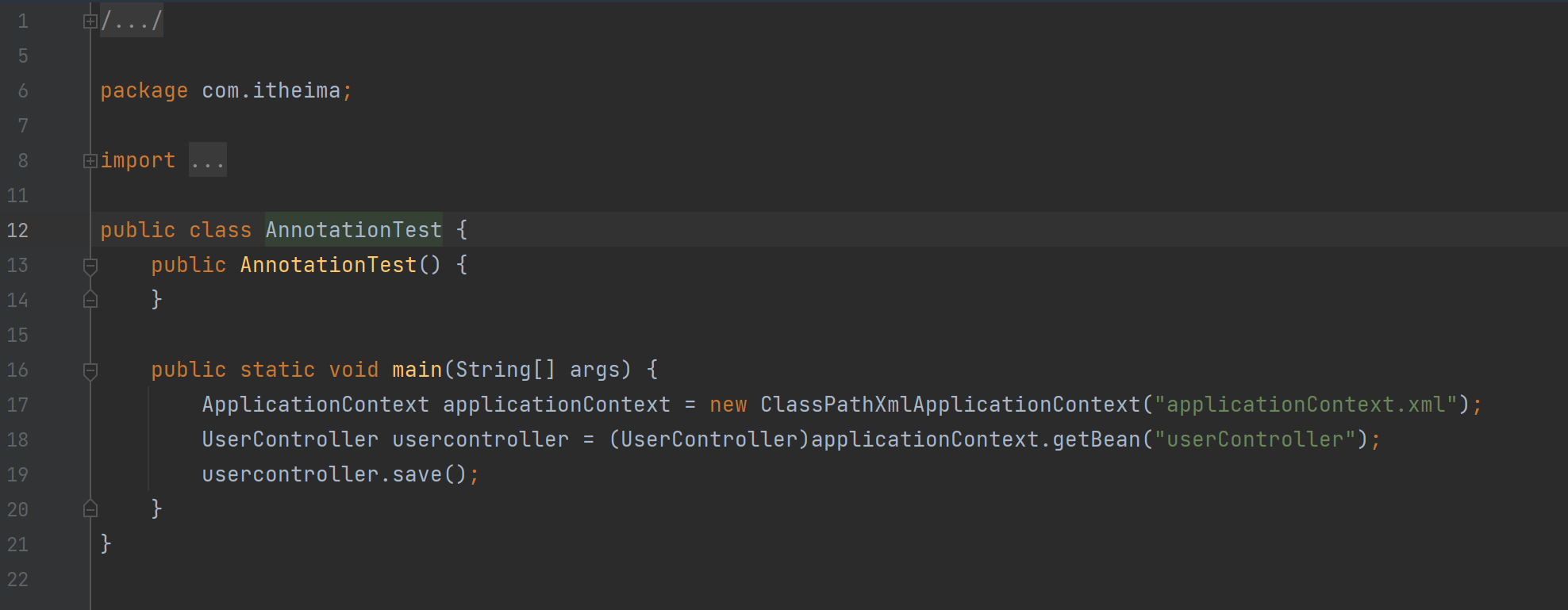
**（5） chapter07项目的com.itheima.dao包下创建UserDaoImpl作为UserDao的实现类，并在UserDaoImpl类中实现UserDao接口中的save()方法。**

**（6）在chapter07项目的com.itheima包下新建service包，在service包下创建UserService接口作为业务逻辑层接口，并在UserService接口中定义save()方法。**

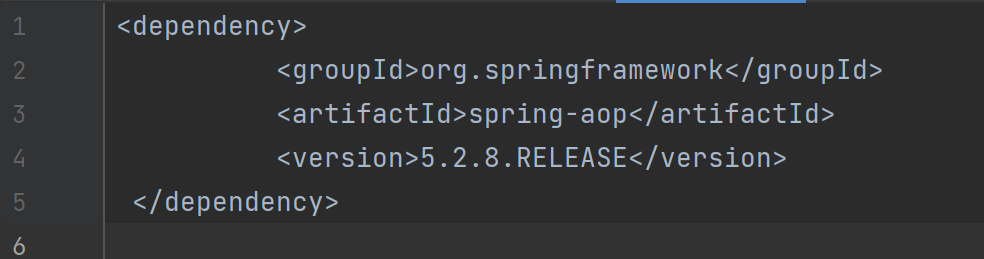
**（7） 在chapter07项目的com.itheima.service包下创建UserServiceImpl作为UserService的实现类，并在UserServiceImpl类中实现UserService接口中的save()方法。**

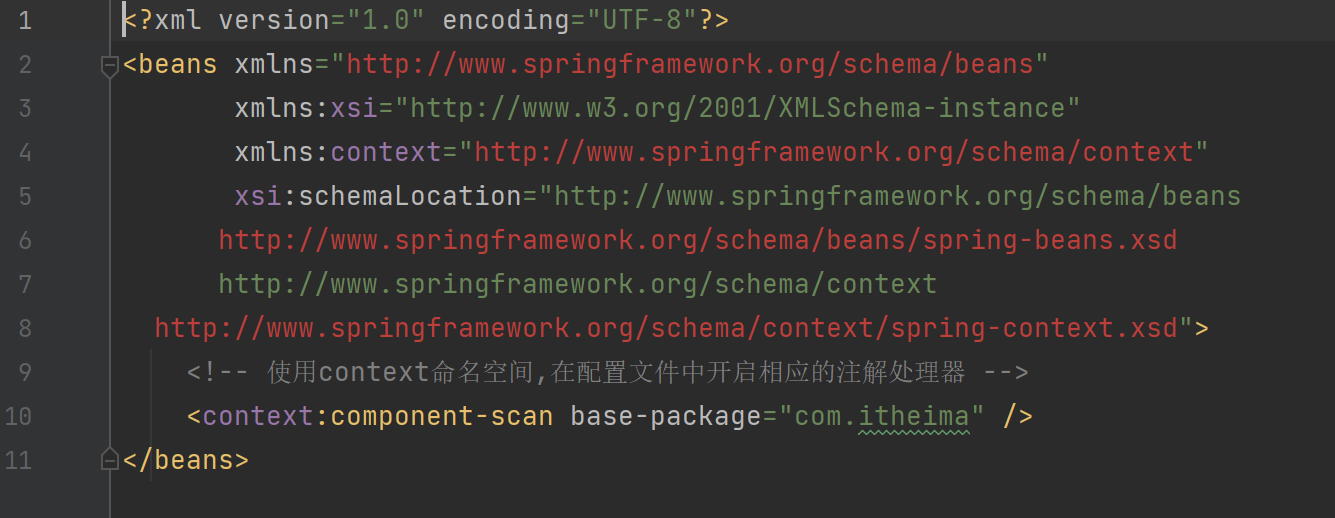


**（8）在chapter07项目的com.itheima的包下新建controller包，在该包下创建UserController类作为控制层。**

**（9）在chapter07项目的com.itheima的包下创建测试类AnnotationTest，在该类中编写测试代码，通过Spring容器加载配置文件并获取UserController实例，然后调用实例中的save()方法。**

**Bean的生命周期**

**（1）在chapter07项目的com.itheima包下创建Student类，在类中定义id和name字段，并使用@PostConstruct指定初始化方法，使用@PreDestroy指定Bean销毁前的方法。**

**（2）在chapter07项目的src\main\resources目录下创建applicationStudent.xml，在该文件中引入Context约束并启动Bean的自动扫描功能。**

**（3）在chapter07项目的com.itheima的包下创建测试类StudentTest，在该类中编写测试代码，通过Spring容器加载配置文件并获取Student实例**