



黑马程序员

《Java EE 企业级应用开发教程 （SSM）（第2版）》

教学设计

课程名称： Java EE 企业级应用开发教程

授课年级： XXXX 年级

授课学期： 第 X 学期

教师姓名： 某某老师

2021年6月



课题名称	第 7 章 Spring 中的 Bean 的管理	计划课时	6 课时
教学引入	第 6 章详细讲解了控制反转和依赖注入，控制反转和依赖注入作为 Spring 核心机制，改变了传统编程习惯，对组件的实例化不再由应用程序完成，转而交由 Spring 容器完成，从而将组件之间的依赖关系进行了解耦。控制反转和依赖注入都是通过 Bean 实现的，Bean 是注册到 Spring 容器中的 Java 类。Bean 由 Spring 进行管理，本章将针对 Bean 的管理进行详细讲解。		
教学目标	<ul style="list-style-type: none"> ● 使学生了解 Spring IoC 容器的原理 ● 使学生掌握 Bean 标签及其属性的使用 ● 使学生熟悉 Bean 的实例化 ● 使学生掌握 Bean 的作用域 ● 使学生掌握 Bean 的装配方式 ● 使学生熟悉 Bean 的生命周期 		
教学重点	<ul style="list-style-type: none"> ● Bean 的配置 ● 基于 XML 的装配 ● 基于注解的装配 		
教学难点	无		
教学方式	课堂教学以 PPT 讲授为主，并结合多媒体进行教学		
教学过程	<p style="text-align: center;">第一课时</p> <p style="text-align: center;">(BeanFactory 接口、ApplicationContext 接口、Bean 的配置)</p> <p>一、创设情景，导入新课</p> <p>Spring 框架的主要功能是通过 Spring 容器实现的，而 Spring 容器是通过提供相应 API 管理开发所需要的各种 Bean。在 Spring 容器的 API 中，最常用的是 BeanFactory 和 ApplicationContext 两个接口。本节课，将对 BeanFactory 和 ApplicationContext 两个接口等内容进行详细讲解。</p> <p>二、新课讲解</p> <p>知识点 1-BeanFactory 接口</p> <p>教师通过多媒体演示 PPT 内容讲解 BeanFactory 接口。</p> <p>BeanFactory 是 Spring 容器最基本的接口，它的实现机制采用的是 Java 经典的工厂模式，BeanFactory 接口提供了创建和管理 Bean 的方法。</p> <p>接下来介绍 BeanFactory 接口的常用方法。</p> <p>有了接口方法，还需要接口的实现类。Spring 提供了几个 BeanFactory 接口的实现类，其中最常用的是 XmlBeanFactory，它可以读取 XML 文件并根据 XML 文件中的配置信息生成 BeanFactory 接口的实例，BeanFactory 接口的实例用于管理 Bean。</p> <p>知识点 2-ApplicationContext 接口</p> <p>教师讲解 ApplicationContext 接口。</p> <p>ApplicationContext 接口建立在 BeanFactory 接口的基础之上，它丰富了</p>		



BeanFactory 接口的特性，例如，添加了对国际化、资源访问、事件传播等方面的支持。

ApplicationContext 接口可以为单例的 Bean 实行预初始化，并根据 <property> 元素执行 setter 方法，单例的 Bean 可以直接使用，提升了程序获取 Bean 实例的性能。

接下来介绍常用的 ApplicationContext 接口的实现类。

知识点 3-Bean 的配置

教师讲解 Bean 的配置。

Spring 容器支持 XML 和 Properties 两种格式的配置文件，在实际开发中，最常用的是 XML 格式的配置文件。XML 是标准的数据传输和存储格式，方便查看和操作数据。在 Spring 中，XML 配置文件的根元素是 <beans>，<beans> 元素包含 <bean> 子元素，每个 <bean> 子元素可以定义一个 Bean，通过 <bean> 元素将 Bean 注册到 Spring 容器中。

接下来介绍 <bean> 元素的常用属性以及 <bean> 元素的常用子元素。

在 XML 配置文件中，一个普通的 Bean 通常只需定义 id(或者 name) 和 class 两个属性，可以通过具体的代码案例进行介绍。

三、归纳总结

教师回顾本节课所讲的内容，可以通过提问的方式引导学生解答问题并给予指导。

四、布置作业

教师通过高校教辅平台 (<http://tch.ityxb.com>) 布置本节课作业以及下节课的预习作业。

第二课时

(构造方法实例化、静态工厂实例化、实例工厂实例化)

一、复习巩固

教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。

二、通过温故知新的方式导入新课

上节课，我们学习了 Spring 容器的 API 中的两个重要接口 BeanFactory、ApplicationContext 接口以及对 Bean 的配置，相信同学们已经对 Spring IOC 容器有了简单的理解。我们知道，Java 是一门面向对象的语言，要想使用某个对象，必须先实例化这个对象。同样，在 Spring 容器中，要想使用某个 Bean，先要实例化 Bean。接下来我们学习 Bean 的实例化的方式。

三、新课讲解

知识点 1-构造方法实例化

教师讲解构造方法实例化。

构造方法实例化是指 Spring 容器通过 Bean 对应类中默认的非参构造方法来实例化 Bean。

下面通过一个案例演示 Spring 容器如何通过构造方法实例化 Bean。

(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.i



theima 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。

知识点 2-静态工厂实例化

教师讲解静态工厂实例化。

使用静态工厂方式实例化 Bean 时，要求开发者定义一个静态工厂的类，用静态工厂类中的方法创建 Bean 的实例，此时，Bean 配置文件中的 class 属性指定的不再是 Bean 实例的实现类，而是静态工厂类。此外，还需要使用 <bean> 元素的 factory-method 属性指定所定义的静态工厂方法。

下面通过一个案例演示如何使用静态工厂方式实例化 Bean。

- (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。

知识点 3-实例工厂实例化

教师讲解实例工厂实例化。

实例工厂实例化 Bean 就是直接创建 Bean 实例。在 XML 配置文件中，不使用 class 属性直接指向 Bean 实例所属的类，而是通过 factory-bean 属性指向为 Bean 配置的实例工厂，并且使用 factory-method 属性指定要调用的实例工厂中的方法。

下面通过一个案例演示如何使用实例工厂方式实例化 Bean。

- (1) 在 chapter07 项目的 com.itheima 包中创建 Bean3 类，该类与 Bean1 类一样，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。

四、归纳总结

教师回顾本节课所讲的内容，可以通过提问的方式引导学生解答问题并给予指导。



五、布置作业

教师通过高校教辅平台 (<http://tch.ityxb.com>) 布置本节课作业以及下节课的预习作业。

第三课时

(singleton 作用域、prototype 作用域、基于 XML 的装配)

一、复习巩固

教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。

二、通过直接导入的方式导入新课

前面我们已经掌握了 Bean 的实例化方式，接下来学习 Bean 的作用域以及 Bean 的装配方式等。

四、新课讲解

知识点 1-singleton 作用域

教师讲解 singleton 作用域。

singleton 是 Spring 容器默认的作用域，当 Bean 的作用域为 singleton 时，Spring 容器只为 Bean 创建一个实例，该实例可以重复使用。Spring 容器管理着 Bean 的生命周期，包括 Bean 的创建、初始化、销毁。因为创建和销毁 Bean 实例会带来一定的系统开销，所以 singleton 作用域可以避免反复创建和销毁实例造成的资源消耗。

下面将通过案例的方式演示 Spring 容器中 singleton 作用域的使用。

- (1) 修改文件 applicationBean1.xml，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。

知识点 2-prototype 作用域

教师讲解 prototype 作用域。

prototype 作用域的 Bean，每次对 Bean 请求时都会创建一个新的 Bean 实例，Spring 容器只负责创建 Bean 实例而不再管理其生命周期。

下面通过案例演示 prototype 作用域的使用，修改配置文件 applicationBean1.xml，将 id 为 bean1 的作用域设置为 prototype。

知识点 3-基于 XML 的装配

教师讲解基于 XML 的装配。

基于 XML 的装配就是读取 XML 配置文件中的信息完成依赖注入，Spring 容器提供了两种基于 XML 的装配方式，属性 setter 方法注入和构造方法注入。下面分别对这两种装配方式进行介绍。

1. 属性 setter 方法注入

在 Spring 实例化 Bean 的过程中，Spring 首先会调用 Bean 的默认构造方法来实例化 Bean 对象，然后通过反射的方式调用 setter 方法来注入属性值。因此，属性 setter 方法注入要求一个 Bean 必须满足以下两点要求。

- Bean 类必须提供一个默认的无参构造方法。



- Bean 类必须为需要注入的属性提供对应的 setter 方法。

2. 构造方法注入

使用构造方法注入时，在配置文件里，需要使用<bean>元素的子元素<constructor-arg>来定义构造方法的参数，可以使用其 value 属性（或子元素）来设置该参数的值。

第四课时

（基于注解的装配、自动装配、Bean 的生命周期）

一、复习巩固

教师通过上节课作业的完成情况，对学生吸收不好的知识点进行再次巩固讲解。

二、通过直接导入的方式导入新课

掌握了 Bean 的作用域以及 Bean 的基于 XML 的装配方式后，接下来学习 Bean 的其他装配方式和 Bean 的生命周期。

五、新课讲解

知识点 1-基于注解的装配

教师讲解基于注解的装配。

在 Spring 中，使用 XML 配置文件可以实现 Bean 的装配工作，但在实际开发中如果 Bean 的数量较多，会导致 XML 配置文件过于臃肿，给后期维护和升级带来一定的困难。为解决此问题，Spring 提供了注解，通过注解也可以实现 Bean 的装配。

再说一下 Spring 的常用注解。

下面通过案例演示如何使用注解来装配 Bean，具体实现步骤如下。

(1) 导入依赖

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

(2) 创建 XML 配置文件

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) 定义实体类

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

(4) 定义 dao 层

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) 实现 dao 层

在 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) 定义 Service 层

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) 实现 Service 层



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) 定义 Controller 层

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) 定义测试类

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。

知识点 2-自动装配

教师讲解自动装配。

除了使用 XML 和注解装配 Bean 外，还有一种常用的装配方式，就是 Bean 自动装配。Spring 的 <bean> 元素中包含一个 autowire 属性，可以通过设置 autowire 属性的值实现 Bean 的自动装配。

接下来讲 autowire 属性的值。

知识点 3-Bean 的生命周期

教师讲解 Bean 的生命周期。

Bean 的生命周期是指 Bean 实例被创建、初始化和销毁的过程。

在 Bean 的两种作用域 singleton 和 prototype 中，Spring 容器对 Bean 的生命周期的管理是不同的。在 singleton 作用域中，Spring 容器可以管理 Bean 的生命周期，控制着 Bean 的创建、初始化和销毁。在 prototype 作用域中，Spring 容器只负责创建 Bean 实例，不会管理其生命周期。

在 Bean 的生命周期中，有两个时间节点尤为重要，这两个时间节点分别是 Bean 实例初始化后和 Bean 实例销毁前，在这两个时间节点通常需要做一些指定操作。因此，常常需要对这两个节点进行监控。监控这两个节点的方式有两种，一种是使用 XML 配置文件，一种是使用注解。

Spring 容器提供了 @PostConstruct 用于监控 Bean 对象初始化节点，提供了 @PreDestroy 用于监控 Bean 对象销毁节点。

下面通过案例演示这两个注解的使用，具体步骤如下。

(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。

(4) 在 IDEA 中启动 StudentTest 类，查看控制台的输出结果。

四、归纳总结

教师回顾本节课所讲的内容，并通过测试题的方式引导学生解答问题并给



	<p>予指导。</p> <p>五、布置作业</p> <p>教师通过高校教辅平台 (http://tch.ityxb.com) 布置本节课作业以及下节课的预习作业。</p> <p style="text-align: center;">第五、六课时（上机练习）</p> <p>上机一：（考察知识点为 BeanFactory 接口、ApplicationContext 接口、Bean 的配置、构造方法实例化、静态工厂实例化、实例工厂实例化、singleton 作用域、prototype 作用域、基于 XML 的装配、基于注解的装配、自动装配、Bean 的生命周期）</p> <p>形式：单独完成</p> <p>题目：</p> <p>编程题：代码演示如何使用注解的方式装配 Bean。</p> <p>要求：</p> <ol style="list-style-type: none">(1) 创建项目名称为 chapter07；(2) 创建配置文件名称为 applicationContext.xml；(3) 创建包名为 com.itheima.XXX，controller 层、service 层、dao 层要分别创建 3 个包；
教学后记	