链滴

Spring 综述

作者：someone33881
原文链接：https://ld246.com/article/1542543097973
来源网站：链滴

关键词：**Spring框架、SpringIOC容器、SpringAOP、SpringMVC、Springboot、设计模式**

一、Spring是什么

Spring框架是一个开源的、轻量级的J2EE开发框架，核心思想是IOC实现松耦合，利用AOP将应用业务逻辑与系统服务分离；

广义上地说，Spring是一系列开源项目的总称，SpringIOC\
pringMVC等只是其中的开源项目而已

二、Spring生态

Spring容器：

专门存放和管理对象及对象间关系的类

https://blog.csdn.net/zorkeaccount/article/details/80818711

SpringIOC：

利用工厂模式+反射实现Spring容器

https://blog.csdn.net/zorkeaccount/article/details/80503941

SpringAOP：

动态代理，即在运行时生成AOP代理对象

https://blog.csdn.net/zorkeaccount/article/details/80517710

SpringMVC:

基于Servlet的一个MVC-web框架，是Spring的一个模块，提供了一种轻度耦合的方式来开发web

用；

** 通过Dispatcher Servlet, ModelAndView 和 View Resolver**，使得开发web应用变得很容易解决的问题领域是网站应用程序或者服务开发——URL路由、Session、模板引擎、静态Web资源等。

Springboot：

由于Spring框架需要进行大量的配置，**Spring Boot引入自动配置的概念，让项目设置变得很容，**降低了项目搭建的复杂度**；

Spring Boot本身并不提供Spring框架的核心特性以及扩展功能，只是用于快速、敏捷地开发新一基于Spring框架的应用程序

Spring设计模式：

简单工厂（静态工厂方法）模式：

其实质就是由一个**工厂类**根据传入的参数，动态决定应该创建哪一个产品类，不属于23种GOF计模式之一；

如**BeanFactory类**,就是根据传入的一个唯一标识来获取bean对象，但传参前还是传参后创建该ean对象视具体情况而定

工厂方法Factory Method模式：

定义一个用于创建对象的**工厂bean接口**，由实现类子类决定实例化哪一个类，也即工厂方法模将一个类的实例化（bean对象生成）延迟到其子类；

如**FactoryBean接口及其各个实现子类**就是典型的工厂方法模式

单例Singleton模式：

保证一个类只有一个实例，并对外提供一个可以访问它的全局访问点；

Spring中**每个bean定义只生成一个对象实例**，有两种模式**饿汉模式**（即类加载时已初始化，缺时默认为该模式，容器启动即实例化容器时，为所有spring配置文件中定义的bean都生成一个实例和**懒汉模式**（即类加载时不初始化，）=>【如可通过静态变量是否初始化实现饿汉和懒汉两种方式】

适配器Adapter模式：

将一个类的接口转换成另外一个接口

Spring中对于aop的处理用到Adapter模式，如因Advisor链需要的是MethodInterceptor对象所以每一个Advisor中的Advice对象都是适配成对应的MethodInterceptor对象

包装器Decortor模式：

动态地给一个对象添加一些额外的职责，增加功能角度而言该模式比生产子类更加灵活；

Spring中用到包装器模式的类如：*Wrapper.java，或者*
ecorator.java

代理Proxy模式：

为其他对象提供一种代理以控制对这个对象的访问，结构上类似于Decortor模式，但Proxy为控

是一种对功能的限制而Decortor是增加职责

Spring中aop的思想就是Proxy模式，比如JdkDynamicAopProxy和Cglib2AopProxy

观察者Observer模式：

定义对象间一种一对多的依赖关系，当一个对象的状态改变时，所有依赖于该对象的其他对象均得到通知并自动更新

Spring中listen的实现，如ApplicationListener

策略Strategy模式：

定义一系列算法，并且将它们一个个封装起来且可以互相替换，使得算法可独立于使用它的客户发生变化

Spring中如SimpleInstantiationStrategy

模板方法Template Method模式：

定义一个操作中算法的骨架，而将一些步骤延迟到子类中；模板方法使得子类可以不改变一个法的结构即可重定义该算法的某些特定步骤，该模式一般是需要继承的（也可以不需要继承的，如JdbTemplate）

Springboot注解：

@SpringBootApplication：

Springboot的入口注解，是多个注解的组合，其中比较重要的是@EnableAutoConfiguration解，即可自动化配置，这是SpringBoot可以方便快捷地新建和启动一个项目的关键

@EnableAutoConfiguration：

该注解比较重要的是导入了可自动化配置导入选择器，@Import({EnableAutoConfigurationImortSelector.class})

有关该springboot注解的详细分析过程可见，springboot自动化配置

三、Spring&SpringMVC&Springboot之间的关系和区别

** Spring**是一个轻量级的控制反转（IOC）和面向切面（AOP）的**容器框架**，通过容器管理Javaean（代替EJB）的配置和声明周期； -> 即**Spring是一个管理bean的容器**，可称之为SpringIOC容器

SpringMVC是一个MVC框架容器，负责controller相关的Bean的管理，且SpringIOC容器的子容；

Springboot本质上就是Spring，只不过做了一些SpringBean的默认配置（**开箱即用、快速启动**）；

通俗地说，就是：**Spring 最初利用“工厂模式”（DI）和“代理模式”（AOP）解耦应用组件。家觉得挺好用，于是按照这种模式搞了一个 MVC框架（一些用Spring 解耦的组件），用开发 web用（ SpringMVC ）。**然后有发现每次开发都写很多样板代码，为了简化工作流程，于是开发出了一**“懒人整合包”（starter）**。

Spring 是一个“引擎”；

Spring MVC 是基于Spring的一个 MVC 框架；

Spring Boot 是基于Spring4的条件注册的一套快速开发整合包。