

00 Spring之旅

C-2 创建: 张林伟, 最后修改: 张林伟 2018-08-27 13:05

为降低Java开发的复杂性，Spring采用 4 中关键策略：

- 基于 POJO 的轻量级和最小侵入性编程

^ 点击展开内容

不会强迫继承框架的类或者实现框架的接口，导致应用和框架绑定。

- 通过依赖注入和面向接口实现松耦合

^ 点击展开内容

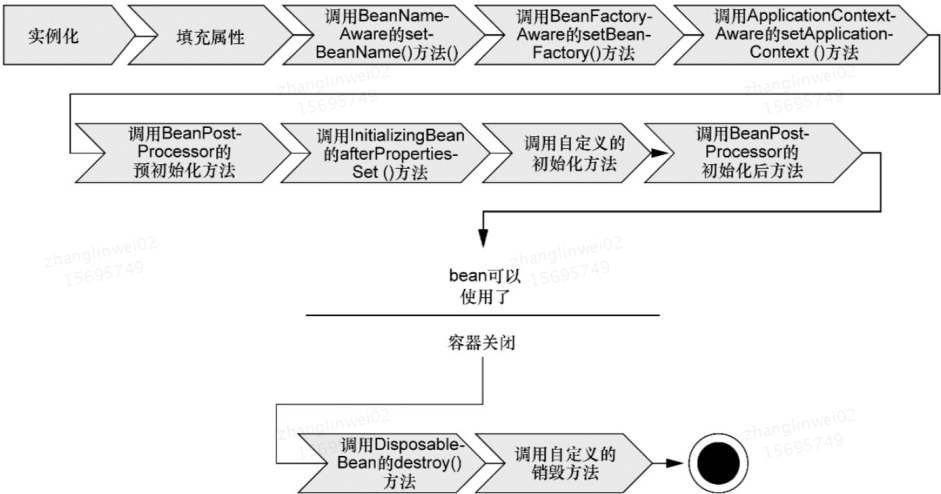
耦合是必须的，但应当被小心谨慎地管理。

面向接口使得对象的依赖关系通过接口串联，而不与具体的实现类耦合在一起。

如果一个对象只通过接口（而不是具体实现或初始化过程）来表明依赖关系，那么这种依赖就能够在对象本身不知情的情况下，用不同的具体实现进行替换。例如测试时使用 mock 实现对依赖进行替换。

- 基于切面和惯例进行声明式编程
- 通过切面和模板减少样板式代码

bean 的生命周期



- 1. Spring对bean进行实例化；
- 2. Spring将值和bean的引用注入到bean对应的属性中；
- 3. 如果bean实现了BeanNameAware接口，Spring将bean的ID传递给setBean-Name () 方法；
- 4. 如果bean实现了BeanFactoryAware接口，Spring将调用setBeanFactory () 方法，将BeanFactory容器实例传入；
- 5. 如果bean实现了ApplicationContextAware接口，Spring将调用setApplicationContext () 方法，将bean所在的应用上下文的引用传入进来；
- 6. 如果bean实现了BeanPostProcessor接口，Spring将调用它们的post-ProcessBeforeInitialization () 方法；
- 7. 如果bean实现了InitializingBean接口，Spring将调用它们的after-PropertiesSet () 方法。类似地，如果bean使用init-method声明了初始化方法，该方法也会被调用；
- 8. 如果bean实现了BeanPostProcessor接口，Spring将调用它们的post-ProcessAfterInitialization () 方法；
- 9. 此时，bean已经准备就绪，可以被应用程序使用了，它们将一直驻留在应用上下文中，直到该应用上下文被销毁；
- 10. 如果bean实现了DisposableBean接口，Spring将调用它的destroy () 接口方法。同样，如果bean使用destroy-method声明了销毁方法，该方法也会被调用。