# 啥是BPP

BBP的全称叫做：BeanPostProcessor，一般我们俗称对象后处理器

简单来说，通过BeanPostProcessor可以对我们的对象进行“加工处理”。

## Bean的生命周期

Spring管理Bean(或者说Bean的生命周期)也是一个常考的知识点，我在秋招也重新整理了一下步骤，因为比较重要，所以还是在这里贴一下吧：

ResouceLoader加载配置信息

BeanDefintionReader解析配置信息，生成一个一个的BeanDefintion

BeanDefintion由BeanDefintionRegistry管理起来

BeanFactoryPostProcessor对配置信息进行加工(也就是处理配置的信息，一般通过PropertyPlaceholderConfigurer来实现)

实例化Bean

如果该Bean配置/实现了InstantiationAwareBean，则调用对应的方法

使用BeanWarpper来完成对象之间的属性配置(依赖)

如果该Bean配置/实现了Aware接口，则调用对应的方法

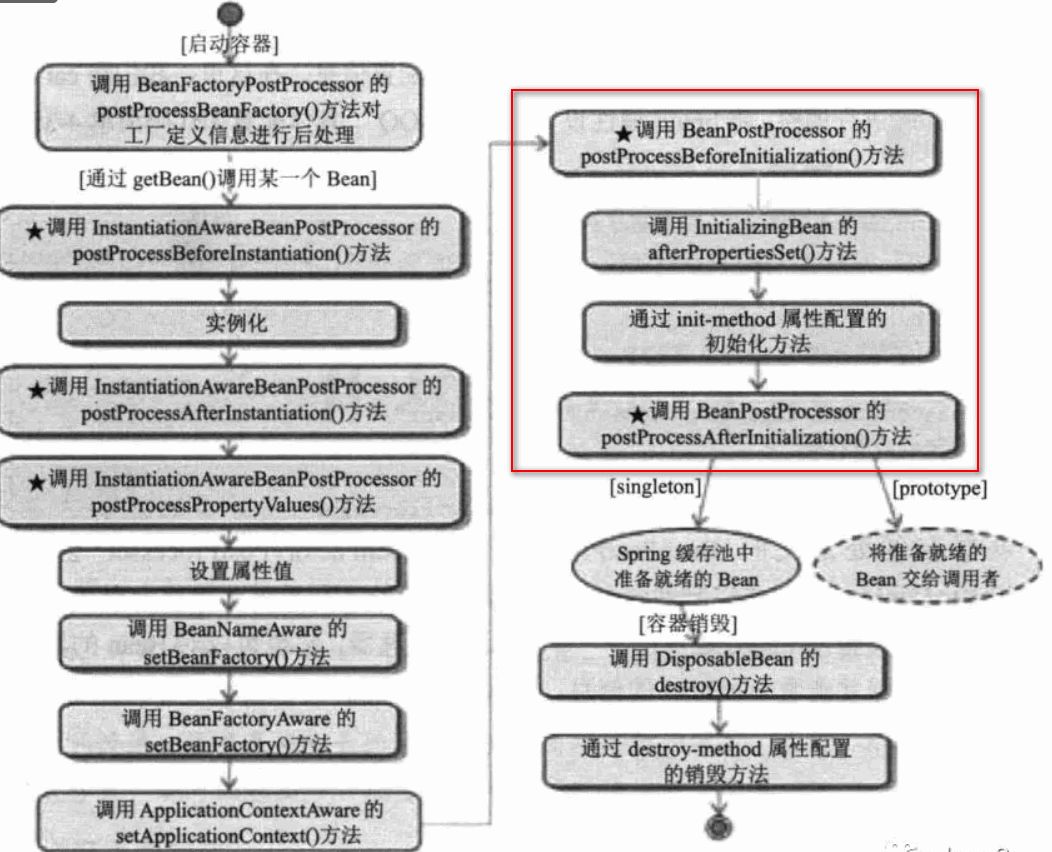
如果该Bean配置了BeanPostProcessor的before方法，则调用

如果该Bean配置了init-method或者实现InstantiationBean，则调用对应的方法

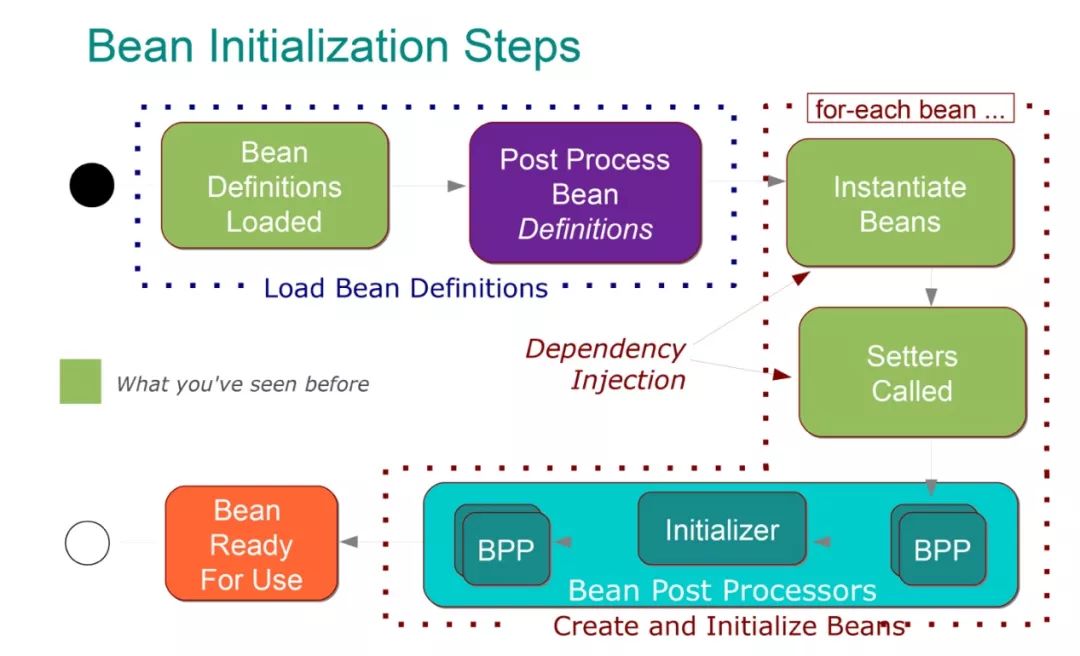
如果该Bean配置了BeanPostProcessor的after方法，则调用

将对象放入到HashMap中

最后如果配置了destroy或者DisposableBean的方法，则执行销毁操作



其中也有关于BPP图片：



## 为什么特意讲BPP？

Spring AOP编程底层通过的是动态代理技术，在调用的时候肯定用的是代理对象。那么Spring是怎么做的呢？

我只需要写一个BPP，在postProcessBeforeInitialization或者postProcessAfterInitialization方法中，对对象进行判断，看他需不需要织入切面逻辑，如果需要，那我就根据这个对象，生成一个代理对象，然后返回这个代理对象，那么最终注入容器的，自然就是代理对象了。

Spring提供了BeanPostProcessor，就是让我们可以对有需要的对象进行“加工处理”啊！

## 认识Spring事务几个重要的接口

在编程式事务中有以下几个重要的了接口：

TransactionDefinition：定义了Spring兼容的事务属性(比如事务隔离级别、事务传播、事务超时、是否只读状态)

TransactionStatus：代表了事务的具体运行状态(获取事务运行状态的信息，也可以通过该接口间接回滚事务等操作)

PlatformTransactionManager：事务管理器接口(定义了一组行为，具体实现交由不同的持久化框架来完成---类比JDBC)

在声明式事务中，除了TransactionStatus和PlatformTransactionManager接口，还有几个重要的接口：

TransactionProxyFactoryBean：生成代理对象

TransactionInterceptor：实现对象的拦截

TransactionAttrubute：事务配置的数据