# Spring 相关面试

## Spring相关面试：



Spring的生命周期

开始方法：doCreate()：

Instantiation -> Populate -> 初始化 -> 销毁

实例化 -> 属性赋值 -> 初始化 -> 销毁

https://www.jianshu.com/p/1dec08d290c1

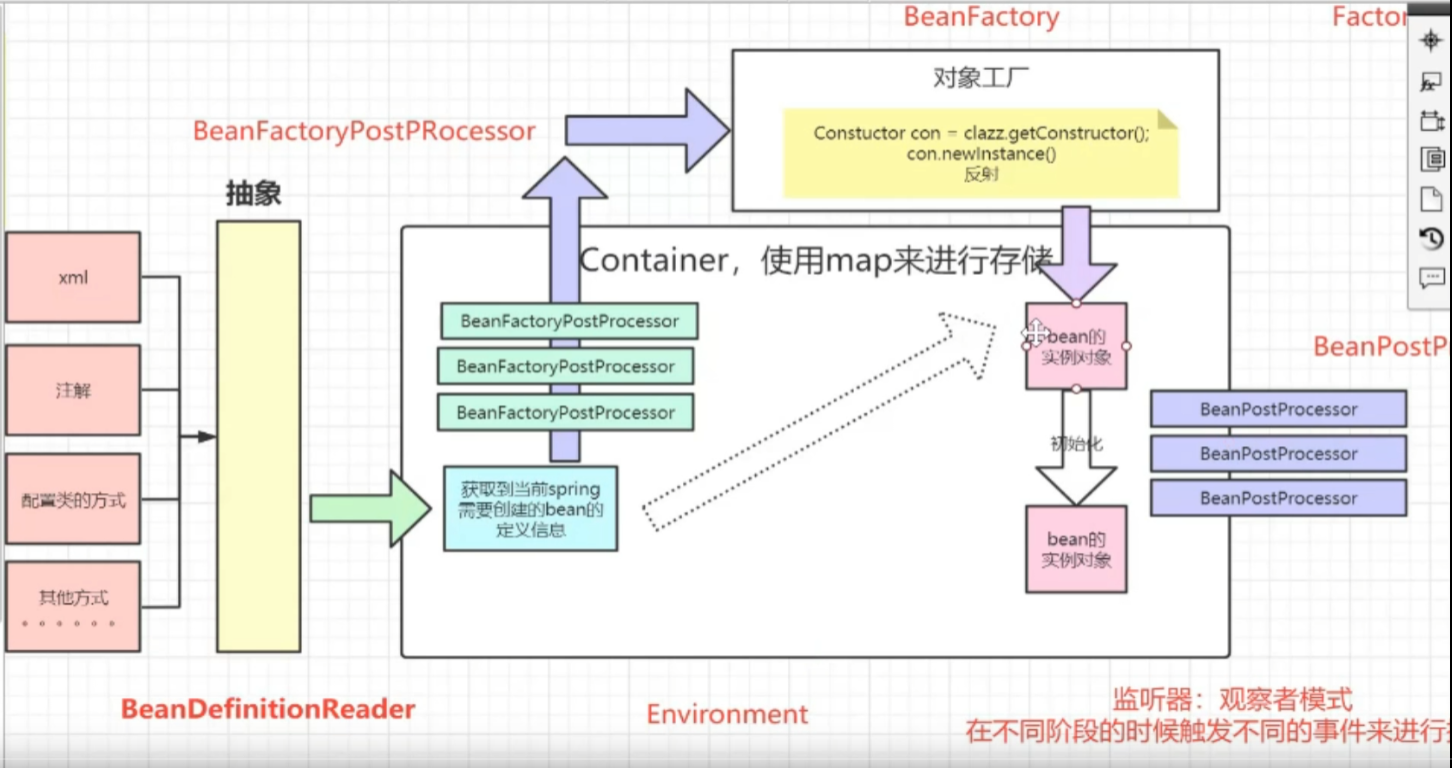
Spring的缓存

https://blog.csdn.net/m0\_43448868/article/details/113578628

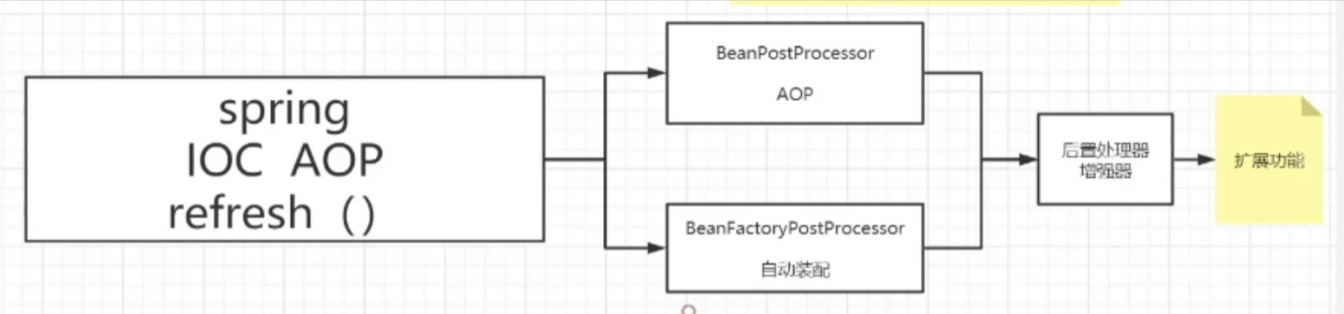
Spring的循环依赖

https://blog.csdn.net/u010853261/article/details/77940767

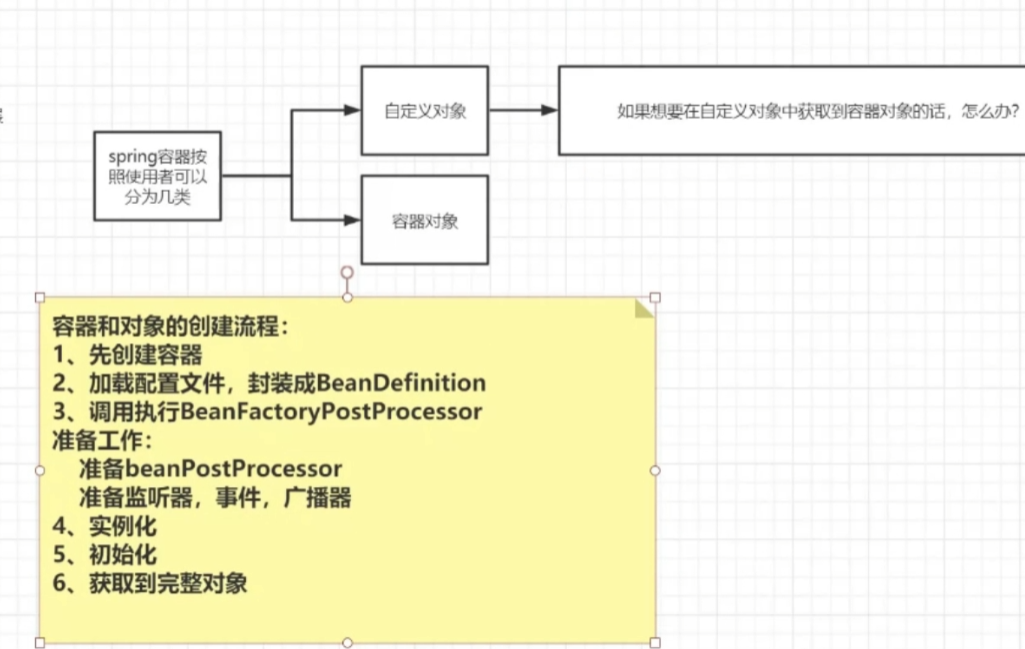
### Spring IOC：

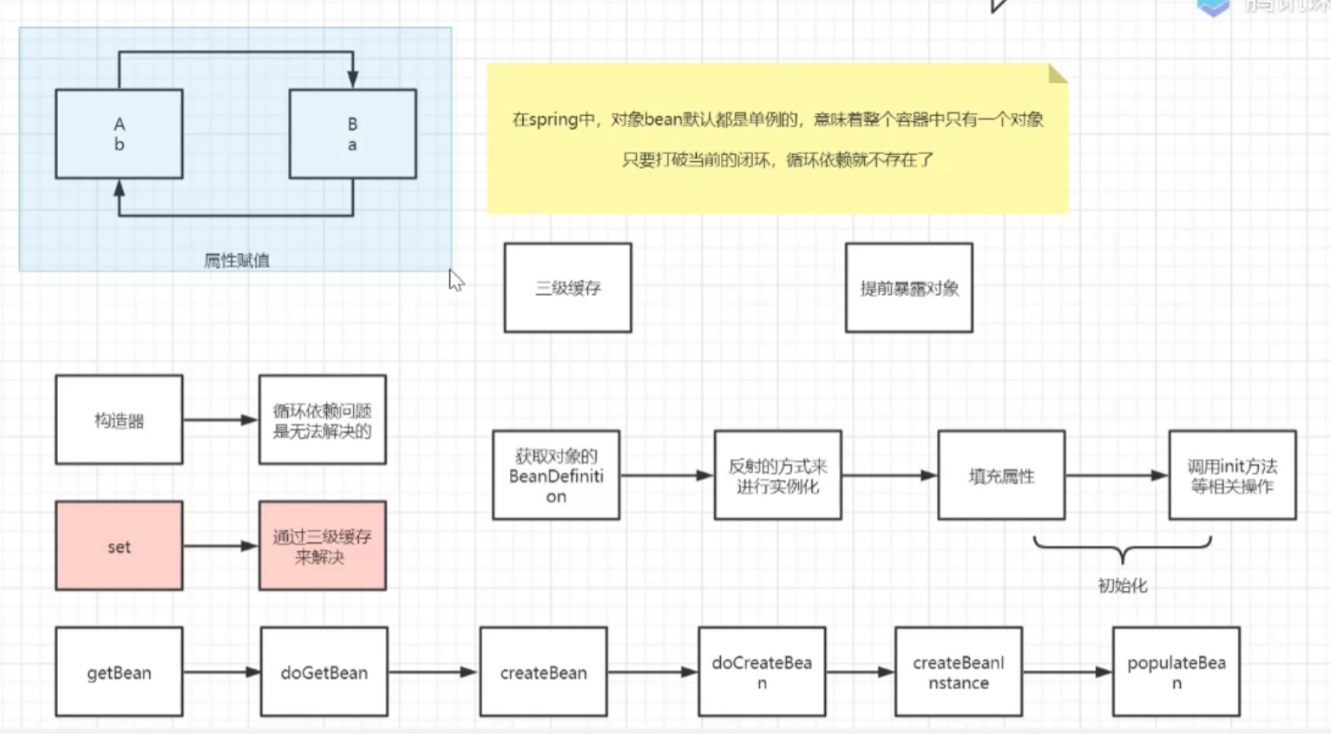


### Spring AOP：



### 核心方法（refresh()）：





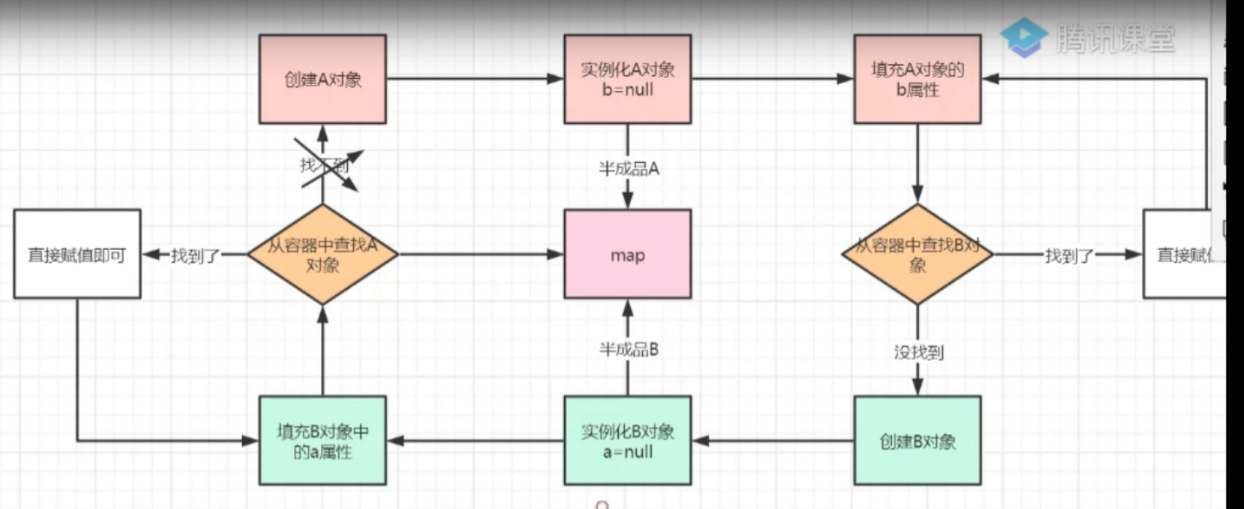
### 创建对象时为什么要有三级缓存

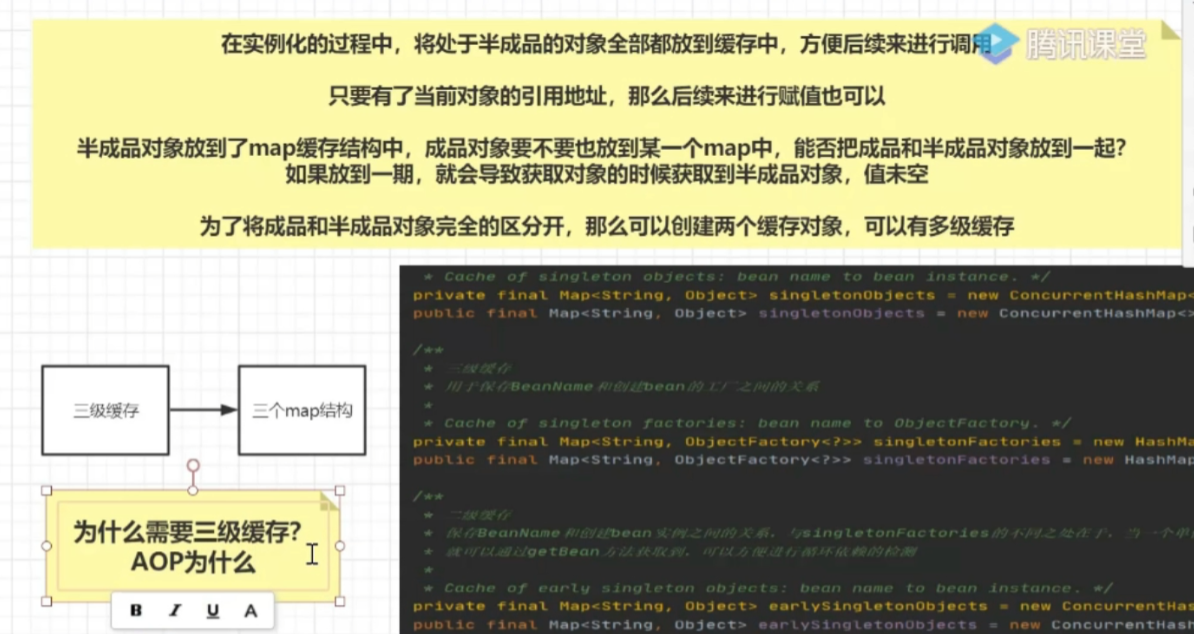
这三级缓存分别指：

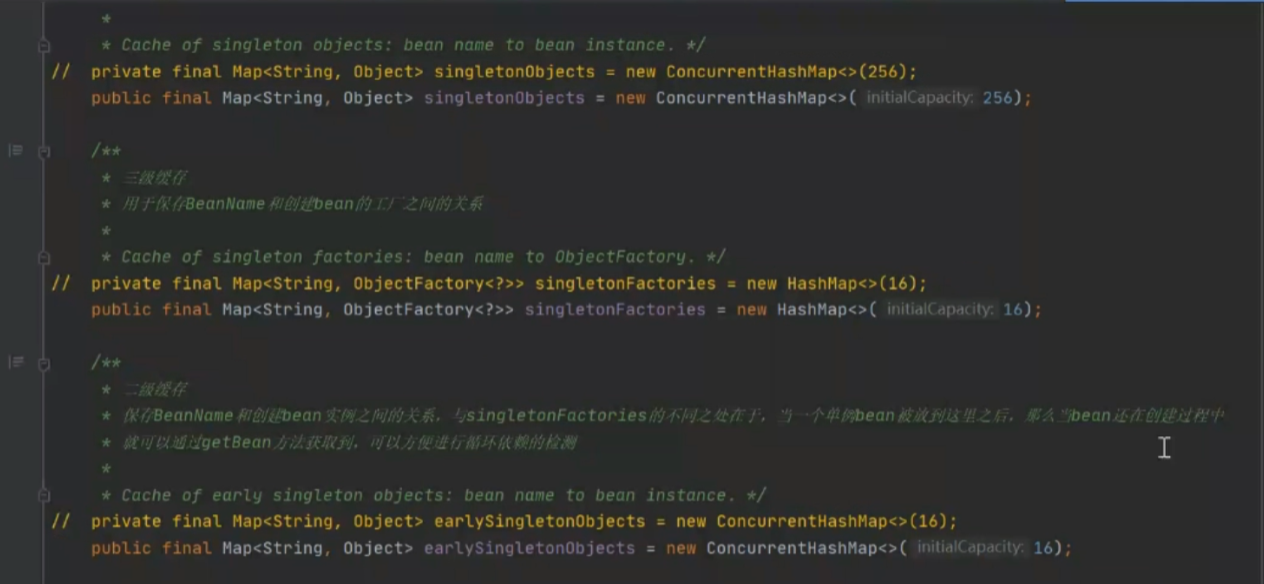
singletonFactories ： 单例对象工厂的cache

earlySingletonObjects ：提前暴光的单例对象的Cache

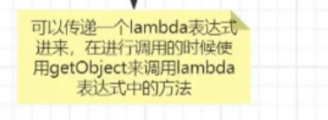
singletonObjects：单例对象的cache

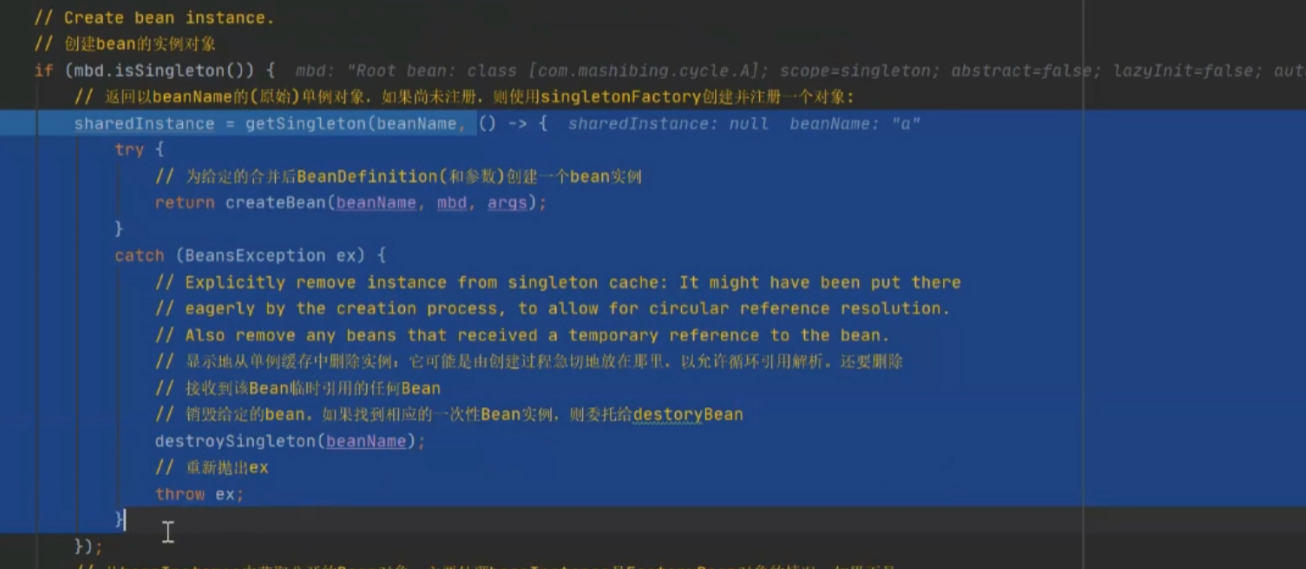


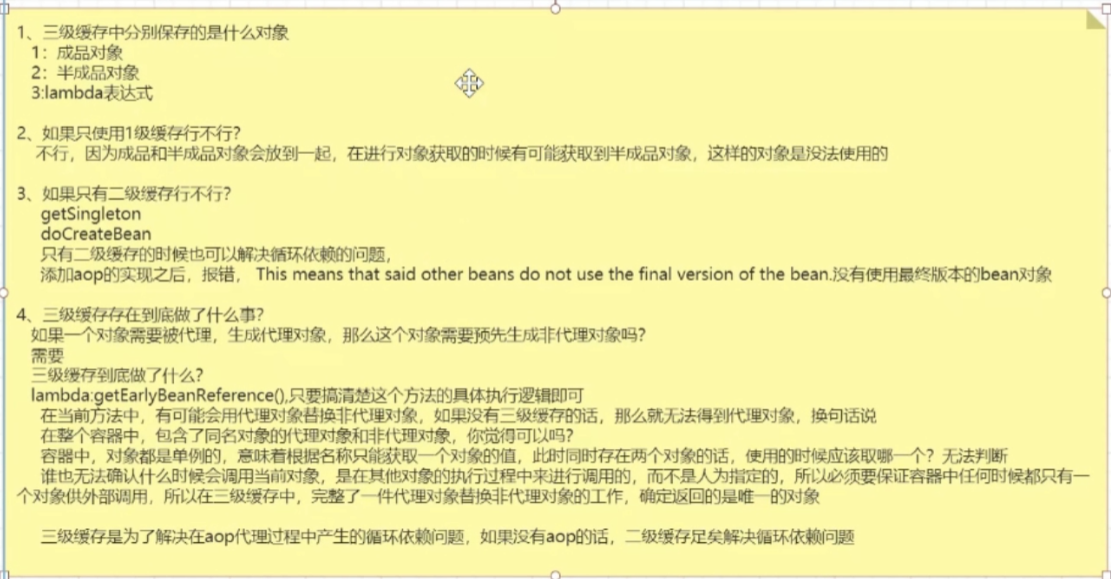


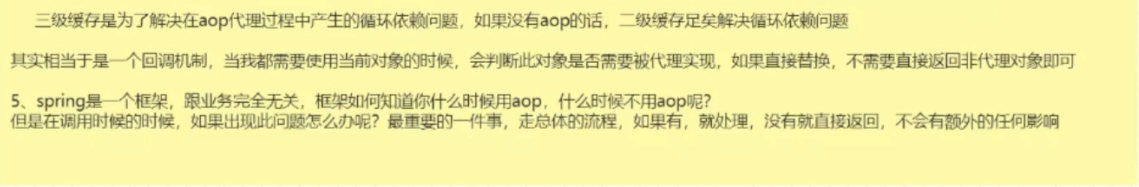


### Spring Bean的初始化：





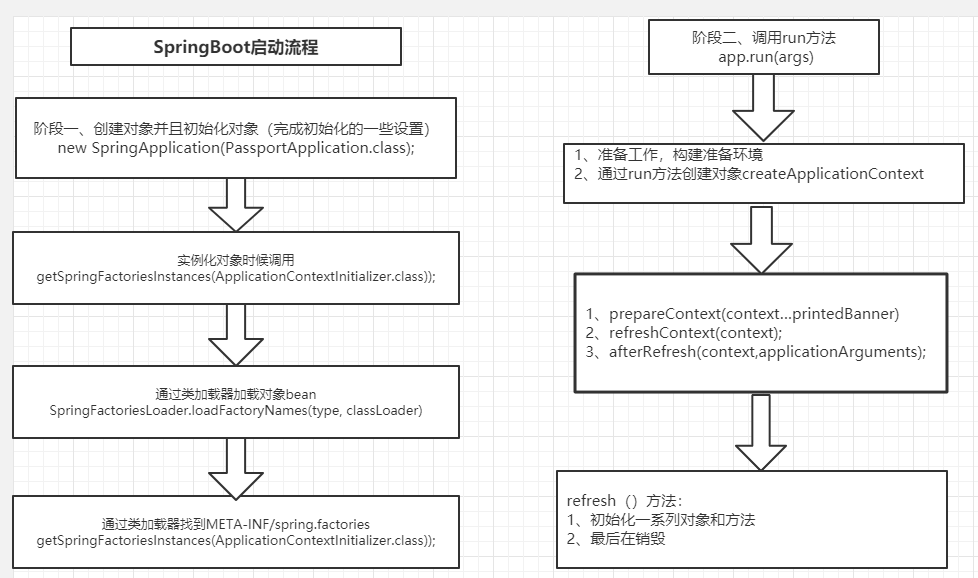




## SpringBoot相关面试：



### SpringBoot启动流程：



1、创建对象SpringApplication（完成初始化的设置）

2、然后通过run方法运行

1、准备工作，构建准备环境

2、prepareContext(context…printedBanner)

2.1、设备环境变量值

2.2、执行某些相关方法

2.3、获取bean工厂

//Add boot specific singleton beans

ConfigurabLeListableBeanFactory beanFactory =context. getBeanFastor();

2.4、加载资源：

Load方法加载资源配置文件

BeanDefinitionLoader loader=

createBeanDefinitionLoader(getBeanDefinitionRegistry(context),sources);

BeanDefinitionLoader：通过xml和注解的方式来注册资源

3、refreshContext(context);

3.1、refresh() 方法：

//\*\*\*调用在上下文中注册为bean的工厂处理器\*\*\*

//Invoke factory processors registered as beans in the context.

invokeBeanFactoryPostProcessors(beanFactory);

3.2、加载注册应用的注解：@Import(AutoConfigurationImportSelector.class)

3.3、自动装配关键方法：processConfigBeanDefinitions

3.4、扫描到注解后: 通过AnnotationConfigutils 开始一些列的加载注解资源以及对注解的处理通过自动注入AutoConfigurationImportSelector来导入注解imports（）方法到selectImports（）

3.5、onRefresh() 创建webServer方法：createWebServer()

4、afterRefresh(context,applicationArguments);

### SpringBoot源码解读：

### SpringBoot自动装配：

**自动装配是从刷新容器开始的，**

**首先要执行到refresh()方法里面去，**

**然后执行invokeBeanFactoryPostProcessos(beanFactory); 自动装配的入口**

**后面就是一系列的解析和加载包括注解该实例化的实例化该解析的解析**

#### 什么是自动装配？：

在之前的项目开发中我们使用SSM框架，进行项目开发的时候需要进行整合，当需要第三方包的时候依赖需要添加额外jar，还要增加很。多额外的配置，使用springboot之后简化了很多配置，当需要什么组件的时候，直接使用starter引入即可，不需要人为再进行整合