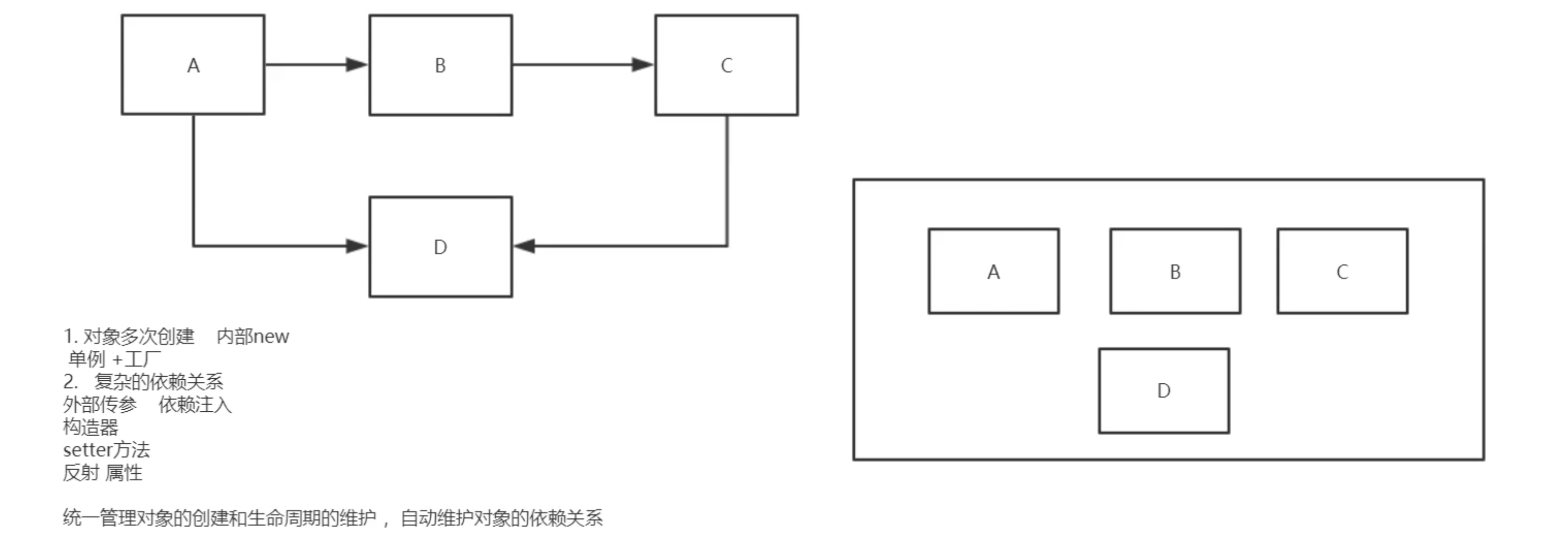
## 零：前言：为什么产生了IOC容器：

我们没有IOC容器的话：

①：在我们比较大的项目中，多个类存在依赖关系，A中需要创建B，B中需要创建C，D中也需要创建B，那么我们项目一旦初始化，就会创建很多个B，造成了内存的浪费! 应对这种问题我们采取的是单例设计模式以及工厂设计模式，让我们创建出来的B只有一个！并且不是通过new创建的！

②：比较复杂的项目会有复杂的依赖关系，我们在修改属性的时候会非常麻烦

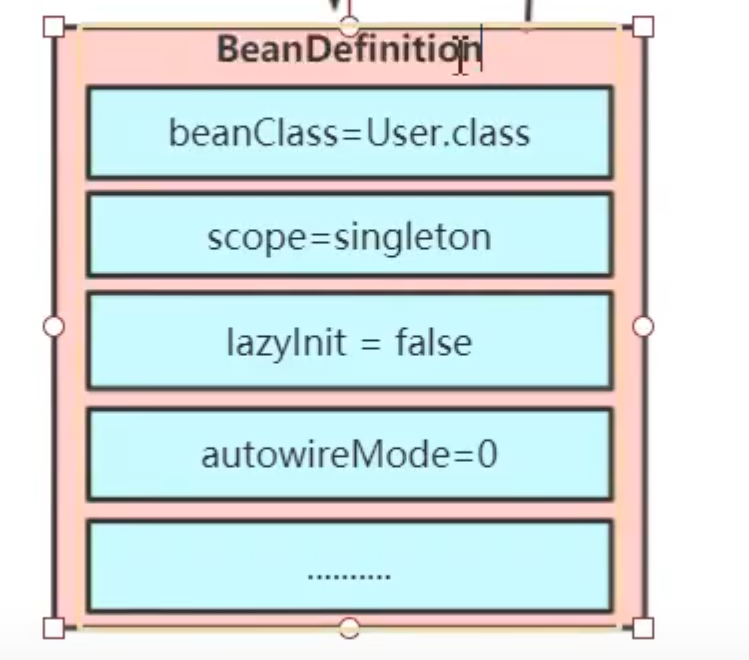
Ioc可以使用外部传参，只需要更改一个地方，所有的对象属性全部变化！



## 一：beandefination和他的注册思路

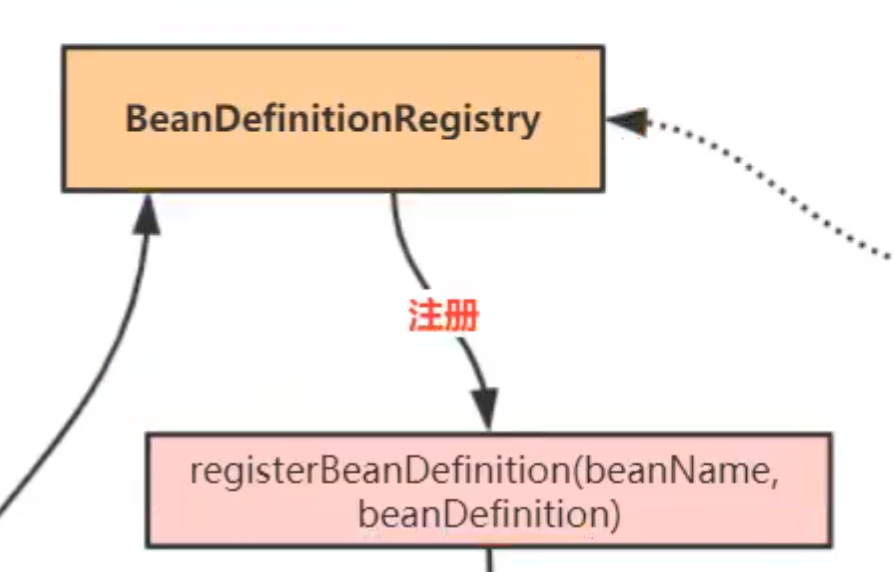
我们注册ioc容器的最关键的一个类就是beanDefination

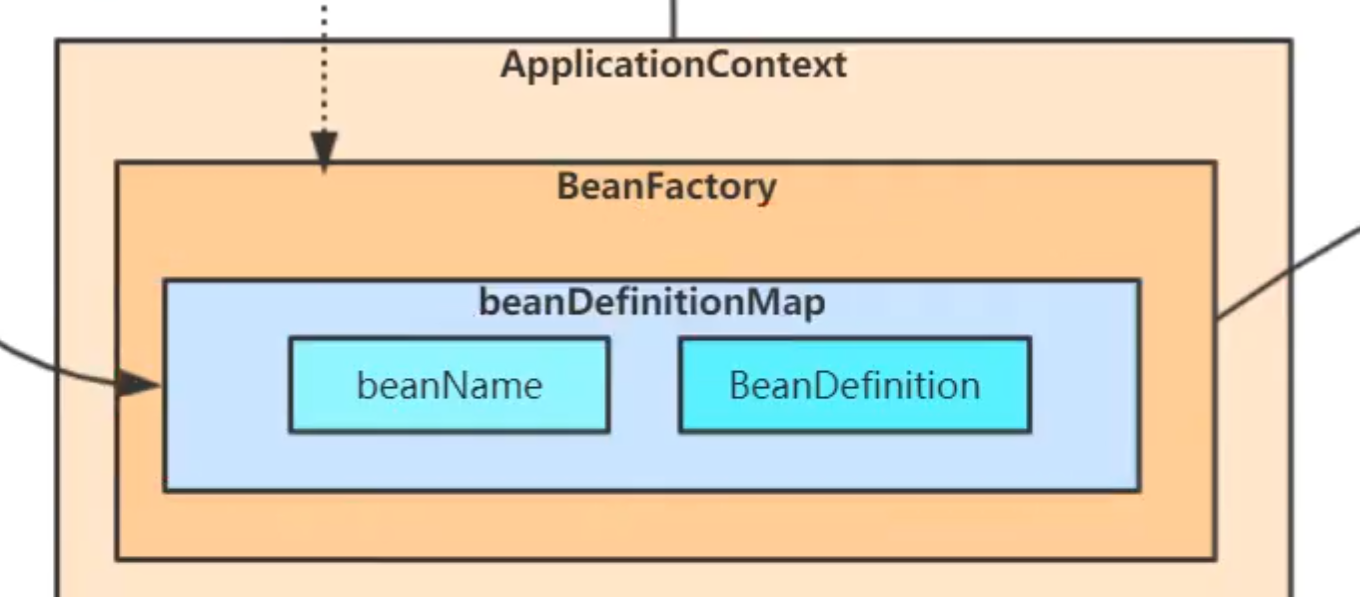
它内部封装了我们要创建的类对象的class字节码文件（仅仅是class，不是对象，具体对象依然需要后面进行反射创建！），同时封装了我们对应的类的各种特点：例如单例多例，懒加载和立即加载等等！



那么beanDefination又是放在哪里的呢，其实就是一个beanDefinationMap集合中，这个集合的键默认值就是我们要注册对象的名称小写第一个字母，值就是beanDefination。

这个将beanDefination放到beanDefinationMap中的过程就是加载过程，由我们beanDefinationRegistery类的registerBeanDefinition方法完成注册！



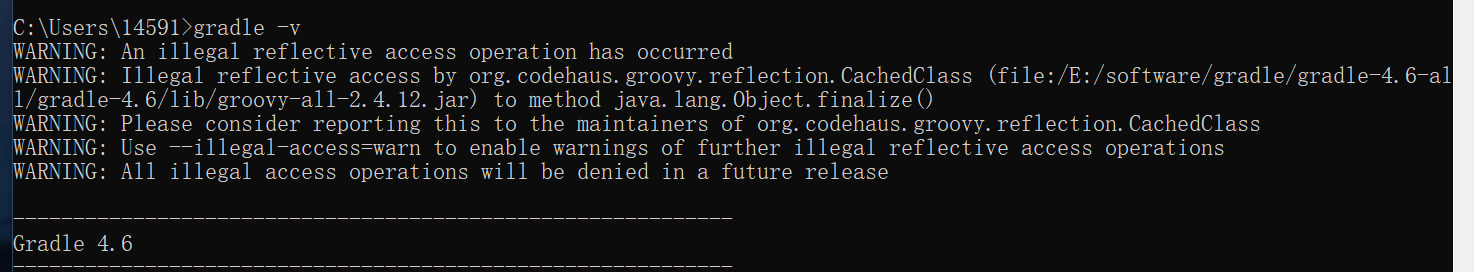


## 二：beandefination和他的注册过程源码

首先准备好我们的环境：

首先安装gradle（类似maven），是一种项目自动化构建的开源工具！

下载gradle，配置好环境变量：



在我们的spring源码路径行执行如下命令：



出现如下build successful说明已经编译成功：

