### Spring框架

Spring是分层的 Java SE/EE应用 full-stack(全栈式) **轻量级**开源框架。

全栈式框架:就是指它对各种主流技术和框架都进行了整合，同时对三层架构都提供了解决方案。

轻量级:所谓轻量级以及重量级的划分主要依据就是看它使用了多少服务、启动时又需要加载多少资源以及耦合度情况等。

**Spring框架以IOC（控制反转）和AOP（面向切面编程）为内核**!

IOC：控制反转，就是把对象的创建权交给spring框架，就是我们不用自己创建对象，在spring框架被启动的时候就帮我们创建好了要使用的对象；所谓控制就是框架帮我们创建以及销毁对象，所谓反转就是对象控制权由开发者在类中手动控制反转到由spring容器控制

AOP：面向切面编程，在不修改源码的情况下，对方法进行增强。例如：动态代理

DI：依赖注入，主要就是通过spring框架来维护持久层与应用层的关系，避免我们自己手动去在应用层中调用持久层

自定义IOC容器中的执行流程:

首先我们要先明确，是由service层代码去调用dao层代码的，而为了避免使用new创建dao层的实现类造成编译期依赖以及使用Class.forName反射机制造成的硬编码问题；我们引入了配置文件+反射的方式，创建了一个工厂类，使用静态代码块的方式，将配置文件中的多个不同的bean，通过它各自的id和class属性放到map表中（流程是先读取配置文件，接下来解析该配置文件，然后编写xpath表达式，通过document.selectNodes(xpath)的方式获取到所有的bean对象，然后一个一个创建对应的反射对象，放入到map中(key为bean对象的id值，value是反射对象)）；然后创建一个静态的方法将map抛出，由service层的实现类获取到，并进行操作。