# ****=== 课程目录 ===****

### ****第一章：Spring源码构建与BeanDefinition源码剖析****

01\_Spring源码的准备工作：环境搭建

02\_Spring源码的准备工作：Spring的简单使用和原理分析

03\_Spring初级容器初始化：Resource到底是个什么玩意儿？

04\_Spring初级容器初始化：忽略指定接口自动装配功能

05\_Spring初级容器初始化：基础数据的准备

06\_Spring初级容器初始化：加载XML的Document

07\_Spring初级容器初始化：XML文件的校验模式

08\_Spring初级容器初始化：获取声明文件和校验类型

09\_Spring初级容器初始化：寻找标签解析的入口

10\_Spring初级容器初始化：bean标签的初步解析

11\_Spring初级容器初始化：[BeanDefinition](https://so.csdn.net/so/search?q=BeanDefinition&spm=1001.2101.3001.7020" \t "https://blog.csdn.net/qq_42046105/article/details/_blank)是什么呢？

12\_Spring初级容器初始化：bean的各种子标签解析

13\_Spring初级容器初始化：BeanDefinition是如何注册到Spring容器的呢？

### **第二章：Spring容器ApplicationContext和BeanFactory**

14\_Spring高级容器初始化：初探容器ApplicationContext初始化

15\_Spring高级容器初始化：初始化环境和容器BeanFactory

16\_Spring高级容器初始化：初步添加扩展功能点

17\_Spring高级容器初始化：BeanFactoryPostProcessor是什么呢？

18\_Spring高级容器初始化：BeanFactoryPostProcessor是如何执行的呢？

19\_Spring高级容器初始化：BeanPostProcessor是如何注册的呢？

20\_Spring高级容器初始化：Spring是如何基于事件驱动的呢？

21\_Spring高级容器初始化：初始化非延迟加载单例bean

### **第三章：bean加载流程与三级缓存解决循环依赖问题**

22\_bean的加载：转换并获取bean的名称

23\_bean的加载：如何通过三级缓存来解决循环依赖（上）

24\_bean的加载：如何通过三级缓存来解决循环依赖（下）

25\_bean的加载：如何通过FactoryBean来实例化bean？

26\_bean的加载：寻找实例化bean的入口

27\_bean的加载：Spring默认是如何实例化bean的？

28\_bean的加载：通过反射来实例化bean

29\_bean的加载：为刚刚实例化的bean填充属性（上）

30\_bean的加载：为刚刚实例化的bean填充属性（下）

31\_bean的加载：看下bean是如何进行初始化的

32\_面试题：IOC和DI之间到底是什么关系？

### **第四章：Spring依赖注入源码剖析**

33\_Spring注解源码解析：注解的本质是什么？

34\_Spring注解源码解析：动手来模拟下@Component注解的功能

35\_Spring注解源码解析：@Component

36\_Spring注解源码解析：@Configuration和@Bean（上）

37\_Spring注解源码解析：@Configuration和@Bean（下）

38\_Spring注解源码解析：@Autowired

39\_Spring注解源码解析：@PostConstruct和@PreDestroy

### **第五章：Spring AOP底层源码剖析**

40\_Spring AOP简介：一个例子告诉你AOP可以解决哪些问题

41\_Spring AOP简介：静态代理、jdk动态代理、cglib动态代理傻傻分不清？

42\_AOP代理的创建：虚晃一枪，实例化前竟然也能创建AOP代理？

43\_AOP代理的创建：真相大白，初始化后完成了AOP代理的创建

44\_AOP代理的创建：@Aspect注解的切面类是怎么找到的？

45\_AOP代理的创建：构建Advisor的前提，怎么获取切面类中声明的增强方法？

46\_AOP代理的创建：增强方法是怎么一步一步构建为Advisor的？

47\_AOP代理的创建：来看下目标类与切点表达式的匹配过程

48\_AOP代理的创建：aspectj是怎么来完成方法级别精准匹配的？

49\_AOP代理的创建：负责创建代理的ProxyFactory是什么？

50\_AOP代理的创建：Spring是如何选择jdk代理和cglib代理的？

### **第六章：Spring AOP中JDK动态代理源码剖析**

51\_JDK代理的创建：深入底层，来看下jdk代理对象的创建过程

52\_JDK代理的执行：调用$Proxy代理对象方法就回调到invoke()是啥原理？

53\_JDK代理的执行：来初步看下invoke()方法的处理流程

54\_JDK代理的执行：invoke()方法是怎么获取到拦截器链的？

55\_JDK代理的执行：怎么为目标方法匹配适用的增强？

56\_JDK代理的执行：来看下Advice的构建过程

57\_JDK代理的执行：Advice和拦截器之间是什么关系？

58\_JDK代理的执行：来初步看下拦截器链的执行流程

59\_JDK代理的执行：拦截器的核心逻辑，递归调用ReflectiveMethodInvocation

60\_JDK代理的执行：拦截器链的本质，责任链模式

### **第七章：Spring AOP中CGLIB代理源码剖析**

61\_CGLIB代理的创建：通过Enhancer来创建代理

62\_CGLIB代理的执行：通过DynamicAdvisedInterceptor来执行增强逻辑

63\_AOP拓展：AnnotationAwareAspectJAutoProxyCreator是怎么注入进来的？（上）

64\_AOP拓展：AnnotationAwareAspectJAutoProxyCreator是怎么注入进来的？（下）

65\_面试题：Spring AOP用到了哪些设计模式？具体体现在哪些地方？

### **第八章：Spring 事务底层源码剖析**

66\_Spring事务：先使用AOP来模拟一下@Transactional的功能

67\_Spring事务：@Transactional注解竟然是基于AOP实现的？

68\_Spring事务：TransactionInterceptor是怎么来控制事务的？

69\_Spring事务：来看下TransactionInterceptor增强的匹配过程

70\_Spring事务：来看下方法级别匹配时，获取事务属性的过程

71\_Spring事务：来看下@Transactional事务代理创建和执行流程

72\_Spring事务：遗漏的细节，BeanFactoryTransactionAttributeSourceAdvisor是怎么注册到IOC容器中的？

73\_面试题：有哪些场景会导致Spring事务失效？