孙哥说分布式课程大纲 V3 版

孙哥寄语：（课前必看）

本课程是由我及课程开发伙伴通过多年 JAVA 教学经验及学员的反馈，进行摸索、积累、总结，最终成型的一套精品课程。本套课程以能让学员更加轻松的学习，更有代入感的学习，更加高效的学习为目标，精心录制而成。希望能让更多学员朋友从中受益，在学习上不走弯路、不浪费更多的时间与精力，边学边用，毕竟我们学习知识技能的目的是要真正用到工作实践中去，生命是有限的，不要把有限的时间浪费在无休无止盲目的学习上，让我来带着大家边学边用，为大家节省宝贵的时间。

知识只有用起来，才能真正的掌握！加入到我们的学习群中，和伙伴们分享学习与工作中遇到的困难，让我与大家一同思考解决办法，共同成长。

一、课程亮点

1、课程名称都一样，但她更有料，更有内涵

不要只看课程名字，或多或少这些课程名字你都接触过，或者这些课程的少部分内容，你都学习过，但是一定没有这套课程讲解的细致和深度。（请看后面的大纲对比）

2、不低于 500 个课时，讲解细致入微

本套课程累计学时超过 500 小时，细致和概括不能两全，在精炼知识点的基础上我们还要保持足够的细致，精炼知识点后必须要有细致的说明与详细的解释，才能让大多数学员知其然，并知其所以然，上手实操无压力。

3、由浅入深，层层递进

所有的课程都是由浅入深，从基础使用，到高级进阶，多年教学经验，精炼内容，干货满满。

4、直接的沟通渠道，我讲什么，由大家决定

我们的课程并不是一程不变，具有灵活性及成长性，对于某门课程中，你感兴趣的知识点可以告诉我，我们会融入进来，进行专门直播讲解答疑。

5、不仅是一套课那么简单

你所购买的不是单单这一套课程，还包括后续一年的会员权益，后续至少还会推出 SpringBoot 大师课程，ES 从使用到精通课程。Redis 一套通关课程等。

6、建立学习圈子，更多增值服务

课程之外，有专门的学习互助圈子，这里有更广的渠道，助力你的学习及工作。例如：会定期内推工作，简历制作，模拟面试等。

7、现货课程，安全保障

这点不多做介绍，课程就在这里，随时可取！

8、一节课只要不到一顿早点钱，不信自己算一算

早鸟购买，优惠满满，限时供应。

9、定制版深造播放器，高清看得见，听的清。

与深造加密深度合作，定制版双高清视频。

10、本课程适合人群广

本课程适合在计算机相关专业在校大学生、毕业生、在职软件开发人员及广大编程爱好者。

11、课程更新周期

可开通两台设备观看课程，每周 4 更。

12、更多好处由您亲身体会！

联系我们提供免费试听课

全网最低价 1999 元

课程持续更新中，

早鸟体验价，

惊爆价！限时抢购！！！

现在购买单课时不足五元！！！

加 v 咨询： suns45 或 fhcxy123

二、讲师介绍

1、**幽默风趣**:在讲课中善于运用幽默和风趣的语言，使得课程更加生动活泼，吸引听众的注意力。

3、**实战经验**:会通过分享自己在实际工作中的经验和案例，帮助听众更好地理解和 2 应用所学的知识。

4、**整体思维**：强调从整体的角度来看问题，注重系统思维和综合运用不同领域的知识和方法。

5、**注重实用性**:讲课风格注重实用性，强调将理论与实践相结合，使听众能够在实际工作中运用所学的知识。

6、**互动性**:会倡导与听众之间的互动，包括提问、讨论、演练等形式，以促使听众更深入地理解和掌握所学的内容。

三、课程简介

课程包含如下内容

1. Spring 源码高级架构课程 ---- 从里到外扒光 Spring

2. Mybatis 源码一套通 ---- 面试无敌，路路通

3. Netty 应用指南 ---- 补足网络通信短板

4. Netty 大师级源码 ----- 全网最全 Netty 源码，时间轮，内存池精讲等

5. RPC 分布式原理设计 ----- 从底层理解 RPC，多种协议一网打进，模拟 SpringCloud

6. Dubbo 技术一套通 ----- 源自阿里, 但在京东内部使用的核心 Java 框架

7. Dubbo 高阶源码 ------ 带着 CT 眼看 Dubbo

8. RocketMQ 应用指南 ------ MQ 的新高地，知其然知其所以然

9. RocketMQ 架构级源码 ------ 抽丝剥茧，提高编程理解能力

10.分布式算法 ------- 常见分布式体现的常见算法分析,

11. SpringCloud 应用实战 ------- 看似普通，但是给你不一样的 Cloud 学习体验

12. SpringCloud 源码 ------- 面试通关装逼新神器

四、课程特点

1、**清晰逻辑**：注重知识的连贯性、逻辑性，讲课概念清晰，逻辑严谨，能够将复杂的 Java 知识以简单易懂的方式呈现帮助学员更好地理解和掌握知识

2、**实用案例**： 善于运用实际案例来支持课程内容，将抽象的 Java 理论与实际应用相结合，帮助学员更好地理解 Java 知识在实际项目中的应用。

3、**生动活泼**：讲课生动活泼，善于运用丰富的语言表达和幽默风格，使课堂氛围轻松愉快，激发学员的学习兴趣和积极参与课程讨论

4、**深入浅出**：能站在学员的角度把知识讲透彻，能够将复杂的 Java 知识讲解得深入浅出，避免使用过多的专业术语和技术细节，使得学员能够更容易理解和掌握 Java 编程的核心概念和技能。

5、**关注学员**：关注学员的学习需求和学习进度，能够根据学员的实际情况进行灵活的教学安排和课程调整，积极回应学员的问题和疑问，帮助学员充分掌握 Java 编程技能，每周都会开一次直播 互相语音现场答疑解惑，照顾基础薄弱的学员，让基础差的听得懂，基础好的搞透彻，每个人都学有所成,学有所获,学有所用。

6、**覆盖广**：覆盖企业主流技术栈、技术体系化、结构化课程深入浅出，从原理源码到企业应用实战，本着授人以渔的原则课程精心设计，基于深、广、宽三维一体设计。

7、**群氛围好**：可以探讨学习，可以在群里面要资料，有大佬讲解，有老师讲解，不懂的问题可以问大佬或者老师

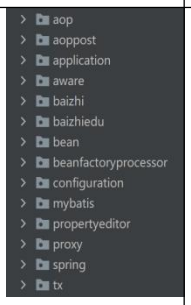
8、**减轻学员学习压力**：课堂最后会完成总结，有完整的课堂笔记，课前会带着学员温故知新。

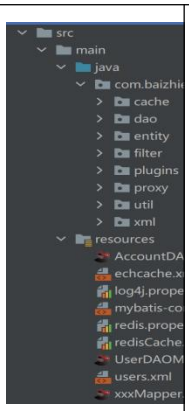
9、**额外收获**：课程不止硬技能培训，还有软技能的培养企业面试指导、职业规划、企业内推。

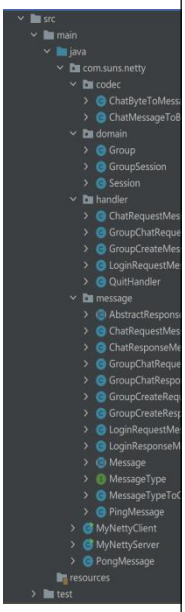
10、**课时非常良心**

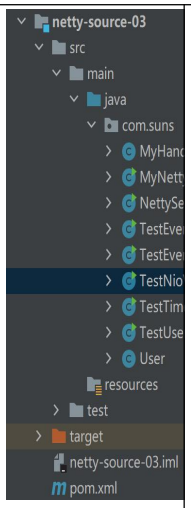
11、**课程价格非常良心**

五、课程大纲介绍

课程模块	实训模块	课程大概内容	培养目标	课时
Spring 源码高级架构课程		1. Spring 创建对象的过程、初始化方法、销毁方法、创建对象的源码 2. doGetBean 的工作流程 3. Spring BeanDefinition 创建注册的过程 4. Spring 的 ApplicationContext(注解版)核心分析 5. prepareBeanFactory()分析 6. BeanPostProcessor 和 BeanFactoryPostProcessor 的讲解 7. ConfigurationClassPostProcessor 作用讲解 8. AutowiredAnnotationBeanPostProcessor 作用讲解 9. CommonAnnotationBeanPostProcessor 作用讲解 10. 代理设计与装饰者设计模式的区别、SpringAOP 含义、本质、是怎么实现代理的、是如何封装 jdk 和 cglib、如何解决内部代理对象调用方法 11. AnnotationAwareAspectJAutoProxyCreator 分析 12. 循环依赖是如何产生的、Spring 是如何解决 13. Spring 事务的基础讲解、以及传播属性的源码介绍、事务是如何起作用的、事务嵌套的源码 14. Spring 与 Mybatis 整合源码的讲解	Spring 创建对象，Spring 常用注解，Spirng 事务，Spring Aop，Spring 整合 Mybatis 对这些有个整体的认知，了解大概的原理。	共 35 集大约 70 个小时

Mybatis 源码一套通		1、Mybatis 介绍 (1) 什么是 Mybatis (2) Mybatis 的设计思想和优点 (3) Mybatis 在 ORM 框架中的定位 2、Mybatis 配置 (1) Mybatis 配置文件的结构和属性 (2) Mybatis 的全局配置 (3) Mybatis 的 Mapper 配置 3、处理 Sql 语句 (1) SqlSource 的作用和实现方式 (2) ParameterHandler 的作用和实现方式 (3) StatementHandler 的作用和实现方式 4、执行查询操作 (1) Executor 的作用和实现方式 (2) CachingExecutor 的作用和实现方式 (3) SimpleExecutor 的作用和实现方式 5、结果映射 (1) ResultMap 的作用和实现方式 (2) TypeHandler 的作用和实现方式 (3) 自定义 TypeHandler 的实现方式 6、插件 (1) 插件的作用和使用方式 (2) Mybatis 提供的常用插件 (3) 自定义插件的实现方式 7、缓存 (1) 二级缓存和一级缓存的区别和使用场景 (2) 缓存的实现原理 (3) Mybatis 提供的缓存实现方式 8、Spring 整合 Mybatis (1) Spring 整合 Mybatis 的方式 (2) Spring 提供的 Mybatis 相关组件 (3) Spring 整合 Mybatis 的最佳实践	了解Mybatis的运行过程，核心对象的作用，以及缓存、拦截器使用、思想等等。	共 17 集大约 34 个小时

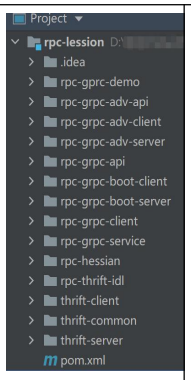
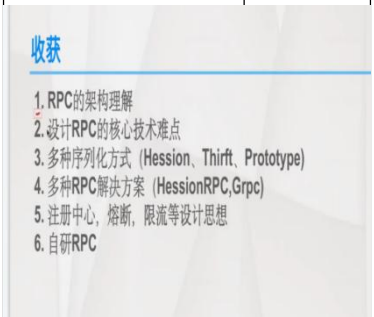
Netty 应用指南	1、java nio 的练习 2、Netty 各个常用知识点的练习 3、Netty 的聊天小项目 	1、全方面的 JavaNio 讲解，从 0-1 仿写 Netty 版的 Java Nio Reactor【讲了市面上好多未提及到的】 2、半包粘包的具体处理讲解以及网络连接常见问题的讲解 3、NioEventLoopGorup、NioEventLoop、DefaultEventLoop 组件的讲解 4、Java Futrue、Netty Promise 的对比讲解 5、Handler、childHandler、ServerBootStrap、BootStrap 各个组件的讲解 以及 Netty 服务端运行过程 6、ByteBuf 和 ByteBuffer 的对比、ByteBuf、ByteBuffer、堆内存、直接内存的讲解以及如何回收内存什么时候回收内存的讲解 7、粘包和分包的本质讲解、Socket 缓冲区和滑动窗口的讲解、write()写入的含义 8、Netty 常见编解码器的讲解 9、Netty Http 编解码器的讲解 10、解码器处理数据过程的讲解、如何自定义协议的讲解。 11、Pipeline 和 Handler 关系 作用的讲解，handler 生命周期方法的讲解，异常的处理等等 12、IdleStateHandler 的讲解、WebSocket 相关 Handler 的讲解。 13、Netty 设置 参数的讲解包括 RCVBUF_ALLOCATOR、SO_RCVBUF & SNDBUF、ALLOCATOR。 14、Socket 缓冲区大小如何控制是池化内存还是非池化内存的解决、如何控制是直接内存还是非直接内存的解决、第一个 Handler 的 Object msg 和 RCVBUF_ALLOCATOR 的关系、SO_BACKLOG 等等参数的讲解 16、SO_KeepAlive 参数的讲解以及与心跳关系的讲解 17、分隔符处理器的讲解 18、从 0-1 实现了一个 netty 聊天小项目	熟悉 Nio 的大概原理，Java 异步任务和 Netty 异步任务的异同点，掌握基本应用，了解每个组件的基本作用，了解经常使用的相关参数，能进行常规 Netty 业务的开发。	共 29 集 大约 58 小时

Netty 大师级源码		1、serverbootStrap.bind()源码分析的讲解 (1)服务器源码启动分析的讲解 (2)EventLoop 的作用以及组成的讲解 (3)Netty 是如何优化 selector 的讲解 2、NioEventLoop.run()方法的讲解 (1) run 方法是如何权衡 io 操作和普通任务的讲解 (2) Netty 是如何解决空轮询的讲解 (3) 如何处理感兴趣事件 (1) ServerSocketChannel 的整个 accept 过程 (2) SocketChannel 读数据的过程 (3) doReadBytes 方法的解析 (4) ContinueReading 方法的解析 (5) 设置循环上限 16 次的原因 2、Netty 可扩缩容的分配器原理和 Unsafe 的实现 (1) 如何识别为堆外内存分配 (2) 动态扩容的原理 (3) Unsafe 的好处 (4) AbstractUnsafe.write、channel.write() 以及 channel.flush()的认知 (5) AbstractUnsafe.write()是怎么将数据交给 Netty 存储的 (6) AbstractUnsafe.write()在过滤 msg 为什么要转化为直接内存 (7) AbstractUnsafe 过滤 msg 是如何实现 3、write 方法、Flush 方法、outboundBuffer 设计的原理分析 (1) pipe.estimatorHandle.size(msg)为什么计算传输的数据大小 (2) ChannelOutboundBuffer 中的存储数据输出到 Sockete 缓冲区会有多少种结果 (3) Flush 方法的原理 (4) Flush0 的操作分析 (5) doWrite 操作分析 (6) Entry 入队时间和修改状态的	熟悉 netty 的大概运行流程，了解 FastThreadLocal 原理以及运行流程， 了解 时间轮原理以及运行流程， 了解 内存池原理以及运行流程以及大概思想。	共 21 集大约 42 个小时

		时机分析 (7) 为什么要设置成可以指定 Entry 对象属性信息所占的字节 (8) OutBoundBuffer 关于 Entry 状态的巧妙设计 (9) addMessage()分析 (10) addFlush()分析 (11) 关于水位线的下降是如何处理 (12) channel.writeAndFlush() 到数据写出的经历过程 4、LineBasedFrameDecoder 的源码分析 (1) LineBasedFrameDecoder 如何完成解码的讲解 (2) LineBasedFrameDecoder 如何处理最后一条消息的讲解 5、Pipeline 的基础介绍、添加 handler 的分析、事件传播 (1) ChannelConext 要包装 channelHandler 的原因的讲解 (2) pipeline.addLast(线程池,名字,handler)分析讲解 (3) 解决 handler 名称冲突的讲解 6、outBound 事件传播 (1) Netty 的 outBound 事件传播源头 (2) Netty 的 outBound 事件是如何在多个 Handler 中进行传播的讲解 7、Java 的 Timer 和 Netty 的时间轮 (1) JDK Timer 的缺点 (2) 小根堆，大根堆 (3) 超过范围外，Netty 的处理讲解和 Kafka 的处理讲解 (4) 时间轮在 Netty 中的体现、时间轮的组成 (5) HashedWheelTimer 构造方法分析 ① 创建时间轮的时候引出时间轮数组大小的设计 (6) newTimeOut(task,time,timeUnit)添加延迟任务的分析 (7) 如何防止线程空转的讲解		

		8、Netty 的时间轮 worker 中的 run 方法分析 (1) Worker 中的 run 线程睡眠处理 +99999 的讲解 (2) Worker 中线程的三种情况 (3) 从 timeouts 队列中取出定时任务，添加链表这个步骤讲解 (4) 一个任务耗时过长时影响了其他定时任务,netty 是怎么处理的讲解 (5) JDK 的 Timer 和 Netty 的事件轮以及 Kafka 时间轮的区别 (6) Netty 时间轮存在的问题 (7) Netty 如何才能提高延时任务的准确性的讲解 (8) 什么情况下 tick 的值会大于 Calculated 的讲解 (9) 9、ThreadLocal 和 FastThreadLocal 的讲解 (1) ThreadLocal 是如何解决 hash 碰撞以及 threadLocal.set() 源码的讲解 (2) FastThreadLocal 在 ThreadLocal 上的优化讲解, 以及处理过程 (3) FastThreadLocal 的 set 方法分析 (4) FastThreadLocal 的 get 方法分析 (5) InternalThreadLocalMap 的 get 方法 (6) 普通 Thread 也可以用 FastThradLocalThread 一定能快吗 (7) 普通 Thread 与 InternalThreadLocalMap 是如何联系的讲解 (8) InternalThreadLocalMap 所对应的 Object[]中 0 号元素, 要存储 Set<FastThreadLocal>的原因讲解 10、内存池讲解一 (1) 内存池本质的讲解		

		(2) 合理使用内存的技巧、netty 合理使用内存的点 (3) 内存池设计的核心讲解 (4) 三种内存池【glibc、tcmalloc、jemalloc】的实现标准 11、内存池讲解二 (1) Netty 的内存规格的讲解 (2) PoolAreana、、PoolSubPage、Chunk 的讲解 (3) tinySubpagePools 和 smallSubpagePools 如何通过位图知道自己的使用情况 (4) PoolChunkList 为什么是多个的原因讲解，使用率各是多少，为什么要做区间，它们中间是如何移动的讲解 12、内存池讲解三 (1) PoolAreana 中 qinit 和 q000 的本质区别 (2) qinit 中会出现使用率 0 的 chunk 的情况讲解 (3) q000 中会出现使用率 0 的 chunk 的情况讲解 (4) PoolChunkList 的升级降级的分析 (5) PoolChunkList chunk 的创建过程的讲解 (6) PoolChunk 如何通过友邻算法对 Page 进行管理的讲解 (7) Netty 申请内存的完整流程的讲解		

RPC 分布式原理设计		1、架构的演变过程，RPC 的概念，TCP 和 HTTP 与 RCP 的关系。【第一讲】 (1) RPC 概念 (2) 架构的演变过程 (3) 单体架构的概念 (4) 单体架构存在的问题 (5) 垂直架构的概念以及优点 (6) HTTP 和 TCP 的关系 (7) 走 HTTP 协议的问题、HTTP 典型的技术实现方案、协议优势 (8) 跨进程间调用的本质、网络通信的内容 (9) TCP 协议的缺点、优点、技术实现方案 2、RPC 设计目标、自定义协议、RPC 通信客户端和服务端要做的事情【第二-1 讲】 (1) RPC 的设计目标、如何完成调用、如何自定义协议 (2) RPC 通信客户端做的事情、RPC 通信服务端做的事情 (3) 如何设计客户端、远端服务方法功能等同于本地方法的原理 (4) Stub、Skeleton 专业术语 3、注册中心的由来和好处，以及辅助 RPC 的技术，以及 Hessian RPC 概念【第二-2 讲】 (1) 注册中心的由来、注册中心的好处 (2) 辅助 RPC 的技术 (3) 学习 HessianRPC 的原因、HessianRPC 的概念、HessianRPC 客户端设计思想、HessianRPC 服务端设计思想 (4) HessianRPC 要求定义接口的原因 4、HessianRPC 开发【第三-1 讲】 (1) HessianRPC 的开发环境搭建、开发步骤讲解 (2) HessianRPC 源码分析、序列		正在录制 目前 36 集， 大约 50 个小时

		化的定义、Hessian 序列化、Hessian 的目的 5、复习 HessianRPC ，讲解 Thrift RPC【第四讲】 (1) RPC 所需技术、特点、数据类型 (2) Thrift 的作用、基本概念、特点、设计思想 ① Thrift 如何解决不同编程语言差异性问题 (3) Thrift 的三个核心组件、调用流程、安装 (4) IDL 语法、注释、命名空间、基本类型、集合类型、stuct 自定义对象 6、Thrift 数据类型介绍，Thrift 开发上【第五-1-2 讲】 (1) 字面值、枚举、异常的定义 (2) 数据和服务的定义、include、生成命令 (3) ThirftRPC 的开发、环境搭建、项目结构的定义 (4) TTranspost、TProtocol、TProcessor、Tserver 作用 (5) Thrift 代码编写 common 模块、服务端、客户端 7、ThriftRPC 各种 Server 源码分析【第六-1-2 讲】 (1) 实战开发思考、TSimpleSever 、TNonblockingServer、TThreadSelectorServer 作用以及源码分析 (2) TThreadSelectorServer 的坑 8、GRPC 的讲解【第七讲】 (1) GRPC 的介绍、核心设计思路 (2) GRPC 与 ThriftRPC 的区别 (3) GRPC 的好处 (4) HTTP2.0 协议介绍、Http1.x 协议介绍		

		① 多少个连接完成一次请求 ② 怎么解决 3 次问题 ③ 为什么 TCP 是长连接、HTTP1 是短连接 ④ HTTP2.0 三个概念的介绍、以及相互的关系 (5) Protobuf Buffers 特点、编译器安装、Idea 插件安装 9、Protobuf 的语法详解【第八讲-1】 (1) 文件格式、版本设定、注释、与 java 相关的语法、逻辑包、导入、基本类型、枚举 (2) 消息 Message 的编号、singular、repeated、定义多个消息、消息嵌套、oneof 10、GRPC 开发实战环境搭建【第八讲-2】 (1) 服务定义、项目结构讲解、api 模块、修改 pom.xml 插件配置 11、GPRC 开发实战【第九讲-1】 (1) GRPC 目录说明 (2) xxx-server 服务端模块开发 (3) xxx-client 客户端模块开发 12、GRPC 四种通信方式【第九讲-2】 (1) 服务端注意事项 (2) GRPC 的四种通信方式 ① 简单 RPC 的特点、语法、特点、代码 ② 服务端流式 RPC 的概念、特点、使用场景、语法、代码、监听异步方式以及和服务端流水 RPC 的区别 ③ 客户端流式 RPC、概念、应用场景、proto		

		定义、服务端开发【第十讲-1】、 客户端开发【第十讲-2】 ④ 双向流式 RPC 的概念、应用场景、编码 (3) GRPC 的三种代理方式 13、SpringBoot 整合 GRPC【第十一讲】 (1) 整合思想、SpringBoot 与 GRPC 是如何封装如何结合的 (2) 如何从根本上解决 Protobuf 依赖问题 (3) 如何去掉 tomcat (4) Spring.application.name 默认值问题 (5) 如果自定义了包扫描策略默认的包扫描是否生效 (6) 实战之服务端搭建环境、服务端开发、客户端搭建环境、客户端开发 14、GRPC 的高级应用之拦截器【十二讲-1】 ① 一元拦截器与拦截器的区别 ② 常见拦截器 ③ Grpc 拦截器 (2) Web 端的 filter 只能拦截服务端的请求、GRPC 客户端和服务端都可以拦截 (3) GRPC 的工作流程、调试技巧 (4) 简单客户端【一元拦截器】介绍、开发、缺点、拦截器请求的四个阶段 (5) 复杂客户端拦截器特点、请求数据和响应数据的拦截代码编写【一元拦截器】【十二讲-2】 15、GPRC 高级应用之 GRPC 流式拦截器【十三讲-1】 (1) 流式拦截器的概念 (2) 一元通信方式和流式通信		

		方式的区别 (3) 客户端、服务端流水拦截器的开发 16、GRPC 高级应用之客户端重试【十四讲】 (1) 客户端重试开发 (2) 命名解析 NameResolver 概念介绍 (3) 注册中心的由来、常见注册中心介绍 (4) GRPC 默认的注册中心、如何进行扩展 17、Consul 注册中心【十五讲-1】 (1) Consul 是什么、作用、好处 (2) Consul 安装 (3) Consul Java 客户端注册编码 (4) 删除服务、健康检查 (5) Client 端从 Consul 获取服务代码编写 18、GRPC 和 Consul 整合【十五讲-2】 (1) 服务端处理、客户端处理【十六讲-1-2】 (2) GRPC 客户端开发、以及客户端与服务端注意事项 19、GRPC 与 SpringBoot 整合的高级应用【十七讲-1】 (1) 拦截器开发 (2) 与注册中心结合 GRPC 替换 SpringCloud OpenFeign ① 服务端开发、注意事项 ② 客户端开发 20、自定义 RPC---【十七讲-2】 (1) RPC 技术回顾 (2) 自研 RPC 开发的技术储备 (3) 协议讲解【十八讲-1】 ① 概念、分层 ② 公有协议、私有协议 ③ 通信组合 (4) 序列化讲解		

		(5) 解决 Thirft 插件不存在重新编译解决问题【十八讲-2】 (6) 编写 JSON、Thrift、Hessian、JDK、Protobuf 编解码对比测试代码【十九讲-1】 (7) 编写 JSON、Thrift、Hessian、JDK、Protobuf 编解码以及序列化封装【十九讲-2】 (8) Netty 与序列化方式进行集成【二十讲-1】 (9) Netty 编解码编写 (10) netty 服务端、netty 客户端开发【二十讲-2】 (11) Netty 封装编解码和自行设计的编解码那一个更好？【思考的问题】 21、注册中心【二十一讲-1】 (1) Zookeeper 简介，zookeeper 的功能、zookeeper 的逻辑结构 (2) Zookeeper 物理结构以及单机和集群的对比【二十一讲-2】 (3) zookeeper 集群注意事项 ① zk 主节点从节点的概念 ② 如何确定 zk 集群众节点的身份 ③ 主节点作用、从节点作用 ④ zk 集群的容错性 ⑤ zk 集群节点数的要求 (4) zookeeper 安装需要的 linux 配置 (5) 单机版安装【二十二讲-1】 (6) 集群版 SSH 解决登录问题【二十二讲-2】 (7) 集群版安装【二十三讲-1】 (8) zookeeper 客户端集群通信方式【二十三讲-2】 (9) 四种节点类型 (10) 四种节点类型的注意实现		

		(11) Zookeeper 客户端使用 ① Zk-cli 命令 ② Java 代码		
Dubbo 技术一套通		1、Dubbo 实战运用 2、Dubbo 的 SPI 机制 3、Dubbo 的 IOC 扩展机制 4、Dubbo SPI 的 AOP 机制 5、Dubbo 的@Adaptive 作用	掌握 Duboo 的正常开发	待录制，内容暂定
Dubbo 高阶源码		1、Dubbo 的 SPI 机制源码 2、Dubbo 的 IOC 扩展机制源码 3、Dubbo SPI 的 AOP 机制源码 4、Dubbo 的@Adaptive 源码	了解 dubbo 如何与 java 的代理结合如何与 SPI 进行结合，如何改善原生 SPI 的问题	待录制，内容暂定

RocketMQ 应用指南		1、MQ 使用场景 2、6 种消息 3、ACL 权限控制 4、消息存储、消息存储结构 5、刷盘机制 6、消息主从复制 7、负载均衡 8、消息重试 9、死信队列 10、如何保证消息幂等 11、使用 RocketMQ 如何保证消息不丢失? 12、使用 RocketMq 保证消息有序 13、使用RocketMq 如何快速处理挤压消息	掌握 RocketMQ 的正常业务开发	待录制，内容暂定
RocketMQ 架构级源码		1、生产者原理 2、生产者启动流程 3、消息发送流程 4、消费者启动机制 5、消费者的 Rebalance 机制 6、消费进度保存机制 7、消费方式 8、消息过滤 9、NameSrv 概述、架构、路由原理 10、Broker 存储机制、索引机制、过期文件删除机制、主从同步机制、关机恢复机制 11、RocketMQ 事务消息与延迟消息机制		待录制，内容暂定
分布式算法				

SpringCloud 应用实战+ SpringCloud 源码		1、Ribbon (1) Ribbon 的基础知识 (2) @LoadBalanced 原理 (3) 饥饿加载 (4) 负载均衡 (5) 扩容负载均衡算法 (6) 扩展负载均衡器 (7) 控制 Ribbon 负载均衡算法范围 (8) 配置异构服务 2、Feign (1) Feign 的基础知识 (2) Feign 的架构 (3) 日志配置 (4) 契约配置 (5) Feign 拦截器 (6) 超时时间配置 (7) 客户端组件配置 (8) GZIP 压缩配置 (9) 编解码配置 (10) Feign 调用原理 3、Nacos (1) Nacos 职责和核心功能 (2) 使用注册中心的原因 (3) 几种远程调用服务的方法 (4) Nacos 是如何存储 instance 实例 (5) Nacos 新版本不用加 @EnableDiscoveryClient 的原因 4、Nacos 源码 (1) Naocs 注册表如何防止多节点读写与并发冲突 (2) Nacos 高并发支持异步与内存队列剖析 (3) Nacos 心跳机制与服务健康检测源码分析 (4) Nacos 服务变动事件发布源码深度剖析 (5) Nacos 服务下线源码深度剖析 (6) Nacos 心跳在集群架构下的设计原理剖析 (7) Nacos 集群节点状态同步		待录制，内容暂定

		源码欧西 (8) Nacos 集群服务新增数据同步源码剖析 (9) Nacos 集群服务状态变化同步源码剖析 (10) Nacos 配置动态刷新原理、Nacos 配置优先级 5、Seata (1) Seata 的基础知识 (2) Seata 各个模式的工作流程 (3) @GlobalTransactional 原理 (4) GlobalTransactionalInterceptor 原理 (5) 实战 6、Gateway (1) Gateway 核心概念 (2) SpringCloud 整合 Gateway 使用 (3) GatewayFilterFactories 过滤器工厂的设置 (4) GlobalFilters 全局过滤器设置 (5) Gateway Cors 跨域设置 (6) Gateway 整合 Sentinel 限流实战 (7) Gateway 网关高可用部署 7、Gateway 源码 (1) WebFlux 核心请求流程分析 (2) Gateway 是如何整理 WebFlux (3) GateWay 路由匹配核心源码分析 (4) GateWay 请求过滤器源码分析 (5) Gateway 设计模式 (6) Gateway 两种 filter 区别 (7) Gateway 工作原理 8、Sentinel (1) Sentinel 的使用场景 (2) Sentinel 的基础概念 (3) Sentinel 各种流控规则		

		(4) Sentinel 整合 RestTemplate (5) Sentinel 整合 Feign (6) @SentinelRestTemplate 原理 (7) Sentinel 使用 Feign 原理 9、Sentinel 源码 10、Hystrix (1) 什么是 Hystrix (2) 雪崩效应 (3) 模拟高并发场景 (4) 请求缓存 (5) 请求合并 (6) Feign 雪崩容错 (7) 服务降级 (8) 服务熔断 (9) 服务隔离 (10) Actuator (11) 监控中心 (12) 聚合监控 11、Hystrix 源码		

六、解决疑惑

问题一、为什么要报课学习？

1、获得专业指导：报课学习可以让你获得来自专业教师的指导和支持，他们可以帮助你更好地理解和掌握所学内容，同时也可以及时回答你在学习过程中遇到的问题。

2、学习更有序：通过报课学习，你通常会按照预定的课程安排逐步学习，这样可以使你的学习更加有序、系统，并且可以帮助你更好地掌握知识点。

3、学习更高效：报课学习通常会有一个明确的学习目标和计划，这有助于你更加集中精力、高效学习。此外，因为你有一个固定的学习时间表，你可能会更有动力去完成每一次作业和任务。

4、聚焦实践：很多课程都会包含实践环节，这可以帮助你将理论知识应用到实际场景中，从而更好地理解和掌握所学内容。

5、与同行互动：报课学习还可以让你与其他学生互动交流，你可以借此机会结识新朋友、分享学习心得，甚至寻找未来的合作伙伴。
6、报课学习可以让你通过专业指导、有序学习、高效实践、互动交流等方式提高自己的技能和知识水平。

问题二、这些技术我又用不到，源码什么看不看不都一个样吗?

虽然你可能不会直接使用或者看懂某些技术或者源码，
但是阅读和学习这些内容仍然是有益的。
首先，通过阅读和学习源码，你可以了解到一些其他人在实现某个功能或解决某个问题时的思路和方法，从中汲取经验和启发，有助于提高自己解决问题的能力。其次，阅读和学习源码也是一个拓宽视野、增加知识储备的过程。尽管某些技术或者源码可能与你当前的工作没有直接关联，但你透彻理解它们可能会帮助你更好地理解整个行业的发展趋势和技术演进方向，为未来做好准备。
最后，阅读和学习源码也可以提高你的学习能力和技术素养，让你成为一个更加全面和有竞争力的程序员。通过不断花时间阅读和学习源码，你可以逐渐提高自己的技术水平，并且在日常工作中更加游刃有余。因此，即使你认为某些技术或者源码与你当前的工作没有直接关联，也建议你保持对它们的关注和学习，从中获取经验和启示，不断提高自己的技术素养。

问题三、为什么市面上已经有了免费视频，我还要做类似的视频呢?

1、专业性：视频清晰明了，更具有逻辑性，易于掌握。【您可以看看试看，自行判断】
2、更深入的学习：相比免费的视频，付费课程通常会提供更加深入、详细和系

统的学习体验，能够让你更好地理解和掌握所学内容。

3、更好的服务和支持：提供更好的售后服务和支持，如专业的答疑、跟踪指导等，让你在学习过程中更加顺畅和愉快。