

1. 必须深入了解的知识点：

- a. java集合数据结构
- b. jvm： gc、内存模型、堆分代
- c. 多线程： volatile、java.util.concurrent包
- d. spring： 启动、时间
- e. redis： 分布式锁、lua保证原子性、集群
- f. 消息队列： kafka
- g. Springcloud
- h. mysql搜索引擎，索引B+tree

2. java基础：

- a. nio模型
- b. java反射
- c. java8特性
- d. java中有哪些集合？各自的数据结构？
- e. 说说你用过的设计模式
- f. jdk以及spring中看过哪些设计模式

3. jvm：

- a. jvm结构，以及每一部分的作用
- b. jvm中哪些部分会出现oom，出现的可能原因
- c. 哪些对象会被存放到老年代？
- d. 什么时候触发full gc？
- e. GC算法分类
- f. 类加载机制以及类加载器，双亲委派机制。
- g. 一个对象被实例化后，在堆中的流转流程（堆结构以及GC流程）
- h. jvm调优，写调优参数。

4. spring：

- a. spring bean的生命周期
- b. 如何实现spring aop？
- c. springboot启动流程？
- d. Spring中BeanFactory和FactoryBean有什么区别，factorybean原理？
- e. spring事件
- f. 作用域spring，如果是request方式，a依赖b，a是单例方式，b是request，回怎么样

5. 多线程：

- a. 乐观锁、悲观锁
- b. 三种分布式锁原理
- c. 线程池原理，java有哪几种线程池？如何创建？
- d. synchronized特性
- e. volatile原理
- f. Threadlocal
- g. ConcurrentHashMap
- h. HashMap为何线程不安全
- i. java.util.concurrent包下有哪些类？

6. 数据结构：

- a. HashMap数据结构？
- b. ConcurrentHashMap数据结构（jdk1.7 &1.8）？
- c. 二叉树、平衡树、红黑树
- d. B-tree、B+tree
- e. TreeMap

7. 微服务：

- a. 画springcloud工作原理图
- b. 描述各个组件的作用及原理
- c. 心跳机制
- d. 限流方法：
- e. 服务熔断、降级与隔离的实现原理

- f. eureka优缺点
- g. dubbo的负载均衡方式

8. 数据库：

- a. mysql存储引擎有哪些？他们的区别？
- b. 数据库隔离级别有哪些，并说下什么是脏读、不可重复读、幻读？
- c. 数据库事务的四大特性ACID？
- d. mysql调优方案？
- e. 在代码中，我们如何实现事务？
- f. 如果在一个事务中，代码业务流程很长，会有什么问题吗？为什么会出现这种问题？

9. zookeeper：

- a. 选举算法
- b. 分布式锁实现原理、优缺点
- c. 服务发现与移除原理

10. redis：

- a. 实现机制
- b. 分布式锁，以及要考虑的问题
- c. 除了加锁，如何使多个redis命令具有原子性（lua脚本）
- d. redis持久化的原理
- e. 分布式锁，使用redis的方式，但是redis是集群的，会有不同步的问题，官方文档是怎么解决的

11. 消息中间件：

- a. 常见消息中间件的区别
- b. kafka的实现原理
- c. RabbitMQ消息模型
- d. RabbitMQ如何保证消息可靠性

12. 分布式：

- a. 介绍几种分布式事务，以及优缺点。
- b. 分布式服务的原则，什么情况下要使用分布式开发。
- c. 使用自增ID和UUID作为主键有什么不同

13. 源码：说说你看过的源码，其中有用到什么思想，以及设计模式？

- a. spring启动时创建容器：AnnotationConfigApplicationContext.refresh()

```
1 public void refresh() throws BeansException, IllegalStateException {
2     synchronized (this.startupShutdownMonitor) {
3         //设置环境配置参数
4         prepareRefresh();
5
6         //创建beanFactory（同步代码块，反射获取bean）
7         ConfigurableListableBeanFactory beanFactory = obtainFreshBeanFactory();
8
9         // 给beanFactory注册一些标准组件，如ClassLoader，BeanPostProcess
10        prepareBeanFactory(beanFactory);
11
12        try {
13            //设置web生命周期管理的Scope
14            postProcessBeanFactory(beanFactory);
15
16            //调用所有BeanFactoryProcessor的postProcessBeanFactory()方法
17            invokeBeanFactoryPostProcessors(beanFactory);
18
19            //注册BeanPostProcessor，BeanPostProcessor作用是用于拦截Bean的创建
20            //如果我们需要在Spring容器完成Bean的实例化、配置和其他的初始化前后添加一些自己的逻辑处理，
21            //我们就可以定义一个或者多个BeanPostProcessor接口的实现，然后注册到容器中。
22            registerBeanPostProcessors(beanFactory);
23
24            //初始化消息Bean
```

```
25         initMessageSource();
26
27         //初始化上下文的事件多播组件，ApplicationEvent触发时由multicaster通知给ApplicationListener
28         initApplicationEventMulticaster();
29
30         //ApplicationContext初始化一些特殊的bean
31         onRefresh();
32
33         // Check for listener beans and register them.
34         //注册事件监听器，事件监听Bean统一注册到multicaster里头，ApplicationEvent事件触发后会由multicaster
35         registerListeners();
36
37         //非延迟加载的单例Bean实例化
38         finishBeanFactoryInitialization(beanFactory);
39
40         //最后一步，发布ContextRefreshedEvent事件
41         finishRefresh();
42     }
43 }
```

b.

14. 网络相关：

- a. tcp三次握手、四次挥手
- b. http机制
- c. https机制

15. 项目亮点：

16. 画架构图：

17. 平时看过什么书？