1. **Java代码质量和规范**
2. **CheckStyle风格**

在对应的IDE中配置相应的checkstyle，推荐Clean Code，重构：改变代码质量两本书

* 1. 代码命名规范，代码中必须以驼峰规则命名
  2. 类，方法，变量的命名，必须规范化，表意明确，不要随意，让人误解，如List<String> list
  3. 一个方法，只做一件事，避免一个方法中啰里啰嗦一大堆，以简洁、明了，方法体最好不超过80行
  4. 不要在类，方法中使用过多的注释，影响整体美观，故再次强调方法的命名规范；不要使用没必要的注释，啰嗦，如Array.sort()—对数组排序

1. **代码质量**

推荐Effective Jave，深入理解JVM虚拟机第二版，了解JVM底层原理，掌握Java中常见数据结构的实现源码，编写高质量的代码

2.1 日志的埋点，避免在代码中大量日志的记录，减少频繁IO

2.2 异常处理，避免一个方法中大部分直接try…catch，代码的一次处理是很浪费性能的，只有在合适的地方使用

2.3 工具类，工具类表示一个项目中整体的工具使用，尽量以普遍，通俗，泛型的形式处理

2.4 设计模式，在适当的场景使用恰当的设计模式，必须掌握常用的设计模式

2.5 枚举使用，在适当，可穷举的情况下，使用枚举，避免直接使用final static 静态变量，枚举的好处，可以自己google

2.6 错误信息的展示，不要在代码中使用硬编码，可以用枚举，常量替代，对于错误信息，最好本地化配置

2.7 内存泄露，避免使用长生命周期对短生命周期的使用

2.8 连接数，合理分配数据库连接池和定时Job的连接数，使其达到平衡

2.9 适当使用算法替换代码中执行效率低得场景

3.0 服务治理，模块化，降低多模块话之间的耦合度，做到可插拔式模块替换及扩展