**A. 强制**

1. **包名小写且使用公司/项目域名反写**，如 com.example.project。
2. **类名使用大驼峰**（PascalCase），且不得以下划线或特殊字符开头。
3. **方法名使用小驼峰**（camelCase），描述性动词开头。
4. **常量使用全大写加下划线**，如 MAX\_BUFFER\_SIZE。
5. **严禁魔法数字和魔法字符串**，必须提取为常量，并加注释。
6. **每个类文件顶部必须包含版权和授权信息**，格式统一。
7. **单行最多 120 字符**，超出需拆分；禁止出现横向滚动条。
8. **所有公共 API、复杂逻辑必须写 Javadoc 注释**，包括参数、返回值、异常。
9. **不得在生产代码中留下 TODO、FIXME**，如有待办需新建 issue 并链接。
10. **所有 SQL 必须使用预编译语句（PreparedStatement、MyBatis 参数占位符）**，防止 SQL 注入。
11. **所有外部请求（数据库、RPC、HTTP）均需设置超时**，并适当重试或降级。
12. **不得捕获 Exception 或 Throwable**，只能捕获特定异常并做好处理或包装后抛出。
13. **所有可复用业务逻辑需编写单元测试，代码覆盖率不低于 80%**。
14. **禁止使用过时（@Deprecated）或被屏蔽的 API**，需及时升级或替换。
15. **代码提交前必须通过静态分析工具（Checkstyle/PMD/SpotBugs）和单元测试**，否则禁止合入主分支。

**B. 推荐**

1. **方法长度尽量控制在 50 行以内**，过长可拆分成私有子方法。
2. **避免多重嵌套（if/for/switch）超过三层**，可使用早返回或策略模式。
3. **使用 Java 8+ 特性**，如 Stream、Optional，但避免过度链式调用导致可读性差。
4. **推荐使用 Lombok 简化 POJO/DTO 的 getter/setter/构造函数**，并开启 IDE 插件支持。
5. **日志使用 SLF4J 接口**，避免直接使用具体实现；日志级别原则：ERROR/WARN/INFO/DEBUG。
6. **Service 层业务方法名推荐以动词+领域对象命名**，如 createOrder、cancelSubscription。
7. **Controller 层返回统一的响应体模型**，包含 code、message、data 三部分。
8. **配置信息（DB、MQ、API 密钥）必须通过配置中心/Register 或 environment variable 管理**，禁止硬编码。
9. **使用依赖注入（Spring @Autowired/@Qualifier）而非手动 new 对象**。
10. **对于跨服务调用，推荐使用异步、消息队列或熔断降级（Hystrix/Resilience4j）**。
11. **代码审查（PR）时应至少两人 Review，且 Review 人不得为代码作者本人。**
12. **推荐使用 Git Flow 或 trunk-based flow 分支模型，并严格执行命名规范（feature/bugfix/release）。**
13. **新功能上线前需在测试环境通过集成测试、性能测试与安全扫描。**
14. **推荐使用 Docker 容器化部署，且镜像体积需尽量瘦化。**
15. **对外接口文档推荐使用 OpenAPI/Swagger 自动生成并同步更新。**
16. **推荐使用统一的异常码规范（例如 6 位数字），并在全局异常处理器中统一返回。**
17. **性能敏感场景下，推荐使用 Caffeine/Redis 缓存并设置合理的过期策略。**
18. **推荐定期组织代码评审会议（Code Review Meeting），分享最佳实践。**

**C. 允许**

1. **可根据项目需要选用 Maven 或 Gradle 构建工具。**
2. **可使用 Guava、Apache Commons 等第三方库，但需评估依赖体积及维护情况。**
3. **在小工具脚本中可适当使用静态导入（static import），但生产代码慎用。**
4. **对于简单的枚举，可不写 Javadoc，但需保证命名清晰。**
5. **测试中可使用 Mockito/PowerMock 等模拟框架。**
6. **微服务之间可使用 REST 或 gRPC，视团队技术栈和性能要求而定。**
7. **可选用 MapStruct、Dozer 等工具进行对象映射。**
8. **可根据项目需求使用事务注解 @Transactional 的不同传播级别。**
9. **允许在非核心路径使用日志占位符 {} 方式拼接。**
10. **允许在开发环境开启 SQL 日志和慢查询日志，生产环境中需关闭或限流输出。**
11. **对于低风险实验性模块，可适当放宽提交前测试覆盖率要求，但必须事后补齐。**
12. **允许团队内部根据项目特点补充细化本规范，但需在 Wiki/文档中心登记并公开。**