

java 技术管理文档

一、命名规范

1. 强制：类名使用 UpperCamelCase 风格（如 UserService），接口名可为名词或形容词（如 Runnable）。
2. 强制：方法名、变量名使用 lowerCamelCase 风格（如 getUsername()）。
3. 强制：常量命名全大写，下划线分隔（如 MAX_RETRY_COUNT）。
4. 强制：包名全小写，使用单数形式（如 com.example.util）。
5. 推荐：接口实现类加 Impl 后缀（如 CacheServiceImpl）。
6. 推荐：工具类以 Utils 结尾（如 StringUtils）。
7. 允许：测试类以 Test 结尾（如 UserServiceTest）。

二、格式规范

8. 强制：缩进采用 2 个空格（Google 规范）或 4 个空格（阿里规范），禁用 Tab。
9. 强制：即使单行代码块也需使用大括号（如 if (condition) { ... }）。
10. 强制：行宽不超过 120 字符（推荐 80~100）。
11. 推荐：类成员顺序为：静态变量 > 实例变量 > 构造方法 > 公有方法 > 私有方法。
12. 允许：数值字面量用下划线分隔（如 1_000_000）。

三、代码结构

13. 强制：@Override 注解必须用于覆写方法。
14. 强制：废弃方法需标注 @Deprecated 并说明替代方案。
15. 推荐：方法长度不超过 30 行，类长度不超过 500 行。
16. 推荐：优先使用卫语句(Guard Clauses)减少嵌套(如 if (condition) return;)。
17. 允许：使用设计模式并在类名中体现（如 OrderFactory）。
18. 强制：避免在循环体内执行远程调用或数据库查询（需批量获取数据后再处理）。

四、编程实践

19. 强制：包装类比较必须用 equals()（如 Integer.valueOf(1).equals(x)）。
20. 强制：禁止在 foreach 循环中修改集合（应使用 Iterator）。
21. 强制：SimpleDateFormat 需定义为线程局部变量（ThreadLocal）。
22. 推荐：常量按功能分类存放（如 CacheConsts、ConfigConsts），禁止大而全的常量类。
23. 推荐：使用 Objects.equals() 避免空指针。
24. 允许：POJO 的布尔属性命名避免 isXxx（用 success 而非 isSuccess）。
25. 强制：资源类（InputStream/Connection）必须使用 try-with-resources 自动关闭。
26. 推荐：优先使用 Enum 替代整型常量表示状态（如 OrderStatus.PAID）。

五、注释与文档

27. 强制：公共类、方法必须有 Javadoc，说明功能、参数及返回值。
28. 强制：所有魔法值（Magic Number）必须定义为常量。
29. 推荐：TODO 注释需标注负责人及截止日期（如 // TODO [LiMing] Fix by

2025-07-01)。

30. 允许：非公共方法可省略 Javadoc，但复杂逻辑需写单行注释。

六、工程实践

• 分层架构与领域模型

31. 强制：遵循分层架构 (Controller → Service → Manager → DAO)，禁止跨层调用。

32. 强制：Controller 层仅做参数校验与路由转发，业务逻辑下沉至 Service 层。

33. 推荐：领域模型转换规范：

层级	对象类型	示例
DAO 层	DO	UserDO
Service 层	DTO/BO	UserDTO
Controller 层	VO	UserVO

34. 允许：简单场景可合并 DTO 与 VO，避免过度转换。

• 并发与事务

35. 强制：高并发场景下，锁粒度应最小化（优先锁区块而非整个方法）。

36. 推荐：线程池需通过 ThreadPoolExecutor 显式创建，避免 Executors 默认方法。

37. 允许：并发修改概率>20%时用悲观锁，否则用乐观锁。

• 数据库与批量操作

38. 强制：批量操作需分组提交（如每 500 条一组）。

39. 强制：查询结果超过 5000 条必须分页。

40. 推荐：DAO 层方法命名规范：

- 获取单个对象：getXxx()
- 获取多个对象：listXxx()
- 插入数据：insertXxx()

• 工程实践规则

41. 强制：单元测试覆盖率核心逻辑≥70%，需包含边界条件测试。

42. 推荐：日志统一使用 SLF4J 门面，禁止直接依赖 Log4j/Logback 实现。

七、异常处理

43. 强制：禁止捕获 Throwable 或 Exception 基类，需捕获具体异常（如 IOException）。

44. 强制：异常信息需包含上下文（如 throw new UserNotFoundException("UserId:" + id)）。

45. 推荐：自定义异常以 Exception 结尾（如 ValidationException）。

关键原则总结：

安全 > 性能：资源关闭、线程安全优先于优化。

显式 > 隐式：避免魔法值、隐式类型转换。

可测性：通过分层和接口设计支持单元测试。