# Java 技术管理规范文档

## a. 强制（必须遵守）

1. 1. 使用驼峰命名法（CamelCase）：类名首字母大写，方法和变量名首字母小写。
2. 2. 一个类一个文件：每个`.java`文件只包含一个顶级公共类。
3. 3. 每个类必须有明确的访问控制符（public、protected、private）。
4. 4. 禁止使用通配符导入（`import xxx.\*`）。
5. 5. 常量必须为 `static final` 且使用全大写+下划线命名。
6. 6. 每个类、接口必须包含 Javadoc 注释。
7. 7. 类名必须为名词，方法名为动词。
8. 8. 异常必须明确捕获并处理，禁止空`catch`块。
9. 9. 不允许提交带有 TODO 的代码进入主分支。
10. 10. 使用 4 个空格缩进，不允许使用 Tab。
11. 11. 代码中不允许出现 System.out.println()，统一使用日志框架（如 SLF4J）。
12. 12. 禁止使用过时（@Deprecated）的API。
13. 13. 禁止在业务代码中使用魔法值，必须提取为常量。
14. 14. 方法长度不超过 100 行，类长度不超过 1000 行。
15. 15. 禁止 catch(Exception) 或 catch(Throwable)，必须捕获具体异常类型。
16. 16. 开关语句（switch）必须包含 default 分支。
17. 17. 禁止在一个类中定义多个 public class（或 interface）。
18. 18. 禁止在构造函数中调用可被子类重写的方法。
19. 19. 集合初始化必须指定泛型参数，禁止使用原始类型（如 `List list = new ArrayList();`）。
20. 20. 禁止出现重复代码（DRY原则）。

## b. 推荐（建议遵守）

1. 1. 优先使用接口而不是具体类进行编程（面向接口编程）。
2. 2. 优先使用不可变对象（如 `final` 修饰），特别是在多线程环境中。
3. 3. 构造函数参数超过 5 个建议使用构建者模式（Builder）。
4. 4. 字段的默认访问权限为 private，提供 getter/setter 访问。
5. 5. 日志输出采用占位符方式，不直接使用字符串拼接。
6. 6. 尽量避免使用静态变量共享状态（除常量外）。
7. 7. 方法参数不超过 4 个，超过建议封装为对象。
8. 8. 工具类应为 final 且包含私有构造函数。
9. 9. 使用 Optional 表达可能为 null 的返回值。
10. 10. 单元测试覆盖率应不低于 80%，重点模块需达 90%+。
11. 11. 枚举类型应当只包含业务无歧义的常量集合。
12. 12. 优先使用 try-with-resources 来管理资源。
13. 13. 使用 Lombok 需严格控制范围，核心逻辑代码禁止使用 @Data。
14. 14. if-else 嵌套不超过 3 层，超过建议重构为策略类或状态模式。
15. 15. 异步操作必须加超时控制，避免资源悬挂。

## c. 允许（可视情况采用）

1. 1. 使用 Stream API 替代传统循环，但需关注可读性和性能。
2. 2. 可使用断言（assert）进行开发阶段的校验，但不用于生产逻辑中。
3. 3. 允许根据项目特性使用注解驱动的开发方式（如 Spring 注解）。
4. 4. 可使用枚举实现策略模式、状态机等高级结构。
5. 5. 代码生成工具（如 MapStruct、Lombok）在代码审查通过后可允许使用。