**Java 技术管理文档（技术规范 V1.0）**

## 编写目的

本技术管理文档是我在学习 Java 编程过程中，根据自己平时写代码的经验、课程所学内容，以及参考一些大厂（如 Google、阿里巴巴）代码规范，整理出的一份简化版 Java 编程规范指南。目的是为了加深对编程规范的理解，也方便团队协作和代码维护。文档按照**强制（a）、推荐（b）、允许（c）**三个等级对常见编码习惯进行了分类，共收录41条具体规则，涵盖命名、格式、编程实践、注释、测试和依赖管理等方面内容。希望通过这次整理，不仅能规范自己的编码习惯，也能为类似课程项目提供参考。

## 命名规范

【a】类名使用大驼峰（PascalCase），必须是名词，如：UserService、OrderDetail。

【a】接口名加I前缀（例：IUserService）。

【a】方法名、变量名使用小驼峰（camelCase），方法名必须为动词或动词短语。

【a】常量名必须全部大写，单词之间以下划线分隔，例如：MAX\_RETRY\_COUNT。

【a】包名必须全小写，使用反域名结构，例如：com.company.project.module。

【b】接口命名推荐使用形容词或名词，如：Readable、Logger，避免加 I 前缀。

【b】测试类命名应以被测类 + Test 命名，如 OrderServiceTest。

【b】布尔类型变量以 is、has、can 开头，如 isValid、hasPermission。

【b】临时变量使用语义明确的短名，如 i、j、tmp 只在小作用域内使用。

## 代码风格与格式化

【a】每行代码不得超过 120 个字符，超过需换行。

【a】使用 4 个空格缩进，不允许使用 Tab字符。

【a】类、方法之间必须使用空行隔开。

【a】左大括号{不得另起一行，必须与语句同行。

【b】在 IDE 中统一使用 Prettier 或 Google Java Format 插件格式化代码。

【b】尽量避免在一行中声明多个变量。

【b】使用枚举（enum）替代魔法值（magic number）。

【b】使用 final 修饰不变引用变量。

【b】尽量使用 Lambda 表达式简化函数式接口实现。

【b】三元运算符尽量只在简单逻辑中使用，避免嵌套。

## 编程实践

【a】不得在业务代码中出现 System.out.println()，统一使用日志框架如 SLF4J。

【a】禁止在项目中使用原生态的 Thread，统一使用线程池（Executors）管理线程。

【a】敏感逻辑必须加空指针检查（如 if (obj != null)）。

【a】禁止捕获 Exception 后不处理，最少应记录日志。

【b】控制类的大小，一个类建议不超过 800 行。

【b】方法的功能保持单一，一个方法不超过 80 行。

【b】对于复杂逻辑应进行模块封装，不应在 Controller 中出现业务逻辑。

【b】使用 Optional 处理可能为空的返回值。

【b】优先使用集合工具类（如 Collections.emptyList()）避免返回 null。

【b】避免在循环中频繁创建对象，注意性能优化。

【c】适度使用注解简化配置（如 Lombok），但应评估可读性和团队接受程度。

【c】可使用记录类型（record）表示简单的数据结构，前提是 Java 版本支持。

## 注释与文档

【a】所有类、接口、枚举必须添加类级 Javadoc 注释。

【a】公共方法必须添加方法注释，包括参数、返回值、异常。

【b】必要时对复杂逻辑添加行内注释，避免晦涩代码。

【b】使用 @since 标注类/方法自哪个版本引入。

## **测试与工具**

【a】新功能上线前必须覆盖单元测试，测试覆盖率不得低于 80%（核心模块≥95%）。

【a】禁止将测试用例代码混入生产代码路径。

【b】使用 JUnit 5 和 Mockito 编写单元测试。

【b】通过 SonarQube 等工具持续进行代码质量扫描。

## 依赖与构建

【a】所有依赖版本必须固定，不得使用 SNAPSHOT 版本发布。

【a】构建工具统一使用 Maven 或 Gradle，不得混用。