

# Java 技术管理规范文档（v1.0）

## 一、命名规范

### a. 强制（必须遵守）

1. 类名必须使用大驼峰（PascalCase），例如：OrderService。
2. 变量名必须使用小驼峰（camelCase），例如：orderId。
3. 常量必须全部大写，单词间用下划线分隔，例如：MAX\_RETRY\_COUNT。
4. 包名必须全部小写，使用反域名格式，例如：  
com.example.project.module。
5. 接口名称必须以 I 为前缀，或以 able、ible 结尾，例如：IUserService  
或 Serializable。
6. 枚举类型必须使用单数形式的类名，内部常量必须大写。
7. 方法名必须动词开头，例如：getUserById()。

### b. 推荐（鼓励遵守）

8. 测试类命名应以被测试类名开头加上 Test 后缀，例如：  
UserServiceTest。
9. 使用简洁有意义的变量名，避免 tmp、foo、bar。
10. 使用复数形式表示集合，例如：users、orderList。
11. 不推荐缩写，除非众所周知（如 URL，ID）。
12. 使用 Constants 或 Enums 封装常量。

### c. 允许（可选遵循）

13. 工具类可使用 Util 或 Helper 结尾，例如：DateUtil。
14. DTO/VO 命名可使用 DTO、VO 后缀，例如：UserDTO。
15. 接口命名可以不带 I，但实现类应以 Impl 结尾，例如：  
UserService/UserServiceImpl。

## 二、代码格式

### a. 强制

- 16. 每行不得超过 100 个字符，建议断行。
- 17. 使用 4 个空格缩进，不使用 Tab。
- 18. 大括号不得省略，必须换行风格如下：

```
1. if (condition) {  
2.     // ...  
3. }
```

- 19. 所有文件必须使用 UTF-8 编码，无 BOM。
- 20. 每个类的开头必须有类级注释，说明作者、创建时间、用途。

## b. 推荐

- 21. 方法之间空一行，增强可读性。
- 22. import 顺序应为：标准库 > 第三方库 > 项目内部，之间用空行隔开。
- 23. 代码块之间适当使用空行，逻辑更清晰。
- 24. 一个类尽量不超过 500 行。
- 25. 方法参数不超过 5 个，超出应封装为对象。

## c. 允许

- 26. 可使用 IDE 格式化工具，但需统一配置。
- 27. 可使用 Lombok，但必须代码审查通过。

# 三、编程规范

## a. 强制

- 28. 禁止出现任何 IDE 自动生成的无用代码（如无效注释、模板代码）。
- 29. 禁止使用魔法值，所有常量必须定义为 static final 常量。
- 30. 必须显式指定访问修饰符（如 private/protected/public）。
- 31. 异常必须捕获或抛出，不能忽略异常。
- 32. 必须使用泛型，禁止使用裸类型，例如：List list。
- 33. 禁止捕获异常后什么都不做，至少应记录日志。
- 34. 不允许使用 System.out.println，必须使用日志框架（如 SLF4J）。

## b. 推荐

- 35. 使用 try-with-resources 关闭资源。
- 36. 合理拆分方法，每个方法控制在 50 行以内。

- 37. 控制类的职责，一个类只处理一个领域问题（SRP 原则）。
- 38. 使用 `final` 修饰不变变量和参数。
- 39. 使用 `Optional` 处理可能为空的返回值。

#### c. 允许

- 40. 可以使用 Java 8 Stream，但需保证性能与可读性。
- 41. 可以使用记录类（record）简化 DTO。

## 四、注释规范

#### a. 强制

- 42. 所有公共 API 必须添加 JavaDoc，包括方法描述、参数说明、返回值说明。
- 43. 所有复杂业务逻辑必须写注释说明其背景和关键逻辑

#### b. 推荐

- 44. 注释尽量简洁准确，避免无意义注释如 `// getter method`。
- 45. 使用 `TODO`, `FIXME` 标签标注待处理问题。
- 46. 为单元测试方法加注释，说明测试目的与覆盖场景。

#### c. 允许

- 47. 可在类/方法中使用注释块（例如 `/** 业务逻辑说明 */`）做逻辑分段说明。

## 五、测试与部署

#### a. 强制

- 48. 所有代码必须编写单元测试，覆盖率不低于 80%。
- 49. 不能在生产代码中保留任何测试逻辑（如 `main` 方法用于测试）。

#### b. 推荐

- 50. 单元测试应遵循 Given-When-Then 模式书写，提升可读性和结构性。