### **Java 技术管理规范**

一、代码风格

1.a 缩进使用 4 个空格（禁用 Tab）。

2.a 类名使用 UpperCamelCase（如 UserService），方法/变量名使用 lowerCamelCase（如 getUserId()）。

3.a 常量名全大写+下划线（如 MAX\_RETRY\_COUNT）。

4.b 每行不超过 120 字符（超长时换行对齐参数）。

5.a if/for 等语句即使单行也需加 {}。

6.c 允许链式调用换行（每行一个方法，. 开头对齐）。

二、注释与文档

7.a 公共类、方法必须用 Javadoc 注释（说明功能、参数、返回值）。

8.b 废弃代码用 @Deprecated 标注并说明替代方案。

9.a 复杂逻辑必须添加行内注释（解释意图，而非重复代码）。

10.c 允许使用 // TODO 标注待办项（需指定负责人/日期）。

三、异常处理

11.a 禁止捕获 Exception 等通用异常（需捕获具体异常如 IOException）。

12.a 异常信息必须包含上下文（如 throw new UserNotFoundException("User ID: " + id)）。

13.b 推荐使用自定义异常（如 PaymentFailedException）。

14.a try-with-resources 管理资源（实现 AutoCloseable 的类）。

四、并发与线程

15.a 线程池必须通过 ThreadPoolExecutor 显式创建（避免 Executors 隐藏风险）。

16.a 共享变量访问必须同步（用 synchronized、Lock 或 Atomic 类）。

17.b 推荐使用 ConcurrentHashMap 代替同步的 HashMap。

18.c 允许使用 volatile 修饰轻量级共享状态。

五、安全与防御

19.a SQL 语句必须使用 PreparedStatement（禁止字符串拼接）。

20.a 用户输入需显式校验（如 StringUtils.isNotBlank(input)）。

21.a 禁止日志输出敏感信息（密码、密钥等）。

22.b 推荐使用 SSL/TLS 加密网络传输。

六、性能优化

23.a 避免在循环中调用 + 拼接字符串（用 StringBuilder）。

24.b 集合初始化指定容量（如 new ArrayList<>(100)）。

25.a 及时关闭数据库连接、IO 流（在 finally 块或 try-with-resources 中）。

26.c 允许缓存频繁访问的只读数据（如 Guava Cache）。

七、OOP 与设计

27.a 类成员变量必须 private（通过 getter/setter 访问）。

28.b 推荐使用组合代替继承（除非是 “is-a” 关系）。

29.a 接口方法不允许 public 修饰（默认隐含）。

30.a 工具类需私有化构造器（避免实例化）：

八、测试规范

31.a 单元测试需覆盖核心逻辑（覆盖率 ≥ 80%）。

32.b 测试方法名应表达意图（如 testTransferMoney\_WhenInsufficientBalance\_ThrowsException()）。

33.a 禁止测试代码连接真实数据库（用内存数据库如 H2）。

34.c 允许使用 Mock 框架（如 Mockito）模拟外部依赖。

九、工程管理

35.a 提交代码前必须通过本地构建（mvn clean install）。

36.a 依赖版本必须固定（禁用 SNAPSHOT 版本上生产）。

37.b 推荐使用 Lombok 减少样板代码（需团队统一）。

38.a 禁止提交调试代码（如 System.out.println()）。

十、其他

39.b 推荐枚举代替整型常量（如 enum Status { ACTIVE, INACTIVE }）。

40.a equals() 和 hashCode() 必须同时重写。

41.c 允许使用 Optional 避免 NullPointerException。

42.a 使用 @Override 注解标记重写方法。