**Java 技术管理规范**

**版本：** 1.1  
**生效日期：** 2025-06-04

**A. 强制（必须遵守）**

1. 所有类、方法必须添加 **@NonNull** 或 **@Nullable** 注解（JSR 305）。
2. 禁止使用 **==** 比较对象（除枚举外），必须用 **equals()**。
3. 单元测试覆盖率 ≥80%，核心模块 ≥90%，未达标禁止合并。
4. 异常处理必须捕获具体类型，禁止 **catch (Exception e)**。
5. 资源操作（IO/DB）必须使用 **try-with-resources**。
6. 所有 API 返回值必须用 **Optional<>** 包装可能为空的结果。
7. 日志必须使用 SLF4J + Logback，禁止 **System.out.println()**。
8. 并发场景必须使用线程安全集合（**ConcurrentHashMap**/**CopyOnWriteArrayList**）。
9. 禁止使用 **new Date()** 或 **SimpleDateFormat**（强制使用 **java.time.\***）。
10. 数据库操作必须使用 PreparedStatement，禁止拼接 SQL。
11. 配置文件必须通过 **Spring Config** 或 **Archaius** 管理，禁止硬编码。
12. 服务接口响应时间超过 500ms 必须优化。
13. 所有公开方法必须用 Javadoc 注释（含参数、返回值、异常说明）。
14. 禁止提交 **//TODO** 或 **//FIXME** 注释。
15. 每行代码不超过 120 字符。

**B. 推荐（最佳实践）**

1. 优先使用构造器注入（**@Autowired** 标注在构造器上）。
2. 使用 **Records**（Java 16+）替代纯数据类。
3. 集合返回值优先返回不可变集合（**Collections.unmodifiableList()**）。
4. 使用 **@VisibleForTesting** 限制测试专用方法作用域。
5. 并发控制优先用 **java.util.concurrent** 而非 **synchronized**。
6. 配置文件格式优先级：YAML > Properties > JSON。
7. 使用 **Builder 模式** 构造复杂对象（Lombok **@Builder** 可用）。
8. 优先使用 **Stream API** 替代循环（保证可读性前提下）。
9. 常量定义必须使用 **static final**，命名全大写（如 **MAX\_RETRIES**）。
10. 跨服务调用设置超时（HTTP ≤3s，RPC ≤1s）。
11. 使用 **AssertJ** 替代 JUnit 原生断言。
12. 优先使用 **var** 声明局部变量（Java 10+）。
13. 定期运行 **SpotBugs** 和 **OWASP Dependency-Check**。
14. 线程池必须自定义命名（通过 **ThreadFactory**）。
15. 使用 **final** 修饰方法参数和局部变量（避免意外修改）。
16. 优先使用 **Enum** 替代整型常量。
17. DTO 类必须实现 **Serializable** 并显式定义 **serialVersionUID**。

**C. 允许（团队自决）**

1. 允许在内部工具中使用 Lombok（但禁止 **@Data**）。
2. 允许使用 **switch** 表达式（Java 14+）。
3. 允许在性能关键路径使用 **sun.misc.Unsafe**（需安全评审）。
4. 允许使用 **CompletableFuture** 处理异步流程。
5. 允许在测试代码中使用 Mockito 或 Spock。
6. 允许使用 **@SuppressWarnings** 抑制警告（需附原因注释）。
7. 允许在工具类中使用静态方法。
8. 允许使用 **Optional.orElseThrow()** 替代显式 **null** 检查。
9. 允许在数据转换场景使用 MapStruct。
10. 允许使用 **-parameters** 编译参数保留方法参数名。

**工具链强制要求**

- 构建工具：Maven ≥ 3.8 或 Gradle ≥ 8.0

- CI/CD：Jenkins 或 GitLab CI（必须包含 Checkstyle/PMD/Test 阶段）

- 依赖管理：禁止 SNAPSHOT 依赖进入生产镜像

- 静态检查：ErrorProne + Checkstyle（Google规则）

- 部署：容器化（Docker + Kubernetes）

- 监控：Micrometer + Prometheus/Grafana

- 堆栈：Spring Boot ≥ 3.0（JDK 17+）或同等企业框架

**执行细则**

* 违反强制条款的代码不予合并
* 推荐条款审计计入 KPI
* 每季度同步 OpenJDK LTS 安全更新  
  文档维护：范奕珩