

Java 编程语言技术管理规范

级别：a、强制；b、推荐；c、允许

一、命名规范

规则	级别	描述
1	a	类名使用大驼峰命名法（PascalCase），且应为名词，如 UserService。
2	a	变量名使用小驼峰命名法（camelCase），不得包含下划线。
3	a	常量名必须全部大写，单词之间使用下划线分隔，如 MAX_SIZE。
4	b	包名一律使用小写，不使用下划线，推荐按公司域名倒序组织结构，如 com.example.utils。
5	b	接口命名以功能名为主，不强制加 I 前缀，如 UserRepository。
6	b	测试类应以被测试类名 + Test 结尾，如 UserServiceTest。

二、格式与结构

规则	级别	描述
7	a	每个文件最多只包含一个顶级公共类。
8	a	每个类必须包含类注释，说明功能、作者和日期。
9	a	使用 4 个空格缩进，禁止使用 tab。
10	b	每行代码不应超过 100 个字符。
11	a	成员变量、构造函数、方法之间用一个空行隔开。
12	a	方法参数或调用方法时，如果参数过多，应分行排列，每行一个参数。
13	b	控制语句（如 if/for）必须使用大括号 {}，即使只有一行。

三、注释规范

规则	级别	描述
14	a	所有公共方法必须添加 Javadoc 注释。

规则	级别	描述
15	b	对于复杂的逻辑或特殊处理，必须写单行注释说明。
16	c	可以使用 TODO 注释标记待完善功能，需说明原因。
17	b	不允许出现无意义注释，如 <code>// getter</code> 方法。
18	a	删除代码前不应注释，而应通过版本控制查看历史。

四、异常与日志处理

规则	级别	描述
19	a	不允许捕获 <code>Exception</code> 或 <code>Throwable</code> ，应捕获具体异常类型。
20	a	日志记录必须使用统一的日志框架（如 <code>SLF4J</code> ）。
21	b	异常抛出应携带上下文信息，方便排查问题。
22	a	不得在生产代码中使用 <code>System.out.println</code> 。
23	b	日志中不得记录敏感信息（如密码、手机号）。

五、面向对象设计规范

规则	级别	描述
24	a	成员变量一律使用 <code>private</code> 访问权限，通过 <code>getter/setter</code> 控制访问。
25	a	禁止在构造函数中执行复杂逻辑，应尽量保持轻量。
26	b	一个类尽量只做一件事，符合单一职责原则。
27	b	不得随意使用 <code>static</code> ，应评估其线程安全性和可测试性。
28	b	不建议使用继承实现代码复用，优先使用组合。

六、集合与泛型

规则	级别	描述
29	a	遍历集合时优先使用增强 <code>for</code> 循环。
30	b	泛型使用时应明确类型，避免使用原始类型（如 <code>List list</code> ）。
31	a	使用集合前应指定容量，防止扩容浪费。
32	b	判断集合是否为空应使用 <code>CollectionUtils.isEmpty</code> 或 <code>isEmpty()</code> 。

七、多线程与并发

规则	级别	描述
33	a	多线程操作共享资源必须进行同步控制。
34	a	不得直接使用 Thread 创建线程，统一使用线程池。
35	b	使用 volatile 前应充分理解其语义，防止误用。
36	b	原子操作优先使用 Atomic 类（如 AtomicInteger）。

八、构建与依赖管理

规则	级别	描述
37	a	必须使用统一的构建工具（如 Maven 或 Gradle）。
38	b	依赖版本需锁定，防止依赖冲突。
39	b	避免在 pom.xml 中引入未使用的依赖项。
40	a	发布版本的代码不得依赖 SNAPSHOT 版本。

九、单元测试与代码质量

规则	级别	描述
41	a	所有业务逻辑类必须至少配有基础的单元测试类。
42	b	单元测试代码与生产代码保持 1:1 的文件结构。
43	a	使用断言库（如 AssertJ / JUnit5 Assertions）进行验证，避免 if 手动判断。
44	b	遵循测试命名规范：方法名 + 条件 + 预期结果，如 login_withValidUser_shouldSucceed()。
45	b	测试方法应保持幂等，不依赖运行顺序或外部环境。

十、安全编码规范

规则	级别	描述
46	a	严禁直接拼接 SQL 语句，必须使用预编译语句（PreparedStatement）或 ORM 框架。

规则	级别	描述
47	a	所有输入参数必须进行有效性校验和类型转换，避免注入漏洞。
48	b	HTTP 请求必须验证用户身份（Token、Session、OAuth 等）。
49	a	业务系统中不得明文存储或传输用户密码。密码加密必须使用盐值哈希算法（如 bcrypt）。
50	b	使用 JWT 时，必须设置过期时间，并避免在 token 中存敏感信息。

十一、RESTful API 规范

规则	级别	描述
51	a	使用标准 HTTP 方法：GET 查询、POST 创建、PUT 修改、DELETE 删除。
52	b	接口路径使用复数资源名，如 /users/123，避免动词，如 /getUserInfo。
53	b	返回结构统一，建议封装为标准格式：code、message、data。
54	b	错误状态应返回标准 HTTP 状态码，如 400、401、403、404、500。
55	b	所有 API 应写清楚文档，建议使用 Swagger/OpenAPI 自动生成。

十二、代码可维护性与可读性

规则	级别	描述
56	a	方法长度不得超过 80 行，超过应拆分子方法。
57	b	类的行数不超过 1000 行，建议模块化分拆。
58	a	方法的参数不超过 5 个，复杂参数应使用对象封装。
59	b	嵌套层级不应超过 3 层，避免深层 if-else。
60	b	注重语义表达，避免使用无意义的变量名，如 a1, temp, data1。

十三、项目结构与模块划分

规则	级别	描述
61	a	项目结构应符合分层架构（如 controller/service/repository）。
62	b	每个模块需明确职责边界，不允许跨模块随意引用内部类。
63	b	公共工具类应集中管理，不得散落在业务模块中。

规则 级别		描述
64	b	domain/entity 层应为 POJO，不应引入业务逻辑。
65	c	可使用 Lombok 简化样板代码，但需评估维护风险。
