**Java语言技术管理文档：**

**代码风格**

a. 强制

1. 必须使用4个空格缩进，禁止使用Tab
2. 每行代码不超过100个字符
3. 类名必须使用大驼峰式命名
4. 方法名和变量名必须使用小驼峰式命名
5. 常量名必须全部大写，单词间用下划线分隔
6. 必须使用大括号包裹所有控制语句体，即使只有一行
7. 必须为所有public类和成员编写Javadoc注释
8. 必须遵循Java标准库的import顺序：java > javax > 第三方 > 项目内部
9. 必须使用@Override注解来标记覆盖的方法
10. 必须使用final修饰不可变变量

b. 推荐

1. 推荐使用空行分隔逻辑相关的代码块
2. 推荐将成员变量声明在类顶部，方法声明在底部
3. 推荐使用增强for循环而非传统for循环
4. 推荐使用try-with-resources处理资源
5. 推荐使用静态导入仅用于常量类
6. 推荐使用Optional而非null作为返回值
7. 推荐使用枚举替代整数常量
8. 推荐使用标准库中的函数式接口
9. 推荐使用var声明局部变量(Java 10+)
10. 推荐使用空行分隔类成员和方法

c. 允许

1. 允许在简单情况下使用三元运算符
2. 允许在switch语句中使用箭头语法(Java 14+)
3. 允许在简单情况下使用方法链
4. 允许在测试代码中使用单字符变量名
5. 允许在lambda表达式中省略参数类型
6. 允许在记录类中使用紧凑构造函数(Java 16+)
7. 允许在模式匹配中使用instanceof(Java 16+)
8. 允许在文本块中使用多行字符串(Java 15+)

**异常处理**

a. 强制

1. 必须捕获特定异常而非通用的Exception
2. 必须处理或记录所有捕获的异常
3. 必须在finally块中释放资源

b. 推荐

1. 推荐使用自定义异常来封装领域特定错误
2. 推荐在异常中包含有意义的错误信息
3. 推荐使用异常链来保留原始异常

c. 允许

1. 允许在测试代码中抛出RuntimeException

**并发编程**

a. 强制

1. 必须同步访问共享可变数据
2. 必须使用线程安全的数据结构或同步机制

b. 推荐

1. 推荐使用ExecutorService而非直接创建线程
2. 推荐使用并发集合而非同步包装器

c. 允许

1. 允许在性能关键路径中使用原子变量

**测试规范**

a. 强制

1. 单元测试类名必须以Test结尾（如UserServiceTest）
2. 测试方法必须使用@Test注解，并且方法名应描述测试行为（如shouldThrowExceptionWhenInputIsNull）
3. 必须使用断言（如assertEquals、assertTrue）而非System.out或日志验证结果
4. 测试代码必须独立运行，不能依赖外部环境（如数据库、网络）
5. 必须使用@BeforeEach和@AfterEach管理测试资源，而非@Before和@After（JUnit 5+）

b. 推荐

1. 推荐使用Mock框架（如Mockito）替代真实依赖
2. 推荐测试覆盖率至少达到80%（关键业务代码应接近100%）
3. 推荐使用参数化测试（@ParameterizedTest）减少重复代码

c. 允许

1. 允许在集成测试中使用真实数据库或外部服务
2. 允许在性能测试中使用@RepeatedTest或@Timeout

**代码质量**

a. 强制

1. 必须使用final修饰不可变变量（包括方法参数）
2. 必须避免在循环内创建对象（如new StringBuilder()）
3. 必须使用Objects.equals()进行对象比较，避免NullPointerException

b. 推荐

1. 推荐使用@Nullable和@NonNull注解（如Lombok或JetBrains）
2. 推荐使用record（Java 16+）替代简单的POJO

c. 允许

1. 允许在性能优化场景下使用StringBuilder而非+拼接字符串
2. 允许在特定场景下使用System.arraycopy()替代循环复制数组

**并发与线程安全**

a. 强制

1. 必须使用ConcurrentHashMap而非synchronizedMap（高并发场景）
2. 必须避免在synchronized块内调用外部方法（防止死锁）