**一、代码风格**

A 使用4个空格缩进，禁止使用Tab字符。

A 类名使用UpperCamelCase，方法/变量名使用lowerCamelCase，常量使用UPPER\_SNAKE\_CASE。

A 每行代码不超过120字符，超长需换行对齐（参数对齐或操作符后换行）。

B 大括号{}遵循K&R风格：左括号不换行，右括号换行。

B 代码块缩进必须一致，IDE需配置自动格式化（如Checkstyle）。

C 允许链式调用换行，但每个方法调用需单独一行。

**二、命名规范**

A 包名全小写，使用单数名词（如 com.example.service）。

A 接口名不加前缀（如 UserService），抽象类以Abstract开头。

B 布尔变量名以is/has/can开头（如 isValid）。

B 测试类以Test结尾（如 UserServiceTest）。

C 允许局部变量使用缩写（如 cnt代替count），但需保持上下文一致。

**三、异常处理**

A 禁止捕获Throwable或Exception（应捕获具体异常如IOException）。

A 异常信息需包含上下文（如 throw new UserNotFoundException("User ID: " + id)）。

B 使用JDK7+的try-with-resources管理资源（如流、连接）。

B 日志记录异常时使用log.error("Context: {}", message, exception)。

C 允许在框架层统一捕获未处理异常（如Spring @ControllerAdvice）。

**四、并发安全**

A 线程池必须通过ThreadPoolExecutor显式创建，禁止使用Executors快捷方法。

A 共享变量需用volatile、锁（ReentrantLock）或原子类（如AtomicInteger）。

B 使用ConcurrentHashMap代替同步的HashMap。

B 避免在锁内调用外部方法（防止死锁）。

C 允许使用ThreadLocal，但需确保及时清理（避免内存泄漏）。

**五、OOP与设计**

A 类成员变量禁止public修饰（用private+Getter/Setter）。

A 覆写方法必须加@Override注解。

B 优先使用组合而非继承。

B 接口方法单一职责（避免“上帝接口”）。

C 允许使用Optional替代null返回值，但禁止用于成员变量。

**六、性能优化**

A 避免在循环内执行数据库查询或远程调用。

A 使用StringBuilder替代字符串拼接（循环内必须使用）。

B 集合初始化指定容量（如 new ArrayList<>(100)）。

B 缓存频繁访问的数据（如Guava Cache），但需定义失效策略。

**七、注释与文档**

A 公共方法必须用Javadoc注释（说明作用、参数、返回值）。

A 弃用代码用@Deprecated标注并说明替代方案。

B 复杂算法需添加行内注释解释逻辑。

C 允许使用//临时注释调试代码，但提交前需移除。

**八、工程与依赖**

A 禁止循环依赖（Maven/Gradle需通过mvn dependency:analyze检测）。

A 第三方库版本统一由父POM管理。

B 使用slf4j作为日志门面，禁止直接调用Log4j/Logback。

B 新项目必须使用JDK LTS版本（如JDK 17+）。

C 允许在测试代码中使用@SuppressWarnings抑制警告。

**九、测试规范**

A 核心逻辑必须覆盖单元测试（JUnit5+Mockito），覆盖率≥80%。

B 测试用例名需明确（如 shouldReturnNullWhenUserNotFound()）。

C 允许集成测试使用内存数据库（如H2）。