一、基础规范

a 包名必须全小写，使用反域名格式（如 com.company.project.module）

a 类名采用大驼峰（UpperCamelCase），接口名可加 I 前缀（如 IUserService）

a 方法/变量名采用小驼峰（lowerCamelCase），常量名全大写+下划线（MAX\_COUNT）

b 推荐使用有意义的英文命名，禁止拼音/无意义缩写（如 a1, var2）

a 所有重写方法必须加 @Override 注解

b 推荐使用 final 修饰不可变变量（局部变量、成员变量）

a 禁止使用 import \*（通配符导入）

二、代码风格

a 缩进必须用4个空格（禁用Tab）

a 大括号换行规则：左括号不换行（if () {），右括号单独一行

b 单行字符数不超过120（推荐80-120）

a 运算符左右必须加空格（如 a + b）

c 允许链式调用换行（每个 . 换行并缩进）

a if/for/while 等即使单行也必须加 {}

b 推荐使用空行分隔代码块（方法间、逻辑段间）

三、面向对象

a 类成员变量必须私有化（private），通过方法访问

b 推荐优先使用组合而非继承

a 工具类（Utils）必须私有化构造器

b 推荐接口中声明默认方法（default）替代抽象类

c 允许使用 record（Java 16+）简化纯数据类

四、异常处理

a 禁止捕获 Exception 后忽略（空 catch 块）

a 自定义异常必须继承 RuntimeException（非受检异常）

b 推荐抛出具体异常而非 Throwable/Exception

a try-with-resources 必须用于资源类（InputStream 等）

c 允许在测试代码中抛出 Exception

五、并发安全

a 多线程共享变量必须用 volatile 或同步机制

b 推荐使用 java.util.concurrent 工具类（如 ThreadPoolExecutor）

a 禁止使用 java.util.Date（改用 java.time.\*）

b 推荐使用不可变对象（Immutable）减少锁竞争

a 线程池必须自定义命名（通过 ThreadFactory）

六、资源与IO

a 流操作后必须在 finally 或 try-with-resources 中关闭

b 推荐使用NIO（Files、Paths）替代传统IO

a 文件路径禁止硬编码（使用配置文件或常量）

七、集合与泛型

a 初始化集合时必须指定容量（如 new ArrayList<>(10)）

b 推荐使用 Collections.emptyList() 替代 new ArrayList()

a 泛型必须显式声明类型（禁止 new ArrayList()）

c 允许使用钻石操作符（<>）简化泛型（Java 7+）

八、现代语法

b 推荐使用 Optional 替代 null 检查

a switch 表达式必须覆盖所有枚举值（或含 default）

b 推荐使用 var 声明局部变量（Java 10+，需明显类型）

a 禁止使用已弃用（@Deprecated）的API

九、测试与文档

a 单元测试类命名：被测试类名 + Test（如 UserServiceTest）

b 推荐使用JUnit 5+ 和 AssertJ 断言库

a 公有方法必须用Javadoc注释（含参数、返回值说明）

c 允许测试方法使用非标准命名（如 test\_validInput()）

十、工具与流程

a 必须使用静态分析工具（如SonarQube、Checkstyle）

b 推荐启用Lombok减少样板代码

a 提交代码前必须通过CI流水线（编译、测试、代码扫描）