# 一、命名规范

* 【a】
* 类名必须使用 PascalCase 命名，如 UserService。
* 方法名使用 camelCase，命名为动词或动宾结构。
* 常量必须全大写，单词间以下划线连接，如 MAX\_COUNT。
* 包名全小写，不能包含下划线，符合域名倒写规则。
* 所有命名不能以下划线或美元符号开始或结束。
* 不允许使用拼音或中文作为变量名。
* 【b】
* 局部变量尽量短小，但要具有语义。
* 包名建议使用公司域名倒写结构（如 com.example.module）。
* 在命名中标明设计模式（如 XxxFactory、XxxProxy）。
* 【c】
* 接口名可加 I 前缀（如 IService）。
* 内部类可不完全遵循公共类命名规范。
* 允许枚举类名带 Enum 后缀，枚举值全大写。

# 二、格式规范

* 【a】
* 使用 4 个空格缩进，禁止使用 Tab。
* 单行长度不超过 120 字符，b换行对齐缩进。
* if/while 等结构必须使用 {} 包裹。
* 左大括号与控制语句同行，不允许换行。
* 操作符两侧必须加空格。
* 所有 @Override 必须加注解。
* 【b】
* 单方法总长度建议不超过 80 行。
* 逻辑不同代码段之间添加空行提升可读性。
* 【c】
* 允许短 lambda 表达式单行展示。
* 接口内部可定义常量，但需明确其通用性。

# 三、注释规范

* 【a】
* 公共类和方法必须写 Javadoc 注释。
* 注释内容必须说明意图而非代码翻译。
* 【b】
* 私有方法、复杂逻辑需加块注释。
* TODO、FIXME 必须标明责任人或任务编号。
* 【c】
* 允许临时代码注释，但合并前需清理。

# 四、编码实践

* 【a】
* 禁止捕获异常后不处理。
* 使用 try-with-resources 管理资源。
* 禁止直接使用 '+' 拼接字符串于循环内。
* 公共方法必须 null 检查或使用 @NonNull。
* 【b】
* 优先使用 Optional 返回可能为空的对象。
* 静态工厂方法优于构造函数创建对象。
* 【c】
* 单方法长度尽量不超 50 行。

# 五、面向对象设计

* 【a】
* 每个类遵循单一职责原则。
* 类成员默认 private，通过方法访问。
* 【b】
* 建议接口隔离，职责拆分清晰。
* 优先使用组合而非继承。
* 【c】
* 允许工具类包含静态方法，但构造器需私有防实例化。

# 六、并发与性能

* 【a】
* 禁止直接 new Thread，统一使用线程池。
* 共享变量必须线程安全。
* 不可在循环中频繁创建对象或打开文件。
* 【b】
* 多线程场景下优先使用不可变对象。
* 【c】
* 根据业务选择锁粒度，如 ReadWriteLock。

# 七、安全规范

* 【a】
* 禁止硬编码密钥、密码等敏感信息。
* 禁止 SQL 拼接，使用预编译或 ORM。
* 【b】
* 建议对所有外部输入做参数合法性校验。
* 【c】
* 允许内部调试接口，但必须标注禁止生产。

# 八、测试与工程化

* 【a】
* 暴露接口必须覆盖单元/集成测试。
* 构建不通过禁止提交代码。
* 【b】
* 建议测试覆盖率 ≥ 80%。
* 【c】
* 开发中允许临时禁用测试，但必须加 TODO 注释。