一、编码规范

A 1.类名使用大驼峰（UpperCamelCase），接口名使用名词或形容词（如 Runnable）

A 2.方法/变量名使用小驼峰（lowerCamelCase），禁止拼音或无意义缩写

A 3.常量名全大写，单词间用下划线连接（MAX\_COUNT）

B 4.包名全小写，使用公司域名反写（com.company.module）

A 5.缩进统一用4个空格，禁止使用Tab

B 6.单行字符不超过120个，超出需换行对齐

A 7. if/for/while 等必须使用大括号，即使只有一行代码

二、异常与日志

A 8.禁止捕获异常后不处理（至少记录日志）

B 9.自定义异常需继承 RuntimeException（避免过度受检异常）

A 10.日志使用SLF4J门面，禁止System.out.println

B 11.日志内容需包含上下文信息（如ID、参数）

A 12.异常信息需明确业务含义（避免只打印 e.getMessage()）

三、并发安全

A 13.线程池必须通过 ThreadPoolExecutor 显式创建（避免Executors默认方法）

B 14.高并发场景下使用 ConcurrentHashMap 代替 Collections.synchronizedMap

A 15.静态变量需声明 final 或保证线程安全

B 16使用 Atomic 类替代 synchronized（如 AtomicInteger）

四、资源与内存

A 17.try-with-resources 必须用于所有 Closeable 资源（JDK7+）

B 18.大对象（如缓存）需软引用（SoftReference）或弱引用（WeakReference）

A 19.避免在循环内创建对象（如字符串拼接用 StringBuilder）

C 20.允许使用@SuppressWarnings但需注释原因

五、API设计

B 21.REST接口返回统一结构体（含code/data/message）

A 22.公开API必须使用Javadoc注释（包括参数、返回值说明）

B 23.接口版本号通过URL路径管理（如/v1/user）

A 24.敏感数据接口必须加密（HTTPS+参数签名）

六、现代语法规范

B 25.使用Optional避免NullPointerException

A 26.equals() 方法必须同时重写 hashCode()

B 27.优先用switch表达式（JDK14+）替代传统switch

A 28. Lambda表达式参数需显式声明类型（避免(a,b)->）

C 29.允许var用于局部变量（JDK10+），但需保证可读性

七、测试与质量

A 30.核心逻辑必须有单元测试（覆盖率≥80%）

B 31.测试类名规范：被测试类名 + Test（如 UserServiceTest）

A 32.禁止提交被@Ignore的测试用例

B 33.使用AssertJ代替JUnit原生断言（提高可读性）

八、工程化

A 34.所有依赖必须显式声明版本号（禁止继承父POM隐式版本）

B 35.多模块项目使用dependencyManagement统一管理依赖

A 36.生产代码禁止snapshot依赖

B 37.使用Lombok减少模板代码，但避免过度使用

九、其他

A 38.日期处理必须用java.time（JDK8+），禁用Date/Calendar

B 39.枚举类名用单数形式（如UserRole而非UserRoles）

A 40.序列化接口（Serializable）必须显式声明serialVersionUID

B 41.工具类应私有化构造器（避免实例化）

C 42.允许方法返回空集合（如Collections.emptyList()），禁止返回null