**Java 技术管理规范文档**

**一、代码风格规范**

**a. 强制**

1. **缩进**：必须使用4个空格作为缩进，禁止使用Tab键
2. **命名**：类名采用UpperCamelCase风格，方法名、变量名采用lowerCamelCase风格
3. **常量**：全部大写，单词间用下划线隔开，如MAX\_STOCK\_COUNT
4. **大括号**：即使是单行语句块也必须使用大括号
5. **每行字符**：单行字符数限制不超过120个，超出需换行
6. **包名**：全部小写，使用公司域名倒置作为前缀(如com.alibaba.xxx)
7. **注释**：所有public类和方法必须有Javadoc注释

**b. 推荐**

1. **空行**：方法之间、不同逻辑段之间推荐使用空行分隔
2. **import**：不使用通配符import(如java.util.\*)
3. **变量声明**：推荐在需要使用时才声明变量，而非方法开始处
4. **行尾空格**：推荐删除所有行尾空格
5. **链式调用**：推荐将链式调用拆分为多行，每行一个方法调用

**c. 允许**

1. **空块**：简单的空块可以写在同一行(如**while (condition) {}**)
2. **多行数组**：简单的数组初始化可以写在一行
3. **三目运算符**：简单的条件判断允许使用三目运算符

**二、编码实践规范**

**a. 强制**

1. **异常处理**：禁止捕获异常后不做任何处理(空的catch块)
2. **资源释放**：必须使用try-with-resources或finally块释放资源
3. **equals**：使用equals比较对象时必须先检查null
4. **可变参数**：方法参数中不允许使用Object作为可变参数类型
5. **静态变量**：非final的静态变量必须通过方法访问
6. **日期时间**：禁止使用java.util.Date和java.util.Calendar，必须使用java.time包
7. **字符串拼接**：循环体内字符串拼接必须使用StringBuilder

**b. 推荐**

1. **日志**：推荐使用SLF4J作为日志门面，Logback或Log4j2作为实现
2. **集合初始化**：推荐在创建集合时指定初始容量
3. **工具类**：推荐将工具类设计为final类，并私有化构造方法
4. **参数校验**：推荐使用Guava Preconditions或Apache Commons Validate进行参数校验
5. **枚举**：固定值集合推荐使用枚举而非常量类

**c. 允许**

1. **方法长度**：简单的方法允许超过50行(如数据转换方法)
2. **循环**：简单的业务逻辑允许使用传统的for循环而非增强for循环
3. **内部类**：简单的回调处理允许使用匿名内部类

**三、面向对象设计**

**a. 强制**

1. **所有字段必须私有化，通过 getter/setter 访问。**
2. **避免过多的 public static 成员。**
3. **不允许在接口中定义实现逻辑（除非 default 方法且必要）。**
4. **抽象类必须包含至少一个抽象方法。**

**b. 推荐**

1. **多态优于 instanceof 和强制转型。**
2. **尽量使用组合代替继承。**
3. **接口职责单一，避免“胖接口”。**

**c. 允许**

1. **允许使用 Builder 模式构造复杂对象。**

**四、性能与安全规范**

**a. 强制**

1. **SQL注入**：必须使用预编译Statement，禁止拼接SQL
2. **密码存储**：必须使用BCrypt等安全算法存储密码
3. **敏感数据**：日志中禁止输出密码、密钥等敏感信息
4. **线程池**：必须使用ThreadPoolExecutor而非直接创建线程

**b. 推荐**

1. **缓存**：推荐使用成熟的缓存框架(如Caffeine、Redis)
2. **连接池**：推荐使用HikariCP等高性能连接池
3. **序列化**：推荐使用JSON或Protocol Buffers而非Java原生序列化

**c. 允许**

1. **反射**：框架代码允许合理使用反射
2. **synchronized**：简单的并发控制允许使用synchronized关键字