**Java 技术管理规范文档**

**规范级别说明：**

* **强制（a）**：必须遵守，违反即为不合格代码。
* **推荐（b）**：建议遵守，违反不会影响功能但可能影响质量。
* **允许（c）**：可选做法，鼓励统一风格。

**一、命名规范**

1. **[a]** 类名使用大驼峰（PascalCase），如 CustomerManager。
2. **[a]** 方法名使用小驼峰（camelCase），如 calculateTotal()。
3. **[a]** 常量使用全大写字母，单词间下划线分隔，如 MAX\_RETRY\_COUNT。
4. **[b]** 接口命名不要加 I 前缀（如 IUserService），应命名为 UserService。
5. **[a]** 包名一律小写，使用反域名格式（如 com.example.project.module）。
6. **[b]** Boolean 类型变量推荐使用 is 开头，如 isActive。
7. **[b]** 枚举命名应使用大写单词，如 SUCCESS, FAILURE。
8. **[b]** 单个字符变量（如 i, j）仅限在临时循环中使用。

**二、格式与缩进**

1. **[a]** 使用 4 个空格缩进，禁止使用制表符（Tab）。
2. **[a]** 每行代码不超过 120 个字符，超过应换行。
3. **[b]** 方法之间空一行，便于阅读。
4. **[b]** 类成员的顺序推荐为：常量 -> 属性 -> 构造方法 -> 公共方法 -> 私有方法。
5. **[b]** 单行 if/else/while/for 等结构也应加 {}。
6. **[a]** 文件末尾必须保留一个空行。
7. **[c]** 方法参数换行时，推荐每行一个参数并对齐。

**三、编码规范**

1. **[a]** 禁止出现魔法值，使用常量代替。
2. **[a]** 严禁空指针风险写法，如 obj.toString()，应使用 Objects.toString(obj)。
3. **[b]** 多个变量声明应拆分为多行。
4. **[a]** 禁止使用 == 比较字符串，应使用 equals() 或 Objects.equals()。
5. **[b]** 不推荐使用过多的嵌套（不超过 3 层为宜）。
6. **[b]** 使用 StringBuilder 替代字符串拼接（特别是在循环中）。
7. **[a]** 禁止捕获 Exception 后不处理或仅打印，应有明确处理策略。
8. **[a]** 所有资源类（如 InputStream、Connection）必须在 finally 或使用 try-with-resources 关闭。
9. **[b]** 尽量使用 final 修饰不变的变量和参数。
10. **[b]** 推荐使用 Optional 处理返回值可能为 null 的情况。

**四、注释规范**

1. **[a]** 所有 public 方法必须添加 Javadoc 注释。
2. **[a]** 注释内容要准确描述代码逻辑，避免无意义注释。
3. **[b]** 复杂逻辑应在代码块上方添加块注释。
4. **[b]** TODO 注释需标明处理人和时间限制，如 // TODO: 优化 by 2025-06-30 @Alice。
5. **[a]** 禁止无效注释或遗留代码，如被注释掉的旧方法。

**五、异常与日志**

1. **[a]** 自定义异常必须继承 RuntimeException。
2. **[b]** 使用日志框架（如 SLF4J），禁止使用 System.out.println()。
3. **[b]** 日志应包含关键信息（如 userId, orderId），便于定位问题。
4. **[a]** 捕获异常后必须记录日志并适当重抛或处理。
5. **[b]** 避免在 finally 中返回数据或抛出异常。

**六、并发与线程安全**

1. **[a]** 禁止使用非线程安全类（如 SimpleDateFormat）作为静态变量。
2. **[b]** 使用线程池代替手动创建线程。
3. **[a]** 共享变量必须使用 volatile、synchronized 或其他并发控制方式保护。
4. **[b]** 避免过度同步，防止死锁。
5. **[c]** 并发类推荐使用 J.U.C 包（如 ConcurrentHashMap, AtomicInteger）。