JAVA技术管理规范文档

2022141461071 宋润博

一、命名规范

1. **[a]** 类名必须使用大驼峰命名法（PascalCase），例如：UserService。
2. **[a]** 方法名、变量名必须使用小驼峰命名法（camelCase），例如：getUserName。
3. **[a]** 常量名必须使用全大写字母，单词间用下划线分隔，例如：MAX\_COUNT。
4. **[b]** 接口命名建议以名词或形容词结尾，例如：Runnable、Serializable。
5. **[b]** 布尔类型的变量命名建议以 is、has、can 开头，例如：isActive、hasPermission。
6. **[a]** 包名必须使用全小写字母，避免使用下划线，例如：com.example.project。
7. **[b]** 枚举类型的命名建议使用大驼峰命名法，枚举常量使用全大写字母，例如：Color.RED。
8. **[a]** 禁止使用拼音或无意义的命名，例如：tmp、test1。
9. **[b]** 避免使用单字符命名，除非在临时变量中使用，例如：i、j。
10. **[c]** 允许在测试方法中使用下划线分隔测试场景，例如：testLogin\_Success。

二、代码结构与格式

1. **[a]** 每个 Java 源文件只能包含一个公共类，且类名必须与文件名一致。
2. **[a]** 源文件必须使用 UTF-8 编码。
3. **[a]** 每行代码不得超过 100 个字符。
4. **[a]** 使用 4 个空格进行缩进，禁止使用制表符（Tab）。
5. **[a]** 大括号必须使用 K&R 风格，即左大括号与语句同行，右大括号独占一行。
6. **[a]** 控制语句（if、for、while 等）必须使用大括号，即使只有一条语句。
7. **[b]** 方法之间使用一个空行分隔，方法内部逻辑段落之间可以使用空行分隔。
8. **[b]** 类的成员变量和方法按照访问权限从高到低排序：public → protected → private。
9. **[b]** 在类中，常量、静态变量、实例变量的声明顺序建议为：常量 → 静态变量 → 实例变量。
10. **[c]** 允许使用空行提高代码的可读性，但应避免过多的空行。

三、注释规范

1. **[a]** 公共类、方法必须使用 Javadoc 注释，描述其功能、参数、返回值等。
2. **[a]** 禁止使用无意义的注释，例如：// TODO、// fix，除非明确说明待办事项。
3. **[b]** 对于复杂的业务逻辑，建议在代码中添加必要的注释，说明实现思路。
4. **[b]** 注释应与代码保持同步，及时更新过时的注释。
5. **[c]** 允许使用块注释（/\* \*/）对代码块进行说明，但应避免嵌套使用。

四、异常处理

1. **[a]** 禁止使用 catch (Exception e) 捕获所有异常，必须针对具体异常进行处理。
2. **[a]** 禁止在 catch 块中忽略异常，必须进行处理或记录日志。
3. **[a]** 禁止在 finally 块中使用 return、throw、break、continue，以避免覆盖异常。
4. **[b]** 建议使用自定义异常类，明确异常的业务含义。
5. **[b]** 建议在异常信息中包含关键参数，便于问题定位。

五、日志规范

1. **[a]** 禁止使用 System.out.println、System.err.println 进行日志输出，必须使用日志框架。
2. **[a]** 日志中禁止输出敏感信息，例如：密码、密钥、身份证号等。
3. **[b]** 建议使用占位符方式输出日志，避免字符串拼接，例如：logger.info("User ID: {}", userId);。
4. **[b]** 建议根据日志级别合理输出日志，避免过多的 debug 或 info 日志。
5. **[c]** 允许在开发阶段使用 debug 日志，但在生产环境中应适当调整日志级别。

六、集合与泛型

1. **[a]** 禁止使用原始类型的集合，必须使用泛型，例如：List<String>。
2. **[a]** 禁止在集合中添加 null 元素。
3. **[b]** 建议根据业务需求选择合适的集合实现类，例如：ArrayList、LinkedList、HashMap。
4. **[b]** 建议使用集合工具类进行集合操作，例如：Collections、Arrays。
5. **[c]** 允许使用第三方集合库，但应评估其稳定性和性能。

七、并发编程

1. **[a]** 禁止使用 Thread、Runnable 创建线程，必须使用线程池。
2. **[a]** 禁止使用 synchronized 锁住字符串常量或全局变量。
3. **[b]** 建议使用 java.util.concurrent 包下的并发工具类，例如：CountDownLatch、Semaphore。
4. **[b]** 建议使用 volatile 关键字保证变量的可见性，但应避免其替代锁机制。
5. **[c]** 允许使用 Lock 接口及其实现类进行显式锁控制，但应注意死锁风险。

八、代码安全

1. **[a]** 禁止在代码中硬编码敏感信息，例如：密码、密钥等。
2. **[a]** 禁止使用过时的加密算法，例如：MD5、DES。
3. **[b]** 建议对外部输入进行校验，防止 SQL 注入、XSS 攻击等安全问题。
4. **[b]** 建议使用安全框架或库处理认证、授权等安全功能。
5. **[c]** 允许使用代码扫描工具进行安全检测，例如：FindBugs、SonarQube。