**Java 语言技术管理规范文档**

**一、基础语法规范（共 10 条）**

| **级别** | **规则编号** | **规范内容** |
| --- | --- | --- |
| a | 1 | 所有 Java 文件必须以 .java 结尾，文件名与 public class 保持一致。 |
| a | 2 | 类、接口、枚举名使用 **大驼峰命名法**，例如 MyServiceImpl。 |
| a | 3 | 方法名、变量名使用 **小驼峰命名法**，例如 calculateSum。 |
| a | 4 | 常量必须使用 **全大写字母加下划线**，例如 MAX\_LENGTH。 |
| a | 5 | 不允许使用魔法值，必须使用常量或枚举代替硬编码数字或字符串。 |
| b | 6 | 每个类文件建议只定义一个顶级类或接口。 |
| b | 7 | 控制语句必须使用大括号包裹，即使只有一行代码。 |
| b | 8 | 使用 switch 时，必须处理 default 分支。 |
| c | 9 | 可使用 var 关键字（Java 10+）简化局部变量声明，但须保证可读性。 |
| c | 10 | static import 可用于简化工具类调用，但不能滥用影响可读性。 |

**二、代码结构规范（共 10 条）**

| **级别** | **规则编号** | **规范内容** |
| --- | --- | --- |
| a | 11 | 包名全部小写，使用反域名命名法，如 com.example.project。 |
| a | 12 | 类的结构顺序建议为：常量 > 字段 > 构造函数 > 公共方法 > 私有方法。 |
| a | 13 | 一个类中不能包含多个 public 类，内部类建议为 private static。 |
| b | 14 | 每个方法不超过 80 行代码；每个类尽量不超过 500 行。 |
| b | 15 | 工具类必须为 final 且构造函数 private。 |
| b | 16 | 非必要不暴露 public 字段，应使用 private + getter/setter。 |
| b | 17 | 日志统一使用 Slf4j 或 Log4j2，不使用 System.out.println。 |
| c | 18 | Builder 模式适用于包含多个可选字段的对象创建。 |
| c | 19 | 可在类上添加注释 @author，但不强制 |
| c | 20 | 接口中可包含 default 方法，但应谨慎使用，避免破坏设计单一性。 |

**三、异常与错误处理（共 6 条）**

| **级别** | **规则编号** | **规范内容** |
| --- | --- | --- |
| a | 21 | 禁止捕获 Throwable，应具体捕获 Exception 或其子类。 |
| a | 22 | 异常必须记录日志，不能吞掉异常信息。 |
| b | 23 | 自定义异常类名称必须以 Exception 结尾。 |
| b | 24 | 应使用运行时异常表示程序逻辑错误，受检异常用于外部依赖错误（如IO）。 |
| c | 25 | 方法可使用 throws 声明异常，但不应随意向外抛出过多异常类型。 |
| c | 26 | 可在项目中引入统一的异常封装处理类（如 ExceptionHandler）。 |

**四、注释与文档（共 6 条）**

| **级别** | **规则编号** | **规范内容** |
| --- | --- | --- |
| a | 27 | 所有 public 类和方法必须添加 Javadoc 注释。 |
| b | 28 | 对于复杂算法或业务逻辑，必须在方法内部使用块注释说明处理过程。 |
| b | 29 | TODO、FIXME 等注释必须注明责任人及处理期限。 |
| b | 30 | 不应出现无意义注释，如 // getter method。 |
| c | 31 | 使用 Markdown 风格的注释（如在接口注释中）可增强可读性。 |
| c | 32 | 可使用第三方插件生成 HTML API 文档（如 Javadoc + Swagger）。 |

**五、并发与集合类使用（共 4 条）**

| **级别** | **规则编号** | **规范内容** |
| --- | --- | --- |
| a | 33 | 多线程共享变量必须加锁或使用原子类控制，禁止使用非线程安全集合。 |
| b | 34 | 推荐使用 ConcurrentHashMap 代替 HashTable。 |
| b | 35 | 遍历集合使用增强 for 循环或 stream，避免使用 iterator.remove()。 |
| c | 36 | 可使用 synchronized、ReentrantLock 等方式控制并发，但需合理设计临界区。 |

**六、项目结构与依赖（共 4 条）**

| **级别** | **规则编号** | **规范内容** |
| --- | --- | --- |
| a | 37 | 项目必须使用构建工具进行管理，如 Maven 或 Gradle，不允许手动维护 jar 包。 |
| b | 38 | 依赖应采用版本锁定（如 Maven 使用 dependencyManagement）。 |
| c | 39 | 可以引入 Lombok 简化代码，但必须规范使用，如统一 @Data 和 @Builder 组合。 |
| c | 40 | 项目中允许使用模块化结构（如 Maven 多模块），建议按照业务领域拆分包结构。 |