**Java 语言技术管理规范**

**1. 代码风格**

**a. 强制**

1. 所有Java源文件必须使用UTF-8编码
2. 使用4个空格缩进，禁止使用Tab字符
3. 每行代码不超过120个字符
4. 类和方法必须有Javadoc注释
5. 大括号必须遵循K&R风格（开括号不换行）

**b. 推荐**

1. 推荐使用Unix风格换行符(LF)
2. 推荐在逗号、冒号后加空格
3. 推荐使用空行分隔逻辑代码块
4. 推荐将多个连续空行缩减为一个空行
5. 推荐使用小驼峰命名法(lowerCamelCase)命名变量和方法

**c. 允许**

1. 允许在链式调用中每个调用点换行
2. 允许在复杂表达式周围使用额外空格提高可读性
3. 允许在switch语句中case块不缩进
4. 允许在数组初始化时使用块状风格或行内风格
5. 允许在简单的lambda表达式周围不使用括号

**2. 命名规范**

**a. 强制**

1. 类名必须使用大驼峰命名法(UpperCamelCase)
2. 常量名必须全部大写，单词间用下划线分隔
3. 包名必须全部小写，不使用下划线
4. 布尔变量/方法必须以is/has/can等开头
5. 禁止使用单个字符作为变量名(循环计数器除外)

**b. 推荐**

1. 推荐使用有意义的名称，避免缩写
2. 推荐接口名称以"I"开头(如IUserService)
3. 推荐测试类以被测试类名开头，以Test结尾
4. 推荐枚举类以Enum结尾
5. 推荐工具类以Util/Utils结尾

**c. 允许**

1. 允许在局部变量中使用简短的名称(如i,j,k)
2. 允许在特定领域使用行业通用缩写
3. 允许在lambda参数中使用单个字母命名
4. 允许在覆盖父类方法时使用相同的参数名
5. 允许在内部类中使用简短的名称

**3. 编程实践**

**a. 强制**

1. 必须处理受检异常，禁止捕获Exception基类
2. 必须使用@Override注解标注覆盖方法
3. 必须为所有公有API编写单元测试
4. 必须使用日志框架而非System.out.println
5. 必须使用try-with-resources处理AutoCloseable资源

**b. 推荐**

1. 推荐使用final修饰不可变变量
2. 推荐使用不可变集合(Collections.unmodifiableXXX)
3. 推荐使用枚举替代整数常量
4. 推荐使用Optional代替返回null
5. 推荐使用工厂方法而非构造器创建对象

**c. 允许**

1. 允许在性能关键代码中使用可变对象
2. 允许在内部类中使用protected访问修饰符
3. 允许在特定场景使用可变参数(varargs)
4. 允许在DTO中使用public字段
5. 允许在测试代码中使用魔法数字

**4. 工具与框架**

**a. 强制**

1. 必须使用Maven或Gradle作为构建工具
2. 必须使用Checkstyle/PMD/Sonar进行代码检查
3. 必须使用JUnit作为测试框架
4. 必须使用SLF4J作为日志门面
5. 必须使用Git作为版本控制系统

**b. 推荐**

1. 推荐使用Lombok减少样板代码
2. 推荐使用Guava工具库
3. 推荐使用AssertJ进行断言
4. 推荐使用Mockito进行模拟测试
5. 推荐使用JaCoCo进行代码覆盖率检查

**c. 允许**

1. 允许在特定项目中使用其他日志实现(如Log4j2)
2. 允许在遗留代码中使用其他测试框架
3. 允许在小型项目中使用其他构建工具
4. 允许在特定场景使用其他集合框架
5. 允许在原型开发中暂时禁用部分检查规则

**5. 性能与安全**

**a. 强制**

1. 必须对用户输入进行验证和过滤
2. 必须使用预编译语句(PreparedStatement)处理SQL
3. 必须对敏感数据进行加密存储
4. 必须关闭数据库连接等资源
5. 必须对集合访问进行边界检查

**b. 推荐**

1. 推荐使用连接池管理数据库连接
2. 推荐使用StringBuilder进行字符串拼接
3. 推荐使用缓存提高性能
4. 推荐使用并发集合替代同步集合
5. 推荐使用对象池技术管理昂贵对象

**c. 允许**

1. 允许在确定安全的情况下使用原生方法
2. 允许在单线程环境下使用非线程安全集合
3. 允许在启动时进行耗时初始化
4. 允许在特定场景使用反射
5. 允许在性能分析指导下进行优化

**6. 文档与注释**

**a. 强制**

1. 必须为所有公有API编写Javadoc
2. 必须为复杂算法编写注释
3. 必须为TODO/FIXME注释添加责任人
4. 必须保持注释与代码同步更新
5. 必须为变更添加变更日志

**b. 推荐**

1. 推荐使用ASCII图表解释复杂逻辑
2. 推荐为重要设计决策添加注释
3. 推荐为测试用例添加预期行为说明
4. 推荐使用版本控制信息替代部分注释
5. 推荐为接口添加使用示例

**c. 允许**

1. 允许在明显代码处省略注释
2. 允许使用非英文注释(但不推荐)
3. 允许在注释中使用Markdown格式
4. 允许在注释中添加外部参考链接
5. 允许在测试代码中使用简略注释