Java 语言技术管理规范

一、代码风格

a. 强制

1. 类名必须使用大驼峰式命名法（UpperCamelCase）

2. 方法名、变量名必须使用小驼峰式命名法（lowerCamelCase）

3. 常量名必须全部大写，单词间用下划线分隔（UPPER\_CASE）

4. 每行代码不得超过120个字符

5. 必须使用4个空格进行缩进，禁止使用Tab

6. 大括号必须遵循K&R风格（左大括号不换行）

7. 所有重写方法必须添加@Override注解

b. 推荐

8. 推荐使用有意义的命名，避免缩写（除非是广泛认可的缩写如max, min）

9. 推荐在运算符前后和逗号后添加空格

10. 推荐将类成员变量集中声明在类顶部

11. 推荐每行只声明一个变量

12. 推荐使用空行分隔逻辑代码块

c. 允许

13. 允许在简单语句中省略大括号（如单行if语句）

14. 允许在链式调用中适当换行对齐

15. 允许在特定情况下使用魔法数字（如for循环中的i<10）

二、编码实践

a. 强制

16. 必须处理所有受检异常（Checked Exception）

17. 必须使用try-with-resources处理所有Closeable资源

18. 必须对可能为null的对象进行null检查

19. 必须使用@Override注解标记重写方法

20. 必须为所有公有类、方法和字段添加文档注释

b. 推荐

21. 推荐使用final修饰不会被修改的变量

22. 推荐优先使用Java标准库而非第三方库

23. 推荐使用枚举代替整数常量

24. 推荐使用StringBuilder进行字符串拼接

25. 推荐使用接口而非具体实现类声明变量

c. 允许

26. 允许在性能关键代码中使用非标准优化

27. 允许在特定场景下使用System.out进行调试

28. 允许在内部工具类中使用反射

三、面向对象

a. 强制

29. 必须遵循单一职责原则（一个类只做一件事）

30. 必须将类成员的可访问性设置为最小（优先private）

31. 必须使用getter/setter访问类成员变量

32. 必须避免在构造方法中调用可重写方法

b. 推荐

33. 推荐优先使用组合而非继承

34. 推荐使用不可变对象（Immutable Objects）

35. 推荐为工具类添加私有构造方法

36. 推荐使用工厂方法替代多个构造方法

c. 允许

37. 允许在特定场景下使用protected访问修饰符

38. 允许在DTO/VO类中使用public字段

四、并发编程

a. 强制

39. 必须避免在同步块中调用外部方法

40. 必须为共享可变数据添加适当的同步控制

41. 必须使用线程池而非直接创建线程

42. 必须避免使用已废弃的Thread方法（如stop, suspend）

b. 推荐

43. 推荐优先使用java.util.concurrent包中的并发工具

44. 推荐使用volatile修饰多线程共享变量

45. 推荐使用不可变对象处理线程间共享数据

46. 推荐使用并发集合而非同步集合

c. 允许

47. 允许在特定场景下使用双重检查锁定模式

48. 允许在性能关键代码中使用原子变量

五、异常处理

a. 强制

49. 必须捕获特定异常而非通用的Exception

50. 必须避免在finally块中使用return语句

51. 必须避免在catch块中忽略异常（至少记录日志）

b. 推荐

52. 推荐使用自定义异常而非通用异常

53. 推荐在异常中包含有意义的错误信息

54. 推荐使用异常链传递原始异常

c. 允许

55. 允许在特定场景下捕获Throwable

56. 允许在测试代码中忽略预期异常

六、工具与测试

a. 强制

57. 必须使用版本控制系统管理代码

58. 必须为所有公有API编写单元测试

59. 必须使用静态代码分析工具检查代码

b. 推荐

60. 推荐使用持续集成系统

61. 推荐使用代码覆盖率工具

62. 推荐使用日志框架而非System.out

c. 允许

63. 允许在原型代码中省略部分测试

64. 允许在特定场景下使用断言（assert）