

# Java 技术管理规范文档

## 一、代码风格规范

### a) 强制

1. 缩进：必须使用 4 个空格进行缩进，禁止使用 Tab 键
2. 大括号：必须使用 K&R 风格，左大括号不换行

```
```java
if (condition) {
    // code
}
...`
```

3. 行长度：单行代码不得超过 120 个字符
4. 包名：必须全部小写，使用公司域名倒序（如 com.company.project）
5. 类名：必须使用大驼峰式命名（UpperCamelCase）
6. 方法名必须使用小驼峰式命名（lowerCamelCase）
7. 常量名：必须全部大写，单词间用下划线分隔（如 MAX\_COUNT）

### b) 推荐

8. 变量名：推荐使用有意义的名称，避免缩写（如用`customerId`而非`cid`）
9. 方法长度：推荐不超过 50 行
10. 类长度：推荐不超过 500 行
11. 参数个数：推荐方法参数不超过 5 个
12. 空行：推荐在方法之间、逻辑块之间添加空行提高可读性

### c) 允许

13. 局部变量名：允许使用简短的名称（如循环中的`i`、`j`）
14. 重载：允许方法重载，但必须保持行为一致

## 二、编码实践规范

### a) 强制

15. 异常处理：必须捕获具体的异常，禁止捕获`Throwable`或`Exception`

```
```java
// 错误
try {...} catch (Exception e) {...}
// 正确
try {...} catch (IOException e) {...}
...`
```

16. 资源关闭：必须使用 try-with-resources 管理所有资源

```
```java
try (InputStream is = new FileInputStream(file)) {
    // code
}
...`
```

17. 空指针检查：必须对可能为 null 的参数进行校验

18. equals 比较：必须使用常量或已知非空对象调用 equals

```
```java
// 错误
str.equals("value");

// 正确
"value".equals(str);
```
```

19. 字符串拼接：必须使用 **StringBuilder** 进行循环内字符串拼接

b) 推荐

20. 不可变性：推荐使用 **final** 修饰不会改变的变量和参数

21. 防御性拷贝：推荐对可变参数和返回值进行防御性拷贝

22. 日志使用：推荐使用 **SLF4J+Logback** 日志框架

23. 集合返回：推荐返回空集合而非 **null**

```
```java
return Collections.emptyList();
```
```

24. 枚举使用：推荐使用枚举替代常量类

c) 允许

25. 方法链：允许适度的方法链式调用

26. 三元运算符：允许在简单条件下使用三元运算符

### 三、面向对象规范

a) 强制

27. 访问控制：必须使用最严格的访问级别（优先 **private**）

28. 继承限制：必须使用 `@Override` 注解重写方法

29. 接口隔离：必须遵循接口隔离原则，避免庞大接口

30. 构造方法：必须保持构造方法简单，不包含业务逻辑

b) 推荐

31. 组合优先：推荐优先使用组合而非继承

32. 依赖注入：推荐使用构造方法注入依赖

33. 不可变对象：推荐设计不可变对象

34. 单一职责：推荐类和方法遵循单一职责原则

c) 允许

35. 工具类：允许使用包含静态方法的工具类（但应避免滥用）

### 四、并发编程规范

a) 强制

36. 线程安全：必须明确标识非线程安全类

37. 同步控制：必须使用更高层次的并发工具（如 `ConcurrentHashMap`）而非 `synchronized`

38. **volatile** 使用：必须正确使用 **volatile** 修饰共享变量

b) 推荐

39. 线程池：推荐使用 `ThreadPoolExecutor` 而非直接创建线程

40. 不可变对象：推荐在多线程环境下使用不可变对象

c) 允许

41. **synchronized**：允许在简单场景下使用 `synchronized` 关键字

## 五、测试规范

a) 强制

42. 单元测试：必须为所有公开方法编写单元测试

43. 测试命名：测试方法名必须清晰表达测试意图

```
```java
@Test
public void shouldReturnFalseWhenInputIsNegative() {...}
```
```

b) 推荐

44. 测试覆盖率：推荐保持至少 80% 的代码覆盖率

45. 测试隔离：推荐每个测试方法独立不依赖其他测试

c) 允许

46. 集成测试：允许在必要时编写集成测试

## 六、工具与框架

a) 强制

47. 构建工具：必须使用 **Maven** 或 **Gradle** 管理项目

48. 代码格式化：必须使用统一配置的 IDE 代码样式

b) 推荐

49. 静态分析：推荐使用 **SonarQube** 进行代码质量分析

50. 代码审查：推荐所有代码必须经过 **Code Review** 才能合并

c) 允许

51. IDE 选择：允许开发者选择自己喜欢的 IDE（但需统一代码样式）