**C++技术管理文档规范**

1. 头文件规范  
   1.1 (a) 所有头文件必须使用#define保护，格式：PROJECT\_PATH\_FILE\_H  
   1.2 (a) 优先使用前置声明，减少#include依赖  
   1.3 (a) 类成员为指针/引用且无需访问定义时，禁止包含头文件  
   1.4 (b) 内联函数实现放在-inl.h文件中  
   1.5 (a) 包含顺序：同名.h > C系统头 > C++标准库 > 第三方库 > 项目内头
2. 命名规则  
   2.1 (a) 宏与枚举：全大写+下划线（MAX\_LENGTH）  
   2.2 (a) 变量名：全小写+下划线（user\_name）  
   2.3 (a) 类成员变量：下划线结尾（count\_）  
   2.4 (b) 普通函数：驼峰式（OpenFile）  
   2.5 (a) 常量：k前缀+驼峰式（kBufferSize）  
   2.6 (a) 类名：驼峰式（MyClass）  
   2.7 (a) 纯接口类：Interface后缀（LoggerInterface）
3. 类设计  
   3.1 (a) 单参数构造函数必须声明explicit  
   3.2 (a) 无自定义构造函数时需显式定义默认构造函数  
   3.3 (a) 禁用拷贝操作时声明为=delete  
   3.4 (a) 优先使用组合而非继承  
   3.5 (a) 禁用多重继承（接口继承除外）  
   3.6 (b) 成员声明顺序：public → protected → private  
   3.7 (a) 基类析构函数必须为virtual或protected
4. 函数规范  
   4.1 (a) 函数参数顺序：输入参数在前（值/const&），输出参数在后（指针）  
   4.2 (a) 函数行数不超过50行  
   4.3 (a) 重载函数必须功能一致  
   4.4 (b) 函数返回值禁止用const修饰（基础类型除外）
5. 代码风格  
   5.1 (a) 所有代码块必须使用大括号（包括单行）  
   5.2 (a) 缩进统一为2空格  
   5.3 (b) 行宽不超过80字符  
   5.4 (a) if/for后必须空格（if (cond)）  
   5.5 (a) switch必须包含default分支  
   5.6 (a) 禁用tab符
6. 注释规范  
   6.1 (a) 文件顶部注明版权、作者、功能概述  
   6.2 (a) 公共API必须说明参数/返回值/异常  
   6.3 (b) 复杂逻辑添加行内注释（解释原因而非行为）  
   6.4 (c) 临时代码用//TODO(作者)：描述标注  
   6.5 (a) 废弃接口用//DEPRECATED标记
7. 工程实践  
   7.1 (a) 禁用C风格错误处理（printf），使用异常/状态码  
   7.2 (b) 异常仅用于致命错误  
   7.3 (a) 使用RAII管理资源（std::lock\_guard）  
   7.4 (a) 禁用C风格强转，使用static\_cast/dynamic\_cast  
   7.5 (a) 共享数据必须用原子类型或互斥锁  
   7.6 (a) 禁用全局非POD变量
8. 文档与维护  
   8.1 (a) 公共头文件必须包含接口文档  
   8.2 (a) 版本更新需记录变更内容  
   8.3 (a) 术语首次出现时标注全称（如IOC）  
   8.4 (b) 代码变更时关联文档需同步更新  
   8.5 (a) 所有API必须提供使用示例
9. 工具要求  
   9.1 (a) 使用clang-format统一格式  
   9.2 (b) 集成Clang-Tidy静态检查  
   9.3 (b) 使用Doxygen生成API文档  
   9.4 (a) CI流水线必须拦截编译警告  
   9.5 (b) 单元测试覆盖率不低于80%
10. 其他规则  
    10.1 (a) 禁止使用C标准库（如malloc）  
    10.2 (c) 允许使用lambda表达式  
    10.3 (a) 所有提交必须通过Code Review  
    10.4 (b) 关键模块需设计评审文档

规范级别统计：  
强制(a)：34条  
推荐(b)：9条  
允许(c)：2条  
总计：45条

执行原则：  
(a)级：CI流水线强制拦截  
(b)级：Code Review必须检查，例外需审批  
(c)级：项目内部保持统一即可