**C++ 技术管理规范**

（基于 Google C++ Style Guide 和业界实践）

**一、格式规范**

1. a 缩进必须使用 4 个空格，禁止使用 Tab 键。

2. a 函数/类定义的大括号换行：

void Function() { // 函数左大括号不换行

if (condition) { // 控制语句左大括号不换行

// ...

}

}

3. a 行宽不超过 120 字符，超长需换行对齐参数或表达式。

4. b 运算符前后加空格（`a = b + c;`），但单目运算符不加（`++i;`）。

5. a 头文件使用 `#pragma once` 或标准宏守卫：

#ifndef PROJECT\_FILENAME\_H

#define PROJECT\_FILENAME\_H

// ...

#endif

**二、命名规范**

6. a 类/结构体/枚举：大驼峰（`MyClass`）。

7. a 函数：大驼峰（`CalculateTotal()`）。

8. a 变量：小驼峰（`userCount`），类成员变量加 `m\_` 前缀（`m\_instanceCount`）。

9. a 常量：k 前缀 + 大驼峰（`kMaxBufferSize`）。

10. a 枚举值：全大写 + 下划线（`COLOR\_RED`）。

11. b 命名空间：全小写 + 下划线（`project\_utils`）。

**三、注释规范**

12. a 公共 API 必须用 Doxygen 风格注释：

/// @brief 计算两个数的和

/// @param a 加数

/// @param b 被加数

/// @return 和

13. a 复杂逻辑必须添加行内注释。

14. b 文件头部注明版权、作者和功能概述。

15. a 废弃代码用 `// DEPRECATED: 替代方案` 标记。

**四、设计规范**

16. a 禁止使用裸指针，资源管理必须用 智能指针（`unique\_ptr`, `shared\_ptr`）。

17. a 禁用 C 风格字符串/数组，用 `std::string` 和 `std::vector`。

18. a 类设计遵循 单一职责原则（一个类一个功能）。

19. a 优先使用 组合 而非继承。

20. a 接口类命名以 `I` 开头（`IReadable`）。

21. a 禁用 C++ 异常（项目统一错误码或 `std::optional`）。

22. a 禁用 RTTI（禁用 `dynamic\_cast`/`typeid`）。

23. a 成员变量初始化：声明时初始化 或 构造函数初始化列表。

24. a 函数参数顺序：输入参数在前，输出参数在后。

**五、内存与性能**

25. a 避免拷贝大对象，传递 const 引用（`const vector<T>&`）。

26. b 循环中避免重复计算（如提前缓存 `size()` 结果）。

27. a 禁止返回 `new` 分配的原始指针（用智能指针封装）。

28. a 移动语义：大对象返回用 值传递 + std::move。

29. b 使用 `reserve()` 预分配容器内存（避免频繁扩容）。

**六、安全与健壮性**

30. a 所有外部输入必须验证（长度/范围/类型）。

31. a 整数运算检查溢出（`if (a > INT\_MAX - b)`）。

32. a 禁用 `rand()`，用 `<random>` 库生成随机数。

33. a 禁用未初始化变量（开启编译选项 `-Wuninitialized`）。

34. a SQL 操作必须参数化（禁止拼接 SQL 字符串）。

**七、现代 C++ 特性**

35. b 优先使用 `constexpr` 替代宏常量。

36. a 用 `nullptr` 代替 `NULL` 或 `0`。

37. b 使用 `auto` 简化迭代器类型（`auto it = vec.begin()`）。

38. b 范围循环：`for (const auto& item : container)`。

39. a 用 `override` 标记重写虚函数（`virtual void foo() override;`）。

40. a 用 `=default` 和 `=delete` 管理特殊成员函数。

**八、其他**

41. a 禁止提交调试代码（如 `printf`/临时文件）。

42. a 单元测试覆盖率 ≥ 80%（核心模块 ≥ 95%）。

43. c 允许 Lambda 表达式，但避免嵌套超过两层。

44. c 允许模板元编程，仅限基础库开发。

45. a 编译警告视为错误（启用 `-Werror`）。

规范级别说明

- 强制 (a)：违反即 CI 失败，禁止合入代码。

- 推荐 (b)：允许例外，但需在代码评审中说明原因。

- 允许 (c)：特定场景可使用，需保证可读性。