C++ 技术管理规范文档

# 一、命名规范（Naming Conventions）

* a. 强制

1. 类名使用 大驼峰命名法，例如 MyClass。
2. 函数名使用 小驼峰命名法，例如 doSomething()。
3. 变量名一律小写，单词之间用下划线 \_ 分隔，例如 user\_id。
4. 宏名必须全大写，单词间用下划线连接，例如 MAX\_BUFFER\_SIZE。
5. 命名不得使用拼音或中英文混合。

* b. 推荐

1. 成员变量以 m\_ 开头，如 m\_count。
2. 指针变量可添加后缀 \_ptr，如 data\_ptr。
3. 常量建议使用 k 前缀，如 kMaxUsers。

* c. 允许

1. 对于局部临时变量可使用短变量名如 i, j, tmp，但仅限于 10 行以内的作用域。

# 二、代码风格（Code Style）

* a. 强制

1. 使用 4 空格缩进，不允许使用 Tab。
2. 每行代码最多不超过 100 个字符。
3. 左大括号 { 必须与控制语句同行，如 if (...) {。
4. 每个函数必须明确其返回类型。
5. 禁止在头文件中使用 using namespace std;。

* b. 推荐

1. 为每个类和函数添加注释，说明用途和边界条件。
2. #include 顺序应为：头文件自身、C++标准库、第三方库、项目内部头文件。
3. 使用前向声明减少头文件依赖。
4. 使用尾逗号（trailing comma）在多行初始化列表中。

* c. 允许

1. 可以根据个人风格选择内联注释或块注释，但应保持统一。
2. 允许使用空行来区分逻辑段落。

# 三、内存与资源管理（Memory & Resource）

* a. 强制

1. 禁止裸指针手动管理堆内存，必须使用 std::unique\_ptr 或 std::shared\_ptr。
2. 析构函数必须为 virtual，当类可能被继承时。
3. 不允许使用 malloc / free，应使用 C++ 的 new / delete 或智能指针。

* b. 推荐

1. 使用 RAII 方式管理资源。
2. 若需要自定义拷贝构造/赋值运算符，也要实现移动构造和移动赋值运算符。
3. 对于多线程共享数据，推荐使用 std::mutex 或 std::lock\_guard。

* c. 允许

1. 允许使用 C 风格数组，但推荐使用 std::array 或 std::vector。

# 四、错误处理（Error Handling）

* a. 强制

1. 不允许忽略函数的返回值。
2. 禁止使用 assert 处理用户输入或运行时错误。
3. 所有 try-catch 必须明确捕获异常类型，禁止使用 catch (...)。

* b. 推荐

1. 错误返回建议封装为 Result 类型或使用 std::optional。
2. 日志系统必须捕捉并输出错误细节。

* c. 允许

1. 允许在性能关键路径使用 errno，但必须检查其值。

# 五、结构与模块（Structure & Design）

* a. 强制

1. 每个类定义必须放在独立头文件中。
2. 禁止循环包含（include cycle）。
3. 所有头文件必须加 include guard 或使用 #pragma once。

* b. 推荐

1. 尽量将函数划分为 .h 和 .cpp 实现，避免头文件暴露实现细节。
2. 模块设计遵循单一职责原则，每个模块只做一件事。

* c. 允许

1. 允许使用模板类，但应限制在头文件中定义。
2. 允许使用 lambda，但不可滥用嵌套 lambda。