CPP规范

一、命名规范

1. 类/结构体：大驼峰（MyClass）

级别：a（强制）

1. 常量：全大写+下划线（MAX\_SIZE）

级别：a（强制）

1. 类成员变量：小驼峰 + 后缀下划线（myVar\_）

级别：a（强制）

1. 普通变量/函数参数：小驼峰（totalCount）

级别：a（强制

1. 函数名：小驼峰（calculateSum()）

级别：b（推荐）

1. 命名空间：全小写（project\_utils）

级别：b（推荐）

1. 枚举值：全大写（COLOR\_RED）

级别：b（推荐）

1. 宏定义：全大写（#define LOG\_ERROR()）

级别：a（强制）

二、格式与风格

1. 缩进：2空格（禁用Tab）

级别：a（强制）

1. 花括号：K&R风格（函数左括号换行，其他不换行）

void foo()

{ // 函数换行

if (cond) { // 控制语句不换行

// ...

}

}

级别：b（推荐）

1. 行长度：≤100字符

级别：b（推荐）

1. 头文件保护：#pragma once

级别：a（强制）

1. 空行分隔：函数/类之间保留1行

级别：b（推荐）

1. 指针/引用符号：贴紧类型（int\* ptr）

级别：b（推荐）

三、内存与资源管理

1. 动态内存：禁用new/delete，使用std::make\_unique/shared

级别：a（强制）

1. 资源释放：RAII模式（构造获取，析构释放）

级别：a（强制）

1. 移动语义：大对象传递优先用std::move

级别：b（推荐）

1. 禁用裸指针：除性能关键模块（需审批）

级别：b（推荐）

四、安全性与健壮性

1. 初始化：所有变量显式初始化

级别：a（强制）

1. 类型转换：禁用C风格强转，用static\_cast<>()等

级别：a（强制）

1. 整数溢出：int → int32\_t/uint64\_t等明确宽度类型

级别：b（推荐）

1. 异常处理：禁用异常（需在文档说明原因）

级别：c（允许）

1. 输入验证：所有外部输入必须校验

级别：a（强制）

五、现代C++特性

1. auto使用：仅用于复杂类型（如迭代器）

级别：b（推荐）

1. 范围for：遍历容器优先用for (auto& item : container)

级别：a（强制）

1. constexpr：常量计算优先使用

级别：b（推荐）

1. Lambda：避免复杂捕获（[=]/[&]需审批）

级别：b（推荐）

六、类设计

1. 成员访问：数据成员一律private

级别：a（强制）

1. 构造函数：多参数用explicit

级别：b（推荐）

1. 默认成员：显式声明=default或=delete

级别：b（推荐）

1. 接口设计：遵循单一职责原则

级别：a（强制）

七、模板与泛型

1. 模板参数：用typename而非class（除类模板）

级别：b（推荐）

1. 概念约束：C++20起用requires

级别：b（推荐）

八、并发与性能

1. 线程安全：共享数据必须加锁（std::mutex）

级别：a（强制）

1. 原子操作：简单共享用std::atomic

级别：b（推荐）

1. 循环优化：避免循环内申请内存

级别：b（推荐）

九、测试与文档

1. 单元测试：公共接口必须覆盖测试

级别：a（强制）

1. Doxygen注释：公共API用///文档注释

级别：b（推荐）

1. 日志输出：关键路径记录状态（如INFO: Init complete）

级别：b（推荐）

1. TODO标记：代码中明确标注// TODO(作者)：优化逻辑

级别：a（强制）