

# C++ 规范

研发

# 原则

- 重质量和效率：功能，不崩溃，性能
- 可读性
- 可扩展性
- 重设计，轻代码
- 重实用，轻技巧

# 工程设计

- 合理拆分项目
- 理解项目关系
- 考虑工程独立性
- 理解动态库，静态库

# 类设计

- 理解业务，设计好接口
- 轻量级
- 合理使用设计模式
- 对外接口设计放在public

# 函数设计

- 函数名体现功能
- 参数不超过5个
- 长度控制在300行
- 合理使用 引用, const

# 指针使用

- 强制智能指针
- UI可使用raw 指针
- 类成员变量显式构造拷贝
- 拒绝浅拷贝，如需要显式说明

# 头文件

- 避免重复引用
- 引用顺序：系统，外部，其它工程，本工程
- 一个头文件对应一个cpp文件
- 杜绝头文件中使用 `using namespace`
- 声明导出

# 注释

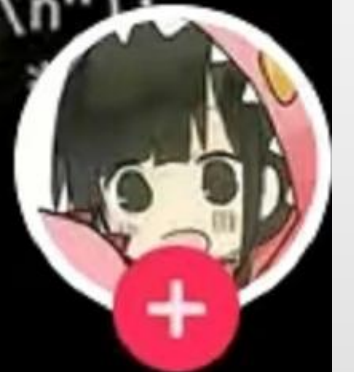
- 文件头写明文件功能，创建者，日期
- 外调函数需写注释，内建函数原则上需要写注释
- 核心复杂功能需写注释



# 哪个代码好? (来源dy)

```
1 #include<stdio.h>
2 main()
3 {
4     int i,j;
5     printf("\n the pattern is \n");
6     for (i=0; i<=4; i++)
7     {
8         for (j=0; j<=i; j++)
9         {
10            printf(" * ");
11        }
12        printf("\n");
13    }
14 }
```

```
1 #include<stdio.h>
2 main()
3 {
4     printf("*\n");
5     printf("* *\n");
6     printf("* * *\n");
7     printf("* * * *\n");
8     printf("* * * * *\n");
9
10 }
```



# 年轻人才纠结代码的优化，老油条只写易于读懂/理解的代码

```
1 #include<stdio.h>
2 main()
3 {
4     int i,j;
5     printf("\n the pattern is \n");
6     for (i=0; i<=4; i++)
7     {
8         for (j=0; j<=i; j++)
9         {
10            printf(" * ");
11        }
12        printf("\n");
13    }
14 }
```

```
1 #include<stdio.h>
2 main()
3 {
4     printf("*\n");
5     printf("* *\n");
6     printf("* * *\n");
7     printf("* * * *\n");
8     printf("* * * * *\n");
9
10 }
```

年轻人

老油条





# Q&A

公众号：多物理场仿真技术