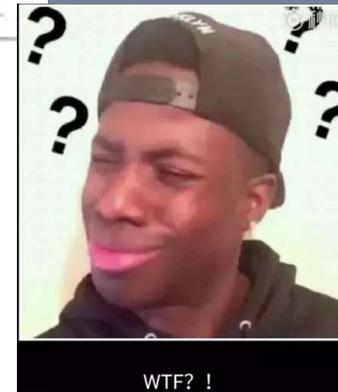
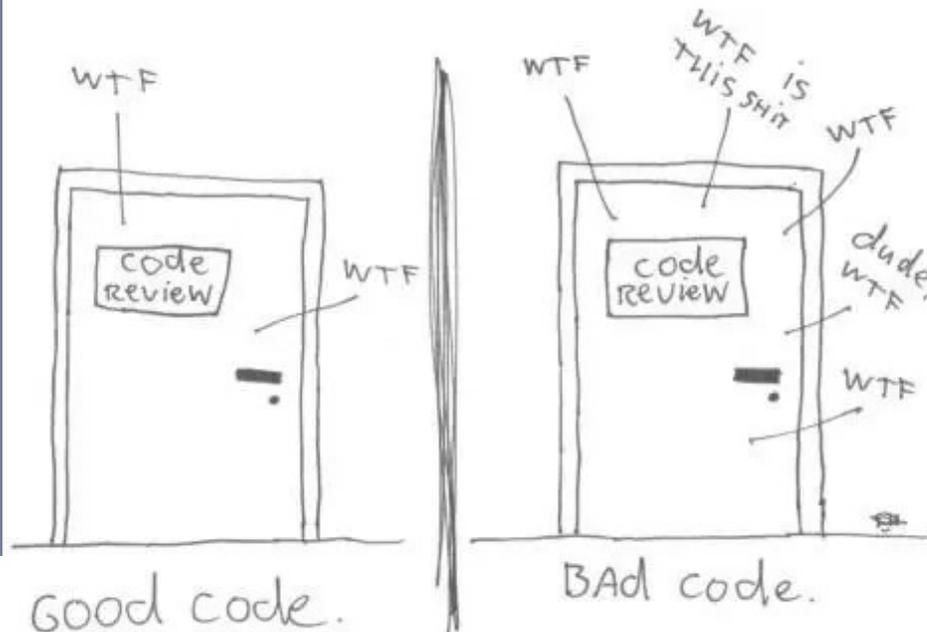


# 代码规范

The ONLY VALID MEASUREMENT  
OF CODE QUALITY: WTFs/MINUTE





## why?

- 每段自己写的代码，都是属于你自己的东西，你自己很容易能看懂，但是别人来看你的代码就没有那么简单了，所以当你把代码规范化之后，他有两点好处：
  - 1.便于别人看你代码
  - 2.降低报错率
  - 3.便于你自己再回去检查代码
  
- Even bad code can function. But if the code isn' t clean, it can bring a development organization to its knees.

by Robert C.Martin

百度翻译：即使是错误的代码也能起作用。但是如果代码不干净，它会让开发组织屈服。 ---罗伯特.马丁

# C/C++ 代码规范

# 1.排版风格

- 程序块缩进，一般是TAB（四个空格），TAB与空格不混用；
- 空格：
  - (1) 逗号，分号只在后面加空格
  - (2) “=”“+=”“-=”等双目运算符前后加空格
  - (3) “!”“++”“--”等单目运算符前后不加空格
  - (4) “->”“.”等关系密切的立即操作符前后不加空格
  - (5) 调用函数关键词后有一空格

# 1.排版风格

- if、while、for、case、default、do等语句 独占一行
- 一行最多写一句语句
- 不同类型操作符混合使用时，用括号 给出优先级
- else if 必须写在同一行
- “{}” 可以但不必要消去

## 2.可理解性

- 注释

- (1) 注释一般在代码上方或者代右方相邻位置（放上方时注意与上面的代码空行分开）
- (2) 常量，变量，数据结构声明
- (3) 头文件，源文件头部注释（文件名、作者、目的、功能...）
- (4) [函数头部注释（同上）](#)
- (5) 公共变量
- (6) 指针
- (7) 重要代码功能
- (8) switch没有break语句的case分支

## 2.可理解性

- 命名

- (1) [标识符缩写](#)

- (2) 缩写准则

- 一致性，长度不超过32字符，非通用要注释

- (3) 变量命名约定

- [作用范围或前缀]+[前缀]+基本类型+变量名

- 例：指针 Pointer

- 前缀：p

- 命名：pThePoin

## 2.可理解性

- 命名

- (4) 宏和常量命名 (单词字母大写, 单词用下划线分开)

- (5) 结构体和结构体成员 (单词字母大写, 单词用下划线分开)

- (6) 枚举 (单词字母大写, 单词用下划线分开; 成员第一个单词相同)

- (7) 类的命名

- COM类 COMMyobjectClass

- COM实现类(COM implewtation class)

- (8) 函数名首字母大写

## 3.可维护性

- [明确逻辑判断](#)
- 预编译条件不应分离
- 宏定义中合并预编译条件
- 宏定义表达式要使用完备的括号
- 使用宏时不允许参数发生变化
- 结构中一行只定义一个元素
- 枚举从小到大定义
- 不要把 “++” “--” 与 “-=” “+=” 组合使用
- 关系密切的代码尽可能相邻

## 3.可维护性

- 原则上行数小于200行
- 语句嵌套层次不得超过5层
- 避免相同代码在多个地方出现，可以用函数调用代替
- 使用强制类型转换
- 避免使用goto语句
- 循环语句中避免使用break和goto
- 功能相近的一组常量最好用枚举定义

## 4.正确性、效率

- 禁止使用未经初始化的变量
- 定义公共指针同时对其初始化
- 2k以上的局部变量声明成静态变量 (static) , 避免发生占用太多栈堆空间, 发生栈堆溢出
- 防止内存操作越界
- 减少没必要指针使用
- 防止精度丢失
- Switch语句中一定有default语句
- 最忙的循环放在最内层
- 循环中无关判断移除

本PPT内容借鉴于博客@此间的年少  
<https://blog.csdn.net/zxxSdsd/article/details/9345737>

**谢谢**

## 逗号, 分号只在后面加空格

- `printf("%d %d %d" , a, b, c);`

### “=”“+=”“-=”等双目运算符前后加空格

- `a *= 2;`
- `a = b ^ 2;`
- `a = b + c;`

## if、while、for、case、default、do等语句独占一行

- `if(...)`
- `{`
- `}`

## “{}”可以但不必要消去

- If(a>b)
- {  
    printf(“ok”);  
}
- If(a>b)  
  printf(“ok”);

## 不同类型操作符混合使用时，用括号给出优先级

如本来是正确的代码：

```
if( year % 4 == 0 || year % 100 != 0 && year % 400 == 0 )
```

如果加上括号，则更清晰。

```
if((year % 4) == 0 || ((year % 100) != 0 && (year % 400) == 0))
```

### 函数头部注释 (同上)

**/\*该函数用于传入两个整形变量，返回这两个变量的和\*/**

**Int sum (int a, int b)**

**{**

**return a+b;**

**}**

## 标识符缩写

### 1) 去掉不在词头的元音字母

screen—scrn

### 2) 单词首字母或者前几个字母

Chanel Activation—ChanActiv

### 3) 标准惯用的缩写形式

Capture The Flag—CTF

## 明确逻辑判断

**While(a)**

{

.....//代码

}

**While(a!=0)**

{

.....//代码

}