

C++代码风格指南v0.1(试用版)

本指南结合实际情况修改自[google开源项目风格指南](#).以下仅列出了**必须**遵守的风格规范,主要包含:

`build`, `legal`, `readability`, `runtime`, `whitespace` 五大类风格规范,更多规范请参考原指南.试用过程中如发现问题,请及时反馈.

[build/header_guard]

- 1.头文件保护:头文件必须有
- 2.必须**大写**
- 3.必须以**_**连接,以**_**结尾,
- 4.`#endif`后面的注释必须遵循**注释的规则**.像这样:

参考这里:<https://zh-google-styleguide.readthedocs.io/en/latest/google-cpp-styleguide/headers/#define>

```
1  #ifndef XXX_YYY_ZZZ_  
2  #define XXX_YYY_ZZZ_  
3  
4  #endif  // XXX_YYY_ZZZ_
```

[build/include]

- 1.禁止使用 `#include "bar.h"`,建议使用 `#include "foo/bar.h"`
- 2.禁止包含.cc和.cpp作为头文件
- 3..cpp文件一定要包含对应的.h头文件(如果存在的话)
- 4.同一文件不能重包含相同的头文件,即同一文件不能写两行重复的include

[build/include_order]

- 1.头文件包含顺序:**相关头文件-C系统文件-C++系统文件-其他库的.h文件-本项目内的.h文件**,头文件包含条件编译时,放在最后
- 2.禁止使用UNIX特殊快捷目录字符.(当前目录)和..(上一级目录)

参考这里:<https://zh-google-styleguide.readthedocs.io/en/latest/google-cpp-styleguide/headers/#include>

[build/namespaces]

- 1.禁止在.h头文件中使用**匿名空间**
- 2.禁止使用using namespace ...指示污染整个命名空间

参考这里:<https://zh-google-styleguide.readthedocs.io/en/latest/google-cpp-styleguide/scoping/#unnamed-namespace-and-static-variables>

[legal/copyright]

1.在每个文件的开头加入版权公告,格式为 Copyright [year] <Copyright Owner>

参考这里:<https://zh-google-styleguide.readthedocs.io/en/latest/google-cpp-styleguide/comments/#id3>

[readability/braces]

1.除了类和结构体,右大括号}后面不应该加分号

2.if/else后面的语句应加大括号

3.if/else语句结构必须使用以下格式,像这样:

```
1  if (condition) { // 圆括号里没有空格.
2      ...
3  } else if (...) { // else 与 if 的右括号同一行.
4      ...
5  } else {
6      ...
7  }
```

4.switch开关语句结构

```
1  switch (var) {
2      case 0: {
3          ...
4          break;
5      }
6      case 1: {
7          ...
8          break;
9      }
10     default: {
11         assert(false);
12     }
13 }
```

5.for循环语句结构

```
1  for (int i = 0; i < kSomeNumber; ++i) {
2      printf("I take it back\n");
3  }
```

6.while循环结构

```
1  while (condition) {
2      ...
3  }
```

参考这里:<https://zh-google-styleguide.readthedocs.io/en/latest/google-cpp-styleguide/formatting/#id7>

[readability/casting]

1. 禁止使用c风格的类型转换,应该使用c++风格的类型转换如 `static_cast<>()`. 不要使用 `int y = (int)x` 或 `int y = int(x)` 等转换方式

参考这里:<https://zh-google-styleguide.readthedocs.io/en/latest/google-cpp-styleguide/others/#id8>

[readability/inheritance]

1. 子类进行虚函数重载时,使用override关键字,不可再加上virtual关键字

参考这里:<https://zh-google-styleguide.readthedocs.io/en/latest/google-cpp-styleguide/classes/#inheritance>

[readability/fn_size]

1. 倾向于编写简短,凝练的函数,如果超过40行,在不影响程序结构的前提下对其进行分割,一般函数体函数体的有效长度不得超过250行,测试函数(Test)函数体有效长度不得超过400行

参考这里:<https://zh-google-styleguide.readthedocs.io/en/latest/google-cpp-styleguide/functions/#id3>

[readability/multiline_comment]

1. 建议使用单行注释//,单行禁止使用多行注释/*...*/,建议:除了文件开头的版权声明,其他位置一律使用单行注释

[readability/multiline_string]

1. 禁止使用多行string类型字符串,如字符串超过一行应使用c++11的raw字符串或者拼接代替

[readability/namespace]

1. namespace应该以 `// namespace ...` 结尾,且应遵循注释规则

2. 匿名空间应以 `// namespace` 或者 `// anonymous namespace` 结尾,且应遵循注释规则

参考这里:<https://zh-google-styleguide.readthedocs.io/en/latest/google-cpp-styleguide/scoping/#namespaces>

[readability/todo]

1. TODO注释的格式必须为 `// TODO(my_username): Stuff`,注意,冒号后面有一个空格,且应遵循注释的一般规则

参考这里:<https://zh-google-styleguide.readthedocs.io/en/latest/google-cpp-styleguide/comments/#todo>

[runtime/explicit]

- 1.单参数构造函数, 使用 `explicit` 关键字禁用隐式类型转换
- 2.无参或者大于一个参数的构造函数,禁止使用 `explicit`

参考这里:<https://zh-google-styleguide.readthedocs.io/en/latest/google-cpp-styleguide/classes/#implicit-conversions>

[runtime/int]

- 1.使用int16/int64等等,不要使用short,long,long long

参考这里:<https://zh-google-styleguide.readthedocs.io/en/latest/google-cpp-styleguide/others/#id11>

[runtime/operator]

- 1.运算符重载要谨慎,不建议使用
- 2.禁止重载 `&&`, `||`, `,` 或一元运算符 `&`

参考这里:<https://zh-google-styleguide.readthedocs.io/en/latest/google-cpp-styleguide/classes/#id10>

[runtime/printf]

- 1.禁止使用printf(),存在安全隐患,应使用snprintf()代替
- 2.用snprintf()代替strcpy()和strcat()

[runtime/string]

- 1.禁止声明string类型的静态/全局变量,应使用c风格的char数组表示

<https://zh-google-styleguide.readthedocs.io/en/latest/google-cpp-styleguide/scoping/#static-and-global-variables>

[runtime/threadsafe_fn]

- 1.不使用线程不安全函数:

```
1 | asctime,ctime,getgrgid,getgrnam,rand,gmtime...
```

使用以下线程安全的函数代替:

```
1 | asctime_r, ctime_r, getgrgid_r, getgrnam_r, rand_r, gmtime_r...
```

[runtime/invalid_increment]

- 1.禁用无效的指针递增 `*count++`,应使用 `++*count`, `(*count)++` or `*count += 1`

[whitespace/blank_line]

总体规则:不要随意在代码中加入空行

- 1.代码块的第一行不应该是空行
- 2.代码块的最后一行不应是空行
- 3.类的访问权限关键字 `public`, `protected`, `private` 后面第一行不应留空行
- 4.类的访问权限关键字 `protected`, `private` 前面一行应该是空行

即类的格式如下：

```
1 | class XXX {  
2 |     public:  
3 |         ...  
4 |         ...  
5 |         // 空行  
6 |     protected:  
7 |         ...  
8 |         ...  
9 |         //空行  
10 | private:  
11 |     ...  
12 |     ...  
13 | };
```

[whitespace/braces]

- 1.函数体的左大括号{总是在最后一个参数同一行的末尾处,且与右圆括号)之间有一个空格
- 2.左大括号之前总是有一个空格

[whitespace/comma]

- 1.逗号后边总是有空格

[whitespace/comments]

- 1.行尾的注释和前面的代码之间至少有两个空格的间隔
- 2.注释内容和注释符号//之间必须有一个空格

[whitespace/end_of_line]

- 1.不要在行尾添加空格

[whitespace/ending_newline]

- 1.文件必须以空行结尾

[whitespace/forcolon]

- 1.range-based的for循环中,冒号前后都要有空格

[whitespace/line_length]

- 1.每一行的长度不能超过120个字符

[whitespace/newline]

- 1.else永远在右大括号}后面,不要另起一行
- 2.else从句不能和else在同一行,而是要另起一行
- 3.一般情况下,一行只能有一条语句(for除外)

[whitespace/operators]

- 1.一般情况下,赋值运算符=前后总是有空格,类似这种情况可以没有: `if ((a=Foo()) == 0)`
- 2.二元操作符前后有空格(+*/>< && ||)
- 3.一元操作符(负号,++,--,!)和参数之间不加空格
- 4.流运算符(>> <<)前后必须有空格

参考这里:<https://zh-google-styleguide.readthedocs.io/en/latest/google-cpp-styleguide/formatting/#id17>

[whitespace/parens]

- 1.参数表的左圆括号(应和函数名在同一行,且与函数名之间没有空格,与参数之间没有空格,声明和调用均适用)
- 2.右圆括号禁止单独占一行,且前面没有空格
- 3.圆括号与内部字符之间不应留空格,这种情况除外:

```
for ( ; foo; bar)      for (foo; bar; )
```

- 4.if/for/while/switch与后面的左圆括号之间有一个空格

- 5.函数参数换行总体原则:对齐,美观

参考这里:<https://zh-google-styleguide.readthedocs.io/en/latest/google-cpp-styleguide/formatting/#id4>

[whitespace/semicolon]

- 1.分号后面有一个空格(行尾除外)
- 2.分号前不加空格

[whitespace/tab]

- 1.禁止使用tab,使用空格替代tab(IDE可设置自动将tab转换为空格)
- 2.采用4空格缩进,不要使用2空格缩进

强烈建议:在c++中,不是万不得已的情况,不建议使用宏

参考这里：<https://zh-google-styleguide.readthedocs.io/en/latest/google-cpp-styleguide/others/#preprocessor-macros>