

高质量C_C编程指南--第2章程序的版式 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/134/2021_2022__E9_AB_98_E8_B4_A8_E9_87_8FC_c97_134238.htm

第2章程序的版式 版式虽然不会影响程序的功能，但会影响可读性。程序的版式追求清晰、美观，是程序风格的重要构成因素。可以把程序的版式比喻为“书法”。好的“书法”可让人对程序一目了然，看得兴致勃勃。差的程序“书法”如螃蟹爬行，让人看得索然无味，更令维护者烦恼有加。请程序员们学习程序的“书法”，弥补大学计算机教育的漏洞，实在很有必要。2.1

空行 空行起着分隔程序段落的作用。空行得体（不过多也 不过少）将使程序的布局更加清晰。空行不会浪费内存，虽然 打印含有空行的程序是会多消耗一些纸张，但是值得。所以 不要舍不得用空行。【规则 2-1-1】在每个类声明之后、每 个函数定义结束之后都要加空行。参见示例 2-1（a）【规 则 2-1-2】在一个函数体内，逻辑上密切相关的语句之间不 加空行，其它地方应加空行分隔。参见示例 2-1（b）// 空 行 void Function1(...) { ... } // 空行 void Function2(...) { ... } // 空行 void Function3(...) { ... } // 空行 while (condition) { statement1. // 空行 if (condition) { statement2. } else { statement3. } // 空行 statement4. } 示例 2-1(a) 函数之间的空行 示例2-1(b) 函 数内部的空行

2.2 代码行 | 【规则 2-2-1】一行代码只做一件事 情，如只定义一个变量，或只写一条语句。这样的代码容 易阅读，并且方便于写注释。 | 【规则 2-2-2】if、for、 while、do 等语句自占一行，执行语句不得紧跟其后。不论 执行语句有多少都要加 {}。这样可以防止书写失误。 示例

2-2 (a) 为风格良好的代码行 , 示例 2-2 (b) 为风格不良的代码行。 int width. // 宽度 int height. // 高度 int depth. // 深度
int width, height, depth. // 宽度高度深度 x = a b. y = c d. z = e f. X
= a b. y = c d. z = e f. if (width { dosomething(). } if (width for
(initialization. condition. 0update) { dosomething(). } // 空行
other(). for (initialization. condition. 0update) dosomething().
other().

示例 2-2(a) 风格良好的代码行 示例2-2(b) 风格不良的代码行

2 【建议 2-2-1】尽可能在定义变量的同时初始化该变量（就近原则）如果变量的引用处和其定义处相隔比较远，变量的初始化很容易被忘记。如果引用了未被初始化的变量，可能会导致程序错误。本建议可以减少隐患。例如

int width = 10. // 定义并初始化width
int height = 10. // 定义并初始化height
int depth = 10. // 定义并初始化depth

2.3 代码行内的空格 | 【规则 2-3-1】关键字之后要留空格。象 const、virtual、inline、case 等关键字之后至少要留一个空格，否则无法辨析关键字。

象 if、for、while 等关键字之后应留一个空格再跟左括号 ‘ (’，以突出关键字。 | 【规则 2-3-2】函数名之后不要留空格，紧跟左括号 ‘ (’，以与关键字区别。 | 【规则 2-3-3】 ‘ (’ 向后紧跟， ‘) ’、 ‘ , ’、 ‘ . ’ 向前紧跟，紧跟处不留空格。 | 【规则 2-3-4】 ‘ , ’ 之后要留空格，如 Function(x, y, z)。如果 ‘ . ’ 不是一行的结束符号，其后要留空格，如 for (initialization. condition. 0update)。 | 【规则 2-3-5】赋值操作符、比较操作符、算术操作符、逻辑操作符、位域操作符，如 “ = ”、 “ = ” “ >= ”、 “

跟左括号 ‘ (’，以突出关键字。 | 【规则 2-3-2】函数名之后不要留空格，紧跟左括号 ‘ (’，以与关键字区别。 | 【规则 2-3-3】 ‘ (’ 向后紧跟， ‘) ’、 ‘ , ’、 ‘ . ’ 向前紧跟，紧跟处不留空格。 | 【规则 2-3-4】 ‘ , ’ 之后要留空格，如 Function(x, y, z)。如果 ‘ . ’ 不是一行的结束符号，其后要留空格，如 for (initialization. condition. 0update)。 | 【规则 2-3-5】赋值操作符、比较操作符、算术操作符、逻辑操作符、位域操作符，如 “ = ”、 “ = ” “ >= ”、 “

跟左括号 ‘ (’，以突出关键字。 | 【规则 2-3-2】函数名之后不要留空格，紧跟左括号 ‘ (’，以与关键字区别。 | 【规则 2-3-3】 ‘ (’ 向后紧跟， ‘) ’、 ‘ , ’、 ‘ . ’ 向前紧跟，紧跟处不留空格。 | 【规则 2-3-4】 ‘ , ’ 之后要留空格，如 Function(x, y, z)。如果 ‘ . ’ 不是一行的结束符号，其后要留空格，如 for (initialization. condition. 0update)。 | 【规则 2-3-5】赋值操作符、比较操作符、算术操作符、逻辑操作符、位域操作符，如 “ = ”、 “ = ” “ >= ”、 “

跟左括号 ‘ (’，以突出关键字。 | 【规则 2-3-2】函数名之后不要留空格，紧跟左括号 ‘ (’，以与关键字区别。 | 【规则 2-3-3】 ‘ (’ 向后紧跟， ‘) ’、 ‘ , ’、 ‘ . ’ 向前紧跟，紧跟处不留空格。 | 【规则 2-3-4】 ‘ , ’ 之后要留空格，如 Function(x, y, z)。如果 ‘ . ’ 不是一行的结束符号，其后要留空格，如 for (initialization. condition. 0update)。 | 【规则 2-3-5】赋值操作符、比较操作符、算术操作符、逻辑操作符、位域操作符，如 “ = ”、 “ = ” “ >= ”、 “

跟左括号 ‘ (’，以突出关键字。 | 【规则 2-3-2】函数名之后不要留空格，紧跟左括号 ‘ (’，以与关键字区别。 | 【规则 2-3-3】 ‘ (’ 向后紧跟， ‘) ’、 ‘ , ’、 ‘ . ’ 向前紧跟，紧跟处不留空格。 | 【规则 2-3-4】 ‘ , ’ 之后要留空格，如 Function(x, y, z)。如果 ‘ . ’ 不是一行的结束符号，其后要留空格，如 for (initialization. condition. 0update)。 | 【规则 2-3-5】赋值操作符、比较操作符、算术操作符、逻辑操作符、位域操作符，如 “ = ”、 “ = ” “ >= ”、 “