



Coding Standard 讲义

2007年7月20日



内容

- ◆ 为什么要有编码规范
- ◆ 不好的编码例子
- ◆ 编码规范介绍
- ◆ 怎么落实编码规范



内容

- ◆ 为什么要有编码规范
- ◆ 不好的编码例子
- ◆ 编码规范介绍
- ◆ 怎么落实编码规范



为什么要有编码规范（1）

◆ 规范是工程管理的基本手段

- “说到的要作到，作到的要说到”。说到的就是规范。
- 编码是软件工程中一个重要的“作”。把“作到”的“说到”，就是编码规范。
- 树立“工程意识”，懂得原理和能够大批量制作质量稳定的产品不同



为什么要有编码规范（2）

◆ 软件特性

- 正确性
- 易读性
- 易维护性
- 可靠性
- 可重用性
-

◆ 编码规范是提高易读性、易维护性、可靠性、可重用性的重要手段之一。



为什么要有编码规范（3）

- ◆ 一个软件的生命周期中，80%的花费在于维护。
- ◆ 几乎没有任何一个软件，在其整个生命周期中，均由最初的开发人员来维护。
- ◆ 编码规范可以改善软件的可读性，可以让程序员尽快而彻底地理解新的代码
- ◆ 在维护期易读性、易维护性、可靠性、可重用性表现出了更高的重要性。
- ◆ iSM V3.4的开发的一个重要任务就是易读性改善。
 - 去掉用注释掉的代码
 - 去掉`#if(0)` 包含的代码



内容

- ◆ 为什么要有编码规范
- ◆ 不好的编码例子
- ◆ 编码规范介绍
- ◆ 怎么落实编码规范

不好的编码例子(1)

◆ 没有界限检查

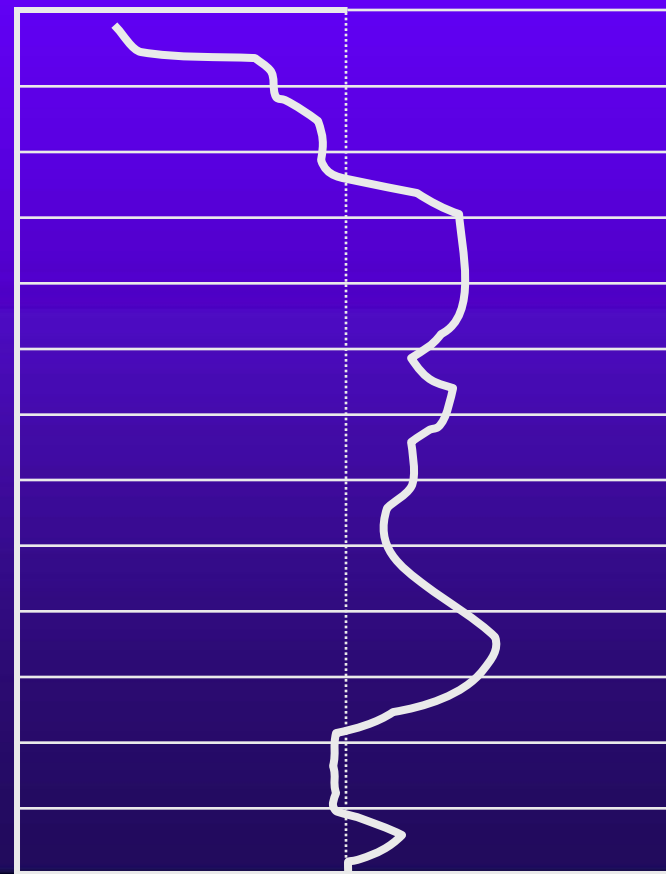
OutPutMark(char * Low, char* High char * Unit)

```
{  
    char  bufHigh[6];  
    char  bufLow[6];  
    int    i;  
  
    for(i=0; i<strlen(High);i++)  
    {  
        bufHigh[i] = High[i];  
    }  
    bufHigh[i]='\0';  
}
```

200

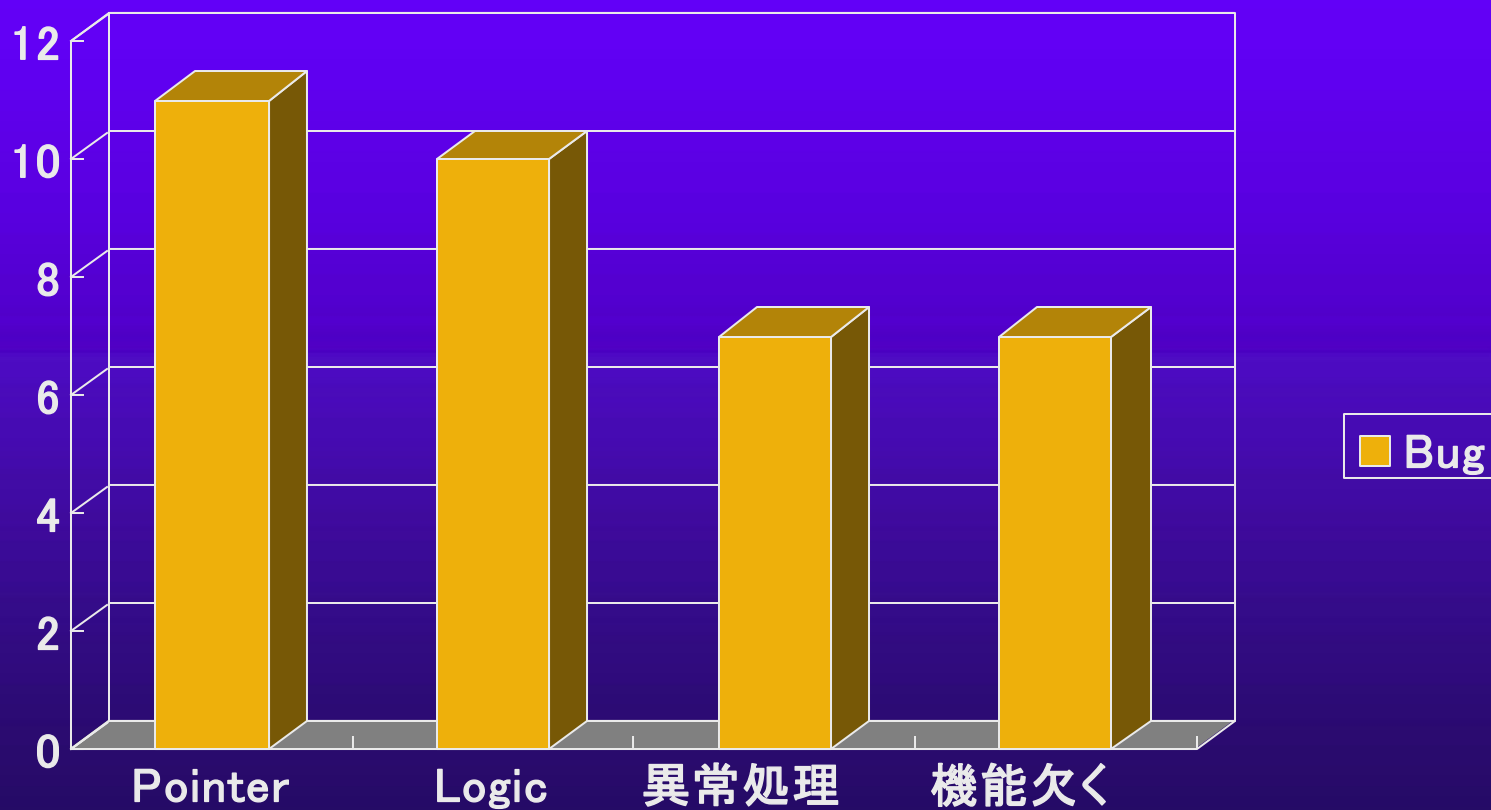
PPS

0



不好的编码例子(2)

JC项目用户发现Bug原因分析（总数50）





内容

- ◆ 为什么要有编码规范
- ◆ 不好的编码例子
- ◆ **编码规范介绍**
- ◆ 怎么落实编码规范

编码规范介绍- C++编码规范

- ◆ 摩托罗拉公司的研发部是国内目前通过CMM5的部门之一
- ◆ 从25个方面以300多条原则列举了C++程序员编程时应该遵守的规范
- ◆ 非常全面，极具参考价值，可以根据这个制订自己部门的规范。





编码规范介绍- 林锐博士的书

- ◆ 《高质量C++/C编程指南》
- ◆ 实际上一本优秀的规范介绍书籍
- ◆ 介绍了11个方面、65个规则
- ◆ 还有一个“C++/C代码审查表”

化境编程界-推荐图书系列

高质量 C++/C 编程指南

文件状态	文件标识:	
<input type="checkbox"/> 草稿文件	当前版本:	1.0
<input checked="" type="checkbox"/> 正式文件	作 者:	林锐 博士
<input type="checkbox"/> 更改正式文件	完成日期:	2001 年 7 月 24 日



编码规范介绍- MISRA (1/4)


- ◆ MISRA C Coding Standard
- ◆ MISRA (The Motor Industry Software Reliability Association 汽车工业软件可靠性联合会)
- ◆ 127条C语言编码标准，通常认为，如果能够完全遵守这些标准，则你的C代码是易读、可靠、可移植和易于维护的。
- ◆ 很多嵌入式开发者都以MISRA C来衡量自己的编码风格



编码规范介绍- MISRA (2/4)

- ◆ Rule 10. 不得残留被注释掉的废代码
- ◆ Rule 11. 所有标识符不超过31字符
- ◆ Rule 12. 不同名空间中的变量名不得相同

```
struct Person {  
    char* name;  
    ...  
};  
char name[32]; (违规)
```




编码规范介绍- MISRA (3/4)

- ◆ Rule 14. 不得使用类型char，必须显示声明为unsigned char或者signed char。
- ◆ Rule 35. 在返回布尔值的表达式中不得出现赋值操作。

我们常用的 `if (!(fp = fopen("fname", "r")))`
`{ /* error */ }`
被禁止。

- ◆ Rule 67. 循环计数器的值不得在循环体内修改。
- ◆ Rule 70. 禁止任何直接和间接的递归函数调用。
- ◆ Rule 82. 每个函数只能有一个退出点。



编码规范介绍- MISRA (4/4)

- ◆ Rule 56. 不得使用goto.
- ◆ Rule 59. if, else if, else, while, do..while, for语句块必须使用{}括起。
- ◆ 更严格的
 - Rule 118. 禁止使用动态堆分配(也就是不得使用malloc, calloc和realloc)。
 - Rule 124. 禁止使用<stdio.h>(不能用printf, scanf了！)
 - Rule 127. 禁止使用<time.h>



编码规范介绍-教育用规范(1/2)

- ◆ <http://www.csdn.com>上的一篇文章“编程修养”
- ◆ 7篇，共介绍了32个规则。容易记忆和遵守
- ◆ 教育期间，不讨论这些规则的合理性，只要求严格执行。
- ◆ 这些规范本身就是“C代码审查表”。



编码规范介绍-教育用规范(2/2)

◆ 规范1

- 文件名: File name、文件描述: Description、创建人: Creator、版本号: Version、修改纪录: Modify recorder。
- 函数名: Function name、参数: Parameter、功能描述: Description、返回值: Return value、抛出异常: Abnormal、作者: Author。

◆ 规范29: 尽量用for而不是while做循环。

◆ 规范30: 不要忽略Warning



内容

- ◆ 为什么要有编码规范
- ◆ 不好的编码例子
- ◆ 编码规范介绍
- ◆ 怎么落实编码规范



怎么落实编码规范（1/2）

- ◆ 编码规范是基本功，好的编码规范在没有成为自己的习惯之前，就需要记忆，一旦不假思索就能从指间蹦出优美的代码，规范也就失去意义了，或者说完成使命了。
- ◆ 通过交叉代码审查CDI来保证。
- ◆ 强制的管理手段。



怎么落实编码规范（2/2）

- ◆ CASE工具帮助进行代码审查。
 - C++Test
 - JTest

再见

