

OI编程基础规范及技巧总结

第1章 - 基础编程规范

何为编程规范？编程规范就是在编写程序需要遵守的一定之规，但并非语言本身所限制必须遵守的规则。我们不按照规范写也不会导致出错，但是代码将会很乱，并在调试或者维护方面带来不必要的麻烦或者隐患。所以说编程规范是我们需要养成的一种良好的写代码习惯。遵守编程规范可以带来什么好处呢？首先，符合编程规范的代码看着很整齐美观，结构清晰，另外良好的编程规范可以有效地减少程序出错的概率，并且方便对代码进行功能上的修改和扩充。

编程规范不是唯一的，像世界上很多知名软件公司都开发了适合自己公司的编程规范，公司内的员工工作时必须遵守该规范。这里我总结了适合Oier们C/C++编程的若干条规范。其中大部分不仅仅适用于OI竞赛。

以下是编程规范具体内容。

1. 代码风格及格式

1.1 宏定义，变量及函数的命名

- 一般情况下，宏定义采用大写字母+下划线方式命名，如果用于替换函数接口不在此例。示例如下：

```
#define DEFAULT_NUMBER 10000
#define EPSILON 1e-6
#define MAX(a,b) ((a)>(b)?(a):(b))
```

- 变量和函数的命名通常采用驼峰式命名或小写字母加下划线命名，变量通常为定语+名词的形式，函数通常为谓语+宾语的形式。驼峰式命名是指除了第一个单词以外每个单词首字母大写，如：

```
int maxValue;
void getResult ();
int addTwoNumbers (int a, int b);
```

小写字母加下划线为c语言经典风格，如：

```
int max_value;
void get_result ();
int add_two_numbers (int a, int b);
```

在编程中可根据喜好选用其中一种，或使用自己特殊的命名风格，但需注意保持一致，不要混用多种风格。

- 命名尽量清晰明了，有明确含义，尽量使用完整单词或大家能理解的缩写（除了类似for循环内的循环变量这样的惯用约定之外），以免引起误解。
- 采用命名加特定前缀，或使用命名空间（namespace）来避免可能的名字冲突。

1.2 空行，缩进及花括号对齐

- 利用空行来分隔意义相对独立的代码段
- 使用空格而非tab缩进，以免在不同编辑器看到不同的缩进效果。（编辑器通常都有将tab键自动转换为若干空格的设定）

- 花括号对齐有多种方式，以下为几种常见的风格：

```
// bsd风格
int max(int a, int b)
{
    if (a > b)
    {
        return a;
    }
    else
    {
        return b;
    }
}

// stroustrup(c++ 创始人)风格
int add(int a, int b)
{
    if (a > b) {
        return a;
    } else {
        return b;
    }
}

// Java风格
int add(int a, int b) {
    if (a > b) {
        return a;
    } else {
        return b;
    }
}

// GNU风格
int add(int a, int b)
{
    if (a > b)
    {
        return a;
    }
    else
    {
        return b;
    }
}
```

可选用自己喜欢的某种风格或自己定义的风格，但需注意保持统一。