软件编程规范

**卷心菜文档编辑器**

**2022年4月**

**目录**

[1. 基本要求 3](#_Toc55836206)

[2. 可读性要求 3](#_Toc55836207)

[3. 结构化要求 4](#_Toc55836208)

[4. 正确性与容错性要求 4](#_Toc55836209)

[5. 可重用性要求 4](#_Toc55836210)

[6. 编程规范 5](#_Toc55836211)

[6.1 命名规则 5](#_Toc55836212)

[6.2 注释 5](#_Toc55836213)

[6.3 未说明的 5](#_Toc55836216)

# 基本要求

1. 程序结构清晰，简单易懂，单个函数的程序行数不得超过300行
2. 代码精简，避免垃圾程序
3. 尽量使用标准库函数和公共函数
4. 使用括号以避免二义性
5. 不要随意定义全局变量，尽量使用局部变量

# 可读性要求

1. 可读性第一，效率第二
2. 保持注释与代码完全一致
3. 每个源程序文件，都有文件头说明，说明规格见规范
4. 每个函数，都有函数头说明，说明规格见规范
5. 主要变量（结构、联合、类或对象）定义或引用时，注释能反映其含义
6. 常量定义（DEFINE）有相应说明
7. 处理过程的每个阶段都有相关注释说明
8. 在典型、重要算法前都有注释
9. 循环、分支层次不要超过五层
10. 注释可以与语句在同一行，也可以在上行
11. 一目了然的语句不加注释
12. 注释的作用范围可以为：定义、引用、条件分支以及一段代码

# 结构化要求

1. 禁止出现两条等价的支路。
2. 禁止GOTO语句。
3. 用 if 语句来强调只执行两组语句中的一组。用 case实现多路分支。
4. 避免从循环引出多个出口。
5. 函数只有一个出口。
6. 不使用条件赋值语句。
7. 避免不必要的分支。
8. 不要轻易用条件分支去替换逻辑表达式。

# 正确性与容错性要求

1. 程序首先是正确，其次是优美。
2. 在编写完一段程序后，应先回头检查。
3. 改一个错误时可能产生新的错误，在修改前首先考虑对其它程序的影响。
4. 所有变量在调用前必须被初始化。
5. 对所有的用户输入，必须进行合法性检查。

# 可重用性要求

1. 重复使用的完成相对独立功能的算法或代码应抽象为公共控件或类。
2. 公共控件或类应考虑OO思想，减少外界联系，考虑独立性或封装性。
3. 公共控件或类应建立使用模板。

# 编程规范

## 6.1命名规则

（1）类名都以UpperCamelCase风格编写。

（2）测试类的命名以它要测试的类的名称开始，以Test结束。

（3）方法名都以lowerCamelCase风格编写。

（4）包名全部小写，连续的单词只是简单地连接起来，不使用下划线。

（5）常量名命名模式为CONSTANT\_CASE，全部字母大写，用下划线分隔单词。

（6）非常量字段名，参数名，局部变量名以lowerCamelCase风格编写。

## 6.2注释

1. 原则上注释要求使用中文；
2. 文件开始注释内容包括：作者名称、时间、模块用途、背景介绍等,复杂的算法需要加上流程说明；
3. 函数注释包括:输入、输出、函数描述、流程处理、全局变量、调用样例等，复杂的函数需要加上变量用途说明；
4. 程序中注释包括：修改时间和作者、方便理解的注释等。

**6.3未说明的**

本文档未说明的事项，可以参考google python style guide，若HTML文档也未说明，则有编码者自行处理。此时只有一个原则：保持一致性。即对问题的处理应该一致的贯穿编写的所有代码段。

引用一：文件开头的注释模板

///////////////////////////////////////////////////////////

// 文件名:

// 创建人:

// 日 期:

// 修改人:

// 日 期:

// 描 述:

// 版 本:

///////////////////////////////////////////////////////////

引用二：程序中的注释模板

///////////////////////////////////////////////////////////

// 注释内容

///////////////////////////////////////////////////////////