规范内容

1 标识符命名及书写规则

1.1 基本规则

1) 这里的标识符是指编程语言中语法对象的名字，它们有常量名、变量名、函

数名、类和类型名、文件名等，标识符的基本语法是以字母开始，由字母数字及下划线组成的单词。

2) 标识符本身最好能够表明其自身的含义，以便于使用和他人阅读。按其在应用中的含义由一个或多个词组成。可以是英文词或中文拼音词。

3) 当标识符由多个词组成时，每个词的第一个字母大写，其余全部小写，常量标识符全部大写。中文词由中文描述含义的每个汉字的头一个拼音字母组成。英文词尽量不缩写，如果有缩写，在同一系统中对同一单词必须使用相同的表示法。

4) 标识的总长度不要超过32个字符。

1.2 特殊约定

有的编程工具或项目开发小组对标识符的命名有自己的规定，例如，把标识符分为两部分：规范标识前缀+含义标识

1) 规范标识前缀用来标明该标识的归类特征， 以便与其它类型的标识互相区别，

例如：字符串变量标识符的前缀为str，某字符串变量可命名为： strExample；文本框对象标识符的前缀为txt，某文本框对象的命名可为： txtExample；

2) 含义标识用来标明该标识所对应的被抽象的实体，以便记忆，上面例子中strExample”的“ Example”就是含义标识。编程工具或项目开发小组有特殊约定的，以他们的约定为准。

1.3 源代码文件标识符命名规则

源代码文件标识符分为两部分，即文件名前缀和后缀。格式、规则如下：

××× \_×× **.**×××

1）前缀部分通常与该文件所表示的内容或作用有关。可以由项目组成员统一约

定。

2） 后缀部分通常表示该文件的类型，可以自己给定，具体的编程环境有特殊规定的以编程环境的规定为准。

3） 前缀和后缀这两部分字符应仅使用字母、数字和下划线。文件标识的长度不能超过32个字符，以便于识别。

2 注释及格式要求

注释总是加在程序的需要一个概括性说明或不易理解或易理解错的地方。注释应

语言简炼、易懂而又准确，所采用的语种首选是中文，如有输入困难、编译环境

限制或特殊需求也可采用英文。

2.1 源代码文件的注释

1) 在文件的头部必须标明程序名称，它所完成的主要功能。

2) 文件的作者，及完成时间。

3) 阶段测试结束后，主要修改活动的修改人、时间、简单原因说明列表。

4) 维护过程中需要修改程序时， 应在被修改语句前面注明修改时间和原因说明。

2.2 函数或过程的注释

1) 在函数头部必须对函数进行功能和参数（值参、变参）说明；

2) 在函数的主体部分， 如算法复杂时， 应以注释的方式对其算法结构作出说明；

3) 函数申请过全局资源且有可能导致资源紧张应加以注明(如内存，文件柄等)；

4) 函数有副作用一定以十分醒目的方式（如加!号等）注明；

5) 函数的长度在100语句行以内（不包括注释），程序有特殊要求时（如速度要求等）可以例外。

2.3 语句的注释

1) 应对不易理解的分支条件表达式加注释；

2) 不易理解的循环，应说明出口条件(有GOTO的程序还应说明入口条件)；

3) 过长的函数实现，应将其语句按实现的功能分段加以概括性说明；

4) 供别的文件或函数调用的函数，绝不应使用全局变量交换数据。

2.4 常量和变量的注释

在常量名字(或有宏机制的语言中的宏)声明后应对该名字作适当注释，注释说明的要点是：

1) 被保存值的含义(必须)；

2) 合法取值的范围(可选)；

3) 全局量需要对以上逐点做充分的说明。

3 缩进规则

3.1 控制结构的缩进

程序应以缩进形式展现程序的块结构和控制结构， 在不影响展示程序结构的前提下尽可能地减少缩进的层次。采用如下两种缩进方式之一：

1）

if (expression )

{

statements

}

else

{

statements

}

2）

if (expression ){

statements

}

else{

statements

}

3.2 缩进的限制

一个程序的宽度如果超出页宽或屏宽，这将是很难读的，所以本规范要求使用折行缩进的方法、合并表达式或编写子程序的方法来限制程序的宽度。

1）任何一个程序最大行宽不得超过80列，超过者应折行书写。

2）建议一个函数的缩进不得超过5级，超过者应将其子块写为子函数；

3）算法或程序本身的特性有特殊要求时，可以超过5级。\_\_