**代码规范**

|  |  |
| --- | --- |
| **类别** | **内容** |
| （1）变量命名 | ①开头字母用变量的类型，其余部分用变量的英文意思或其英文意思的缩写,尽量避免用中文的拼音,要求单词的第一个字母应大写。 |
| ②对非通用的变量，在定义时加入注释说明，变量定义尽量可能放在函数的开始处 |
| ③对struct、union、class变量的命名要求定义的类型用大写，并要加上前缀 |
| ④对const变量，要求在变量的命名规则前加入c\_ |
| （2）函数命名 | ①尽量用英文表达出函数完成的功能 |
| ②遵循动宾结构的命名法则，函数名中动词在前,并在命名前加入函数的前缀 |
| ③函数名的长度不得少于8个字母 |
| （3）注释 | ①必须在各功能模块的每一主要部分之前添加块注释，注释每一组语句，在循环、流程的各分支等，尽可能多加以注释 |
| ②循环、条件、选择等位置必须注释 |
| ③对于前后顺序不能颠倒的情况，建议在注释中增加序号 |
| （4）代码组织 | ①实现类似功能的文件应该放在同一个文件夹中或者同一个项目中 |
| ②尽量保证单级调用关系 |
| （5）类名 | 首字母大写 |
| （6）静态常量 | 全大写用下划线分割 |
| （7）代码格式 | 使用tab缩进源代码 |
| （8）域格式 | ①每行只能声明一个域 |
| ②域的声明用空行隔开 |
| （9）缩进风格(java) | ①大括号的开始在代码块开始的行尾 |
| ②闭合在和代码块同一缩进的行首 |
| ③同一层次的代码要保持整齐 |
| （10）空格的使用 | ①表示分割时用一个空格 |
| ②运算符两边用一个空格隔开 |
| ③逗号语句后如不换行，紧跟一个空格 |
| （11）代码嵌套 | 减少代码嵌套层次 |
| （12）变量的作用域 | 能用局部变量的，不要使用实例变量，能用实例变量的，不要使用类变量。 |
| （13）声明变量 | ①每次声明一个变量，不要使用组合声明 |
|  |  |
| ②当需要时才声明，尽快完成初始化 |
| （14）数字型的字面值 | long类型的字面值使用大写L为后缀，不要使用小写l，避免和1混淆 |
| （15）不用的代码和引用 | 删除 |
| （16）空行的使用 | ①连续两行的空行代表更大的语义分割 |
| ②方法之间用空行分割 |
| ③域之间用空行分割 |
| ④超过十行的代码如果还不用空行分割，就会增加阅读困难 |
| （17）线程 | ①防止过多的同步。 |
| ②避免从synchronized Method调用synchronized Method |
| （18）数据库、SDE | ①数据库连接、SDE连接必须关闭，包括在异常状态下。 |
| ②数据库对象、SDE对象必须关闭，包括在异常状态下。 |
| ③数据库操作异常必须捕获，并正确处理。 |
| （19）Jsp、Servlet | ①JSP文件的命名采用完整的英文描述，并且包括一个动词，第一个字母小写。 |
| ②Servlet的命名一般对应于所服务的对象加后缀Service来命名。 |
| ③JSP文件中不写复杂的业务逻辑，包括业务对象和数据库的操作。 |
| （20）异常 | ①要求对有异常抛出的方法都执行了try...catch..finally保护。 |
| ②生成Exception Class的Object后不抛出。建议抛出系统自定义的Exception Object。 |
| ③在catch Block中必须进行处理。 |
| ④不继承Error、Throwable Class。 |
| （21）String对象 | ①一个文件中不要出现多个相同的字符串。 |
| ②比较字符串时，使用equals()。 |
| ③确认equals左边的运算符为非空。 |
| ④String 与Object的转换中，使用目标对象的转换用的Method |
| ⑤不使用系统依赖符号。包括目录分隔符[/,\]和换行符[\r\n] |
| ⑥字符串连接使用“+”耗费资源，在频繁字符串连接时建议使用StringBuffer对象。（Jdk1.4、Jdk1.5） |