**Python代码评审标准**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 目的 | 指导你进行有效的代码复查 | # | # | # | # | 累计 | 累计% |
| 一般性说明 | 在完成每个复查步骤之后，将发现的某个类型的缺陷的个数记录在右面的栏目中。如果该步骤没有发现缺陷，就在右面的栏目中打个表示检查无误的交叉符号（X）。在开始复查下一个程序单元之前，要按照检查表完成对程序、函数的检查。 |  |  |  |  |  |  |
| 完整性 | 验证设计的所有功能都已经编码 |  |  |  |  |  |  |
| import | 验证import语句是完全的 |  |  |  |  |  |  |
| 初始化 | 检查变量和参数的初始化：   * 在程序的开始； * 在每个循环的开始； * 在函数/过程的入口 |  |  |  |  |  |  |
| 行长度 | 检查程序每行的长度：   * 是否每行不超过80个字符 |  |  |  |  |  |  |
| 名字 | 检查名字的拼写和使用：   * 是否前后一致？ * 结构和类中变量的引用是否使用了‘。’ |  |  |  |  |  |  |
| 字符串 | 检查所有的字符串：   * 是用指针来标记的； * 是以NULL结尾的 |  |  |  |  |  |  |
| 空格 | 检查所有的程序：   * 括号内是否有空格 * 逗号, 分号, 冒号前面是否有空格 * 是否用空格来垂直对齐多行间的标记, |  |  |  |  |  |  |
| 输出格式 | 检查输出格式：   * 换行是否合适？ * 间隔是否合适？ |  |  |  |  |  |  |
| {}对 | 保证检查{}是适当的并且是成对的 |  |  |  |  |  |  |
| 逻辑  操作符 | 验证==，=，||等逻辑操作符的使用是合适的。检查每个逻辑函数（）是合适的 |  |  |  |  |  |  |
| 逐行检查 | 检查每一行代码：   * 指令的语法是否正确？ * 表点是否正确、符号是否正确？ |  |  |  |  |  |  |
| 标准 | 保证所有代码符合编程标准 |  |  |  |  |  |  |
| 文件的打开与关闭 | 验证所有的文件：   * 是合适地声明的； * 是合适地打开的； * 是合适地关闭的； |  |  |  |  |  |  |
| 前面检查 | 前面检查以发现系统问题和非期望 |  |  |  |  |  |  |
| 总计 |  |  |  |  |  |  |  |