|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 姓名 | | 学号 | 班级 | | |
| 邓棋 | | 2018081062 | 软工182 | | |
| 程序名称 | 简单图书管理程序 | | | | |
| 自查方法 | 就多文件组织、排版、注释、标示符命名以及代码效率和安全性五个方面进行检查，记录检查结果，了解自己需改进的方面。 | | | | |
| 一级指标 | 二级指标 | | | 说明不达标之处 | |
| 文件组织  结构 | 采用多文件组织 | | | 无不达标处 | |
| 每个文件前有文件头部注释，列出文件名、作者、日期、描述、修改说明和版本声明 | | | 没有修改说明金额版本声明 | |
| 排版 | 采用缩进风格，严格采用阶梯层次组织程序码。  函数或过程的开始、结构的定义及循环、判断等语句中的代码都要采用缩进风格，case语句下的情况处理语句也要遵从语句缩进要求。 | | | 无不达标处，均已采用缩进 | |
| 相对独立的程序块之间、变量说明之后须加空行 | | | 无不达标处，均已加空行 | |
| 一行只写一条语句，长语句或表达式适当分行 | | | 无不达标处，均已适当分行 | |
| 在两个以上的关键字、变量、常量进行对等操作时，它们之间的操作符之前、之后或者前后要加空格；进行非对等操作时，如果是关系密切的立即操作符（如－>），后不应加空格 | | | 无不达标处，均已加空格分隔开 | |
| 程序块的分界符（如C语言的大括号'{'和'}'）应各独占一行并且位于同一列，同时与引用它们的语句左对齐 | | | 无不达标处 | |
| 注释 | 函数头部注释，列出：函数的目的/功能、输入参数、输出参数、返回值、调用关系（函数、表）等。 | | | 均有这些注释 | |
| 全局变量要有较详细的注释，包括对其功能、取值范围、哪些函数或过程存取它以及存取时注意事项等的说明。 | | | 有功能注释，没有注意事项等 | |
| 对变量的定义和分支语句（条件分支、循环语句等）必须编写注释 | | | 只有部分分支有简单注释 | |
| 注释的内容要清楚、明了，含义准确，防止注释二义性，并与其描述的代码相近 | | | 注释简洁、明了 | |
| 标示符  命名 | 标识符（变量名、函数名、符号常量等）的命名要清晰、明了，有明确含义，同时使用完整的单词或大家基本可以理解的缩写，避免使人产生误解——尽量采用采用英文单词或全部中文全拼表示 | | | 无不达标处 | |
| 一级指标 | 二级指标 | | | 说明不达标之处 | |
| 标示符  命名 | 自己特有的命名风格，要自始至终保持一致，可采用驼峰命名法、匈牙利命名法、帕斯卡命名法等 | | | 均采用帕斯卡命名法，无不达标处 | |
| 文件名的命名要求表达出文件的内容，要求文件名的长度不得少于5个字母，严禁使用象file1,myfile之类的文件名 | | | 无不达标处，文件名均是功能的英文简写 | |
| 代码效率及  安全性 | 循环体内工作量最小化，多重循环中，应将最忙的循环放在最内层 | | | 无不达标处 |
| 过程/ 函数中分配的内存，在过程/ 函数退出之前要释放 | | | 无不达标处，均已释放 |
| 防止内存操作越界，防止差1 错误 | | | 无不达标处 |
| 浮点表达式不应该测试其是否相等或者不相等，for控制表达式中不要包含任何浮点类型等 | | | 无不达标处 |
| 打开编译器的所有告警开关对程序进行编译，无编译错误或警告 | | | 无编译错误或警告 |
| 小结 | （*小结自查情况，哪些方面做的比较好？在哪些方面还需改进的？*）  大家都很积极，效率也很高，很认真地听别人讲程序；但是好像大家都只专注于自己的程序，提问时没有对别人的代码提出针对性的问题，问题过于笼统。 | | |  |