

Java 程序设计 课程设计方案

一、选题与内容要求

1. 学生自拟题目，可以在如下参考选题中选择：
 - 信息管理系统
 - 需要给出一个完善的需求背景和方案，基于架空场景是不允许的
 - 比如实现一个好用的北航图书馆或者完善教务管理系统
 - 多线程和网络编程
 - 后端
 - 比如基于 Java 的 web 服务端应用（软工后端）
 - 如果是纯服务端的方案，至少要能够通过某种方式展示你的功能
 - 造轮子
 - 比如自己写一个基于 Java 的 JSON 库
 - 实用产品
 - 比如基于 Java 的在线聊天室
 - 实用小工具
 - 比如一个实用的代码编辑器
 - 比如一个支持画图的图像编辑工具
 - 比如一个支持复杂功能的科学计算工具
 - 比如一个支持识别磁盘文件并提供预览功能的文件管理器
 - 游戏相关
 - 比如格斗类、飞行射击类、横版过关类
 - 比如为特定游戏提供一个 AI
 - 造轮子与其他杂项
 - 比如实现一个动画库
 - 比如为某产品定制 SDK、或基于产品与开源项目编写插件
 - 比如基于“酷 Q”，为其定制一个聊天插件并实现聊天机器人
 - 安卓开发
2. 限定使用 Java 编写，项目结构要合理
- 如果是编写后端应用，那么不限制配套前端的编码语言
3. 程序编写要有意义，能完成一定的功能（比如：游戏、计算等），功能范围不要求很全面，但是每一个功能都要实现地相对完整
- 比如提供画图软件，你可以不支持多种图像格式文件的解析，但是对于某一个你决定提供支持的格式，必须要对打开、导入、编辑、导出等功能都完整实现。
4. 代码需要经过测试，至少不应该在正常使用中出现内容逻辑的错误
5. 代码需要遵循 javadoc 标准编写代码注释
6. 允许使用第三方库，但是要在文档（见后文）指明选择该库的原因
- 比如实现更好看的 GUI
- 提供更稳定的 I/O 操作、网络相应的处理
7. 工作量的要求参见后文

二、提交要求

1. DDL 是 16 周周日晚 (2020 年 6 月 7 日)
 - a) 没有延期的余地, 不交即0分
2. 以小组为单位提交, 提交地点是云平台, 每组只需要组长提交
3. 提交一个描述项目的文档, 内容包含:
 - a) 小组成员的名字、学号、在项目中的任务职责
 - b) 编写项目的目的 (或项目想要达成的目标)
 - c) 项目运行环境, 包括硬件种类、JDK 版本、操作系统
 - d) 类图形式表现的类设计
 - e) 使用到了比较高级的算法或计数, 请吹出来
 - f) 涉及到的文件读写 (或网络请求), 需要指明文件 (或请求) 的解析规则 (协议)
 - g) 使用到的第三方库, 何处使用, 为何使用
4. 需要提供一个使用说明书, 内容包含:
 - a) 对于程序各部分使用方式的引导
 - b) 使用时可能发生的错误、错误原因与解决方案
5. 需要提供一个展示产品的 PPT
 - a) 如果安排展示, 会选择在 16 周周日上机时展示项目的 DEMO
 - b) 不一定会安排展示, 但是 PPT 必须要有

三、评分依据

以原创为前提, 提倡结合实际进行开发, 杜绝拷贝和代做程序, 一旦发现, 整组记 0

- 请尊重他人的代码成果, 使用第三方库必须要指明来源

以下是具体评分标准

1. 工作量 (约30%)
 - 根据小组人数粗略裁定, 建议在 2k~3k 行代码, 安卓开发及完善教务管理系统不适用此标准
 - 请不要刻意使用大量空行、大量注释、大量冗余代码来充数
1. 代码结构的设计和使用技术的水平 (约10+20%)
3. 产品的使用体验 (约15%)
 - 易用性
 - 错误处理与反馈
4. 各个文档内容的完整性 (约10%)
5. 小组成员分工的合理度 (约5%)

四、一些建议

这一部分不要求必须实现, 但是为了分数着想建议根据自己的项目酌情考虑。

- 对于给出的使用说明书, 用户至少能够在只根据界面上的内容和提示, 以及阅读使用说明书的情况下就能够正常使用所有功能。不应该让用户去猜测下一步要如何操作, 也不应该报出让用户摸不着头脑的错误和异常。通俗的说, 找几个其他同学, 让他们试用, 试用者体验时不会喷“什么垃圾玩意儿这么难用”, 那么就差不多了。
- 对于各种情况, 应该提供正确的错误处理 (构建新文件、超时重试、反馈错误后退出等)。
 - 在任何情况下, 程序不应该因为Exception 而爆炸。
 - 对于使用菜单式CLI的同学, Ctrl+Z 与 Ctrl+C 这种强制炸程序的操作可以不考虑。
 - 对于使用GUI的同学, 对于任务管理器和kill 这种强制杀程序的操作可以不考虑。
 - 对于使用GUI的同学, 当执行窗体关闭时, 应该能够对用户操作进行确认, 根据设计对于部分数据进行存储。
- 美观。这是一个很主观的标准, 因此不会有很多约束, 但是至少别辣眼睛吧:)。
- 通过连接数据库或文件读写来实现持久化。

- 项目结构合理，可以参考SpringMVC。
- 所有源文件，按照javadoc 的标准格式打上注释。
- 按照软工课的标准，给出完整的测试计划，至少包括单元测试和集成测试。对于单元测试，还应该给出测试代码并指明测试框架。
- 给出一个类图（或设计文档）描述你的代码设计。设计上的东西很难评价，遵循面向对象设计准则、高内聚低耦合等说法很难映射到具体的代码上。