

# 《软件系统设计》- 迭代一

---

项目背景

迭代一功能需求

1. 题目读取

2. 评分

提示

目录结构

提交方式

截止时间-2024.3.31 23:59:59

## 项目背景

作为一名软件工程专业的学生，大家对于在线评测系统（Online Judge, OJ）一定都不陌生，借助 OJ 平台，用户可以提交自己的代码，让系统测试程序能否在给定的测试用例下正常运行，从而检验程序的正确性、空间与时间消耗等信息。学生能够借助 OJ 平台练习编程，提高自己的编程能力，教师也可以借助 OJ 系统对学生的编程能力进行考核。

在本次课程项目中，我们希望大家结合在软件系统设计课上学到的面向对象设计原则及设计模式等内容，设计并实现一个类似OJ系统的项目，该项目将通过三次迭代完成，在每一次迭代中，我们将提供给你若干功能需求，你需要进行恰当的设计，完成代码编写，并提交**设计文档与项目代码**。

## 迭代一功能需求

在迭代一中，你需要实现 **题目读取** 与 **评分** 两个功能。

为了简化需求，使你更加关注系统的功能设计，我们并不需要你为你的系统设计用户界面、网络连接等等，只需要通过文件读写来完成所有的需求。（也就是说，本次我们实现的其实是Offline Judge）

在本次迭代中，我们将提供给你一个文件夹路径，其中包含两个子文件夹，分别是exams与answers，其中exams文件夹中包含若干考试文件，格式为JSON或XML，answers中包含若干回答文件，格式为JSON，你需要在读取exams文件夹内所有考试文件后为answers文件夹内的每一份回答文件打分，并将打分结果输出为csv文件。

你可以在**教学立方**的**附件**中获取到项目的初始框架与测试用例（seoj.zip），请在提供的框架上进行迭代开发，另外请不要改动Main类中main函数的接口（可以改动实现），我们后续将通过该类对你的代码进行测试。

## 1. 题目读取

在exams目录下，每一个文件对应于一次考试，你的系统应该支持分别从XML与JSON文件中读取一次测试的所有题目信息，文件中给出了考试的基本信息，包括考试名称、开始时间、结束时间和本次考试的题目列表等信息。

属性名	含义	示例
id	考试编号	1
title	考试名称	oj01
startTime	考试开始时间，以时间戳的形式给定（ms）	1704902400000
endTime	考试结束时间	1710417965206
questions	题目列表	见下表

在本次迭代中，我们要求你实现三种类型的题目，分别是 **单选题**、**多选题** 与 **编程题**。每类题目会包含若干属性。

属性名	含义	示例
id	题目编号	1
type	题目类型  1- 单选题  2- 多选题  3- 编程题	oj01
question	题目内容	下面哪些是C语言的关键字
points	本题分值	20

除了这些每个题目都具备的属性，还有一些不同题目所特有的属性

属性名	含义	示例
-----	----	----

options	对于单选题与多选题，提供若干选项	[ "char", "int", "float", "BigDecimal" ]
answer	对于单选题与多选题，提供正确答案，答案标号从0开始，多选题提供答案列表，用户回答时按照选项顺序以ABCD标号作答	0 [0, 1, 2]
scoreMode	对于多选题，允许设定给分模式，选项为nothing、fix、partial，含义见下文	fix
fixScore	当多选题的scoreMode设置为fix时提供该选项，表示少答应得的部分分	10
partialScore	当多选题的scoreMode设置为partial时提供该选项，表示少答时按选项确定部分分	[5, 10, 5]

多选题应当可以实现三种不同的给分模式，分别是 多答、错答、漏答均不得分（nothing）、多答不得分，少答得固定部分分（fix）、多答不得分、少答按照选项得部分分（partial）

比如一道多选题的答案是 ABD

在nothing给分模式下，答 ABD 可得points对应分数，答其他均不得分。

在fix给分模式下，答 A 、 B 、 D 、 AB 、 AD 、 BD 均可得部分分（fixScore）。

在partial给分模式下，按照partialScore给定的部分分确定得分，比如partialScore为[5, 10, 5]，当用户回答BD时得10+5=15分。

在本次迭代中，我们不要求实现对编程题的评分，这一功能将在后续迭代中进行实现，在本次迭代中，无论用户是否回答了编程题，以及回答是否正确，你都应该将编程题视为回答正确，并给对应的编程题满分。

## 2. 评分

在answer目录下，每一个文件对应一个用户在一个考试下的作答，文件中给出了考试id、学生id、提交时间与作答信息，我们确保不会出现一个用户多次作答同一考试的情况，你不需要考虑这一点。

你需要遍历answer目录下的每一份作答，为每一份作答打分，并将打分结果输出为csv文件，当提交时间早于考试开始时间或晚于考试结束时间时，**该份作答均得0分**。

csv文件中应包含三列，分别是examId、stuid与score，你需要为csv文件输出表头。

## 提示

1. 在每一次迭代中，你都需要认真思考功能需求，给出恰当的设计，你应该在灵活性、可扩展性与避免过度设计之间做出恰当的权衡。
2. 如何实现XML与JSON文件的读取并不是我们这次作业关注的重点，你并不需要自行完成XML与JSON文件的解析，我们也不推荐你自行解析这两种文件，这偏离了我们的设计重点，你可以按照你自己的偏好，使用任意类库来帮助实现。如果你不了解如何使用程序读取这两类文件的话，我们推荐你查询如下类库的相关资料。

在Java中读取JSON文件：fastjson、Gson、Jackson

在Java中读取XML文件：Jdom、Dom4j、Sax、Dom

## 目录结构

项目根目录应当包含doc文件夹，用于存放每一次迭代的设计文档，在第一次迭代中你需要在doc文件夹内创建子文件夹iter1，并在该文件夹内提交迭代一的设计文档，你可以使用任何格式文件编辑设计文档，但最终需要导出为PDF格式文件。

在设计文档末尾，你需要添加功能演示部分，体现系统实现满足了所有功能需求。

## 提交方式

1. 你需要使用git来管理你的代码，请在开发的过程中遵守git的提交规范，我们会通过git提交记录跟踪你的设计过程，每次迭代至少需要有一次提交记录，并且将每次迭代的最终版本的git提交信息设置为**iter{n}\_finish**（如第一次迭代为iter1\_finish）。
2. 每下一次迭代需要在上一次迭代的最终版上继续更改，在三次迭代结束后我们会检查完整的三次迭代的git提交信息。
3. 除了本地的git追踪外，你还需要在<https://git.nju.edu.cn/>上创建git仓库进行远程的备份，仓库命名

为**software\_design\_学号**，并将仓库设置为private，在项目三次迭代结束后，我们会收集所有同学的项目链接，并对你的项目实现与文档进行检查。

4. 前两次迭代结束时，你仅需将仓库中所有内容打包为zip压缩包，提交到教学立方上。
5. 为了控制提交大小，请务必通过 `.gitignore` 过滤build、target等产物目录。