

《软件系统设计》- 迭代二

1. 总览

2. 功能需求

2.1. 整体流程

2.2. 代码（编程题作答结果）的预处理

2.3. 代码（编程题作答结果）的执行

2.4. 并发需求

3. 测试用例

关于编程题测试用例的一些额外规定：

4. 评分

5. 提交方式

截止时间-2024.4.21 23:59:59

1. 总览

在迭代一中，大家已经实现了一个初步的在线评测系统（Online Judge），并已经实现了部分功能，本次迭代需要大家上次作业的基础上进行功能的扩展，最终提交**设计文档与项目代码**。

在迭代一中，大家已经实现了题目读取和评分两个功能。本次迭代的功能需求则主要围绕**编程题的评测**展开，编程题的评测需要借助有限的测试用例进行评测，之后判定这段代码是否正确。

2. 功能需求

2.1. 整体流程

我们首先给出本次迭代**要求的编程题评测流程**，整体流程可分为以下两个部分：

1. 代码（编程题作答结果）的预处理：对编译型语言（如Java，C），要对保存在文件中的代码进行编译。对解释型语言（如Python），则可以忽略这一步骤。
2. 代码（编程题作答结果）的执行：执行上一步预处理之后的可执行文件，并输入相应的参数，最终捕获输出，和答案进行对比，完成评测。

本次迭代只需要实现Java语言的编程题评测，但仍需要考虑到未来可能的变更，因此在这设计时我们希望大家考虑到不同语言的评测需求，**同时设计接口和实现类**，而不是只设计具体的实现类。

下面给出具体的流程细节要求。

2.2. 代码（编程题作答结果）的预处理

预处理阶段，我们建议的方式是获取到编程题作答结果所在的文件路径后，在java中调用命令行命令将作答结果编译成class。对编译好的class文件路径，我们不作要求。

注：当编译报错时，本题给0分。

2.3. 代码（编程题作答结果）的执行

执行阶段，我们建议的方式同样是在java中通过调用命令行命令执行编译后的class文件，为简化流程，编程题的输入参数直接通过命令行参数传递给class文件即可，参数传递和代码执行后的输出可通过**重定向或临时文件的形式**，具体我们不作要求。

注：当运行报错时，本题给0分。

2.4. 并发需求

并发在现实中是一种很重要的编程手段，但同时也是non-trivial的，本次迭代除了上述的两个功能需求之外，我们还希望大家自己设计一个线程池来加速编程题的评测，注意，线程池**不可以使用java库中自带的版本**，当然，我们要求实现的线程池并不复杂，总体来说，大家设计的线程池只需要满足以下几个条件即可：

1. 固定的、有限个的线程创建，为了统一，我们指定线程数量为5。在线程池创建之初就直接创建好固定的线程并放在线程池之中。（虽然现实中的线程池中的线程都是懒加载式，但是为了简化，我们直接去掉这一特性。）
2. 自动的任务调度。不要担心，我们要求的任务调度不会很复杂，只需要线程池能够自动将任务分配到现有的空闲线程上即可。
3. 线程池和任务类型需要是解耦的，也就是说，一个线程池类可以执行任意种任务，在本次迭代中，需要同时支持代码的预处理和执行。

注：为了完成并发需求，大家需要了解一点java中的一些多线程编程知识，如synchronized关键字以及thread-safe的一些数据结构等等。

3. 测试用例

关于测试用例部分，迭代一中的规定仍然不变：

在迭代一中，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。

关于编程题测试用例的一些额外规定：

- 编程题的测试用例在exams文件夹中的考试文件中，具体为编程题的**samples属性**，测试用例只包含**input**和**output**两项，input为需要传递给编程题代码的参数，output则为期望的输出。
- 编程题的答案保存在文件中，文件名则被包含在学生的答案中，为了统一和简化，**编程题的答案文件会被放置在answers/code-answers文件夹下，类名为Solution{xx}，只包含main函数，所需参数由main函数的参数传入。**

本次迭代需要在上次的代码上迭代开发，因此本次迭代我们只提供测试用例，大家只需要将本次的测试用例替换掉迭代一中的即可，也就是把解压后的resources文件夹替换掉src/test/resources。

4. 评分

编程题需要测试用例全部通过，任一一个测试用例不过均为0分。

其他评分的具体细则和迭代一保持一致

5. 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。

