Due Date

See due date in Canvas

Score

50

Problem Description

In this programming assignment, you will build a Java-based simulation tool based on the live poll idea (mentimeter) we used during the class. This is going to be a pure desktop-based console program, WITHOUT web interactions or graphical UIs required.

The basic functions of the system include:

1) Configure the question types and candidate answers. It should support 2 types of questions: 1) multiple-choice question (more than one choice can be chosen including one choice); 2) single choice question (only one choice can be chosen). For each type, you can configure the candidate answers, such as "A", "B", "C", "D", or "1. Right", "2. Wrong", etc.

2) Allow students to submit answers. Every student has a unique ID (String), and can submit an answer to the iVote service.

3) The Voting Service can be first configured with a given question type and candidate answers. Then, it can start accepting submissions from students. The important thing to keep in mind about accepting submissions is that each student can only submit one answer. If multiple submissions are received from the same student, only the last submission will be counted.

4) The Voting Service can also output the statistics of the submission results. You don't have to use the graphical interface. Simply use the standard output (System.out) to print out the count for each answer. For example, "A : 5, B : 12", or "1. Right : 15, 2. Wrong : 20". Feel free to customize the format for your output.

5) You need a SimulationDriver to automatically simulate the whole process. Basically, the driver will 1) create a question type and configure the answers; 2) configure the question for iVote Service; 3) randomly generate a number students and the answers; 4) submit all the students' answers to iVote Service; 5) call the Voting Service output function to display the result.

You should write the program in Java, and follow the important Java OOP principles we have learned so far. The minimum core concepts (classes) you probably need to have are:

1) Student
2) Question
3) VotingService
4) SimulationDriver

Of course, other classes, subclasses or interfaces may be needed depending on how you design and implement the system.

`SimulationDriver` should contain a `main()` method that triggers the whole process. You can randomly generate the IDs and answers for a group of students to simulate the submission process and check the results.

This programming assignment is very open. For the details that are not clearly specified above, feel free to make your own design decisions, but you should always clarify your decisions in your code with comments. Also, you can email me for any clarification about the requirements.

Submission Directions

Everyone should checkin your code in GitHub. You should use your own GitHub account, and create a project repository for your own assignment. **To submit your assignment, simply submit the public URL to your GitHub repo in Canvas.**

Getting Help

Please let me know if you need to meet to discuss any problems that you may have.