CMPS 360, Fall 2016
Programming Project #2

Due Date: As indicated on CMPS 360 Moodle site.

The coding for the solution to the following problem may be done individually or by partnering with another CMPS 360 student. You may ask for help on this project from the instructor or grader, but otherwise work on this project should be done individually or within the team. You may use your class notes, text, on-line Java tutorials, or talk about Java with anyone, but the code you write for this project must be your own. The project is to be created using NetBeans 8.1 and Java SDK version 8.

Programming Contest Bookkeeping System

Programming contest contestants (individuals or teams) are ranked in a variety of ways. A common way is to rank contestants on the number of problems solved correctly. The more problems solved, the higher the ranking. If more than one contestant solves the same number of problems, ranking within that number of problems solved is done by the total time taken to solve the problems. The shorter the total time, the higher the ranking. A bookkeeping system is used to keep track of the number of problems solved and the time taken to solve them for all contestants.

When the contest starts, the clock starts on all problems. When a solution to a particular problem is submitted by a contestant, the clock stops on that problem (only for that contestant!) while the solution is being judged. A correct solution is added to the contestant's number of problems solved and the time on clock for that problem remains fixed. If a solution is judged as incorrect, the contestant is notified and the clock restarted for that problem (only for that contestant!).

At the end of the contest, the number of solved problems per contestant and the total time taken by each contestant to produce correct solutions is used to create a rating of all contestants.

Create an object oriented system with a JavaFX user interface to serve as a programming contest bookkeeping system.

Project Requirements

- At program start, the user (the contest bookkeeper) enters the number of teams (maximum of 50), the number of programming problems available (maximum of 10) and the duration of the contest in minutes (maximum of 180).
- Teams are to be identified by consecutive numbers only. For example, if there are 18 teams, teams will be numbered from 1 to 18.
- The bookkeeper starts the contest.
- When the contest starts, clocks on all problems start and the main contest timer starts.
- When the contest is over (i. e. when time has expired), the clocks on all unsolved or unpaused problems halt.
- Judging will finish after the contest ends. The bookkeeping system must allow paused problems that are judged correct to be marked correct after time has expired.

- The interface shows elapsed time for any problem of any team when the controls of the problem are visible.
- The time remaining in the contestis always visible in the interface.
- A current listing of the state of all problems by team must be available for the bookkeeper to chose to view. Hint: See JavaFX class *TableView* (*javafx.scene.control.TableView*).
- The final ranking of all contestants is output to a file in human readable form at the discretion of the bookkeeper.

Tip:

Consider allowing the user to select a team then having the interface show the state of all problems for that team.

Additional Requirements

(a) Identifiers must be descriptive, i. e. must self document. The only exception granted is in the case of a "for variable", that is a variable created as a simple counter as in the control variable in a "for" loop statement.

(b) Indention of all code blocks (compound statements, anything in braces), including single statements following selection or while statements, is required. NetBeans will do this fairly automatically as you type if your syntax is correct. ALT-SHIFT-F will re-format a whole file (if your syntax is correct), which can help make it easier to debug logic in functions, loops, ifs and switches.

(c) The main ".java" file [the one with the method *public static void main(String[] args)*] of your project must contain this minimal documentation:

// *Your Name(s)*
// *Your CLID(s)*
// CMPS 360
// Programming Project : #
// Due Date : *due date*
// Program Description: *brief description of actions of your code*
// Certificate of Authenticity: (Choose one of the two following forms:)

I (we) certify that the code in the method functions including method function *main* of this project are entirely my own work.

{or}

I (we) certify that the code in the method functions including method function *main* of this project are entirely my own work., but I received some assistance from {*name*}. *Follow this with a description of the type of assistance. (For example, if you consulted a book, and your solution incorporates ideas found in the book, give appropriate credit; that is, include a bibliographical reference.) Note: You do not have to list the text, the author of the text or the instructors examples.*

Submitting

Clean the project, then zip it, naming the tar "p2_*your-clid*.zip". Upload the zip to Moodle.

Helpful Hint: Keep a backup copies of your project folder! Teams may want to use a version control system. (select Tools, Options, Team, Versioning, then choose) Also search the NetBeans Help system for "versioning", then scroll to the bottom of the first page for links to how to set up said control.