CSCI 1130A/B Introduction to Computing Using Java 2019-2020 First Term

Department of Computer Science and Engineering

The Chinese University of Hong Kong

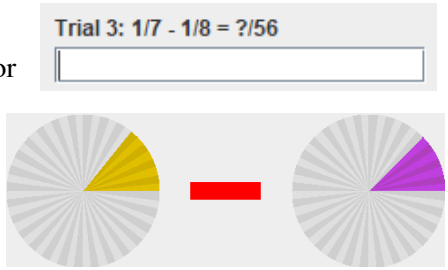
Assignment 3: Pie Sharing

Due date: 23 October 2019 (Wed) Full mark: 100

Expected normal time spent: 8 hours

Aims:

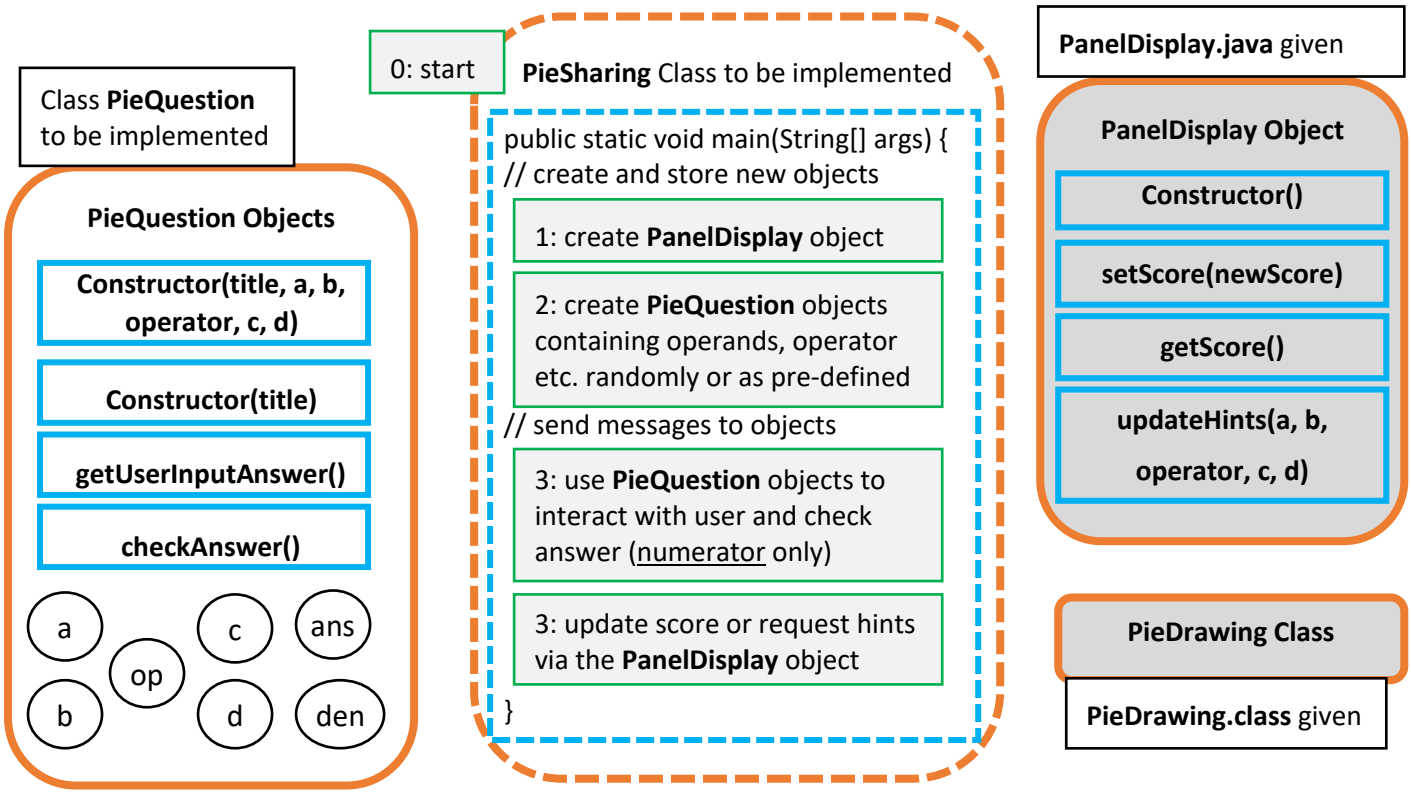
1. Practice defining classes, as well as creating and using objects.
2. Practice random number generation.
3. Practice interacting with user via JOptionPane dialogs.



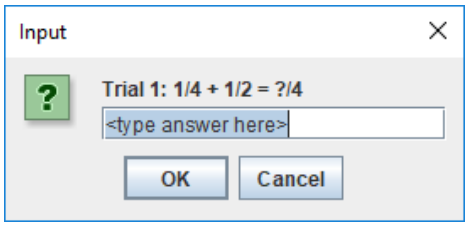
We are going to build a simple game called ***Pie Sharing*** with GUI dialogs for teaching basic fractions arithmetic within one same pie. Each Fractional Number Quiz is abstracted (represented) as an object named *PieQuestion*. *PanelDisplay* class is given for presenting scores and hints. The main class *PieSharing* links up all the objects and implements the workflow.

Procedure:

1. Create a new NetBeans project **`PieSharing`** with a new class **`exercise.PieSharing`**. Create another class **`PieQuestion`** under the package exercise. You just have to complete these two classes. The source code for the class **`PanelDisplay`** and the compiled byte codes for another class **`PieDrawing`** are given for your use.
2. Here is the flow of the whole application, explained in an OO diagram:



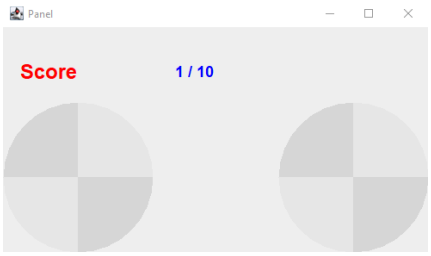
- Each `PieQuestion` object keeps several internal states in its fields, as shown in the diagram above. Operands *a*, *b*, *c* and *d* are integers and the denominator should be in the range of **2 – 8** inclusively, the numerator should be in the range of **1 – 7** inclusively. Operator represents one of these two operations on the operands: $\frac{a}{b}+\frac{c}{d}$ and $\frac{a}{b}-\frac{c}{d}$ while all input and resultant fractional numbers *f* ($\frac{a}{b}, \frac{c}{d}$ or $\frac{ans}{den}$) should be between **0** to **1** exclusive, i.e. $0 < f < 1$. $\frac{a}{b}, \frac{c}{d}$ and $\frac{ans}{den}$ should be in its reduced, or simplified form, e.g. $\frac{2}{4}$ should be reduced to $\frac{1}{2}$. (Hints: fraction reduction can be done by locating the Greatest Common Divisor (gcd), one practical approach is Euclidean algorithm (https://en.wikipedia.org/wiki/Euclidean_algorithm). For example, $\frac{5}{10}+\frac{1}{4}$ or $\frac{3}{2}-\frac{3}{4}$ are invalid.
- Define a constructor which receives parameters: title which is a *String*, *int* operands a, b, c, and d, as well as a *char* representing the operator either '+', '–', in the PieQuestion can check its validity:
  - First, it shall check if the operands a, b, c, d, are within proper range.
  - Then, it shall check if the char operator is valid. All non-compliant char symbols will be changed to '+', i.e., addition, by default.
  - Finally, according to the operation, calculate the expected answer, so that ans and den can be obtained and check if they are within proper range.
  - If any of the above conditions are not satisfied, set the question to $\frac{1}{4}+\frac{1}{2}$ by default.
- Define another constructor which receives just one parameter: a title String only. All the rest are ***randomly generated***, **within the proper aforementioned range and settings**. Pay particular attention to the calculated answer, for example, $\frac{9}{10}+\frac{1}{2}$ or, $\frac{1}{10}-\frac{1}{2}$ will produce out of range answers.
- Define a method **`getUserInputAnswer()`** which returns a String. It prompts the user with a **`JOptionPane`** input dialog in this form:




```
title: a/b op c/d = ?/den
<type answer here>
```

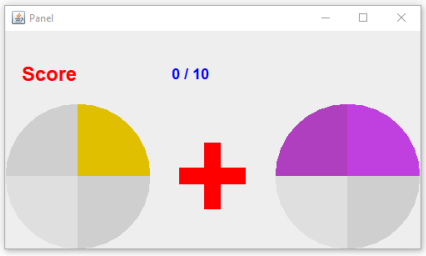
If the user clicks the Close or the Cancel button, it will receive a **`null`** String. Then the method should prompt the user again until getting a proper String response from the user. Using a default icon for the input dialog shall be good enough.

- Define another method **`checkAnswer()`** which accepts a String parameter and returns a **`boolean`** as result. You may use Integer.parseInt() with proper exception handling to convert a String to an integer value for comparison. Alternatively, you may convert the expected answer to a String representation for comparison. To compare the content of two String objects, you may use the *equals()* method.



  - If the answer (for non-trial questions) is correct, send a message to the PanelDisplay to add one point to the score.

- However, if the answer is wrong, send a message to the PanelDisplay to update the hint. User shall be given **ONE and only ONE** extra chance to answer another time by prompting for user input again. E.g. the right figure shall be shown by sending *updateHints(1, 4, '+', 1, 2)* to the PanelDisplay.



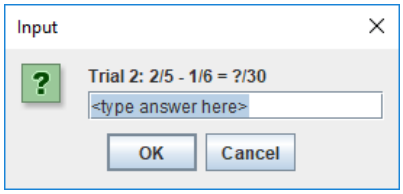
- Before showing the next question, display **NO** hints (clear the colored slices) like initially.
- You may define some other fields and methods that you need.

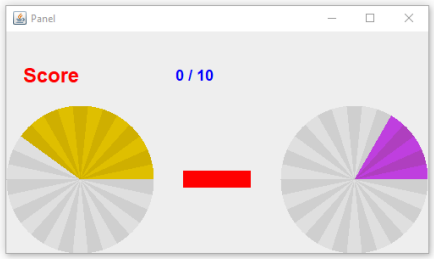
3. Define the main class `PieSharing` according to the steps described in the above OO diagram.
   - Initially it shall create and display a score board of 0/10 score and two empty pies. Then, it prompts the user with 3 pre-defined PieQuestion as trials, with sequential titles "Trial 1" through "Trial 3:

     a) $\frac{1}{4}+\frac{1}{2}=\frac{3}{4}$,

     b) $\frac{2}{5}-\frac{1}{6}=\frac{7}{30}$, and

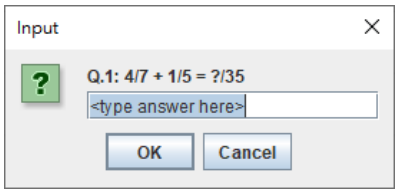
     c) $\frac{1}{7}-\frac{1}{8}=\frac{1}{56}$.





     For trials, no score is added on receiving a correct answer. But hints are shown on receiving an incorrect one and then it prompts user to input a second time.

   - Then, it prompts the user **10** randomly generated PieQuestion one by one, with sequential titles "Q.1" through "Q.10". Prompt the user for responses, update score or show hints for each PieQuestion by updating the PanelDisplay and collect an extra round of user input as required.



   - After answering all 10 questions properly, the user may click the *Close* button of the PanelDisplay to terminate the application.

Procedure:

1. **OO Design and Implementation Requirements:**
   - The given source file **`PanelDisplay.java`**, should be downloaded and saved to the **`src\exercise\`** folder of your NetBeans project **`PieSharing`**. NetBeans will detect it automatically in the package exercise and show the class **`PanelDisplay`** in your project.
   - After downloading the other 2 given files **`PieDrawing.class`** and **`PieDrawing$Slice.class`**, you have to create a new **`lib\exercise\`** folder inside your NetBeans project **`PieSharing`**. Then, right-click on your project in NetBeans, select the *"Libraries"* category, click on the *"Add JAR/Folder"*, and then select the **`lib`** folder. NetBeans will show the **`PieDrawing`** classes in your project automatically.

- Apart from the main() method in class **`PieSharing`**, there should **NOt** be any other static fields and/or methods.
- There should be three classes (excluding those compiled ones) in the whole Java Application project.
- You are expected to infer the type of the fields, methods and parameters.
- You are expected to use proper access modifiers.

2. Build the project (press the function key ***[F11]*** on the keyboard). If there are errors, don't panic. Check the error, correct it and re-compile. Feel tired? Take a rest.
3. You may insert `println()` statements in your work to inspect variables and intermediate results.
4. When you finish and there is no more error, you are ready to try out the program by pressing the function key ***[F6]*** on the keyboard. Enjoy your work!

Submission:

1. **Locate** your NetBeans project folder, e.g. **H:\PieSharing**.
2. ZIP the project folder **PieSharing** and Submit the file **PieSharing.zip** via our Online Assignment Collection Box on Blackboard **https://blackboard.cuhk.edu.hk**

Marking Scheme and Notes:

1. The submitted program should be free of any typing mistakes, compilation errors and warnings.
2. Comment/remark, indentation, style is under assessment in every programming assignments unless specified otherwise. Variable naming, proper indentation for code blocks and adequate comments are important. ***Include also your personal particulars and your academic honesty declaration statement in a header comment block of each source file.***
3. Remember to do your submission before 6:00 p.m. of the due date. No late submission would be accepted.
4. If you submit multiple times, **ONLY** the content and time-stamp of the **latest** one would be counted. You may delete (i.e. take back) your attached file and re-submit. We **ONLY** take into account the last submission.

Score distribution is as follows:

Category	Sub-category	Score
1. Naming	Project naming	3
	Package naming	3
	File naming	3
2. Personal Information		5
3. Coding Style	Indentation	3
	Comment	3
4. Compilation	Compilation error	-5 each; -10 max
5. Running	Runtime error	-5 each; -10 max
6. OO		10
7. Random		10
8. Result	Base Cases	10
	Test Case	30
	Total	**100**

University Guideline for Plagiarism:

Attention is drawn to University policy and regulations on honesty in academic work, and to the disciplinary guidelines and procedures applicable to breaches of such policy and regulations.
Details may be found at http://www.cuhk.edu.hk/policy/academichonesty/. With each assignment, students are required to submit a statement that they are aware of these policies, regulations, guidelines and procedures.

With each assignment, students are required to submit a statement that they are aware of these policies, regulations, guidelines and procedures, in a header comment block.

Faculty of Engineering Guidelines to Academic Honesty:

MUST read: https://www.erg.cuhk.edu.hk/erg/AcademicHonesty
(you may need to access via CUHK campus network/ CUHK1x/ CUHK VPN)