Applitools Software Engineer Support - Home Exercise

The attached folder contains a Calculator Java class implementation of a basic naive calculator **[Calculator.java]** with the following public methods:

1) **public double** add(Double a, Double b)
2) **public double** subtract(Double a, Double b)

The name of the calculator is passed as an attribute of the constructor when an instance is created.

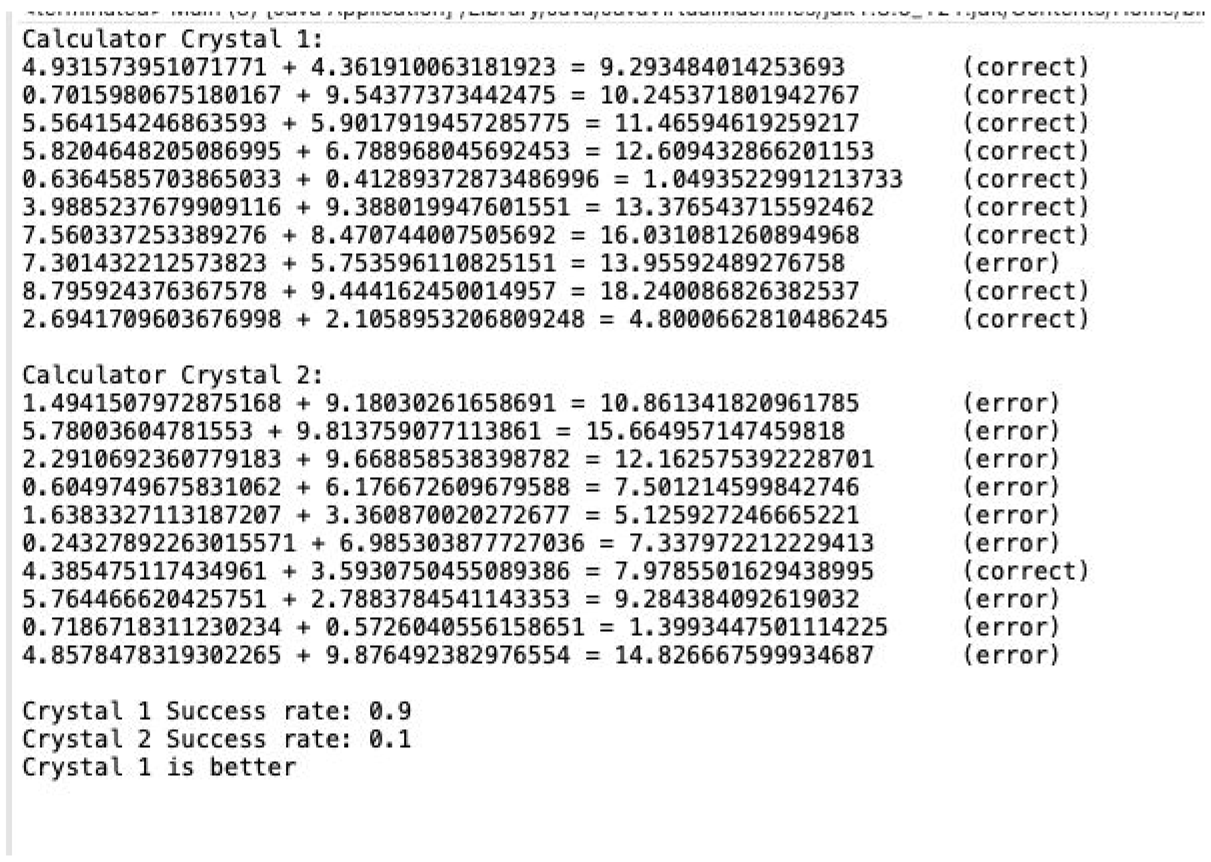
Background Story:
This implementation of the calculator is built to showcase an advanced futuristic technology based on Quantum calculations, which in some cases skews the result due to a small quantum error.

The accuracy of the calculations is derived from the type of crystal that is used to store the Qbits.

Assignment:
Build a testing project, to test which one of the two crystals is better to be used (has a higher accuracy rate).
Within your implementation, generate two instances of calculators with the following calculator names:

1) Crystal 1
2) Crystal 2

In order to test the accuracy of each calculator, 20 pairs of random numbers should be generated. An accurate result should be considered a “Successful Calculation” and a skewed result should be considered an “Error”.
At the end of the 20 samples, each calculator test should produce a “Success Rate” calculated as $\frac{\textit{Number of Successful calculations}}{20}$.
The output of the program should print the calculations, whether they are correct or an error, the success rates of the calculators and which Crystal is better to be used (the one with the higher success rate) to the console (see example below for 10 samples):



```
Calculator Crystal 1:
4.931573951071771 + 4.361910063181923 = 9.293484014253693       (correct)
0.7015980675180167 + 9.54377373442475 = 10.245371801942767      (correct)
5.564154246863593 + 5.9017919457285775 = 11.46594619259217      (correct)
5.8204648205086995 + 6.788968045692453 = 12.609432866201153     (correct)
0.6364585703865033 + 0.41289372873486996 = 1.0493522991213733   (correct)
3.9885237679909116 + 9.388019947601551 = 13.376543715592462     (correct)
7.560337253389276 + 8.470744007505692 = 16.031081260894968      (correct)
7.301432212573823 + 5.753596110825151 = 13.95592489276758       (error)
8.795924376367578 + 9.444162450014957 = 18.240086826382537      (correct)
2.6941709603676998 + 2.1058953206809248 = 4.8000662810486245    (correct)

Calculator Crystal 2:
1.4941507972875168 + 9.18030261658691 = 10.861341820961785      (error)
5.78003604781553 + 9.813759077113861 = 15.664957147459818       (error)
2.2910692360779183 + 9.668858538398782 = 12.162575392228701     (error)
0.6049749675831062 + 6.176672609679588 = 7.501214599842746      (error)
1.6383327113187207 + 3.360870020272677 = 5.125927246665221      (error)
0.24327892263015571 + 6.985303877727036 = 7.337972212229413     (error)
4.385475117434961 + 3.5930750455089386 = 7.9785501629438995     (correct)
5.764466620425751 + 2.7883784541143353 = 9.284384092619032      (error)
0.7186718311230234 + 0.5726040556158651 = 1.3993447501114225    (error)
4.8578478319302265 + 9.876492382976554 = 14.826667599934687     (error)

Crystal 1 Success rate: 0.9
Crystal 2 Success rate: 0.1
Crystal 1 is better
```

The implementation should:

1) Be written in Java.
2) Follow Object-Oriented concepts and be written **as modular and as scalable as possible**.
3) Include a separate thread running for each calculator (parallel runs).
4) Be completed **by yourself** - there is a high correlation between succeeding on this assignment and succeeding in the follow-up interview, as long as no assistance from others is needed. Please feel free to use Google for anything you need and avoid seeking help from friends :)

The final project should be submitted as a Zipped Folder after running a project “clean”.

Good luck! :)