Applitools- Home Exercise

In this home exercise, I have been asked to implement a java project that checks which one of two generated calculators is more accurate using tests.

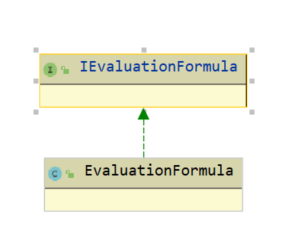
The code should be modular and parallel.

In this project, I tried to think about possible changes that customers may request in the future like adding more mathematical actions, change tests, change prints, change rate formulas, etc.

I use Tester as the main instance that handles the tests process where other components can be replaced according to changing demands like:

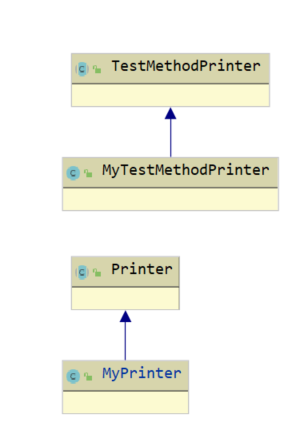
* EvaluationFormula- contains interface name IEvaluationFormula with method evaluate(int success, int fails).

Users can implement evaluation as he wishes by creating a class that implements this interface. For example, EvaluationFormula.java that use success/(fails + success) formula.



* Printer -responsible for printing the report. Using an abstract class Printer that has a printByFormat method that can be changed by demand.

I implemented MyPrinter class as an example that matches with the given example in the instructions.



* TestMethodPrinter is responsible for printing a single test sample in the required format.

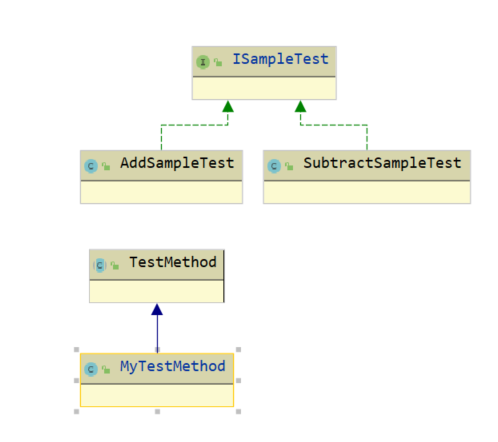
MyTestMethodPrinter is an example of an implementation that prints the test with parenthesis of correct/error.

I used a Strategy design pattern so I could support additional mathematical operations like multiply, division, etc.

ISampleTest is an interface that contains test() method and activates in runtime no matter which action tests now (add, subtract ..).

AddSampleTest.java and SubstractMethod.java implement this interface.

In case the user wants to expand the calculator action he just needs to add another class that implements ISampleTest, no need changing the code.



* TestMethod is an abstract class that responsible for determining test script – how many samples, which actions tests, counting failures and successes, add to StringBuilder object every sample that test in the chosen format for printing to console later.

I located every component in a different package to create isolation and capsulation.

The code runs parallel using thread for each calculator.

I used a list to maintain the tested calculators so there will be no limit to compere just two calculators.

Name: Sahar Ben Baruch.