

Exercise 1

Calculator : Add feature. Work in pairs. The implementer is lazy and writes ugly code. Tester forces generalized solution and is the designer of the API. Can the tester force internal quality of the code?

 Overtime our daily habits become refined skills and abilities,
 and translate into our expertise.

— *Brain Fitness : Peak Performance*

Shorter Version

TDD Kata- an exercise in coding, refactoring and test-first, that you should apply daily for at least 15 minutes.

- Try not to read ahead.
- Do one task at a time. The trick is to learn to work incrementally.
- Make sure you only test for correct inputs. There is no need to test for invalid inputs for this kata.

String Calculator

1. Create a simple String calculator with a method `int add(string numbers)`
 1. The method can take 0, 1 or 2 numbers, and will return their sum (for an empty string it will return 0) for example "" or "1" or "1,2"
 2. Start with the simplest test case of an empty string and move to 1 and two numbers

3. Remember to solve things as simply as possible so that you force yourself to write tests you did not think about
4. Remember to refactor after each passing test
2. Allow the Add method to handle an unknown amount of numbers
3. Allow the Add method to handle new lines between numbers (instead of commas).
 1. the following input is ok: "1\n2,3" (will equal 6)
 2. the following input is NOT ok: "1,\n" (not need to prove it - just clarifying)
1. Support different delimiters
2. to change a delimiter, the beginning of the string will contain a separate line that looks like this: "//
[delimiter]\n[numbers...]" for example "//;\n1;2" should return three where the default delimiter is ',' .
3. the first line is optional. all existing scenarios should still be supported