

String Calculator Kata

Description

This kata was created by Roy Osherove and is designed to help you learn test-first coding and refactoring. Try not to read ahead – do one task at a time. Work incrementally. Do as many steps as you can in a 30 minute period. Try the kata again from scratch until you can complete the entire thing within 30 minutes.

Kata Steps

1. Create a String calculator with a method `int Add(string numbers)`
 - a. The method can take 0, 1, or 2 numbers and will return their sum.
 - b. An empty string will return 0.
 - c. Example inputs: `""`, `"1"`, or `"1,2"`
 - d. Start with the simplest test case of an empty string. Then 1 number. Then 2 numbers.
 - e. Remember to solve things as simply as possible, forcing yourself to write tests for things you didn't think about.
 - f. Remember to refactor after each passing test.
2. Allow the Add method to handle an unknown number of arguments/numbers.
3. Allow the Add method to handle new lines between numbers (instead of commas).
 - a. Example: `"1\n2,3"` should return 6.
 - b. Example: `"1,\n"` is invalid, but you don't need a test for this case.
 - c. Only test correct inputs – there is no need to deal with invalid inputs for this kata.
4. Allow the Add method to handle a different delimiter:
 - a. To change the delimiter, the beginning of the string will contain a separate line that looks like this: `//[delimiter]\n[numbers]`
 - b. Example: `///
\n1;2"` should return 3 (the delimiter is ;)
 - c. This first line is optional; all existing scenarios (using `,` or `\n`) should work as before.
5. Calling Add with a negative number will throw an exception "Negatives not allowed: " listing all negative numbers that were in the list of numbers.
 - a. Example `"-1,2"` throws "Negatives not allowed: -1"
 - b. Example `"2,-4,3,-5"` throws "Negatives not allowed: -4,-5"
6. Numbers bigger than 1000 should be ignored.
 - a. Example: `"1001,2"` returns 2
7. Delimiters can be of any length, using this syntax: `//[***]\n1***2***3"` returns 6.
8. Allow multiple delimiters, using this syntax: `//[*][%]\n1*2%3"` returns 6.
9. Handle multiple delimiters of any length.

Provided by:

Source: <http://osherove.com/tdd-kata-1/>