# String Calculator

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

You can download a more readable version of the Kata here.

## Before you start:

- 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 signature:

   _____

   int Add(string numbers)

   _____

   The method can take up to two numbers, separated by commas, and will return their sum.

   for example “” or “1” or “1,2” as inputs.

   (for an empty string it will return 0)

   Hints:

   _____

   - Start with the simplest test case of an empty string and move to one and two numbers

   - Remember to solve things as simply as possible so that you force yourself to write tests you did not think about

   - 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)

      _____.

4. Support different delimiters

   1. 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 ‘;’ .
   2. the first line is optional. all existing scenarios should still be supported

      _____

5. Calling Add with a negative number will throw an exception “negatives not allowed” - and the negative that was passed.

   if there are multiple negatives, show all of them in the exception message.

   _____

   STOP HERE if you are a beginner. Continue if you can finish the steps so far in less than 30 minutes.

   _____

6. Numbers bigger than 1000 should be ignored, so adding 2 + 1001 = 2

   _____

7. Delimiters can be of any length with the following format: “//[delimiter]\n” for example: “//[***]\n1***2***3” should return 6

   _____

8. Allow multiple delimiters like this: “//[delim1][delim2]\n” for example “//[*][%]\n1*2%3” should return 6.

   _____

9. make sure you can also handle multiple delimiters with length longer than one char

_____

HERE ARE SEVERAL RUN THROUGHS OF THE KATA BY DIFFERENT PEOPLE IN VARIOUS LANGUAGES:



**Latest Blog Posts**

Co-Ops: Enabling True Continuous Delivery with Coope...
"Co-Ops" is a term that one of the attendees at a talk I was giving about cooperative pipelines came up with. I wish I knew his name. I asked him to email me afterwards so I can credit him but he n...Read more
Dec 6, 2019


Video: The Pipeline Driven Organization - Enabling T...
Read more
Nov 20, 2019


Continuous Delivery Israel Meetup - First Meeting Ju...
I'm pleased to announce that we will have the first ever meeting Continuous Delivery Israel meetup , in the AWS Floor28 offices in Tel Aviv, June 24th, 2019, at 6PM. The Group's page is located at ...Read more
Jun 2, 2019

InfoQ Book Q&A on Elastic Leadership
I had the privilege of being interview by Ben Linders over at InfoQ regarding my book on Elastic Leadership. You can read the article over here. Some of the main takeaways from it: A team can be ei...Read more
May 30, 2019

"Pipeline Driven" vs. "We have Jenkins Jobs"
High performing and continuously delivering organizations such as Netflix, Amazon, Google, Wix and others don't just have pipelines. They don't just automate stuff and put jobs on it. That's the ea...Read more
May 3, 2019


The Elastic Leadership Model in a picture
I realized that there isn't one picture that shows the entire elastic leadership model with the right amount of details out there. So I made one. Would love any feedback in the comments!Read more
Mar 22, 2019

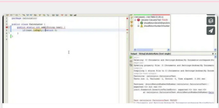

Delivery Pipelines and Discovery Pipelines
In a previous post I discussed the notion of having two seperate pipelines. I want to revisit this, with a bit more details. When we're talking about Continuous Delivery, we're very interested in t...Read more
Feb 20, 2019

Red Pipelines and Build Whisperers: Getting to a Tru...
People wonder why it takes such a long time to transform the way a large organization works. Here's just a small nugget, a small tip of a small iceberg, in a small corner of a large building in a l...Read more
Feb 18, 2019

Johanna Rothman Interview - Agile, a Decade Later
12 years ago I interviewed Johanna Rothman About managing people and her then new book "Behind closed doors: secrets of great management". A few weeks ago we ran again into each other on twitter an...Read more
Feb 16, 2019

A Pipeline Friendly Layered Testing Strategy P...

SUBMIT YOUR OWN IMPLEMENTATION



**Andrea Francia**

LINK TO VIDEO
https://vimeo.com/8506325

WHICH KATA?
String Calculator (Kata 1)

CODING LANGUAGE
Java

TEST FRAMEWORK USED
LamdaJ

CODE REPOSITORY URL

MY SITE/BLOG/TWITTER

LAST MODIFIED
24 June 2019

**Corey Heines**

LINK TO VIDEO
https://vimeo.com/7961506

WHICH KATA?
String Calculator (Kata 1)

CODING LANGUAGE
Ruby

TEST FRAMEWORK USED
RSpec

CODE REPOSITORY URL

MY SITE/BLOG/TWITTER

LAST MODIFIED
25 June 2019

**Dave Foley**

LINK TO VIDEO
https://anotherdave.wordpress.com/2010/01/13/calculator-tdd-ka...

WHICH KATA?
String Calculator (Kata 1)

CODING LANGUAGE
Javascript

TEST FRAMEWORK USED

CODE REPOSITORY URL

MY SITE/BLOG/TWITTER

LAST MODIFIED
25 June 2019

**Giordano Scalzo**

LINK TO VIDEO
https://vimeo.com/8326880

WHICH KATA?
String Calculator (Kata 1)

CODING LANGUAGE
Scala

TEST FRAMEWORK USED

CODE REPOSITORY URL

MY SITE/BLOG/TWITTER

Download CSV View larger version