

TDD Exercise

Purpose

Learn to use Test Driven Development to create a program.

Description

This is a individual assignment.

1. Create a GitHub repository called StringCalculator
2. Share this repository with the teachers account: hap2000
3. Programming environment:
 - Programming language: JavaScript
 - Testing library: Jest
 - Jest can be added using NPM
 - Once installed, do:
 1. `npm init`
 2. `npm install --save-dev jest`
4. **After each TDD cycle you need to create git commit with descriptive message.** If a commit is missing for that step, no score is given for that step
5. Make sure there are only files that are applicable for the project in the repository
6. When the project is finished you turn in the url to the GitHub repository in Canvas

In this exercise you are going to solve a popular Coding Kata called String Calculator. This kata was created by Roy Osheroove and is designed to help you learn TDD coding and refactoring.

Try not to read ahead – do one task at a time. Work incrementally.

Kata Steps

1. Create a String calculator with the function `Add(numbers)`
 1. The method can take 0, 1, or 2 numbers as a string, and will return their sum.
 2. An empty string will return 0.
 3. Example inputs: `""`, `"1"`, or `"1,2"`
 4. Start with the simplest test case of an empty string. Then 1 number. Then 2 numbers.
 5. Remember to solve things as simply as possible, forcing yourself to write tests for things you didn't think about.
 6. 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).
 1. Example: `"1\n2,3"` should return 6.
 2. Example: `"1,\n"` is invalid, but you don't need a test for this case.
 3. Only test correct inputs – there is no need to deal with invalid inputs for this kata.
4. Calling Add with a negative number will throw an exception "Negatives not allowed: " listing all negative numbers that were in the list of numbers.
 1. Example `"-1,2"` throws "Negatives not allowed: -1"
 2. Example `"2,-4,3,-5"` throws "Negatives not allowed: -4,-5"

5. Numbers bigger than 1000 should be ignored.
 - Example: "1001,2" returns 2
6. Allow the Add method to handle a different delimiter:
 1. To change the delimiter, the beginning of the string will contain a separate line that looks like this: "[delimiter]\n[numbers]"
 2. Example: "[delimiter]\n1;2" should return 3 (the delimiter is ;)
 3. This first line is optional; all existing scenarios (using , or \n) should work as before.

It is recommended that you practice your TDD skills by redoing this exercise several times or other TDD katas (search for TDD Kata). You can try to do as many steps as you can in a 60 minute period. I recommend that you try the kata again from scratch until you can complete the entire thing within 30 minutes. **In this exercise you need to finish all steps.**