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 String Calculator
2. Share this repository with the teachers account: hap2000
3. Programming language is Java
4. After each TDD circle you need to commit to git with descriptive message.
5. When the project is finished you turn in the url to the GitHub repository in MySchool.

In this exercise you are going to solve a popular Coding Kata called String Calculator. This kata was created by Roy Osherove 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 a method int Add(string numbers)
    i. The method can take 0, 1, or 2 numbers and will return their sum.
    ii. An empty string will return 0.

    iii. Example inputs: `“”, “1”, or “1,2”`
    iv. Start with the simplest test case of an empty string. Then 1 number. Then 2 numbers.
    v. Remember to solve things as simply as possible, forcing yourself to write tests for things you didn’t think about.
    vi. 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).
    i. Example: `“1\n2,3”` should return 6.
    ii. Example: `“1,\n”` is invalid, but you don’t need a test for this case.
    iii. 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.
    i. Example `“–1,2”` throws “Negatives not allowed: -1”
    ii. 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:
    i. To change the delimiter, the beginning of the string will contain a separate line that looks like this: `“//[delimiter]\n[numbers]”`
    ii. Example: `“//;\n1;2”` should return 3 (the delimiter is ;)
    iii. This first line is optional; all existing scenarios (using , or \n) should work as before.

> It is recommended that you practice you 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.**