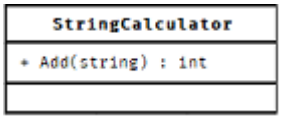
String Calculator

Rules

1. Strictly practice TDD: Red, Green, Refactor
2. Clean Code is required (Intention-revealing names, DRY, SOLID, etc.)
3. No need to check the validity of the input string

The 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, then move to one and two numbers
2. Allow the Add method to handle an unknown amount of numbers
3. Allow the Add method to handle new lines between numbers (in addition to commas).
    1. the following input is ok: "1\n2,3" (will equal 6)
    2. the following input DOES NOT need to be handled: "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 specifying the custom delimiter. This input looks like this: "//{delimiter}\n{numbers...}" (Note that the curly braces are representing the sections of the input and are not input formatting).
    2. For example: "//;\n1;2" should return a result of 3 because the delimiter is now ';'.
    3. The first line is optional (all existing scenarios should still be supported).
    4. Do not worry about supporting the specification of '\n' as an explicit custom delimiter. New lines should always be supported as delimiters in your number string.
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
6. Numbers bigger than 1000 should be ignored, so adding 2 + 1001 = 2

Bonus

1. Delimiters can be of any length with the following format: "//[{delimiter}]\n{numbers...}"
    1. For example: "//[***]\n1***2***3" should return 6.
    2. Note that the square brackets are required around the multiple character delimiter.
    3. A square bracket is not a valid delimiter.
2. Allow multiple delimiters like this: "//[{delim1}][{delim2}]\n{numbers...}"
    1. For example "//[*][%]\n1*2%3" should return 6.
    2. Note that once again the square brackets are required around each custom delimiter.
3. Make sure you can also handle multiple delimiters with length longer than one char

Linked List

Implement your own generic linked list without using built in classes.

Allow for:

- Insert: Insert a node at any position
- Delete: Delete a node at any position
- PrintList: Print the string representation of each node in the list.

Design Patterns

1. List your top 3 Design Patterns
2. Pick your favourite and briefly explain the pattern and its use case.
3. Draw a UML diagram to show the basic structure of the pattern.

*Alternative

Provide us with code you have written that you are allowed to show us and which you think best represent your knowledge and abilities.