KataRomanCalculator

About this Kata

CodersDojoSweden, running in Linköping, was looking for a new Kata which was sufficiently simple to get beginners up and running. ThomasNilsson suggested a calculator using Roman numerals.

Problem Description

"As a Roman Bookkeeper I want to add Roman numbers because doing it manually is too tedious." Given the Roman numerals, (IVXLCDM which means one, five, ten, fifty, hundred, fivehundred and a thousand respectively), create two numbers and add them. As we are in Rome there is no such thing as decimals or int, we need to do this with the strings. An example would be "XIV" + "LX" = "LXXIV"

There are some rules to a Roman number:

- Numerals can be concatenated to form a larger numeral ("XX" + "II" = "XXII")
- If a lesser numeral is put before a bigger it means subtraction of the lesser from the bigger ("IV" means four, "CM" means ninehundred)
- If the numeral is I, X or C you can't have more than three ("II" + "II" = "IV")
- If the numeral is V, L or D you can't have more than one ("D" + "D" = "M")

Clues

String grouping and concatenation is key to solving this kata. But remember the rule that lesser numerals can preceede bigger ones.

Suggested Test Cases

The first test case(s) are pretty easy. I won't put the test cases here since I feel this kata is one of the better ones I have done when it comes to practicing finding out what the next test case is.

Comments from those who are working on this Kata

When I invented it I did not know how well it would work. I was pleasantly surprised to see it work out to be a kata very focused on finding test cases. -- ThomasNilsson

---

This kata does not immediately show the "behaviour" aspect as does e.g. the Bowling kata. By this I mean the behaviour of a class under test, as opposed to testdriving a single method. However, in the Linköping dojo, we decided on adding this aspect by changing the story to "As a Polish born Roman Accountant..." and thereby introducing the Polish reverse notation to the calculation. The first test would then become:

```
public void testOnePlusOneShouldEqualTwo() {
   RomanPolishCalculator calculator = new RomanPolishCalculator();
   calculator.enter("I");
   calculator.enter("I");
   assertEquals("II", calculator.add());
}
```

And we also decided to add this aspect on an already running partial solution in the middle of the kata. This creates the situation of refactoring the API into another structure. We'll see how this turns out. -- ThomasNilsson

---

Here is a CoffeeScript? implementation (both forward and backwards).

I've just been learning CS and working with Node.js, npm, and Grunt. Here's a link to my GitHub? where it's hosted : https://github.com/tebriel/RomanNumeralConverterKata

--ChrisMoultrie?.
