

Introdução ao Teste Unitário

Exercícios Casos de Teste

- 1) Considere a classe “RomanNumeral” que segue. Ela possui um único método chamado “convert” com o propósito de converter numerais romanos em seu equivalente decimal. Projete casos de teste que você julgue testar adequadamente esta classe.

```
public class RomanNumeral {
    private static Map<Character, Integer> map;

    static {
        map = new HashMap<>();
        map.put('I', 1);
        map.put('V', 5);
        map.put('X', 10);
        map.put('L', 50);
        map.put('C', 100);
        map.put('D', 500);
        map.put('M', 1000);
    }

    public int convert(String s) {
        int convertedNumber = 0;
        for (int i = 0; i < s.length(); i++) {
            int currentNumber = map.get(s.charAt(i));
            int next = i+1 < s.length() ? map.get(s.charAt(i + 1)) : 0;

            if (currentNumber >= next) {
                convertedNumber += currentNumber;
            } else {
                convertedNumber -= currentNumber;
            }
        }
        return convertedNumber;
    }
}
```

- 2) A classe “Calculator” tem um método chamado “evaluate” que recebe por parâmetro uma string com um somatório de inteiros no formato “[inteiro] + [inteiro] + ... + [inteiro]” (entenda-se inteiro como um valor inteiro positivo). Ex: “54 + 22 + 32 + 4 + 105 + 2”. O método retorna o valor do somatório. Projete casos de teste que você julgue testar adequadamente esta classe.

```
public class Calculator {
    public int evaluate(String expression) {
        int sum = 0;
        for (String summand: expression.split("\\+"))
            sum += Integer.valueOf(summand);
        return sum;
    }
}
```

- 3) A interface abaixo define uma fila de inteiros de capacidade limitada. Projete um conjunto de casos de teste para testar adequadamente uma classe.

```
public interface BoundedIntQueue {
    // retorna a capacidade da fila
    int capacity();
    // retorna a quantidade de elementos na fila
    int size();
    // retorna true se a fila está vazia
    boolean isEmpty();
    // retorna true se a fila está cheia
    boolean isFull();
    // insere um elemento no fim da
    // fila (gera QueueFullException se a fila está cheia)
    void addLast(int newElement);
    // retira o primeiro da fila (gera
    // QueueEmptyException se a fila está vazia)
    int removeFirst();
}
```