

## Primeira Lista de Exercícios

### Exercício 1: A partir da especificação:

O programa TRI lê três valores inteiros que representam os lados de um triângulo. A partir dos valores, o programa informa se os lados formam um triângulo isósceles, escaleno ou equilátero. Condição: A soma de dois lados tem que ser maior que o terceiro lado.

- a. Aplique o critério Particionamento em Classes de Equivalência (PCE).
- b. Gere casos de teste para o critério PCE.
- c. Aplique o critério Análise do Valor Limite.
- d. Gere casos de teste para o critério Análise do Valor Limite de modo que sejam complementares aos casos de teste gerados em (b).

### Exercício 2: Considere o método **search()** com a seguinte especificação:

```
1. Public static ints search(List list, Object element)
2. //effect: if list or element is null throw NullPointerException
3. //else if element is in the list, return an index of element in
   list;
4. //else return -1
5. //for example, search([3,3,1],3), return 0 or 1
6. //           search([1,7,5],2), return -1
```

- a) Quais partições de equivalência devem ser geradas?
- b) Gere casos para o critério PCE.

**Exercício 3:** Considere um programa chamado FizzBuzz, que dado um número inteiro  $n$ , retorna-o como uma string composta de  $n$  seguido de "!". Portanto, para a entrada 13 o resultado do programa é "13!". Quando o número for divisível por 3, o programa usa "Fizz" em vez do número, e se o número for divisível por 5, o programa usa "Buzz"; caso  $n$  seja divisível por 3 e 5, o programa usa "FizzBuzz".

Um testador iniciante está tentando criar o máximo de testes possível para o programa FizzBuzz e apresentou os seguintes dados de entrada:

- T1 = 15
- T2 = 30
- T3 = 6
- T4 = 3
- T5 = 25.

Usando os conceitos do teste funcional (particionamento de equivalência e valor limite), quais dados de entrada poderiam ser removidos? Esses dados de entrada são suficientes para testar todas as características do programa? Justifique suas respostas aplicando o PCE e Valor Limite.