Verificação e Validação de Software I

Teste Baseado em Especificação

Exercícios Geração de Casos de Teste por Particionamento

1) O que vem a ser uma partição?
    a. Um conjunto de resultados produzidos por um método?
    b. Um conjunto de resultados que é produzido por uma entrada em diferentes métodos?
    c. Um conjunto de entradas onde todas fazem o método se comportar da mesma maneira?
    d. Um conjunto de entradas que gera exatamente o mesmo resultado em qualquer método?

2) Existe um programa chamado FizzBuzz. Ele faz o seguinte: dado um inteiro n, retorna a string formada pelo número seguido de uma "!". Então o inteiro 4 é convertido em "4!". Entretanto se o número é divisível por 3 use "Fizz" no lugar do número, se o número for divisível por 5 use "Buzz" e se for divisível tanto por 3 como por 5, use "FizzBuzz". Uma testadora novata está tentando desesperadamente encontrar tantos casos de teste quanto puder para o programa FizzBuzz. Ela apresentou os seguintes valores de entrada para compor os casos de teste:
    - T1 = 15
    - T2 = 30
    - T3 = 8
    - T4 = 6
    - T5 = 25

    Qual destes valores poderia ser removido sem que o conjunto deixe de ser um bom conjunto de teste? Que conceito podemos usar para determinar que valores podem ser removidos?

3) Analise uma versão ligeiramente modificada do Javadoc do método *put* da classe *HashMap* da API de Java:

```
/**
 * Puts the supplied value into the Map,
 * mapped by the supplied key.
 * If the key is already on the map, its
 * value will be replaced by the new value.
 *
 * NOTE: Nulls are not accepted as keys;
 *  a RuntimeException is thrown when key is null.
 *
 * @param key the key used to locate the value
 * @param value the value to be stored in the HashMap
 * @return the prior mapping of the key, or null if there was none.
 */
public V put(K key, V value) {
  // implementation here
}
```

Aplique o método de partição de categoria. Qual o menor conjunto adequado de partições? Gere um conjunto de casos de teste baseado nestas partições.

4) Os códigos de CEP em um determinado país são sempre compostos de 4 números seguidos de duas letras, por exemplo “2628CD”. Os números estão no intervalo [1000, 4000] e as letras no intervalo [C,M]. Considere um programa que recebe duas entradas: um inteiro (para os 4 números) e uma string (para as duas letras) e retorne true (se o CEP for válido) ou false (se o CEP for inválido). Um testador trouxe as seguintes partições:

a.	[0,999]
b.	[1000, 4000]
c.	[2001, 3500]
d.	[3500, 3999]
e.	[4001, 9999]
f.	[A-C]
g.	[C-M]
h.	[N-Z]

Observe que para um intervalo [a,b] todos os números incluindo a e b pertencem ao domínio. O mesmo ocorre com as letras, tais como [A-Z].
Qual dessas partições são válidas (e boas), isto é, quais podem ser usadas como partições para o processo de geração de casos de teste? Nomeie cada uma das partições válidas identificando a forma como exercitam o programa. Gere um conjunto de casos de teste baseado nestas partições.

5) Analise a seguinte versão ligeiramente modificada do Javadoc do método *add* da classe *HashSet* da API de Java.

```
/**
 * Adds the specified element to this set if it
 * is not already present.
 * If this set already contains the element,
 * the call leaves the set unchanged
 * and returns false.
 *
 * If the specified element is NULL, the call leaves the
 * set unchanged and returns false.
 *
 * @param e element to be added to this set
 * @return true if this set did not already contain
 *   the specified element
 */
public boolean add(E e) {
    // implementation here
}
```

Aplique o método de partição da categoria. Qual o menor e mais adequado conjunto de partições para o parâmetro de entrada *e*? Gere um conjunto de casos de teste baseado nestas partições.

6) Qual das seguintes afirmações é falsa sobre a aplicação do método de partição de categoria sobre o método que segue?

```
/**
 * Puts the supplied value into the Map,
 * mapped by the supplied key.
 * If the key is already on the map, its
 * value will be replaced by the new value.
 *
 * NOTE: Nulls are not accepted as keys;
 *  a RuntimeException is thrown when key is null.
 *
 * @param key the key used to locate the value
 * @param value the value to be stored in the HashMap
 * @return the prior mapping of the key,
 *  or null if there was none.
 */
public V put(K key, V value) {
  // implementation here
}
```

a) A especificação não define qualquer detalhe sobre o parâmetro de entrada *value*, e assim, a experiência deve ser usada para particionar o mesmo, por exemplo, sendo *null* ou *not null*.
b) A quantidade de testes gerados pelo método de partição de categoria pode crescer rapidamente na medida que a partições escolhidas para cada categoria são depois combinadas uma a uma. Isso não é um problema prático para o método *put* porque a quantidade de categorias e partições é pequena.
c) Em uma linguagem orientada a objetos, além de usar os parâmetros de entrada para explorar partições, devemos considerar também o estado interno do objeto (isto é, os atributos das classes), pois eles podem afetar o comportamento do método.
d) Com a informação nas mãos, não é possível executar o método de partição de categoria porque o código fonte é necessário para o último passo deste método: acrescentar restrições.

7) Um programa chamado *find,* que encontra as ocorrências de um padrão em um arquivo, possui a seguinte sintaxe: *find <pattern> <file>*. Um testador, depois de ler a especificação e seguir o método de partição de categorias encontrou as seguintes partições:

- Tamanho do padrão: vazio, um único caractere, muitos caracteres, padrão com mais de uma linha no arquivo;
- Aspas: o padrão está entre aspas, o padrão não está entre aspas, o padrão está entre aspas inadequadamente;
- Nome do arquivo: um bom nome de arquivo, nenhum arquivo com este nome, nome de arquivo omitido;
- Ocorrências no arquivo: nenhuma, mais de uma, exatamente uma
- Ocorrências em uma linha, assumindo que a linha contém o padrão: uma, mais de uma

Entretanto o número de combinações é muito alto. Que ações podemos tomar para reduzir o número de combinações?