

Respostas:

1) C

2)

Utiliza-se o conceito de partição ou classe de equivalência para determinar quais testes podem ser removidos. De acordo com este conceito apenas um valor precisa ser testado para cada classe de equivalência.

Podemos agrupar os testes de acordo com suas classes:

- Divisíveis por 3 e 5: T1, T2
- Divisíveis apenas por 3 e não por 5: T4
- Divisíveis por 5 e não por 3: T5
- Não divisível por 3 ou 5: T3

Precisamos apenas de um teste para a partição “divisíveis por 3 e 5”.

3)

Dois parâmetros: chave e valor.

O comportamento de retorno do método não depende do valor, mas da chave. Logo, precisamos definir diferentes características para a chave.

- a. A chave pode estar presente no mapa ou não
- b. A chave pode ser nula

Os requisitos não fornecem muitos parâmetros ou características então teremos de adicionar restrições.

As partições são as possibilidades para cada chave já que o valor não importa.

- a. Nova chave
- b. Chave existente
- c. Chave nula

Contudo, o estado atual do *HashMap* é algo que provavelmente pode influenciar a descoberta de defeitos pela nossa experiência com estruturas de dados:

- a. Estrutura vazia
- b. Estrutura com um par (chave,valor)
- c. Estrutura com vários pares (chave,valor)

Estado do <i>HashMap</i>	Chave	Valor	Resultado
()	“a”	1	null
()	null	1	RuntimeException
(“a”,1)	“b”	2	null

("a",1)	"a"	0	1
("a",1), ("b",2)	"c"	3	null
("a",1), ("b",2)	"a"	0	1

4)

Analisa-se cada partição fornecida e identifica-se se é uma partição válida:

- a) Partição válida: números inválidos que são muito pequenos
- b) Partição válida: números válidos
- c) Partição inválida: contém alguns números válidos, mas o intervalo é muito pequeno para conter toda a partição
- d) Partição inválida: mesma razão da letra c
- e) Partição válida: números inválidos que são muito grandes
- f) Partição inválida: contém tanto letras válidas como inválidas (o C está incluído no domínio)
- g) Partição válida: letras válidas
- h) Partição válida: letras inválidas, fora do intervalo de letras válidas

Número	String	Resultado
500	"B"	False
500	"H"	False
500	"R"	False
2500	"B"	False
2500	"H"	True
2500	"R"	False
5000	"B"	False
5000	"H"	False
5000	"R"	False

5)

Com relação ao elemento:

- a. elemento não presente no conjunto
- b. elemento já está presente no conjunto
- c. elemento NULL

Contudo, o estado atual do Set é algo que provavelmente pode influenciar a descoberta de defeitos pela nossa experiência com estruturas de dados:

- a. conjunto vazio
- b. conjunto unitário
- c. conjunto com vários elementos

Estado do Set	Elemento	Resultado
{}	"a"	True
{"a"}	"b"	True
{"a","b"}	"c"	True
{"a"}	"a"	False
{"a","b"}	"a"	False
{}	null	False

6)

D é a incorreta. Esta é uma técnica funcional. Não existe necessidade de código fonte.

7)

Ações possíveis:

- a. Pode-se tratar o tamanho do padrão "vazio" como excepcional e assim testar apenas uma vez.
- b. Podemos restringir as opções nas "ocorrências em uma única linha" caso as ocorrências para "ocorrências no arquivo" sejam apenas uma. Se não houver ocorrências no arquivo não faz sentido testar ocorrências na linha.
- c. Pode-se tratar o padrão "aspas incorretas" como excepcional e então testar apenas uma vez.