

2ª FREQUÊNCIA

Cotação:

8 janeiro 2018 - Qualidade de Software - 2018/19, Duração:120 minutos

- Não é permitida a consulta de livros e de apontamentos.
- Não se esclarecem dúvidas durante a Frequência. Se tiver dúvidas, indique a sua interpretação.
- Utilize uma caligrafia legível.

Número	
Nome	

1. [2 valores] Defina a metodologia de teste “Black Box”, e descreve as técnicas associadas ao “*Input Coverage*”.

Resposta:

2. [3 valores] Defina o que são Inspeções enquanto técnica de garantia de Qualidade de Software. Descreva as diferenças entre Inspeção formal e informal, e indique ainda a diferença entre Inspeção e Teste.

Resposta:

3. [5 valores] Considere o seguinte predicado:  $p = (a \wedge b) \vee (b \wedge c) \vee (a \wedge c)$

3.1. Preencha na tabela seguinte os valores calculados para  $p, p_a, p_b, p_c$

#	$a$	$b$	$c$	$p$	$p_a$	$p_b$	$p_c$
1	T	T	T				
2	T	T	F				
3	T	F	T				
4	T	F	F				
5	F	T	T				
6	F	T	F				
7	F	F	T				
8	F	F	F				

3.2. Indique:

3.2.1. Os pares que; para cada cláusula, verificam a General Active Clause Coverage (GACC)

3.2.2. Os pares que; para cada cláusula, verificam a Restricted Active Clause Coverage (RACC)

3.2.3. Os tuplos que; para cada cláusula, verificam a General Inactive Clause Coverage (GICC)

3.2.4. Os tuplos que; para cada cláusula, verificam a Restricted Inactive Clause Coverage (RICC)

4. [1 Valor cada] Indique na tabela seguinte as respostas a cada pergunta:

4.1	4.2	4.3	4.4	4.5

4.1 Qual das frases seguintes representa correctamente a diferença entre testes e *debugging*?

- A) Os testes identificam a origem de defeitos; *debugging* analisam os defeitos e propõem acções preventivas.
- B) Os testes apresentam as falhas causadas pelos defeitos; *debugging* detecta, analisa, e remove as causas das falhas no software.
- C) Os testes removem defeitos; *debugging* identifica as causas das falhas.
- D) Os testes previnem as causas das falhas; *debugging* elimina as falhas.

4.2 Como podem ser implementados os testes de caixa-branca durante os testes de aceitação?

- A) Para verificar se quantidades elevadas de dados podem ser transferidas entre sistemas.
- B) Para verificar se todas as instruções constantes no programa são executadas.
- C) Para verificar se todos os fluxos de processo estão cobertos.
- D) Para verificar todas as ligações entre páginas web.

4.3 Qual das seguintes opções é uma técnica de teste de caixa-preta?

- A) Técnica baseada na análise da arquitectura do software.
- B) Técnica baseada na análise do funcionamento do software de acordo com as suas especificações de *design*.
- C) Técnica baseada na análise do uso esperado do software.
- D) Técnica baseada na análise dos requisitos formais do software.

4.4 Considere um sistema de controlo de velocidade com as seguintes características:

Se a velocidade é igual ou inferior a 50km/h, nada acontece

Se a velocidade é superior a 50km/h, mas igual ou inferior a 55km/h, o condutor recebe um aviso

Se a velocidade é superior a 55km/h, mas igual ou inferior a 60km/h, o condutor é autuado

Se a velocidade é superior a 60km/h, o condutor fica inibido de conduzir (apreensão da carta de condução)

Assim, qual será o conjunto de teste mais adequado tendo em conta as especificações do sistema?

- A) 0, 49, 50, 54, 59, 60.
- B) 50, 55, 60
- C) 49, 50, 54, 55, 60, 62.
- D) 50, 51, 55, 60, 61

4.5 Qual das seguintes opções melhor descreve a divisão de tarefas entre gestor e técnico de testes.

- A) O gestor de testes define o plano de testes, e escolhe os *standards* a adoptar, enquanto o técnico de testes determina as ferramentas de teste a serem usadas.
- B) O gestor de testes planifica, organiza, e supervisiona as actividades referentes aos testes, enquanto o técnico de testes especifica e executa os testes.
- C) O gestor de testes planifica, e supervisiona as actividades referentes aos testes, enquanto o técnico de testes especifica e decide acerca das ferramentas de automação de testes a adoptar.
- D) O gestor de testes planifica, organiza, e especifica os testes, enquanto o técnico de testes prioriza e executa os testes.

5. [5 Valores] Considere o seguinte programa que efectua o cálculo de comissões de acordo com o tipo de conta associada a uma determinada transacção. Aplique técnicas de *Refactoring* por forma a optimizar/clarificar o programa.

```
1  class Account {
2      float principal;
3      float rate;
4      int daysActive;
5      int accountType;
6
7      public static final int STANDARD = 0;
8      public static final int BUDGET = 1;
9      public static final int PREMIUM = 2;
10     public static final int PREMIUM_PLUS = 3;
11 }
12
13 float calculateFee(Account accounts[]) {
14     float totalFee = 0;
15     Account account;
16     for (int i = 0; i < accounts.length; i++) {
17         account = accounts[i];
18         if ( account.accountType == Account.PREMIUM ||
19             account.accountType == Account.PREMIUM_PLUS ) {
20             totalFee += .0125 * (account.principal *
21                 Math.ex(account.rate * (account.daysActive/365.25) )
22                 - account.principal );
23         }
24     }
25     return totalFee;
26 }
```

Resposta:

