



Universidade Federal do Ceará
Campus de Crateús
Sistemas de Informação & Ciência da Computação

ENGENHARIA DE SOFTWARE [CRT0040]
TRABALHO EM DUPLA #1

(entregar em sala de aula as **respostas em sequência e escritas à mão**)

Prof. Allysson Alex Araújo

allysson.araujo@crateus.ufc.br

<http://crateus.ufc.br/allysson>

Nome completo e matrícula do aluno(a) 1:

Nome completo e matrícula do aluno(a) 2:

Atenção: seja **detalhista** nas suas respostas. Fiquem à vontade para anexar/colar ao trabalho imagens externas (post-its, figuras, etc).

1) (0,5 ponto) Com base no seguinte [post](#) sobre SOLID, elabore um exemplo para cada princípio esclarecendo a aplicação do mesmo. Ou seja, demonstre um exemplo e explique como o princípio se aplica ao mesmo.

2) (0,5 ponto) Explique as seguintes conceitual e cite um exemplo para cada:

- a) Integridade conceitual;
- b) Ocultamento de informação;
- c) Coesão;
- d) Acoplamento;
- e) Extensibilidade;
- f) Encapsulamento;
- g) Modularização;
- h) Abstração;
- i) Polimorfismo.

3) (0,5 ponto) Calcule o CBO e LCOM da seguinte classe:

```
1 class A extends B {  
2  
3     C f1, f2, f3;  
4  
5     void m1(D p) {  
6         "usa f1 e f2"  
7     }  
8  
9     void m2(E p) {  
10        "usa f2 e f3"  
11    }  
12  
13    void m3(F p) {  
14        "usa f3"  
15    }  
16 }  
17
```

4) (0,5 ponto) O que é um Padrão Arquitetural? Cite dois motivos que tornam a utilização de tal prática útil para a construção de um software.

5) (0,5 ponto) Para cada um dos seguintes padrões a) Cliente-servidor, b) MVC, c) Microsserviços e d) Arquitetura Orientado à Eventos, detalhe os principais conceitos, elabore uma figura de apoio e, em seguida, contextualize em detalhes um exemplo para cada.

6) (0,5 ponto) Explique o que significa desacoplamento no espaço e desacoplamento no tempo. Por que arquiteturas baseadas em filas de mensagens e arquiteturas Publish/Subscribe oferecem essas formas de desacoplamento?

7) (0,5 ponto) Para cada um dos itens abaixo explique em detalhes, diferencie e forneça dois exemplos:

- a) Verificação e validação de software;
- b) Teste funcional e teste não funcional;
- c) Teste dinâmico e teste estático;
- d) Teste de unidade, teste de integração e teste de sistema;
- e) Testes de aceitação, teste de regressão e testes de sistema;
- f) Técnicas e critérios de teste;

8) (0,5 ponto) Seja a seguinte função. Observe que ela possui quatro comandos, sendo dois deles if. Logo, esses dois ifs geram quatro branches:

```

1 public class Teste {
2
3     void f(int x, int y) {
4         if (x > 0) {
5             x = 2 * x;
6             if (y > 0) {
7                 y = 2 * y;
8             }
9         }
10    }

```

Supondo o código acima, preencha a próxima tabela, com os valores da cobertura de comandos e cobertura de branches obtidos com os testes especificados na primeira coluna (ou seja, a primeira coluna define as chamadas da função f que o teste realiza). Em seguida, explique sua resposta.

Chamadas da função feitas pela suíte de testes	Cobertura de comandos	Cobertura de branches
f(0,0)	1/4 = 25%	1/4 = 25%
f(1,1)		
f(0,1)		
f(0,0) e f(1,1)		

9) (0,5 ponto) Suponha que um programador escreveu o teste a seguir para a classe ArrayList de Java. Como você irá perceber, no código são usados diversos System.out.println. Ou seja, no fundo, ele é um teste manual, pois o desenvolvedor tem que conferir o seu resultado manualmente. Reescreva então cada um dos testes (de 1 a 6) como um teste de unidade, usando a sintaxe e os comandos do JUnit. Observação: se quiser executar o código, ele está disponível neste [link](#).

```

1 import java.util.List;
2 import java.util.ArrayList;
3
4 public class Main{
5
6     public static void main(String[] args) {
7
8         // teste 1
9         List<Integer> s = new ArrayList<Integer>();
10        System.out.println(s.isEmpty());
11
12        // teste 2
13        s = new ArrayList<Integer>();
14        s.add(1);
15        System.out.println(s.isEmpty());
16
17        // teste 3
18        s = new ArrayList<Integer>();
19        s.add(1);
20        s.add(2);
21        s.add(3);
22        System.out.println(s.size());
23        System.out.println(s.get(0));
24        System.out.println(s.get(1));
25        System.out.println(s.get(2));
26
27        // teste 4
28        s = new ArrayList<Integer>();
29        s.add(1);
30        s.add(2);
31        s.add(3);
32        int elem = s.remove(2);
33        System.out.println(elem);
34        System.out.println(s.get(0));
35        System.out.println(s.get(1));
36
37        // teste 5
38        s = new ArrayList<Integer>();
39        s.add(1);
40        s.remove(0);
41        System.out.println(s.size());
42        System.out.println(s.isEmpty());
43
44        // teste 6
45        try {
46            s = new ArrayList<Integer>();
47            s.add(1);
48            s.add(2);
49            s.remove(2);
50        }
51
52        catch (IndexOutOfBoundsException e) {
53            System.out.println("IndexOutOfBound");
54        }
55    }
56 }
57

```

10) (0,5 ponto) As leis de Lehman, como a noção de que a mudança é contínua, descrevem uma série de considerações provenientes de estudos, de longo prazo, de evolução de sistema. Escolha as 3 leis que você julga mais importantes e justifique sua resposta.

11) (0,5 ponto) Explique a diferença entre débito técnico, code smell e refactoring. Em seguida, forneça dois exemplos para cada.

12) (0,5 ponto) Faça um resumo de forma visual do seguinte [post](#) sobre Clean Code.

13) (0,5 ponto) Defina em detalhes e descreva os objetivos de DevOps e, em seguida, diferencie Integração Contínua, Entrega Contínua e Deployment Contínuo.

14) (0,5 ponto) Explique e diferencie as vantagens e desvantagens ao se usar uma abordagem monorepo e uma abordagem multirepo.

15) (0,5 ponto) Pesquise sobre Gerência de Configuração e Infraestrutura como Código e, finalmente, explique os principais conceitos e qual a relação de ambos com a área de DevOps.

16) (0,5 ponto) O que você mais gostou nessa segunda parte da disciplina? Simule que você está numa entrevista de emprego e explique o referido conceito, tanto do ponto de vista teórico, quanto prático.

17) (0,5 ponto) Após consumir o material da Sala de Aula Invertida S4, S5 e S6 faça uma análise interpretativa de no mínimo 15 linhas sobre as principais lições aprendidas extraídas do material contextualizando a entrega de software de qualidade.

18) (1,5 ponto) De forma alinhada ao projeto (seja considerando o Projeto Integrador IV para os alunos matriculados na referida disciplina OU do cenário da aplicação bancária disponível neste [link](#) para os demais alunos), elabore um [plano de casos de testes](#) para utilize este [template](#) para descrever os casos de teste. **Atenção: essa questão, em específico, por se tratar de uma tabela, pode ser entregue de forma impressa e grampeada com o presente trabalho.**