

Exercício de Revisão

INFORMAÇÕES DOCENTE						
CURSO:	DISCIPLINA:	TURNO	MANHÃ	TARDE	NOITE	PERÍODO/SALA:
ENGENHARIA DE SOFTWARE	PROJETO DE SOFTWARE		x		x	4º
PROFESSOR (A): João Paulo Carneiro Aramuni						

1) Sobre o diagrama de implantação da UML, analise as afirmativas abaixo e dê valores Verdadeiro (V) ou Falso (F).

() Mostra uma representação gráfica para um conjunto de classes

() Define como as máquinas estão conectadas e através de quais protocolos se comunicarão.

() Não deve ser utilizado quando o sistema a ser modelado for ser executado sobre múltiplas camadas.

Assinale a alternativa que apresenta a sequência correta de cima para baixo.

- A. V, V, F
- B. V, F, V
- C. F, F, V
- D. F, V, F

2. Considere as seguintes sentenças sobre diagramas de pacotes:

I. Os pacotes podem ser membros de outros pacotes, de modo que você obtém uma estrutura hierárquica.

II. Um pacote na UML é uma construção que permite agrupar classes, outros elementos não são permitidos no agrupamento.

III. A representação visual de um pacote em um diagrama de pacotes é tipicamente um cubo.

Assinale a alternativa correta:

- A. Apenas I está correta.
- B. Apenas I e II estão corretas.
- C. Apenas I e III estão corretas.
- D. Apenas II está correta.
- E. Todas estão corretas.

3. A função a seguir calcula o seno ou o cosseno de seu argumento.

```
float sin_or_cos(double x, int op) {  
    if (op == 1)  
        "calcula e retorna seno de x"  
    else  
        "calcula e retorna cosseno de x"  
}
```

Sobre essa função, avalie as asserções a seguir:

- I. A função abaixo possui um problema de coesão, isto é, ele implementa mais de uma funcionalidade.
- II. A função viola o Princípio de Demeter, visto que ela acessa objetos passados como parâmetros

Assinale a alternativa correta:

- A. Apenas I está correta.
- B. Apenas II está correta.
- C. I e II estão corretas.
- D. Nenhuma das alternativas está correta.

4. No que diz respeito à manutenção e reengenharia de software, um termo define o processo de alterar o código-fonte, de modo que não altere o comportamento externo e ainda melhore a sua estrutura interna. É uma técnica disciplinada de limpar e organizar o código, e por consequência, minimizar a chance de introduzir novos bugs. Esse termo é conhecido como:

- A. elicitação.
- B. refatoração.
- C. recodificação.
- D. replicação.

5. Considere as seguintes qualidades do projeto de um sistema de informação:

(I) Coesão, (II) Acoplamento, (III) Reutilização.

Para obter uma boa qualidade de projeto, o projetista deve tentar:

- A. minimizar I, II e III;
- B. minimizar I, II e maximizar III;

- C. minimizar I, maximizar II e III;
- D. maximizar I, minimizar II e maximizar III;
- E. maximizar I, II e III.

6. Assinale a opção correta quanto à abordagem conceitual de abstração sob o paradigma de programação orientada a objetos.

A. As abstrações, idealmente, caracterizam-se por não serem grandes demais em comparação aos módulos, pois senão elas se tornam multifuncionais e de difícil compreensão. Como consequência, a abstração deve ser implementada apenas no nível de estruturas de dados necessários para se atingir o objetivo pretendido.

B. Abstração consiste em uma linguagem puramente lógica. A motivação para isso veio em parte da vontade de se reconciliar o uso da lógica como uma linguagem declarativa de representação do conhecimento com a representação procedimental do conhecimento.

C. Abstração é uma linguagem declarativa que permite acesso à base de dados mediante a utilização da teoria dos conjuntos e da álgebra relacional como fundamento de seu funcionamento.

D. Abstração é um conceito segundo o qual o sistema ou software é dividido em partes distintas. Compõe o ferramental necessário para um programa mais legível com uma melhor manutenção e melhor desempenho por meio da programação orientada a objetos.

E. Abstração é a habilidade de se concentrar nos aspectos essenciais de um contexto qualquer, ignorando características menos importantes ou acidentais. Em modelagem orientada a objetos, uma classe é uma abstração de entidades existentes no domínio do sistema de software.

Questões V/F

Avalie as sentenças sobre Padrões Arquiteturais a seguir em verdadeiras (V) ou falsas (F). **Justifique** as sentenças falsas

1. () Uma Arquitetura em Camadas torna mais difícil a manutenção e evolução de um sistema de software.

Justificativa:

2. () Em uma arquitetura Três Camadas temos as seguintes camadas: interface com o usuário, aplicação (ou lógica de negócio) e banco de dados.

Justificativa:

3. () A camada de Visão mais o Modelo constitui a interface gráfica de sistemas MVC.

Justificativa:

4. () Em Arquiteturas MVC, classes de Modelo não podem ser usadas por várias Visões.

Justificativa:

5. () São vantagens de MVC: separação entre interface gráfica e modelo; especialização do trabalho de desenvolvimento; e testabilidade.

Justificativa:

6. () Microserviços devem compartilhar o mesmo banco de dados, por questões de segurança.

Justificativa:

7. () Dentre outras vantagens, microserviços oferecem independência de tecnologia, tolerância a falhas parciais e uma menor latência na comunicação entre os componentes de uma aplicação.

Justificativa:

8. () Big ball of mud é um padrão arquitetural que descreve como organizar a comunicação entre módulos de um sistema.

Justificativa:

9. () Na versão original de MVC (adotada em Smalltalk-80), um Controlador é um tratador de eventos de interfaces gráficas, tais como eventos gerados pelo mouse, teclado, etc.

Justificativa:

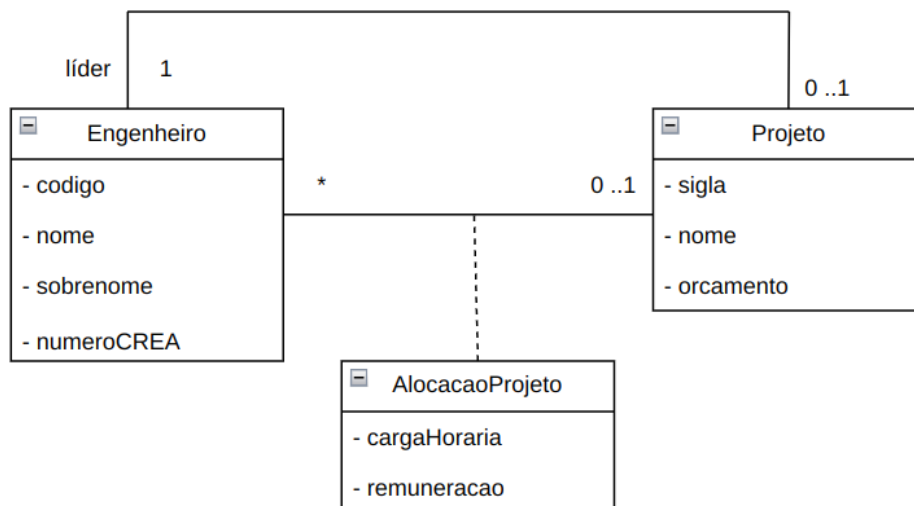
10. () Microserviços devem ser fortemente coesos e fracamente acoplados.

Justificativa:

Diagramas

1) Considere os diagramas de classes a seguir. Realize o mapeamento deste diagrama para o modelo relacional. Para tanto, utilize o seguinte formato: **Tabela 1 (Campo 1 (PK), campo 2, campo 3 (FK))**.

Engenheiros podem liderar um projeto por vez dentro da empresa, enquanto cada projeto tem, necessariamente, um só um engenheiro líder. O diagrama também informa que pode haver diversos engenheiros trabalhando no projeto, embora não possa haver um engenheiro trabalhando em mais de um projeto.



Engenheiro (codigo (PK), nome, sobrenome, numeroCREA)

Projeto (id (PK), sigla, nome, orcamento, idEngenheiroLider (FK))

AlocacaoProjeto(id (PK), idEmpregado (FK), idProjeto (FK), cargaHoraria, remuneracao)

2) Elabore um diagrama de componentes para uma aplicação comercial, em que:

- Uma “Caixa Registradora” pode se conectar com um componente “Servidor de Vendas”, usando uma interface de mensagens de vendas;
- Como a rede não é confiável, um componente “Fila de Mensagens” é introduzido para que a caixa possa se comunicar com o “Servidor de Vendas”, quando a rede estiver ativa, e se comunicar com uma fila, quando a rede estiver desativada;
- A “Fila de Mensagens” se comunicará então com o componente “Servidor de Vendas”, quando a rede se tornar disponível;
- Como resultado, o componente “Fila de Mensagens” fornece uma interface para se comunicar com a “Caixa Registradora” e exige essa interface para se comunicar com o “Servidor de Vendas”;
- O “Servidor de Vendas” exige uma interface para se comunicar com o “Sistema de Contabilidade”.

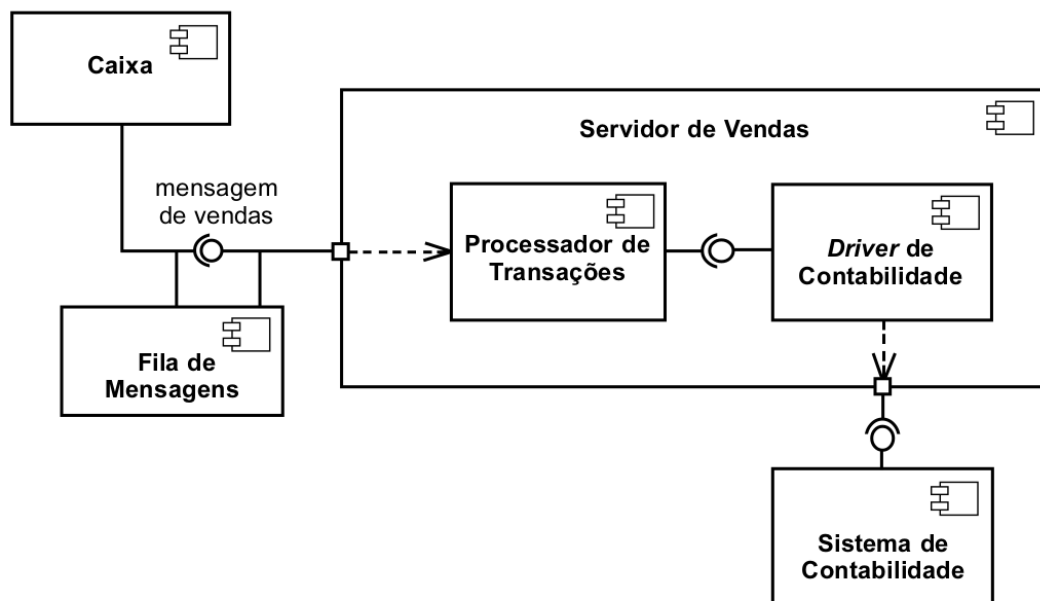


FIGURA 14.2 Um exemplo de diagrama de componentes.

Fonte: Martin Fowler, UML Essencial. (obs: sem parte de dentro do servidor de vendas)