



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO PAMPA – UNIPAMPA

CONCURSO PÚBLICO
NÍVEL INTERMEDIÁRIO

TARDE

CADERNO DE PROVA
PARTE II
CONHECIMENTOS ESPECÍFICOS

CARGO 49:
TÉCNICO DE TECNOLOGIA DA INFORMAÇÃO
ÁREA: PROGRAMAÇÃO DE SISTEMAS

ATENÇÃO!

Leia atentamente as instruções constantes na capa da Parte I do seu caderno de prova.

- 1 Nesta Parte II do seu caderno de prova, confira inicialmente se os seus dados pessoais e se os dados identificadores do seu cargo transcritos acima coincidem com o que está registrado em sua **folha de respostas** e em cada página numerada desta Parte II do seu caderno. Caso o caderno esteja incompleto, tenha qualquer defeito, ou apresente divergência quanto aos seus dados pessoais ou aos dados identificadores do seu cargo, solicite ao fiscal de sala mais próximo que tome as providências cabíveis, pois não serão aceitas reclamações posteriores nesse sentido.
- 2 Quando autorizado pelo chefe de sala, no momento da identificação, escreva, no espaço apropriado da **folha de respostas**, com a sua caligrafia usual, a seguinte frase:

Saber é compreender as coisas que mais convêm.

OBSERVAÇÕES

- Não serão objeto de conhecimento recursos em desacordo com o estabelecido em edital.
- Informações adicionais: telefone 0(XX) 61 3448-0100; Internet — www.cespe.unb.br.
- É permitida a reprodução deste material apenas para fins didáticos, desde que citada a fonte.

CONHECIMENTOS ESPECÍFICOS

A linguagem SQL tornou-se um padrão para o sistema de banco de dados relacional e pode ser considerada como uma das principais razões para o sucesso desse sistema. A respeito das características da linguagem SQL, julgue os itens a seguir.

- 51** A padronização da linguagem SQL ocorreu após trabalho conjunto do ANSI e da ISO, o que resultou na primeira versão denominada SQL-86.
- 52** No padrão SQL-86, as tabelas e os componentes que pertencem a uma mesma aplicação são agrupados em um conceito denominado esquema.
- 53** Os tipos de dados no padrão SQL-92, disponíveis para atributos, incluem somente dados numéricos, *string* de caracteres e data.
- 54** Forte extensão do padrão SQL-92 é a incorporação das características de orientação a objeto.
- 55** O tipo de dados abstrado na SQL3 possibilita ao usuário criar tipos semelhantes ao conceito de classe.

```
1 procedure SORT (A , n)
2   for i ← 1 to n do
3     j ← i
4     for k ← j + 1 to n do
5       if A(k) < A(j) then j ← k
6     end
7     t ← A(i); A(i) ← A(j); A(j) ← t
8   end
9 end SORT
```

Com base no algoritmo acima, julgue os itens seguintes.

- 56** O procedimento `SORT (A , n)` classifica em ordem crescente um conjunto $n \geq 1$ de números inteiros e o resultado permanece em `A (1:n)` assim que $A(1) \leq A(2) \leq \dots \leq A(n)$.
- 57** Quando $i = 1$, o laço das linhas 4 até 6 é executado para $k = 2, 3, \dots, n$. O valor `A` para j é determinado assim que $A(j) \geq A(k)$, para $k = 2, \dots, n$.
- 58** Quando $i = j$ e $j = 1$, a permuta da linha 7 não terá efeito. Esta situação só ocorre quando, na execução do procedimento `SORT (A , n)`, $n = 1$.

O processo de depuração de um *software* ocorre como consequência de teste bem-sucedido, isto é, quando um caso de teste descobre um erro, a depuração é a ação que resulta na reparação do erro. A esse respeito, julgue os itens a seguir.

- 59** O processo de depuração pode ser corretamente considerado como um caso de teste de *software*.
- 60** O processo de depuração tem sempre como resultado a identificação da causa do problema, e, como consequência, a correção do problema.
- 61** A categoria de depuração denominada força bruta é provavelmente o método mais comum e menos eficiente para isolar a causa de um erro de *software*.
- 62** O *software* `Jprobe ThreadAnalyzer` auxilia na avaliação de problemas de trilha, condições de bloqueios, de parada e de execução, que podem gerar várias armadilhas para o desempenho de aplicações Java.

Julgue os próximos itens, a respeito do modelo de dados entidade-relacionamento (E-R), que tem por base a percepção do mundo real como um conjunto de objetos básicos, chamados entidades, e do relacionamento entre eles.

- 63** Se E_1, E_2, \dots, E_n são conjuntos de entidades, obrigatoriamente distintas, então um conjunto de relacionamentos é um subconjunto de $\{(e_1, e_2, \dots, e_n)$, tal que $e_1 \in E_1, e_2 \in E_2, \dots, e_n \in E_n\}$, em que (e_1, e_2, \dots, e_n) são relacionamentos.
- 64** No modelo E-R, é sempre possível recompor um conjunto de relacionamentos não binários (n -ésimos, com $n > 2$) por um número de conjuntos de relacionamentos binários distintos.
- 65** No modelo E-R, é possível expressar relacionamentos entre relacionamentos.
- 66** Em uma entidade no modelo E-R, sempre há atributos suficientes para formar uma chave primária.
- 67** No modelo E-R estendido, existem várias opções para mapear os conceitos generalização/especialização em relações.
- 68** O processo de normalização desconsidera a dependência transitiva entre os atributos não chave.

Um sistema gerenciador de banco de dados (SGBD) é uma coleção de programas que possibilita que os usuários criem e mantenham um banco de dados. Acerca desse assunto, julgue os itens que se seguem.

- 69** No MySQL 5.4, as tabelas do tipo InnoDB permitem transações, compatíveis com as propriedades ACID, com *commit*, *rollback* e recuperação em caso de falhas.
- 70** No arquivo `postgres.conf`, é registrada a autorização de acesso de outros computadores, em uma rede local, ao servidor SGBD PostgreSQL 8.0.
- 71** No PostgreSQL 8.4.0, a consistência dos dados é mantida por meio de um modelo multiversão, tal que as transações de consulta ao um banco de dados acessam dados congelados (*snapshot*) em um instante do tempo (uma versão de banco de dados), independentemente do estado atual dos dados no banco de dados. Esse modelo garante a consistência dos dados, quando transações de atualizações ocorrem de forma concorrente.

Estruturas de dados retratam as relações lógicas existentes entre os dados, de modo análogo ao uso de um modelo matemático para espelhar alguns aspectos de uma realidade física. Essas estruturas são armazenadas na memória do computador e manipuladas por meio de operações que as transformem e devem preservar as relações lógicas existentes entre os dados e permitir que as operações sejam descritas por procedimentos simples e eficientes.

Paulo Veloso *et al.* *Estruturas de dados*. Rio de Janeiro: Campus, 1985, p. 12 (com adaptações).

Tendo o texto acima como referência inicial, julgue os itens a seguir.

- 72** Uma lista linear é um conjunto de $n \geq 0$ nós x_1, x_2, \dots, x_n , organizados estruturalmente de forma a refletir as suas posições relativas, tal que, se $n > 0$, então x_1 é o primeiro nó; para $1 < k < n$, o nó x_k é precedido pelo nó x_{k-1} e é seguido do x_{k+1} ; e x_n é o último nó. Quando $n = 1$, a lista é vazia.
- 73** A classe `Lista` a seguir, escrita na linguagem Java, implementa uma estrutura e operações de uma fila.

```
public class Lista {
    private static class Celula {
        Object item;
        Celula prox;
    }
    private Celula atual;
    private int tam;
    public Lista () {
        atual = null;
    }
    public void armazena (Object x) {
        Celula aux = atual;
        atual = new Celula();
        atual.item = x;
        atual.prox = aux;
        tam ++;
    }
    public Object retira () {
        if(vazia())
            return null;
        Object item = atual.item;
        atual = atual.prox;
        tam --;
        return item;
    }
    public boolean vazia() {
        return (atual == null);
    }
    public int tamanho () {
        return tam;
    }
}
```

Um arquivo é organizado logicamente como uma sequência de registros, que são mapeados em blocos de discos no sistema operacional. O tamanho do registro pode variar, embora os blocos tenham tamanho fixo determinado pelas propriedades físicas do disco e pelo sistema operacional. Em um banco de dados relacional, as tuplas de relações distintas têm geralmente tamanhos diferentes. A respeito desse assunto, julgue os próximos itens.

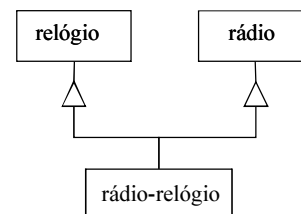
- 74** Uma maneira simples de implementar registro de tamanho variável é anexar um símbolo especial de fim de registro no final de cada registro. Porém, essa estratégia apresenta algumas desvantagens, como: não é fácil reutilizar o espaço ocupado por um registro apagado anteriormente; e, em geral, não há espaço para que os registros possam crescer.
- 75** Uma abordagem para o mapeamento de banco de dados em arquivos é utilizar diversos arquivos e armazenar em determinado arquivo apenas registros de tamanho fixo. Esses arquivos são mais fáceis de implementar que os arquivos de registros de tamanho variáveis.

Acerca de arquitetura cliente servidor multicamadas, julgue os itens subsequentes.

- 76** A arquitetura cliente servidor multicamadas prevê a implementação de camadas intermediárias entre as camadas de aplicação e de dados.
- 77** Na arquitetura cliente servidor multicamadas, os clientes são chamados de *thin clients*, pois a maior parte do processamento da lógica de aplicação reside no cliente.

Com relação a conceitos e aplicações do paradigma orientado a objeto, julgue os itens que se seguem.

- 78** A abstração consiste no isolamento das características dos objetos que se deseja representar no contexto de uma aplicação.
- 79** Por meio do encapsulamento, disponibiliza-se um objeto e suas funcionalidades, de tal modo que os objetos não necessitam expor sua construção interna.
- 80** O mecanismo de herança permite o reúso de código por classes derivadas. Tais classes podem acrescentar novas características, no entanto não pode estender as funcionalidades das classes originais.
- 81** A figura a seguir ilustra o conceito de herança múltipla, na qual uma classe pode herdar características e funcionalidades de mais de uma classe base.



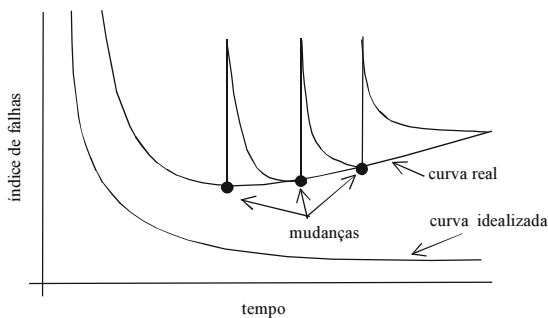
- 82** O polimorfismo não evita a escrita de um comando condicional explícito quando o comportamento dos objetos varia de acordo com seus tipos.
- 83** A sobrecarga permite a existência de vários métodos com nome diferentes, desde que tenham a mesma assinatura.
- 84** Classes possuem métodos que são funções membro associadas que realizam ações sobre os valores dos atributos, não modificando o estado dos objetos.

Muitas organizações desenvolvem *software* sem utilizar nenhum processo de desenvolvimento, porque os processos tradicionais não são adequados às suas realidades. As metodologias tradicionais, denominadas orientadas a documentação diferem bastante das atuais.

Koscianski e Soares. *Qualidade de software: aprenda as metodologias e técnicas mais modernas para o desenvolvimento de software*. 2.ª ed. São Paulo: Novatec, 2007, p. 191 (com adaptações).

No que se refere a modelos de processo de desenvolvimento de *software*, julgue os itens a seguir.

- 85 O modelo em cascata sugere uma abordagem sistemática e sequencial para o desenvolvimento de *software*. Sua natureza linear leva a estados de bloqueio nos quais, para que nova etapa seja iniciada, é necessário que a documentação associada à fase anterior tenha sido aprovada.
- 86 O modelo espiral admite retorno às fases anteriores de desenvolvimento, suportando ainda a execução paralela de fases.
- 87 O modelo incremental entrega uma série de versões por meio da aplicação de sequências paralelas, possibilitando entrega mais rápida do produto final.
- 88 O uso de uma abordagem de construção embasada em componentes faz que o desenvolvimento no modelo RAD (*rapid application development*) seja considerado mais rápido.
- 89 A prototipagem é um exemplo de modelo evolucionário, no qual a cada iteração se apresenta um produto totalmente operacional.
- 90 O modelo evolucionário em espiral combina a prototipagem com os aspectos controlados e sistemáticos do modelo em cascata.

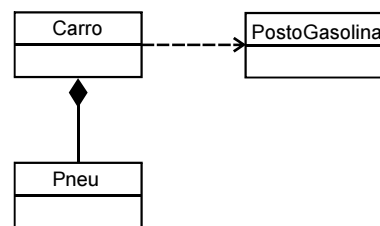


Considerando a figura acima, julgue os itens subsequentes a respeito de conceitos de desenvolvimento e de manutenção de sistemas e aplicações.

- 91 A amplitude dos dentes e a inclinação da curva real da figura mostrada representa o processo de manutenção de *software*. Tal fato ocorre porque, quando as manutenções são feitas, a probabilidade de que novos defeitos sejam introduzidos aumenta, gerando picos na curva do índice de falhas.
- 92 A métrica pontos de função tem como finalidade aferir o tamanho dos projetos de desenvolvimento e a manutenção de *software*.
- 93 A técnica de refatoração é uma manutenção de *software*, uma vez que altera o comportamento observável da estrutura interna do *software*.

Acerca de paradigma orientado a objeto e de desenvolvimento e manutenção de projeto, julgue os seguintes itens.

- 94 No contexto de engenharia de *software* orientada a objetos, componentes contêm um conjunto de classes colaborativas. No entanto, existem casos em que um componente pode conter uma única classe.
- 95 Durante o projeto de um componente, as classes que o compõem são elaboradas para incluir todos os atributos e todas as operações relevantes à sua implementação. As interfaces que habilitam a comunicação e a colaboração interclasses não necessitam ser definidas no projeto do componente.
- 96 A coesão implica que um componente ou classe encapsule somente atributos e operações fortemente relacionadas entre si.
- 97 Em projetos orientados a objeto, o acoplamento não pode ser utilizado como métrica, embora seja uma característica mensurável, em que as conexões físicas entre os elementos do projeto representam o acoplamento dentro de um sistema.



Considerando a figura acima, julgue os itens que se seguem, a respeito do paradigma orientado a objeto.

- 98 A associação entre as classes Pneu e Carro representa uma agregação, demonstrando que um carro tem pneus como partes.
- 99 A associação entre as classes Carro e PostoGasolina representa o fato de que Carro implementa a interface PostoGasolina. Essa associação é também conhecida como dependência.
- 100 A associação entre as classes Pneu e Carro poderia ser substituída pela associação \diamond —, sem que com isso a semântica do modelo fosse alterada.

Julgue os itens de 101 a 107, a respeito de tecnologias *web*.

- 101 *Webservices* utilizam três elementos básicos: SOAP, uma linguagem dependente de plataforma e com base em XML; WSDL, utilizado para descrever e localizar *Webservices*; e UDDI, um diretório para armazenar informações a respeito de *Webservices*, como interfaces descritas em SOAP, oferecendo comunicação via WSDL.
- 102 Ajax é um grupo de tecnologias que consiste de uma combinação, entre outros, de um método para troca de dados assíncrona entre navegador e servidor, evitando recarga de páginas. O objeto XMLHttpRequest é comumente utilizado, mas pode-se usar um objeto IFrame ou adicionar dinamicamente a tag `<script>`.
- 103 O trecho de documento XML a seguir está sintaticamente correto.

```
<?xml version="1.0"?>
<recado id=523>
<para dia =29>Maria</para>
<de>Joana</de>
</recado>
```
- 104 O texto dentro de uma seção PCDATA é ignorado pelo parser.

105 DHTML é um termo usado para descrever as tecnologias para criar páginas *web* dinâmicas e interativas como, por exemplo, JavaScript, CSS (*cascading style sheets*) e ASP.

106 Existem três formas de incluir uma folha de estilo CSS: em uma folha de estilo externa, em que cada página *web* deve possuir uma declaração do arquivo CSS na *tag* `<head>`; em uma folha de estilo interna, em que os estilos são definidos diretamente no cabeçalho, na *tag* `<head>`; diretamente em *tags* que estejam no corpo (`<body>`) da página.

107 A propriedade `border-width:20px 10px 40px 1px` define uma borda com o seguinte estilo: superior = 20 pixels; inferior = 10 pixels; esquerda = 40 pixels; direita = 1 pixel.

A respeito de linguagens de programação *web*, julgue os itens subsequentes.

108 Em PHP, uma variável tem de ser declarada previamente, com um tipo determinado, antes que qualquer valor seja adicionado a ela.

109 O trecho de código PHP a seguir define que um arquivo será lido linha a linha.

```
<?php
$arq=fopen("benvindo.txt","r") or exit("Nao
foi possivel abrir o arquivo!");
while (!feof($arq))
{
    echo fgetc($arq);
}
fclose($arq);
?>
```

110 JavaScript, ao contrário de HTML, faz distinção entre maiúsculas e minúsculas.

111 Em JavaScript, *scripts* na seção de cabeçalho (`<head>`) são carregados antes de qualquer uso e serão executados quando forem chamados ou quando um evento for disparado.

112 Em HTML, quando o atributo `name` é utilizado em um elemento `a`, uma âncora nomeada é definida dentro do documento HTML.

113 Documentos HTML consistem em elementos HTML, que podem ter atributos. Estes elementos não podem ter conteúdo vazio.

114 A execução do trecho HTML a seguir, em um navegador, resultará na abertura da página `outra.html` em uma nova janela do navegador.

```
<html>
<body>
<a href="outra.html"
target="_blank">Pagina</a>
</body>
</html>
```

Julgue os itens seguintes, a respeito de interface gráfica, ergonomia e usabilidade.

115 A usabilidade de sistemas pode ser definida como as características que permitem que o usuário alcance seus objetivos e satisfaça suas necessidades dentro de um contexto de utilização determinado. Desempenho e satisfação do usuário são especificados e medidos a partir do grau de realização de objetivos perseguidos na interação (eficiência) pelos recursos alocados para alcançar esses objetivos (eficácia) e pelo grau de aceitação do produto pelo usuário (satisfação).

116 Usabilidade pode ser dividida nas seguintes metas: ser eficaz no uso (eficácia); ser eficiente no uso (eficiência); ser segura no uso (segurança); ser de boa utilidade (utilidade); ser fácil de aprender (*learnability*); e ser fácil de lembrar como se usa (*memorability*).

117 Entre os critérios usados para identificar e classificar corretamente as qualidades e problemas ergonômicos de interfaces, estão a compatibilidade, a homogeneidade, a concisão, a flexibilidade, o *feedback*, a carga informacional, o controle explícito e a gestão de erros.

118 O teste de usabilidade é uma técnica formal que pode envolver usuários designados para desenvolver tarefas típicas e críticas. Há coleta de dados e posterior análise. O teste de usabilidade caracteriza-se por utilizar diferentes técnicas, que são voltadas, em sua maioria, para a avaliação da ergonomia dos sistemas interativos. Entre essas técnicas, estão a inspeção baseada em padrões, guias de estilos ou guias de recomendações; e a avaliação heurística.

119 Um problema de usabilidade é identificado quando um aspecto da interface está em desacordo com as características dos usuários e da maneira pela qual ele realiza sua tarefa. Um problema de ergonomia é observado em determinadas circunstâncias, quando uma característica do sistema interativo ocasiona a perda de tempo, compromete a qualidade da tarefa ou inviabiliza sua realização.

120 Existem diversos estilos de interação do usuário com interfaces, entre os quais, manipulação direta; seleção de *menu*; preenchimento de formulários; linguagens de comandos; e linguagem natural.

