

# CONCURSO PÚBLICO



Maio - 2006

## Tecnologia da Informação

### Leia estas instruções:

<b>1</b>	Escreva seu Número de Inscrição neste retângulo:	
<b>2</b>	Este Caderno contém 40 questões de múltipla escolha.	
<b>3</b>	Quando o Fiscal autorizar, confira se este Caderno está completo e se não apresenta imperfeição gráfica que dificulte a leitura. Se você verificar algum problema, peça imediatamente ao Fiscal que substitua o Caderno.	
<b>4</b>	Cada questão apresenta quatro opções de resposta, das quais só uma é correta.	
<b>5</b>	Interpretar as questões faz parte da avaliação; portanto, não adianta pedir esclarecimentos aos Fiscais.	
<b>6</b>	Utilize qualquer espaço deste Caderno para rascunhos e não destaque nenhuma folha.	
<b>7</b>	Você dispõe de, no máximo, quatro horas para responder às questões e preencher a Folha de Respostas.	
<b>8</b>	O preenchimento da Folha de Respostas é de sua inteira responsabilidade.	
<b>9</b>	Antes de retirar-se definitivamente da sala, devolva ao Fiscal a Folha de Respostas e este Caderno.	





- 06.** Uma das funções desempenhadas pela camada de enlace dos dados no TCP/IP é
- A)** realizar controle de fluxo para permitir que receptores lentos sejam inundados de quadros por emissores rápidos.
  - B)** organizar os dados recebidos da camada de rede em quadros a serem transmitidos na rede física e vice-versa.
  - C)** realizar o endereçamento de aplicações (processos) no emissor e no receptor.
  - D)** organizar armazenamento temporário de segmentos no emissor e no receptor.
- 07.** A função dos protocolos de roteamento é construir as tabelas de roteamento completas nos diversos roteadores de uma rede, por meio da troca de mensagens entre eles. Há dois tipos de protocolos: IGP (*Interior Gateway Protocol*) e EGP (*Exterior Gateway Protocol*). Pertence ao tipo EGP:
- A)** BGP (*Border Gateway Protocol*)
  - B)** RIP (*Routing Information Protocol*)
  - C)** IGRP (*Interior Gateway Routing Protocol*)
  - D)** OSPF (*Open Shortest Path First*)
- 08.** Em relação aos protocolos de transporte, pode-se afirmar:
- A)** O protocolo UDP é considerado confiável, embora seja simples.
  - B)** O protocolo UDP é orientado a datagrama, não orientado a conexão, e não executa controle de fluxo nem controle de erro nem de seqüenciamento.
  - C)** O protocolo TCP é considerado não-confiável.
  - D)** O protocolo TCP é considerado confiável, porém não efetua controle de erro nem de seqüenciamento.
- 09.** De acordo com a arquitetura TCP/IP, de endereçamento universal, é correto afirmar que um endereço IP
- A)** compõe-se de uma quadra de números naturais na faixa de 1 (um) a 255.
  - B)** compõe-se de uma trinca de números naturais na faixa de 0 (zero) a 255.
  - C)** garante a unicidade de uma rede na Internet.
  - D)** garante a unicidade de um computador na Internet.
- 10.** Em relação ao endereço de *LoopBack*, é correto afirmar:
- A)** Quando o computador está sem conexão de rede, esse endereço é automaticamente selecionado.
  - B)** Quando uma aplicação usa esse endereço como destino, o *software* do protocolo TCP/IP devolve os dados com geração de tráfego na rede.
  - C)** É uma forma simples de fazer com que um cliente local fale com o servidor local correspondente, porém o programa cliente e/ou o programa servidor devem ser alterados.
  - D)** É usado para testes do TCP/IP e para comunicação interprocessos em uma máquina local.

11. O modelo TCP/IP difere do modelo OSI, porque tem duas camadas que se formam a partir da fusão de algumas camadas. As duas camadas são:
- A) transporte e rede
  - B) aplicação e transporte
  - C) aplicação e rede
  - D) transporte e aplicação
12. A camada de aplicação é formada pelos protocolos utilizados pelas diversas aplicações do modelo TCP/IP. Esses protocolos são:
- A) FTP – UDP – TCP – DNS
  - B) UDP – SMTP – POP3 – DNS
  - C) FTP – SMTP – TCP – DNS
  - D) FTP – SMTP – POP3 – DNS
13. Em relação à configuração de serviços DHCP (*Dynamic Host Configuration Protocol*), é correto afirmar:
- A) Um servidor DHCP pode implementar políticas de alocação de endereços e opções DHCP somente de forma dinâmica.
  - B) Um servidor DHCP, respondendo a uma solicitação de parâmetros de um cliente, oferece uma opção de endereço IP, dentre as que estiverem disponíveis, para o solicitante. Porém, isso não caracteriza *leasing* ou arrendamento.
  - C) A solicitação, ao servidor DHCP, de re-arrendamento dos parâmetros obtidos será facultada ao servidor.
  - D) A atribuição dinâmica de endereços IP é feita por meio do protocolo DHCP. O uso e a configuração desse protocolo, tanto da parte do cliente como da do servidor, são simples.
14. Em relação à configuração de serviços, é correto afirmar:
- A) NIS é um serviço, desenvolvido pela *Sun Microsystems*, para distribuição de informações por uma rede. Ele torna a rede transparente para o usuário, não importando onde o usuário esteja trabalhando.
  - B) SMTP é um protocolo para enviar mensagens de *E-mail* entre usuários. A maioria dos sistemas de *E-mail* utiliza o SMTP, pertencente à camada de transporte.
  - C) O serviço FTP é muito utilizado em uma rede para transferência de arquivos entre máquinas servidoras e clientes. A grande vantagem desse serviço é a segurança.
  - D) O serviço HTTP utiliza o protocolo UDP para estabelecer uma conexão com um cliente (geralmente via *browser*). O servidor examina a requisição HTTP e responde ao cliente com um documento HTML, que pode conter figura, som, VRML ou outras linguagens, como, por exemplo, Java.

15. Em relação às ferramentas IDS (Sistemas de Detecção de Intrusos), pode-se afirmar:

- A) As ferramentas IDS, baseadas em rede, fazem suas monitorações nos cabeçalhos dos pacotes e também no campo de dados destes, possibilitando ataques no nível de aplicação (para pacotes TCP e UDP).
- B) As ferramentas IDS utilizam criptografia para tornar a monitoração mais efetiva em cabeçalhos e campo de dados dos pacotes.
- C) Os ataques que ocorrem no nível de aplicação podem ser usados tanto para uma invasão como para indisponibilização do serviço (DoS). Dessa forma, os sistemas de detecção de intrusos terão como registrar o ataque.
- D) A implementação de IDSs em redes baseadas em *switching*, também conhecidas como redes comutadas, não obteve sucesso devido à grande dificuldade de implantação.

16. Considere os seguintes trechos de pseudo-código estruturado:

```
I.  var
    inteiros A, B;
    ler(A);
    ler(B);
    se A ≥ B então
        enquanto A ≥ B fazer
            imprimir(A);
            A := A - 1;
        fim-enquanto;
    senão
        enquanto B > A fazer
            imprimir(B);
            B := B - 1;
        fim-enquanto;
    fim-se;
```

```
III. var
    inteiros A, B;
    ler(A);
    ler(B);
    se A > B então
        enquanto A > B fazer
            imprimir(A);
            A := A - 1;
        fim-enquanto;
    senão
        enquanto B ≥ A fazer
            imprimir(B);
            B := B - 1;
        fim-enquanto;
    fim-se;
```

```
II.  var
    inteiros A, B;
    ler(A);
    ler(B);
    se A ≥ B então
        repetir
            imprimir(A);
            A := A - 1;
        até A < B;
    senão
        repetir
            imprimir(B);
            B := B - 1;
        até B ≤ A;
    fim-se;
```

```
IV.  var
    inteiros A, B;
    ler(A);
    ler(B);
    se A > B então
        repetir
            imprimir(A);
            A := A - 1;
        até A ≤ B;
    senão
        repetir
            imprimir(B);
            B := B - 1;
        até B < A;
    fim-se;
```

Dois trechos de pseudo-código estruturado são considerados equivalentes entre si quando, para os mesmos dados de entrada, eles produzem as mesmas saídas.

Em relação aos trechos I, II, III e IV acima, pode-se afirmar que

- A) I e III são equivalentes entre si e II e IV são equivalentes entre si.
- B) I e II são equivalentes entre si e III e IV são equivalentes entre si.
- C) I e IV são equivalentes entre si e II e III são equivalentes entre si.
- D) I, II, III e IV são equivalentes entre si.

17. As afirmativas a seguir referem-se à programação estruturada:

I	O uso do comando <i>GOTO</i> é permitido de forma irrestrita nesse tipo de programação.
II	Os programas são organizados em blocos que possuem um ponto de entrada e um ponto de saída.
III	O uso de comandos de atribuição só pode ser feito dentro de estruturas de repetição.

Assinale a opção correta em relação a essas afirmativas.

- A) Apenas a I e a III são verdadeiras.                      C) Apenas a II é verdadeira.  
B) Apenas a III é verdadeira.                                D) Apenas a I e a II são verdadeiras.

18. Na programação orientada-a-objetos, a noção de polimorfismo se refere

- A) à possibilidade de definição de várias formas de dados dentro de uma mesma classe.  
B) à possibilidade de definição do mesmo método para objetos de classes diferentes.  
C) à natureza não estruturada de certos programas.  
D) à utilização de métodos de nomes diferentes dentro de uma mesma classe.

19. Na programação orientada-a-objetos, a noção de herança simples se refere

- A) à redefinição de métodos dentro de uma subclasse, com o mesmo nome de algum método de suas superclasses.  
B) à definição aninhada de métodos (métodos definidos dentro de métodos) dentro de uma mesma classe.  
C) ao uso de operações e atributos de uma classe por parte das suas subclasses.  
D) ao uso de tipos dinâmicos para as variáveis de um programa.

20. Na programação orientada-a-objetos, a noção de encapsulamento diz respeito a:

- A) definição de dados e operações dentro de uma mesma unidade sintática.  
B) possibilidade de definir variáveis locais dos métodos.  
C) tratamento de exceções no corpo de cada método ou classe.  
D) manipulação de referências (apontadores) a objetos de maneira restrita.

**As questões 21 e 22 usam os seguintes procedimentos e funções, referentes às operações em estruturas de dados pilha:**

- *criaPilha(P)* – procedimento que cria uma nova pilha de inteiros, vazia, associando-a ao identificador *P*.
- *empilha(V, P)* – procedimento que empilha na pilha *P* o valor inteiro resultante de avaliar a expressão *V*.
- *desempilha(P)* – procedimento que desempilha o valor inteiro que aparece no topo da pilha *P*. Esse valor é descartado.
- *topo(P)* – função que retorna o valor que aparece no topo da pilha *P*. O valor não é desempilhado.
- *vazia?(P)* – função que devolve *verdadeiro*, caso a pilha *P* esteja vazia, e *falso*, caso contrário.

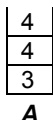
Ao analisar as opções de resposta, considere que o valor numérico inferior representa a base da pilha, enquanto o valor na posição superior representa o topo. Cada pilha será representada com o seu nome logo abaixo.

21. Seja a seqüência de operações:

```
criaPilha(A);  
empilha(3, A);  
empilha(3, A);  
desempilha(A);  
empilha(4, A);  
empilha(topo(A), A);
```

A pilha A, resultante da execução da seqüência de operações acima, é:

A)



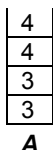
B)



C)



D)



22. Seja o seguinte algoritmo estruturado:

```
criaPilha(A);  
criaPilha(B);  
se vazia?(A) então  
    criaPilha(C);  
fim-se;  
empilha(30, A);  
empilha(31, B);  
se vazia?(C) então  
    empilha(topo(A) + topo(B), C);  
fim-se;  
empilha(20, A);  
empilha(21, B);  
enquanto vazia?(B) = verdadeiro fazer  
    desempilha(B);  
fim-enquanto;  
enquanto vazia?(A) = falso fazer  
    desempilha(A);  
fim-enquanto;  
empilha(21, A);
```

A configuração das pilhas A, B e C, resultante da execução desse algoritmo, é:



24. Seja a seqüência de operações:

```

criaFila(A);
insere(3, A);
insere(3, A);
retira(A);
insere(4, A);
insere(próximo(A), A);

```

A fila A, resultante da execução da seqüência de operações acima, é:

- A) A: → 

4	4	3	3
---	---	---	---

 →
- B) A: → 

4	4	3
---	---	---

 →
- C) A: → 

3	4	4
---	---	---

 →
- D) A: → 

3	4	3
---	---	---

 →

25. Seja o seguinte algoritmo estruturado:

```

criaFila(A);
criaFila(B);
se vazia?(A) então
    criaFila(C);
fim-se;
insere(30, A);
insere(31, B);
se vazia?(C) então
    insere(próximo(A) + próximo(B), C);
fim-se;
se próximo(A) >= 30 então insere(20, A);
    senão insere(20, B);
fim-se;
insere(21, B);
enquanto vazia?(B) = falso fazer
    retira(B);
fim-enquanto;
enquanto vazia?(A) = verdadeiro fazer
    retira(A);
fim-enquanto;
insere(21, A);

```

A configuração das filas A, B e C, resultante da execução do algoritmo acima, é:

- A) A: → 

		21
--	--	----

 →
- B: → 

21	20	31
----	----	----

 →
- C: → 

		61
--	--	----

 →
- B) A: → 

21	20	30
----	----	----

 →
- B: → 

--	--	--

 →
- C: → 

		61
--	--	----

 →

C)

A: → 

20	30	21
----	----	----

 →

B: → 

		31
--	--	----

 →

C: → 

		52
--	--	----

 →

D)

A: → 

		21
--	--	----

 →

B: → 

21	20	31
----	----	----

 →

C: → 

		52
--	--	----

 →

26. Considere as seguintes afirmativas, referentes às estruturas de dados árvore:

I	Todo nodo da árvore tem um nodo pai.
II	Os nodos sem filhos são chamados “folhas” da árvore.
III	Dados dois nodos internos quaisquer da árvore, a quantidade de filhos de cada um deles deve ser a mesma.

Assinale a opção correta em relação às afirmativas acima:

A) Somente a I e a II são verdadeiras.

B) Somente a II é verdadeira.

C) Somente a I e a III são verdadeiras.

D) Somente a III é verdadeira.

27. Considere os seguintes algoritmos, que definem funções para se percorrer uma árvore binária. Nesses algoritmos, são usadas operações de seleção dos componentes de uma árvore binária. Assim, a operação “filho\_esquerdo” devolve a subárvore esquerda do argumento (essa subárvore pode ser vazia); a operação “filho\_direito” devolve a subárvore direita do argumento (essa subárvore pode ser vazia); a operação “informação\_da\_raiz” devolve a informação contida no nodo raiz da árvore. Suponha que a operação “processar” realiza o processamento da informação contida no seu parâmetro.

(i) função percorrer1( T : árvoreBinária)

Início

T1 := filho\_esquerdo(T);

T2 := filho\_direito(T);

Info := Informação\_da\_raiz(T);

processar(Info);

se T1 não é vazio então

percorrer1(T1);

fim-se;

se T2 não é vazio então

percorrer1(T2);

fim-se;

fim.

(ii) função percorrer2( T : árvoreBinária)

Início

T1 := filho\_esquerdo(T);

T2 := filho\_direito(T);

```

Info := Informação_da_raiz(T);

se T1 não é vazio então
    percorrer2(T1);
fim-se;
se T2 não é vazio então
    percorrer2(T2);
fim-se;
processar(Info);
fim.

```

(iii) função percorrer3( T : árvoreBinária)  
Início

```

T1 := filho_esquerdo(T);
T2 := filho_direito(T);
Info := Informação_da_raiz(T);

se T1 não é vazio então
    percorrer3(T1);
fim-se;
processar(Info);
se T2 não é vazio então
    percorrer3(T2);
fim-se;
fim.

```

Os algoritmos (i), (ii) e (iii) acima são chamados, **respectivamente**, percursos

- A) pré-ordem, em ordem, pós-ordem.
- B) pós-ordem, pré-ordem, em ordem.
- C) em ordem, pré-ordem, pós-ordem.
- D) pré-ordem, pós-ordem, em ordem.

28. Acerca do termo “memória virtual”, é correto afirmar:

- A) Permite a utilização de um espaço de memória cujo tamanho não é necessariamente limitado ao tamanho da memória principal de um computador.
- B) Permite que seja movido um bloco da memória principal para a secundária por demanda.
- C) Trabalha sempre com paginação de blocos de memória, pois é a única forma de que dispõe para controlar a transferência deles.
- D) Aloca os blocos de memória sempre na mesma posição.

29. Considere o seguinte algoritmo (pseudo-código estruturado):

```

função Mistério(N : inteiro) retorna inteiro;
início
    se N ≤ 1
        então retorna 1;
        senão retorna (2 + Mistério(N – 2));
    fim-se;
fim;

```

A chamada à função Mistério (10) retorna o número inteiro:

- A) 12
- B) 10
- C) 9
- D) 11

30. Considere as seguintes afirmativas, em relação a algoritmos e programação estruturada:

I	Algoritmo pode se definido como “um procedimento computacional bem determinado que aceita um ou mais valores como entrada e produz um ou mais valores como saída”.
II	Algoritmos só podem ser descritos usando-se estruturas de controle oriundas da programação estruturada, como atribuição, iteração e seleção.
III	Um algoritmo é uma definição da solução de um problema computacional. Ele é de nível mais alto que o de um programa, já que não se preocupa com aspectos de implementação, como, por exemplo, a representação dos dados na memória do computador.

Assinale a opção correta em relação às afirmativas acima:

- A) Somente a I e a III são verdadeiras.
- B) Somente a II é verdadeira.
- C) Somente a III é verdadeira.
- D) Somente a I e a II são verdadeiras.

31. Considere os seguintes trechos de algoritmos:

- I.  $X \leftarrow 3 * Y + 6;$
- II. enquanto  $j \geq k$  fazer:  
    total  $\leftarrow$  total + j;  
    j  $\leftarrow$  j + 1;  
fim-enquanto
- III. se vendas < 1000 então  
    imprimir( “movimento fraco” );  
senão  
    imprimir( “movimento bom” );  
fim-se;

As estruturas de controle principais dos trechos de algoritmos I, II e III são, **respectivamente**,

- A) soma, comparação e impressão.
- B) atribuição, iteração e condicional.
- C) associação, interação e verificação de condições.
- D) operacionalização aritmética, totalização cumulativa e impressão condicional.

32. No ANSI SQL, as funções agregadas (AVG, MIN, MAX, COUNT e SUM) **não** podem ser usadas com a seguinte cláusula:

- A) DISTINCT
- B) SELECT
- C) FROM
- D) WHERE

33. Analise o seguinte código SQL:

```
SELECT VENDEDOR, CLIENTE, VALOR, DATA, NOTAFISCAL FROM VENDA  
ORDER BY 2, 3, 5
```

Em relação ao código, é correto afirmar:

- A) A tabela Venda será ordenada pelos campos Vendedor, Valor e NotaFiscal.
- B) A tabela Venda será ordenada pelos campos Cliente, Valor e NotaFiscal.
- C) A cláusula ORDER BY não permite a utilização de números.
- D) A cláusula ORDER BY só permite o uso de um campo identificado através de numeração.

34. Considere o código SQL abaixo:

```
INSERT INTO EX_ALUNO (MATRICULA,DISCIPLINA,MEDIA)  
SELECT MATRICULA,DISCIPLINA,(NOTA1 + NOTA2 + NOTA3) / 3 FROM CURSA  
WHERE NOTA1 IS NOT NULL  
AND NOTA2 IS NOT NULL  
AND NOTA3 IS NOT NULL
```

Em relação a esse código, é correto afirmar:

- A) Será incluído na tabela EX\_ALUNO todo aluno que tiver as três notas lançadas na tabela CURSA.
- B) Será incluído na tabela CURSA todo aluno que tiver as três notas lançadas na tabela EX\_ALUNO.
- C) Serão incluídas na tabela EX\_ALUNO as informações referentes aos campos NOTA1, NOTA2 e NOTA3.
- D) Será incluído na tabela EX\_ALUNO somente a MEDIA dos campos NOTA1, NOTA2 e NOTA3.

35. De acordo com os conceitos de *constraint* (restrições), a diferença básica entre uma *constraint* UNIQUE e uma *constraint* PRIMARY KEY é:

- A) PRIMARY KEY somente pode ser definida para campos numéricos, e UNIQUE não.
- B) UNIQUE cria um índice, e PRIMARY KEY não.
- C) PRIMARY KEY pode ser referenciada por uma *constraint* FOREIGN KEY, e UNIQUE não.
- D) UNIQUE permite valores NULL, e PRIMARY KEY não.

36. O seguinte trecho de programa Java foi tirado de Thinking in Java, 3rd ed. Revision 4.0 (<http://www.mindview.net/Books/TIJ/>):

```
// HelloDate.java  
import java.util.*;  
  
public class HelloDate {  
    public static void main(String[] args) {  
        System.out.println("Hello, it's: ");  
        System.out.println(new Date());  
    }  
}
```

A execução desse programa resulta na impressão de uma mensagem no dispositivo de saída padrão. Essa mensagem contém os dizeres "Hello, it's: ", seguidos da data e da hora nas quais o programa foi

- A) escrito. C) executado.  
B) compilado pela última vez. D) armazenado na memória do computador.

37. Considere as seguintes afirmativas, referentes às operações com valores de dados na linguagem de programação Java:

I. A seqüência de comandos

```
int x = 0, y = 2, z = 5;  
String sString = "x + y + z ";  
System.out.println(sString + x + y + z);
```

imprime o valor 7 na saída padrão.

II. O trecho de código

```
if (false < true)  
    System.out.println("false < true");  
else  
    System.out.println("true < false");
```

imprime sempre a mensagem "false < true", pois o valor-verdade `false` é definido pela linguagem como sendo sempre menor que o valor-verdade `true`.

III. Supondo-se que a variável `i` tenha sido declarada como do tipo inteiro, a avaliação da expressão

```
i < 0 ? -i : i
```

devolverá o valor absoluto do número armazenado na variável `i` (o valor absoluto de um número é aquele que não considera o sinal do número, sendo sempre positivo).

Assinale a opção correta, em relação a essas afirmativas:

- A) Apenas a III é verdadeira. C) Apenas a I e a II são verdadeiras.  
B) Apenas a II é verdadeira. D) Apenas a I e a III são verdadeiras.

38. Considere as seguintes afirmativas, referentes aos comandos da linguagem de programação Java:

I. O comando `continue`, quando usado dentro de um comando de iteração, interrompe a execução da iteração da seguinte forma: o controle é transferido para o início do comando de iteração e uma nova iteração é iniciada.

II. Os comandos

```
while(true)
```

e

```
for(;;)
```

são equivalentes.

III. O comando `switch` (que implementa a seleção entre múltiplas alternativas a partir do resultado da avaliação de uma expressão) pode ser usado com valores seletores de qualquer tipo primitivo da linguagem.

IV. `goto` é uma palavra reservada da linguagem Java.

Assinale a opção correta em relação a essas afirmativas.

- A) Somente a I, a II e a III são verdadeiras.
- B) Somente a I, a II e a IV são verdadeiras.
- C) Somente a II e a III são verdadeiras.
- D) Somente a II e a IV são verdadeiras.

39. Considere as seguintes afirmativas, referentes à linguagem de programação Delphi:

I	Foi construída pela adição de primitivas de orientação a objetos (entre outras) à linguagem de programação imperativa Pascal.
II	Permite a construção de subprogramas genéricos e classes parametrizadas.
III	Seu projeto original inclui a interface gráfica com o usuário (GUI).

Assinale a opção correta em relação às afirmativas acima:

- A) Só a II é verdadeira.
- B) Só a I e a II são verdadeiras.
- C) Só a I e a III são verdadeiras.
- D) Só a III é verdadeira.

40. Sobre o Diagrama de Atividades da UML (*Unified Modeling Language*), é correto afirmar:

- A) Mostra um comportamento que especifica as seqüências de estados pelos quais um objeto passa durante seu tempo de vida em resposta a eventos.
- B) Fornece a visão externa do sistema e suas interações com o mundo exterior, representando uma visão de alto nível de funcionalidade.
- C) É o diagrama mais importante na modelagem orientada-a-objetos e representa a essência da linguagem.
- D) É um caso especial de diagramas de estado, em que todos os estados têm uma ação interna e nenhuma transição tem um evento de entrada.

