



CENTRO DE SOLUÇÕES EM GOVERNO ELETRÔNICO

CONCURSO PÚBLICO PARA PROVIMENTOS DE VAGAS EM CARGOS DE NÍVEL SUPERIOR E MÉDIO

TÉCNICO DE COMPUTAÇÃO - TC PROGRAMADOR JAVA

INSTRUÇÕES

Leia atentamente e cumpra rigorosamente as instruções que seguem, pois elas são parte integrante das provas e das normas que regem este Concurso Público.

1. Verifique se o cargo constante na capa deste caderno é aquele para o qual realizou a inscrição.
2. Cada questão oferece 5 (cinco) alternativas de respostas, representadas pelas letras **A, B, C, D** e **E**, sendo apenas 1 (uma) a resposta correta.
3. O tempo para a realização da prova é de 4 horas, incluindo o preenchimento da grade de respostas. O candidato só poderá retirar-se do recinto da prova teórico-objetiva após transcorrida 1 hora e 30 minutos de seu início. Os dois últimos candidatos deverão retirar-se da sala de prova ao mesmo tempo, devendo assinar a Ata de Prova.
4. Nenhuma informação sobre as instruções e/ou sobre o conteúdo das questões será dada pelo fiscal, pois são parte integrante da prova.
5. No caderno de prova, o candidato poderá rabiscar, riscar, calcular, etc.
6. Os gabaritos preliminares da prova objetiva serão divulgados no dia 20/11/2012, até às 23h59min, nos sites www.fundatec.org.br e www.procergs.rs.gov.br.
7. Certifique-se de que este caderno contém 60 (sessenta) questões. Caso contrário, solicite ao fiscal da sala a sua substituição.



CONHECIMENTOS ESPECÍFICOS

QUESTÃO 01 – Sobre algumas APIs da plataforma J2EE 1.4, analise as assertivas abaixo:

- I. JavaServer Pages (JSP) é uma tecnologia que roda no cliente e é responsável por gerar páginas web.
- II. Um componente Enterprise JavaBeans (EJB) é utilizado para implementar módulos de lógica de negócios.
- III. A API JDBC permite que comandos SQL sejam executados a partir da linguagem Java.

Quais estão corretas?

- A) Apenas I.
- B) Apenas I e II.
- C) Apenas II e III.
- D) Apenas I e III.
- E) I, II e III.

QUESTÃO 02 – Sobre Enterprise JavaBeans (EJB) do J2EE 1.3, analise as assertivas abaixo:

- I. Os EJBs não podem realizar operações de entrada e saída de arquivos.
- II. O uso de variáveis estáticas é permitido nos EJBs.
- III. Os EJBs podem atuar como servidor de rede, recebendo conexões.

Quais estão corretas?

- A) Apenas I.
- B) Apenas II.
- C) Apenas III.
- D) Apenas I e II.
- E) Apenas II e III.

QUESTÃO 03 – Sobre os *Servlets* do J2EE 1.4, analise as assertivas a seguir:

- I. Um *servlet* é uma classe da linguagem de programação Java.
- II. Pode ser usado para manipular requisições HTTP.
- III. Um *servlet* escrito para rodar num servidor Windows deve ser reescrito quando portado para rodar num servidor Linux.

Quais estão corretas?

- A) Apenas I.
- B) Apenas II.
- C) Apenas III.
- D) Apenas I e II.
- E) I, II e III.

QUESTÃO 04 – Sobre a API JNDI (do inglês, *Java Naming and Directory Interface*) do J2EE 1.3, analise as assertivas a seguir:

- I. É usada para localizar recursos que são registrados através do servidor J2EE.
- II. A JNDI também é um serviço de atribuição de nomes.
- III. O LDAP (do inglês, *Lightweight Directory Access Protocol*) é um serviço de atribuição de nomes suportado pelo JNDI.

Quais estão corretas?

- A) Apenas I.
- B) Apenas I e II.
- C) Apenas II e III.
- D) Apenas I e III.
- E) I, II e III.

QUESTÃO 05 – Considerando que a variável inteira *v* (tipo *int*) armazena o valor 4, associe as expressões apresentadas na Coluna 1 ao respectivo resultado listados na Coluna 2. Considere que as operações sejam executadas independentemente uma das outras.

COLUNA 1

1. $v >> 2$
2. $v | 0x01$
3. $v \% 5$

COLUNA 2

- () 1
- () 5
- () 4

A ordem correta de preenchimento dos parênteses, de cima para baixo, é:

- A) 1 – 2 – 3.
- B) 2 – 1 – 3.
- C) 2 – 3 – 1.
- D) 1 – 3 – 2.
- E) 3 – 2 – 1.

QUESTÃO 06 – Sobre conceitos de orientação a objetos em Java, analise as assertivas abaixo:

- I. Não é permitido a uma classe estender (usando a palavra-chave *extends*) mais de uma classe base.
- II. Uma classe pode implementar (usando a palavra-chave *implements*) mais de uma interface.
- III. Uma classe declarada com o modificador *final* não pode ser estendida.

Quais estão corretas?

- A) Apenas I.
- B) Apenas II.
- C) Apenas I e II.
- D) Apenas II e III.
- E) I, II e III.

QUESTÃO 07 – Considere o seguinte trecho de código Java:

```
int i = 1;
int j = i++;
int k = ++i;
int m = --k;
int n = --j + m++;
```

Qual o valor armazenado na variável *n* após a execução desse trecho de código?

- A) 0
- B) 1
- C) 2
- D) 3
- E) 4

QUESTÃO 08 – Analise o seguinte trecho de código Java:

```
int[] array = {1,2,3,4,5};

for ( int i = 0; i < array.length - 1; i++ ) {
    array[i] = array[i+1];
    array[i+1] = array[i];
} // end for
```

Após a execução desse trecho, qual será o conteúdo da variável *array*?

- A) 1, 2, 3, 4, 5
- B) 2, 3, 4, 5, 6
- C) 2, 3, 4, 5, 5
- D) 2, 3, 4, 5, 0
- E) 2, 3, 4, 5, null

QUESTÃO 09 – Sobre alguns conceitos de orientação a objetos em Java, analise as assertivas abaixo:

- I. Um método que é declarado com o modificador *final* não pode ser sobrescrito em uma subclasse.
- II. Quando se chama um método através de uma referência para superclasse, Java executa o método sobrescrito respectivo na subclasse associada com o objeto.
- III. A instanciação de objetos de classes abstratas é permitido, mas não recomendado.

Quais estão corretas?

- A) Apenas I.
- B) Apenas I e II.
- C) Apenas I e III.
- D) Apenas II e III.
- E) I, II e III.

QUESTÃO 10 – Uma maneira correta de representar a equação algébrica abaixo em código Java é dada pela alternativa:

$$\frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

- A) `(-b + Math.sqrt(Math.pow(b,2) - 4*a*c)) / 2*a`
- B) `(-b + Math.sqrt(b*b - 4*a*c)) / (2*a)`
- C) `-b + Math.sqrt(b*b - 4*a*c) / (2*a)`
- D) `(-b + Math.sqrt(b*b - 4*a*c)) / 2*a`
- E) `(-b + Math.sqrt(Math.pow(b,2) - 4*a*c)) / 2*a`

QUESTÃO 11 – Sobre *multithreading* e a classe *Thread* em Java, analise as seguintes assertivas:

- I. A palavra-chave *synchronized* na definição de um método indica que esse método só pode ser acessado por uma *thread* de cada vez.
- II. O método *join* permite uma *thread* esperar pelo término de outra.
- III. Dois métodos *synchronized* diferentes num mesmo objeto podem ser acessados simultaneamente por duas *threads* distintas.

Quais estão corretas?

- A) Apenas I.
- B) Apenas II.
- C) Apenas I e II.
- D) Apenas I e III.
- E) Apenas II e III.

QUESTÃO 12 – Analise as assertivas abaixo sobre exceções verificadas e não-verificadas em Java.

- I. As exceções verificadas (em inglês, *checked exceptions*) precisam ser listadas direta ou indiretamente (através de uma superclasse) na cláusula *throws* do método que as dispara.
- II. As subclasses da classe *RuntimeException* são consideradas exceções verificadas (em inglês, *checked exceptions*).
- III. Um método de uma subclasse que sobrescreve um método de uma superclasse não pode listar exceções verificadas em sua cláusula *throws* que não sejam as exceções (ou subclasses dessas) listadas na cláusula *throws* do método da superclasse que foi sobrescrito.

Quais estão corretas?

- A) Apenas I.
- B) Apenas II.
- C) Apenas III.
- D) Apenas I e III.
- E) Apenas II e III.

QUESTÃO 13 – Considere a seguinte hierarquia de classes Java.

```

abstract class Letra {
    public void imprime() { System.out.println("Sou uma letra.");}
}

abstract class Consoante extends Letra {
    public void imprime() { super.imprime(); System.out.println("Sou uma consoante.");}
}

abstract class Vogal extends Letra {
    public void imprime() { super.imprime(); System.out.println("Sou uma vogal.");}
}

class A extends Vogal {
    public void imprime() {super.imprime(); System.out.println("A");}
}

class B extends Consoante {
    public void imprime() {super.imprime(); System.out.println("B");}
}
    
```

Qual alternativa apresenta um trecho de código e saída correspondentes?

A)
Código:

```

Letra letra = new B();
letra.imprime();
    
```

Saída:

```

Sou uma letra.
Sou uma consoante.
B
    
```

B)
Código:

```

Consoante consoante = new B();
consoante.imprime();
    
```

Saída:

```

Sou uma letra.
Sou uma consoante.
    
```

C)
Código:

```

A a = new A();
a.imprime();
    
```

Saída:

```

A
    
```

D)
Código:

```

Vogal a = new A();
a.imprime();
    
```

Saída:

```

A
Sou uma vogal.
Sou uma letra.
    
```

E)
Código:

```

Letra letra = new A();
letra.imprime();
    
```

Saída:

```

Sou uma letra.
    
```

QUESTÃO 14 – Selecione a alternativa que apresenta um método Java que calcula e retorna o somatório de todos os elementos com índice par de um vetor (*array*) de inteiros.

A)

```
int fa(int[] v) {
    int resultado = 0;
    for ( int i = 0; i < v.length; i++ ) {
        resultado += v[i];
    }
    return resultado;
}
```

B)

```
int fb(int[] v) {
    int resultado = 0;
    for ( int i = 0; i < v.length; i++ ) {
        if ( i % 2 == 1 )
            resultado += v[i];
    }
    return resultado;
}
```

C)

```
int fc(int[] v) {
    int resultado = 0;
    for ( int i : v ) {
        resultado += i;
    }
    return resultado;
}
```

D)

```
int fd(int[] v) {
    int resultado = 0;
    boolean flag = true;
    for ( int i : v ) {
        if ( flag )
            resultado += i;
        flag = !flag;
    }
    return resultado;
}
```

E)

```
int fe(int[] v) {
    int resultado = 0;
    int i = 0;
    while ( ++i < v.length ) {
        if ( i % 2 == 0 )
            resultado += v[i];
    }
    return resultado;
}
```

QUESTÃO 15 – Sobre o método construtor em Java, assinale a alternativa correta.

- A) Pode retornar um valor.
- B) É chamado quando uma instância da classe é destruída pelo coletor de lixo.
- C) Não pode receber parâmetros.
- D) Não pode ser sobrecarregado.
- E) Precisa ter o mesmo nome da classe.

QUESTÃO 16 – Sobre passagem de parâmetros em Java, analise as assertivas abaixo.

- I. Java possui apenas o mecanismo de passagem de parâmetro por valor (em inglês, *by value*).
- II. Não é possível criar métodos que modificam o conteúdo de variáveis numéricas (*int*, *double*) passadas como parâmetro.
- III. A declaração de método *void foo(String... s) {}* é válida e indica um método que pode receber um número variável de parâmetros.

Quais estão corretas?

- A) Apenas I.
- B) Apenas II.
- C) Apenas I e II.
- D) Apenas II e III.
- E) I, II e III.

QUESTÃO 17 – Relacione os níveis de controle de acesso de membros de classes em Java na Coluna 1 com sua respectiva descrição na Coluna 2.

COLUMNA 1

1. public
2. private
3. protected
4. package

COLUMNA 2

- () Acesso é permitido na própria classe e em todas as classes no mesmo pacote.
- () Acesso é permitido apenas na própria classe.
- () Acesso é permitido na própria classe bem como em todas as outras classes.
- () Acesso é permitido na própria classe bem como em todas as subclasses e classes no mesmo pacote.

A ordem correta de preenchimento dos parênteses, de cima para baixo, é

- A) 4 – 2 – 1 – 3.
- B) 3 – 2 – 1 – 4.
- C) 4 – 1 – 2 – 3.
- D) 3 – 1 – 2 – 4.
- E) 3 – 1 – 4 – 2.

Questão 18 - Selecione a alternativa que melhor e mais amplamente descreve a seguinte construção Java:

```
public static void metodo(List<? extends A> list) { /* ... */ }
```

- A) Uma declaração de classe que estende a classe A.
- B) Um método que recebe como parâmetro uma lista de objetos do tipo A ou que sejam subclasses de A.
- C) Uma declaração de classe genérica.
- D) Um método que recebe como parâmetro uma lista de instâncias da classe A.
- E) Um método que recebe como parâmetro uma lista de qualquer tipo de objetos.

QUESTÃO 19 – Sobre a palavra-chave *super* em Java, analise as assertivas a seguir:

- I. Pode ser usada numa subclasse para chamar o construtor da superclasse imediata.
- II. Pode ser usada, seguida do operador ponto, para acessar membros da superclasse imediata.
- III. Não é possível encadear referências a superclasses como em *super.super.toString()*.

- A) Apenas II.
- B) Apenas III.
- C) Apenas I e II.
- D) Apenas II e III.
- E) I, II e III.

QUESTÃO 20 – Suponha que *i*, *j*, *k* e *m* sejam variáveis do tipo *int* e que *i* = 2, *j* = 5, *k* = 9 e *m* = -8. Preencha os parênteses com V para as comparações avaliadas para *true* e com F as comparações avaliadas para *false*.

- () *i* >= 2 && *m* > -2
- () 3*3 == *k* || *j* != 5
- () *m* < -10 || !(*k* != -*m* + 1)
- () *k* >= *j*? false : true

A ordem correta de preenchimento dos parênteses, de cima para baixo, é

- A) F – V – V – F.
- B) F – V – V – V.
- C) F – F – F – V.
- D) V – V – F – F.
- E) V – F – F – F.

QUESTÃO 21 – Sobre *arrays* em Java, analise as assertivas que seguem:

- I. Quando um *array* é declarado, os elementos (valores) são inicializados com 0 (zero) para *arrays* de números como *int[]* e *double[]*; *false* para *arrays* do tipo *boolean[]*; ou *null* para um *array* de referências para objetos.
- II. O último índice válido de um *array* é igual ao tamanho (propriedade *length*) desse *array*.
- III. Tentar acessar um índice fora dos limites do *array* é um erro detectado em tempo de compilação.

Quais estão corretas?

- A) Apenas I.
- B) Apenas II.
- C) Apenas III.
- D) Apenas I e II.
- E) I, II e III.

QUESTÃO 22 – Sobre o método finalizador *finalize*, analise as assertivas a seguir:

- I. Quando um objeto é finalizado, os métodos finalizadores (*finalize*) das superclasses são automaticamente chamados pela máquina virtual.
- II. Não recebe nenhum parâmetro.
- III. É chamado pelo coletor de lixo.

Quais estão corretas?

- A) Apenas I.
- B) Apenas III.
- C) Apenas I e II.
- D) Apenas I e III.
- E) Apenas II e III.

QUESTÃO 23 – Relacione os métodos da classe padrão Math indicados na Coluna 1 com a respectiva descrição apresentada na Coluna 2.

COLUNA 1

1. abs(x)
2. ceil(x)
3. floor(x)
4. sqrt(x)

COLUNA 2

- () Retorna o valor absoluto de x.
- () Retorna o valor de x arredondado para o menor inteiro não menor que x.
- () Retorna o valor de x arredondado para o maior inteiro não maior que x.
- () Retorna a raiz quadrada de x.

A ordem correta de preenchimento dos parênteses, de cima para baixo, é:

- A) 1 – 3 – 2 – 4.
- B) 4 – 2 – 3 – 1.
- C) 1 – 2 – 3 – 4.
- D) 4 – 3 – 2 – 1.
- E) 2 – 1 – 3 – 4.

QUESTÃO 24 – Sobre interfaces em Java, analise as seguintes assertivas:

- I. Interfaces provêm implementação padrão para métodos que podem ou não serem sobrescritos nas classes que a implementam.
- II. Em uma interface, todo atributo é implicitamente final e estático.
- III. Uma classe pode não pode implementar mais de uma interface.

Quais estão corretas?

- A) Apenas I.
- B) Apenas II.
- C) Apenas III.
- D) Apenas I e II.
- E) Apenas I e III.

QUESTÃO 25 – Sobre sobrecarga em Java, analise as assertivas a seguir:

- I. Java possui mecanismo de sobrecarga de operadores.
- II. É possível sobrecarregar um método com mesma assinatura, mas com tipo de retorno diferente.
- III. Métodos sobrecarregados precisam ter o mesmo número de parâmetros.

Quais estão INCORRETAS?

- A) Apenas I.
- B) Apenas II.
- C) Apenas III.
- D) Apenas II e III.
- E) I, II e III.

QUESTÃO 26 – Considere algumas interfaces de estrutura de coleção apresentadas na Coluna 1 e relacione-as com a respectiva descrição apresentada na Coluna 2.

COLUNA 1

1. Set.
2. List.
3. Map.
4. Queue.

COLUNA 2

- () Em geral, uma coleção primeiro a entrar, primeiro a sair.
- () Pode conter elementos duplicados.
- () Associa chaves a valores e não pode conter chaves duplicadas.
- () Uma coleção que não contém duplicadas.

A ordem correta de preenchimento dos parênteses, de cima para baixo, é:

- A) 4 – 2 – 1 – 3.
- B) 2 – 4 – 3 – 1.
- C) 2 – 3 – 4 – 1.
- D) 4 – 2 – 3 – 1.
- E) 3 – 4 – 1 – 2.

QUESTÃO 27 – Considere alguns algoritmos de estrutura de coleções apresentados na Coluna 1 e relacione-os com sua respectiva descrição apresentada na Coluna 2.

COLUNA 1

1. disjoint.
2. frequency.
3. shuffle.
4. binarysearch.

COLUNA 2

- () Calcula quantos elementos na coleção são iguais ao elemento especificado.
- () Determina se duas coleções não têm nenhum elemento em comum.
- () Localiza um objeto em uma List.
- () Ordena aleatoriamente os elementos de uma List.

A ordem correta de preenchimento dos parênteses, de cima para baixo, é:

- A) 4 – 1 – 2 – 3.
- B) 2 – 1 – 4 – 3.
- C) 3 – 1 – 4 – 2.
- D) 2 – 4 – 1 – 3.
- E) 3 – 4 – 1 – 2.

QUESTÃO 28 – Considere o seguinte trecho de código Java:

```

for ( int i = 0; i < 5; i++ ) {
    for ( int j = 0; j < i+1; j++ ) {
        System.out.print("*");
    } // end for
    System.out.println();
} // end for
    
```

O texto impresso na saída padrão é correspondente à alternativa

A)

 **
 *

B)
 *
 **

C)
 *
 *
 *
 *
 *

D)

E)

 **
 *
 **

QUESTÃO 29 – Considere o programa Java apresentado na caixa abaixo.

```
import java.util.*;

class App {
    private final String first, last;

    public App(String first, String last) {
        this.first = first;
        this.last = last;
    }

    public boolean equals(Object o) {
        if ( !(o instanceof App))
            return false;
        App n = (App)o;
        return n.first.equals(first) && n.last.equals(last);
    }

    public String toString() {
        return first + ", " + last;
    }

    static public void main(String [] args) {
        Set<App> s = new HashSet<App>();
        s.add(new App("A", "B"));
        System.out.println(s.contains(new App("A", "B")));
    }
}
```

Qual a mensagem que poderá ser impressa na saída padrão?

- A) A, B
- B) null
- C) false
- D) A
- E) B

QUESTÃO 30 – Considere a precedência de operadores em Java e preencha os parênteses com números de 1 a 5, em que 1 indica o operador de maior precedência (que é executado primeiro) e 5 o operador de menor precedência.

- () - (subtração)
- () ++ (pós-incrementado)
- () && (e condicional)
- () || (ou condicional)
- () / (divisão)

A ordem correta de preenchimento dos parênteses, de cima para baixo, é:

- A) 1 – 3 – 2 – 5 – 4.
- B) 5 – 4 – 1 – 3 – 2.
- C) 3 – 1 – 4 – 5 – 2.
- D) 1 – 3 – 4 – 2 – 5.
- E) 3 – 1 – 2 – 5 – 4.

QUESTÃO 31 – Considere o seguinte trecho de código Java:

```
int[] arrayv = {2, 3, 5, 7, 11, 13, 17};

int[] arrayi = {0, 3, 2, 1};
int[] arrayj = {1, 6, 4, 5};

int s = 0;
for ( int i = 0; i < arrayi.length; i++ )
    s += arrayv[arrayi[i]] + arrayv[arrayj[i]];

System.out.println(s);
```

Qual será o valor escrito na saída padrão?

- A) 11
- B) 29
- C) 37
- D) 56
- E) 61

QUESTÃO 32 – Qual método Java calcula corretamente o fatorial de um número inteiro positivo?

A)

```
int fa(int n) {
    return n * fa(n-1);
}
```

B)

```
int fb(int n) {
    return n!;
}
```

C)

```
int fc(int n) {
    int result;
    for ( int i = 0; i < n; i++)
        result *= i;
    return result;
}
```

D)

```
int fd(int n) {
    int result = 1;
    while(n > 0)
        result *= n--;
    return result;
}
```

E)

```
int fe(int n) {
    return n >= 0 ? n * fe(n-1) : 1;
}
```

QUESTÃO 33 – Sobre a *serialização* de objetos em Java, analise as assertivas abaixo:

- I. A palavra-chave *transient* indica que um atributo de classe não deve ser *serializado*.
- II. O sistema de *serialização* de Java não suporta referência circular entre objetos (e.g. A referencia B que por sua vez referencia A).
- III. Objetos salvos e recuperados utilizando Object Streams (ObjectOutputStream e ObjectInputStream) precisam implementar a interface Serializable.

Quais estão corretas?

- A) Apenas I.
- B) Apenas II.
- C) Apenas III.
- D) Apenas I e II.
- E) Apenas I e III.

QUESTÃO 34 – Qual mensagem será impressa pelo seguinte trecho de código Java na saída padrão?

```
int a = 10;

if ( a <= 0 )
    System.out.println("A");
    if ( a > 10)
        System.out.println("B");
else
    System.out.println("C");

switch (a) {
    case 0x10:
        System.out.println("D"); break;
    case 0xA:
        break;
    default:
        System.out.println("E");
}
```

- A) A
- B) B
- C) C
- D) D
- E) E

QUESTÃO 35 – Considere a função recursiva *f* apresentada na caixa abaixo:

```
long f(long n) {
    if ( n == 0 || n == 1)
        return n;
    else
        return f(n-1) + f(n-2);
}
```

Qual o valor de *f*(5)?

- A) 2
- B) 3
- C) 4
- D) 5
- E) 8

QUESTÃO 36 – Analise o seguinte trecho de código Java:

```
class MyException extends Exception {
}

class App {
    static public void main(String [] args) {
        try {
            throw new MyException();
            System.out.println("Hello World");
        } catch (Exception e) {
            /*...*/
        } catch (MyException e) {
            /*...*/
        } finally {
            /*...*/
        }
    }
}
```

Considerando o trecho de código de Java acima, assinale a alternativa correta.

- A) É um programa Java válido.
- B) A mensagem *Hello World* será escrita na saída padrão.
- C) O bloco *finally* será executado devido ao lançamento da exceção *MyException* dentro do bloco *try*.
- D) Ao ser compilado, o compilador Java indicará um erro de compilação.
- E) O bloco *catch (MyException e)* será executado.

QUESTÃO 37 – Sobre expressões regulares e a classe *java.util.regex.Pattern* de Java, analise o seguinte programa Java:

```
import java.util.regex.Pattern;

class App {
    static public void main(String[] args) {
        String pattern = "a+[bc]*";

        int i = 0;
        i += Pattern.matches(pattern, "aaaaab")? 1 : 0;
        i += Pattern.matches(pattern, "a")? 1 : 0;
        i += Pattern.matches(pattern, "bcbc")? 1 : 0;
        i += Pattern.matches(pattern, "aaccbc")? 1 : 0;
        i += Pattern.matches(pattern, "cbbbc")? 1 : 0;

        System.out.println(i);
    }
}
```

Qual será o valor escrito na saída padrão?

- A) 1
- B) 2
- C) 3
- D) 4
- E) 5

QUESTÃO 38 – Sobre expressões regulares e a classe `java.util.regex.Pattern` de Java, considere que se deseja casar a sequência de caracteres apresentada na caixa a seguir:

```
<a href='http://www.fundatec.org.br'>Fundatec</a>
```

Qual padrão (expressão regular) poderá ser usado para identificar esse tipo de sequência através da classe `Pattern` de Java?

- A) `<a\\s*href\\s+=\\s+'.*'\\s+>.*`
- B) `<a\\s+href\\s*=\\s*'.*'\\s*>.*`
- C) `<a\\s+href\\s*=\\s*'.*'\\s*>[a-zA-Z0-9]*`
- D) `<a\\s+href\\s+=\\s+'.*'\\s+>.*`
- E) `<a\\s*href\\s*=\\s*'.*'\\s*>.*`

QUESTÃO 39 – Considere o método *recursivo* apresentado na caixa abaixo.

```
long f(long n) {
    if ( n == 0 )
        return 0;
    else
        return n + f(n-1);
}
```

Supondo $n > 0$, tal método pode ser traduzido para o método *não-recursivo* apresentado em qual alternativa?

A)

```
long fa(long n) {
    return n==0 ? 0 : n + fa(n-1);
}
```

B)

```
long fb(long n) {
    int r = 0;
    while(n-- > 0)
        r += n+1;
    return r;
}
```

C)

```
long fc(long n) {
    long r = 0;
    for ( int i = 0; i < n; i++)
        r += i;
    return r;
}
```

D)

```
long fd(long n) {
    if ( n > 0 )
        return n + fd(n-1);
    else
        return 0;
}
```

E)

```
long fe(long n) {
    return (2*(n-1))/2;
}
```

QUESTÃO 40 – Preencha os parênteses quanto à quantidade de *bits* utilizada em Java para armazenar o tipo de dado primitivo indicado.

- () int
- () short
- () long
- () byte

A ordem correta de preenchimento dos parênteses, de cima para baixo, é:

- A) 32 - 24 - 48 - 8
- B) 4 - 2 - 8 - 1
- C) 32 - 16 - 64 - 8
- D) 4 - 3 - 6 - 1
- E) 32 - 16 - 48 - 8

QUESTÃO 41 – Sobre objetos imutáveis, analise as assertivas a seguir:

- I. Os objetos da classe `String` são exemplo de objetos imutáveis.
- II. Objetos imutáveis são considerados seguros de serem usados em programação concorrente (*thread-safe*).
- III. O modificador `immutable` é usado antes da declaração da classe para defini-la imutável.

Quais estão corretas?

- A) Apenas I.
- B) Apenas II.
- C) Apenas I e II.
- D) Apenas I e III.
- E) I, II e III.

QUESTÃO 42 – Considere o método *int compareTo(String)* da classe *String* de Java e analise as seguintes assertivas:

- I. Apenas os valores -1, 0 e 1 são retornados.
- II. A expressão `"Fundatec".compareTo("Brasil")` é avaliada como -1.
- III. A expressão `"Fundatec".compareTo("fundatec")` é avaliada como 0.

Quais estão INCORRETAS?

- A) Apenas I.
- B) Apenas II.
- C) Apenas I e II.
- D) Apenas I e III.
- E) I, II e III.

QUESTÃO 43 – Analise o seguinte programa Java.

```
class App {  
  
    static private void concatena(String s1, String s2) {  
        s1 = s1 + s2;  
        s2 = "";  
    }  
  
    static public void main(String[] args) {  
        String s1 = "Fundatec";  
        String s2 = "Concursos";  
  
        concatena(s1, s2);  
  
        System.out.println(s1 + " " + s2);  
    }  
  
}
```

Ao executar tal programa, o que será escrito na saída padrão?

- A) Fundatec
- B) Fundatec Concursos
- C) Fundatec Concursos Concursos
- D) Concursos
- E) Concursos Concursos

QUESTÃO 44 – Analise o seguinte aplicativo Java:

```

import java.awt.*;
import javax.swing.*;

public class App extends JFrame {
    public App() {
        super("Fundatec");

        JPanel panel = new JPanel();
        panel.setLayout( new BorderLayout() );

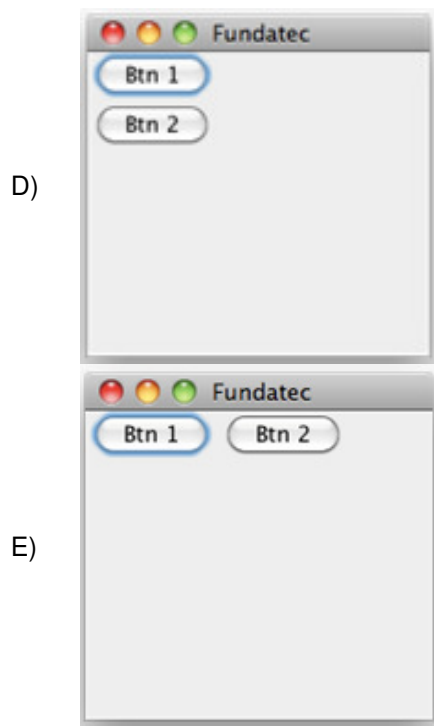
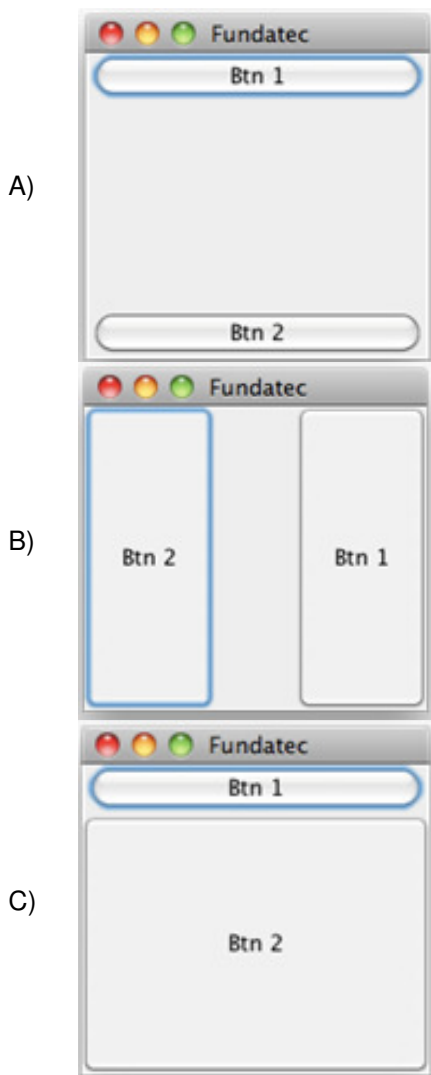
        panel.add( new JButton("Btn 1"), BorderLayout.NORTH );
        panel.add( new JButton("Btn 2"), BorderLayout.SOUTH );

        add(panel);
    }

    static public void main(String[] args) {
        App app = new App();
        app.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        app.setSize( 200, 200 );
        app.setVisible( true );
    }
}

```

Qual alternativa apresenta uma imagem que melhor representa a interface gráfica criada por tal aplicativo?



QUESTÃO 45 – Analise o programa Java a seguir:

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class App extends JFrame {
    public App() {
        super("Fundatec");

        JPanel panel = new JPanel();
        panel.setLayout( new FlowLayout() );

        JButton btn = new JButton("Ok");
        btn.addMouseListener( new MouseListener() {
            public void mouseEntered(MouseEvent e) {
                System.out.print("A");
            }

            public void mouseExited(MouseEvent e) {
                System.out.print("B");
            }

            public void mouseReleased(MouseEvent e) {
                System.out.print("C");
            }

            public void mousePressed(MouseEvent e) {
                System.out.print("D");
            }

            public void mouseClicked(MouseEvent e) {
                System.out.print("E");
            }
        });

        panel.add(btn);
        add(panel);
    }

    static public void main(String[] args) {
        App app = new App();
        app.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        app.setSize( 200, 75 );
        app.setVisible( true );
    }
}
```

Qual mensagem será impressa na saída padrão ao se dirigir o mouse e clicar no botão "Ok"?

- A) E
- B) ADCE
- C) DCE
- D) DE
- E) B

QUESTÃO 46 – Considere que o arquivo texto *input.txt* tenha o seguinte conteúdo:

```
Lista de Países
EUA 100 0
Japão 90 1
China 80 1
Alemanha 70 0
Brasil 60 1
```

E que o seguinte aplicativo Java é utilizado para lê-lo (suponha que o arquivo esteja no mesmo diretório e acessível ao aplicativo).

```
import java.io.File;
import java.io.FileNotFoundException;
import java.util.Scanner;

public class App {

    public static void main(String[] args) {
        File file = new File("input.txt");

        try {

            Scanner scanner = new Scanner(file);
            scanner.nextLine();

            int count = 0;

            while (scanner.hasNextLine()) {
                String country = scanner.next();
                int value = scanner.nextInt();
                int enable = scanner.nextInt();

                if ( enable == 1 )
                    count += value;
            }
            scanner.close();

            System.out.println(count);

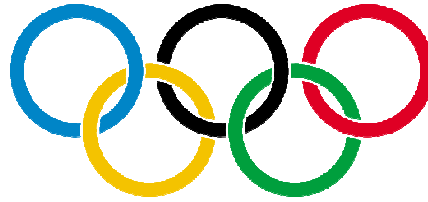
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }

    }
}
```

Qual valor será escrito na saída padrão?

- A) 400
- B) 230
- C) 0
- D) 3
- E) 160

QUESTÃO 47 – Suponha que se queira desenhar círculos arranjados de forma similar aos anéis Olímpicos mostrado na figura a seguir:



Para tal, utiliza-se o trecho de código Java no quadro abaixo.

```
public void paintComponent(Graphics graphics) {
    Graphics2D g = (Graphics2D) graphics;

    final double RADIUS = 10;
    final double[] X = { x0, x1, x2, x3, x4 };
    final double[] Y = { y0, y1, y2, y3, y4 };

    for ( int i = 0; i < X.length; i++ ) {
        g.draw( new Ellipse2D.Double(X[i],Y[i],RADIUS*2,RADIUS*2));
    }
}
```

Quais valores podem ser atribuídos às variáveis x0, y0, x1, y1, x2, y2, x3, y3, x4 e y4, nesta ordem, para obter círculos dispostos de forma similar a disposição dos anéis Olímpicos?

- A) 10, 10, 30, 10, 50, 10, 70, 20, 90 e 20.
- B) 10, 10, 30, 10, 50, 10, 20, 20, 40 e 20.
- C) 90, 20, 70, 20, 50, 10, 30, 10, 10 e 10.
- D) 10, 10, 30, 10, 50, 10, 20, 30, 40 e 30.
- E) 40, 30, 20, 30, 50, 10, 30, 10, 10 e 10.

Para responder às questões 48 a 55, considere as figuras abaixo

id	nome	grupo	salario	sexo
1	João	1	1050.00	1
2	Maria	2	3000.00	0
3	José	NULL	1210.00	1
4	Paulo	1	2300.00	1
5	Ana	1	NULL	0
6	Cris	2	5900.00	0
7	Marco	1	NULL	1
8	Fernanda	1	2689.00	0
9	Pedro	NULL	1876.00	1
10	Madalena	2	3070.00	0

Figura 1 - Tabela de dados *pessoal*.

id	nome
1	Grupo A
2	Grupo B
3	Grupo C

Figura 2 - Tabela de dados *grupos*.

QUESTÃO 48 – Considerando a tabela *peessoal* mostrada na Figura 1, quantos registros retornam o comando SQL na caixa abaixo?

```
SELECT grupo, AVG(salario) FROM peessoal GROUP BY grupo;
```

- A) Nenhum
- B) 1
- C) 2
- D) 3
- E) 10

QUESTÃO 49 – Considerando a tabela *peessoal* mostrada na Figura 1, qual o valor retornado pela consulta SQL na caixa abaixo?

```
SELECT COUNT(*) FROM peessoal WHERE sexo = 0 AND salario >= 1000;
```

- A) 0
- B) 4
- C) 5
- D) 8
- E) 10

QUESTÃO 50 – Considerando a tabela *peessoal* mostrada na Figura 1, quantos registros a consulta SQL abaixo retornará?

```
SELECT * FROM  
    peessoal AS t0, peessoal AS t1  
WHERE  
    t0.sexo = 0 AND t0.sexo != t1.sexo;
```

- A) 5
- B) 10
- C) 25
- D) 50
- E) 100

QUESTÃO 51 – Considere a tabela *peessoal* mostrada na Figura 1 e que o campo *id* é chave-primária dessa tabela. Analise o comando SQL na caixa abaixo e marque a alternativa correta.

```
INSERT INTO peessoal  
(id, nome, grupo, salario, sexo)  
VALUES  
(4, "Jacó", 2, 999, 1);
```

- A) Após a execução, a tabela *peessoal* conterà 11 registros.
- B) O registro com o campo *id* igual a 4 será atualizado.
- C) Contém erro de sintaxe.
- D) Após a execução, a tabela *peessoal* conterà 9 registros.
- E) O novo registro não será inserido devido à chave-primária duplicada.

QUESTÃO 52 – Considere a tabela *peessoal* mostrada na Figura 1 e a tabela *grupos* mostrada na Figura 2. Quantos registros a consulta SQL na caixa abaixo retornará?

```
SELECT * FROM peessoal RIGHT JOIN grupos ON peessoal.grupo = grupos.id;
```

- A) 9
- B) 10
- C) 11
- D) 20
- E) 30

QUESTÃO 53 – Considere a tabela *peessoal* mostrada na Figura 1. Deseja-se utilizar uma consulta SQL para atualizar todos e somente os registros cujo campo grupo é nulo (NULL), alterando o valor do campo grupo para 3. Para tanto, é possível utilizar qual comando SQL?

- A) UPDATE peessoal SET grupo = 3;
- B) UPDATE peessoal SET grupo = 3 WHERE grupo = 0;
- C) UPDATE peessoal SET grupo = 3 WHERE grupo IS NULL;
- D) UPDATE peessoal SET grupo = 3 WHERE grupo IS NOT NULL;
- E) UPDATE peessoal SET grupo = 3 WHERE grupo = NULL;

QUESTÃO 54 – Considere a tabela *peessoal* mostrada na Figura 1 e a tabela *grupos* mostrada na Figura 2. Quais registros, referenciados pelo valor do campo *id*, são retornados pela consulta SQL abaixo, considerando a ordem em que são retornados?

```
SELECT * FROM
    peessoal, grupos
WHERE
    peessoal.grupo = grupos.id AND
    peessoal.sexo = 1 AND
    grupos.id = 1
ORDER BY
    peessoal.salario;
```

- A) 4, 9, 3, 1, 7
- B) 7, 1, 3, 9, 4
- C) 7, 1, 4
- D) 4, 1, 7
- E) A consulta retorna nenhum (zero) registro.

QUESTÃO 55 – Considere a tabela *pessoal* mostrada na Figura 1, a tabela *grupos* mostrada na Figura 2 e, ainda, o aplicativo Java mostrado na caixa abaixo.

```
import java.sql.*;

public class App {
    public static void main(String[] args) {
        try {
            Connection conn;

            Class.forName("com.mysql.jdbc.Driver").newInstance();
            String url = "jdbc:mysql://localhost/fundatec";
            conn =
                DriverManager.getConnection(
                    url,
                    "fundatec",
                    "86GnTfZp7emDxQZ9"
                );

            Statement stmt = conn.createStatement();

            String sql =
                "SELECT * FROM pessoal WHERE sexo = ? AND salario > ?";

            PreparedStatement pst = conn.prepareStatement(sql);

            pst.setInt(1, 0);
            pst.setDouble(2, 3000.00);

            ResultSet rs = pst.executeQuery();

            int total = 0;

            while (rs.next()) {
                total += rs.getDouble("salario");
            }

            System.out.println(total);

            conn.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Supondo que os *drivers* estão disponíveis e que a conexão com o banco de dados MySQL é estabelecida sem problemas, qual o valor impresso na saída padrão?

- A) 0
- B) 5900
- C) 14659
- D) 8970
- E) 6436

QUESTÃO 56 – Sobre chaves-primárias e chaves-estrangeiras em banco de dados relacional SQL, analise as seguintes assertivas:

- I. Uma chave-primária é composta de apenas um campo.
- II. Registros diferentes, numa mesma tabela, não podem ter o mesmo valor de chave-primária.
- III. Uma chave-estrangeira pode relacionar campos de uma tabela apenas com a chave-primária de outra tabela ou da mesma tabela.

Quais estão corretas?

- A) Apenas I.
- B) Apenas II.
- C) Apenas III.
- D) Apenas I e II.
- E) I, II e III.

QUESTÃO 57 – Associe os níveis de visibilidade apresentados na Coluna 1 aos respectivos símbolos sugeridos pela UML apresentados na Coluna 2.

COLUNA 1

- 1. privada (*private*)
- 2. pacote (*package*)
- 3. pública (*public*)
- 4. protegida (*protected*)

COLUNA 2

- () #
- () ~
- () +
- () -

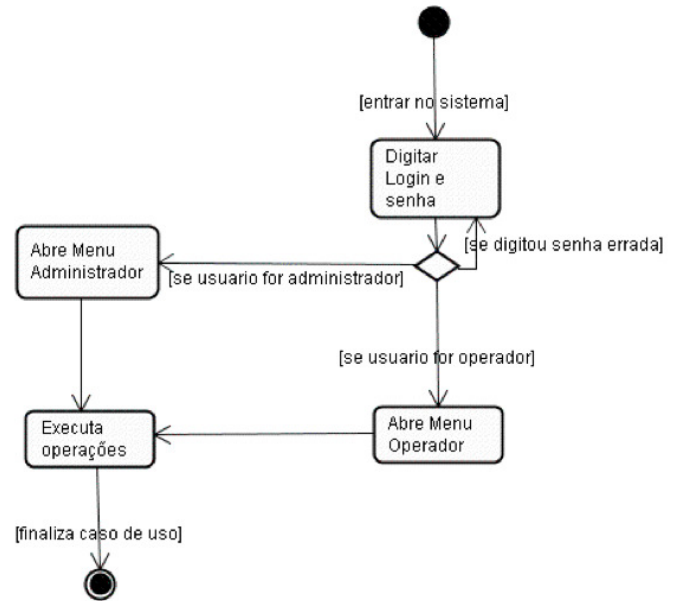
A ordem correta de preenchimento dos parênteses, de cima para baixo, é

- A) 2 – 4 – 3 – 1.
- B) 4 – 2 – 1 – 3.
- C) 4 – 2 – 3 – 1.
- D) 3 – 2 – 4 – 1.
- E) 3 – 1 – 4 – 2.

QUESTÃO 58 – Sobre diagramas de Caso de Uso, assinale a alternativa correta.

- A) Representam a lógica exigida para implementar comportamentos do sistema.
- B) Identificam as expectativas dos clientes para o sistema.
- C) Identificam explicitamente os dados que são passados como parte das interações.
- D) Identificam quais eventos um objeto responderá ou não.
- E) Documentam claramente os requisitos de temporização que controlam a mudança de estado.

QUESTÃO 59 – No contexto de UML, a imagem abaixo representa um diagrama de



- A) Atividade.
- B) Implantação.
- C) Classe.
- D) Sequência.
- E) Caso de Uso.

QUESTÃO 60 – No contexto de UML, assinale a alternativa que descreve um *artefato* mais precisamente.

- A) Detalhe da especificação de um elemento, acrescentado à sua notação gráfica básica.
- B) Uma computação executável que resulta em alteração do estado do sistema ou no retorno de um valor.
- C) Um elemento físico existente em tempo de execução que representa um recurso computacional, geralmente dispendido de pelo menos alguma memória e, na maioria das vezes, capacidade de processamento.
- D) Um conjunto de informações utilizado ou produzido por um processo de desenvolvimento de software ou sistema existente.
- E) A especificação de uma computação executável.