

Analise o trecho de código abaixo.

```
public class Onibus {
    private int poltronas = 32;
    private int passageiros = 0;

    public void embarcarPassageiro(int numero) {
        if ((passageiros+numero)<poltronas)
            passageiros +=numero;
    }
    public int retornarPoltronasVagas() {
        return poltronas - passageiros;
    }
    public static void main(String[] args) {
        Onibus bus1 = new Onibus();
        bus1.embarcarPassageiro(20);

        Onibus bus2 = bus1;
        bus2.embarcarPassageiro(15);

        System.out.print (bus1.retornarPoltronasVagas()+" e ");
        System.out.print (bus2.retornarPoltronasVagas()+".");
    }
}
```

01. A saída impressa gerada pelo programa é

- a) 12 e 12.
- b) 12 e 17.
- c) 0 e 0.
- d) 17 e 17.

02. A palavra-chave *final* no Java é utilizada para:

- a) encerrar a execução da aplicação.
- b) declarar constantes.
- c) declarar classes que não podem ser instanciadas.
- d) tratar exceções.

03. A respeito dos construtores em Java, analise as seguintes afirmações:

- I. Os construtores são métodos especiais que permitem iniciar as variáveis de instância do objeto.
- II. Um construtor não pode chamar métodos que foram definidos na classe, esta operação gera um erro de compilação.
- III. Uma classe pode ter vários construtores desde que declarados com assinaturas diferentes.

Estão CORRETAS as afirmativas

- a) I e II apenas.
- b) I e III apenas.
- c) II e III apenas.
- d) I, II e III.

04. A respeito de variáveis e métodos *static* em Java, marque a alternativa CORRETA:

- a) Os métodos declarados como *static* são chamados métodos de instância, sendo necessário instanciar um objeto para invocá-los.
- b) Os métodos declarados como *static* são chamados métodos de classe, não sendo necessário instanciar um objeto para invocá-los.
- c) Uma variável *static* tem escopo de classe, ou seja, todos os objetos da classe têm uma cópia própria dos dados da variável.
- d) Uma variável *static* é também chamada constante, ou seja, depois da declaração e atribuição de valor, esta variável não pode ter seu conteúdo alterado durante a execução do programa.

05. Marque a alternativa que apresenta a declaração CORRETA de um vetor de inteiros no Java.

- a) `int vetor[] = new int{3};`
- b) `int vetor[3] = new int;`
- c) `int vetor = {5, 10, 15};`
- d) `int vetor = new {5, 10, 15};`

06. A respeito dos modificadores de acesso no Java, marque a alternativa CORRETA:

- a) Uma variável ou método da classe declarados com o modificador de acesso *final* somente podem ser acessados pelos métodos da própria classe.
- b) Uma variável ou método da classe declarados com o modificador de acesso *protected* podem ser acessados por métodos da própria classe e por métodos de uma classe derivada.
- c) Uma variável ou método da classe declarados com o modificador de acesso *public* somente podem ser acessados por classes pertencentes ao mesmo pacote.
- d) *final*, *public*, *protected* e *void* são exemplos de modificadores de acesso válidos no Java.

07. Marque a alternativa que resulta **EM ERRO** de compilação no Java:

- a) `float saldo = 150;`
- b) `float saldo = 150.0;`
- c) `float saldo = 150.0f;`
- d) `float saldo = (float) 150.0;`

08. A respeito de herança no Java, analise as seguintes afirmações:

- I. O Java não suporta herança múltipla.
- II. Quando uma subclasse redefine um método herdado pela superclasse, utilizando a mesma assinatura de método, diz-se que a subclasse sobrecarrega esse método de superclasse.
- III. O construtor de uma subclasse pode chamar o construtor da superclasse, explicitamente, através da palavra-chave *super*.

Estão CORRETAS as afirmativas

- a) I e II apenas.
- b) I e III apenas.
- c) II e III apenas.
- d) I, II e III.

Analise o trecho de código abaixo.

```
public class ProgramaVetor {
    public static void main(String[] args) {
        int v[] = {45,12,9,1,22,34,78,4};
        int i,j,z;
        i = j = z = 0;

        for (i=0; i<v.length-1;i++){
            for (j=i; j<v.length;j++){
                if (v[i]>v[j])
                    z = v[i];
                    v[i] = v[j];
                    v[j] = z;
            }
        }
        for (i=0;i<v.length;i++){
            System.out.print(v[i]);
            if ((i+1)!=v.length)
                System.out.print(", ");
        }
    }
}
```

09. Qual a saída impressa gerada pelo programa?

- a) 1, 4, 9, 12, 22, 34, 45, 78.
- b) 78, 45, 34, 22, 12, 9, 4, 1.
- c) 4, 78, 12, 12, 12, 12, 12, 12.
- d) 4, 34, 22, 1, 0, 0, 0, 0.

10. Analise as seguintes afirmações sobre *interfaces* no Java:

- I. Uma *interface* define um conjunto de métodos que a classe deve oferecer.
- II. As *interfaces* não especificam implementação, somente a assinatura dos métodos.
- III. Os programadores podem declarar classes que implementam uma ou mais *interfaces*.

Estão CORRETAS as afirmativas

- a) I e II apenas.
- b) I e III apenas.
- c) II e III apenas.
- d) I, II e III.

Analise o trecho de código abaixo.

```
public class ManipulaVetor {
    public void modificaVetor(int a[]){
        int i;
        for (i=0;i<a.length;i++){
            modificaElemento(a[i]);
        }
    }
    public void modificaElemento(int x){
        if (x%2==0)
            x *= x;
        else
            x += x;
    }
    public static void main(String[] args) {
        int v[] = {1,2,3,4,5};
        ManipulaVetor mv = new ManipulaVetor();
        mv.modificaVetor(v);
        for (int i=0;i<v.length-1;i++){
            System.out.print(v[i]+" ");
        }
        System.out.print(v[v.length-1]);
    }
}
```

11. Qual a saída impressa gerada pelo programa?

- a) 2, 4, 6, 16, 10.
- b) 1, 4, 9, 16, 25.
- c) 2, 8, 18, 32, 50.
- d) 1, 2, 3, 4, 5.

12. Em Java, uma classe declarada com a palavra-chave *final*

- a) não pode ser instanciada.
- b) não pode ser estendida.
- c) possui somente métodos abstratos.
- d) não possui método construtor.

13. Analise as seguintes afirmações sobre classes abstratas no Java:

- I. Gera um erro de compilação instanciar um objeto de uma classe abstrata, a menos que todos os métodos da classe abstrata sejam métodos concretos.
- II. Uma subclasse concreta, derivada de uma superclasse abstrata, deve fornecer implementação para todos os métodos abstratos dessa superclasse.
- III. Uma classe abstrata pode conter métodos não abstratos, ou seja, métodos que possuem implementação.

Estão CORRETAS as afirmativas

- a) I e II apenas.
- b) I e III apenas.
- c) II e III apenas.
- d) I, II e III.

14. Sobre o tratamento de exceções em Java, marque a alternativa que apresenta **erro** de compilação:

<p>a) <code>try { //Código } catch(Exception e) { //Código } finally { //Código }</code></p>	<p>b) <code>try { //Código } catch() { //Código } finally { //Código }</code></p>
<p>c) <code>try { //Código } catch(Exception e) { //Código }</code></p>	<p>d) <code>try { //Código } finally { //Código }</code></p>

15. Sobre o tratamento de exceções em Java, marque a alternativa CORRETA:

- a) Se uma exceção ocorrer em um bloco *try*, este termina imediatamente e o fluxo de execução é transferido para o primeiro bloco *catch*, cujo parâmetro de exceção corresponde ao tipo de exceção lançada.
- b) Todas as classes de exceções herdam de *Exception* e os programadores não podem criar classes próprias para tratar exceções.
- c) Um bloco *finally* não executará somente se o bloco *try* alcançar uma instrução *return*, *break* ou *continue*.
- d) Após uma exceção ser tratada, o fluxo de execução do programa retornará à primeira instrução depois do ponto de lançamento da exceção.

Para responder as questões 16, 17 e 18, considere o trecho de código abaixo.

<pre>package prova; public class Pessoa { String nome; int idade; void exibirDados(){ System.out.println("Nome: " + nome + "\nIdade: " + idade); } }</pre>	<pre>package prova; public class Aluno extends Pessoa { int matricula; double nota1, nota2; double retonarMedia(){ return (nota1+nota2)/2; } double retonarMedia(double f){ return (nota1+nota2)/2 * f; } void exibirDados(){ super.exibirDados(); System.out.println("Matr.: " + matricula + "\nMédia: " + retonarMedia()); } }</pre>
--	--

16. Marque a alternativa CORRETA:

- a) A classe Pessoa é uma especialização da classe Aluno.
- b) A classe Pessoa é superclasse da classe Aluno.
- c) A classe Aluno é superclasse da classe Pessoa.
- d) O relacionamento de herança entre as classes Pessoa e Aluno permite dizer que um objeto pessoa é um tipo de aluno.

17. Marque a alternativa CORRETA:

- a) A instrução *super.exibirDados()* na classe *Aluno* chama o método *exibirDados()* da classe *Pessoa*.
- b) Os atributos nome e idade da classe *Pessoa* não são herdados pela classe *Aluno*.
- c) Um objeto da classe *Pessoa* pode chamar o método *retornarMedia()* da classe *Aluno* de forma direta.
- d) A declaração de dois métodos *retornarMedia()*, ambos com mesma assinatura, gera de erro de compilação na classe *Aluno*.

18. Marque a alternativa CORRETA:

- a) O método *exibirDados()* da classe *Aluno* sobrecarrega o método *exibirDados()* da classe *Pessoa*.
- b) O método *exibirDados()* da classe *Pessoa* sobrecarrega o método *exibirDados()* da classe *Aluno*.
- c) O método *exibirDados()* da classe *Aluno* sobrescreve o método *exibirDados()* da classe *Pessoa*.
- d) O método *exibirDados()* da classe *Pessoa* sobrescreve o método *exibirDados()* da classe *Aluno*.

19. JDBC é um conjunto de classes e *interfaces* para acesso a bancos de dados relacionais no Java. Marque a alternativa que indica um objeto que é utilizado para submeter uma consulta SQL a um banco de dados:

- a) *Statement*
- b) *ResultSet*
- c) *Connection*
- d) *DriverManager*

20. A respeito dos *servlets*, marque a alternativa CORRETA:

- a) A classe *HttpServlet* define métodos *doGet* e *doPut* para responder a solicitações *get* e *put* de um cliente.
- b) Quando o contêiner de *servlet* termina o *servlet*, o método *finalize* do *servlet* é chamado para liberar recursos desse *servlet*.
- c) Os métodos *doGet* e *doPost* recebem como argumento um objeto *HttpServletResponse* e um objeto *HttpServletRequest* que permite interação entre cliente e servidor.
- d) Normalmente os *servlets* são utilizados no lado cliente de um aplicativo de rede.

21. Em PHP, considerando o operador *bitwise* `<<` (conhecido como *shift left*). Observe o código fonte a seguir:

```
<?php
  $x = 100 << 2;
  echo $x;
?>
```

Assinale a alternativa CORRETA que corresponde ao que será impresso na tela após a execução do programa:

- a) 200
- b) 400
- c) 50
- d) 25

22. O código fonte PHP a seguir contém um operador de comparação.

```
<?php
$x = 1;
$y = "1";
if($x === $y)
    echo '$x e $y são iguais.';
else
    echo '$x e $y não são iguais.';
?>
```

Assinale a alternativa CORRETA após a execução do código fonte:

- a) Será impresso na tela a mensagem: \$x e \$y não são iguais.
- b) Será impresso na tela a mensagem: \$x e \$y são iguais.
- c) Será impresso na tela a mensagem: 1 e 1 são iguais.
- d) PHP reconhece apenas o operador ==, sendo que === causará um erro.

23. Analise o código fonte a seguir:

```
<?php
$d = 12;
$e = 0;
$r = "erro de divisão por zero";
#$r = $d / $e;
echo "resultado $r.";
?>
```

Assinale a alternativa CORRETA que corresponde ao que ocorrerá após a execução do programa:

- a) O programa não irá executar até o fim, apresentará uma advertência de divisão por zero e sua execução será abortada.
- b) Será impressa na tela a mensagem: resultado erro de divisão por zero.
- c) Será impressa na tela a mensagem: resultado 0.
- d) Será impressa na tela a mensagem: resultado 12.

24. O uso de sessões é importante quando desejamos manter os dados de *login* de um usuário através das várias páginas de um *web site*, a fim de que o usuário não precise realizar *login* novamente em cada página que visita. Assinale a alternativa CORRETA que corresponde à função que deve ser chamada para criar uma sessão ou restaurar os seus dados:

- a) *serialize()*
- b) *\$_SESSION*
- c) *session_destroy()*
- d) *session_start()*

25. Assinale a alternativa CORRETA que apresenta os vetores superglobais do PHP usados para receber dados de um formulário ou via URL:

- a) \$FORM e \$URL
- b) \$_FORM e \$_URL
- c) \$_GLOBAL e \$_LOCAL
- d) \$_POST e \$_GET

26. Considere as seguintes afirmações sobre os tipos de índice de um vetor (*array*):

- I. Em PHP o índice de um *array* pode ser tanto um valor numérico quanto um *string*.
- II. Um índice do tipo *string* é chamado de chave associativa.
- III. Um *array* pode possuir índices numéricos e *strings* ao mesmo tempo.

Estão CORRETAS as afirmações:

- a) I e II apenas.
- b) I e III apenas.
- c) II e III apenas.
- d) I, II e III.

27. Observe o código fonte PHP a seguir:

```
<?php
for($x=1;$x<10;$x++){
    if($x%2==0)
        break;
    echo (" $x <br />");
}
?>
```

Considere as seguintes afirmações:

- I. A execução do programa vai gerar uma lista de números pares.
- II. A execução do programa irá imprimir o número 1 na tela.
- III. O comando *break* irá interromper o laço na segunda iteração do programa.

Estão CORRETAS as afirmações:

- a) I e II apenas.
- b) I e III apenas.
- c) II e III apenas.
- d) I, II e III.

28. Considere as seguintes afirmações sobre estruturas de controle em PHP:

- I. Em PHP, um bloco de seleção múltipla pode conter os comandos: *switch*, *case*, *break*.
- II. Em PHP, um bloco seleção múltipla pode conter o comando *default*.
- III. Em PHP são comandos de seleção: *if*, *while*.

Estão CORRETAS as afirmações:

- a) I e II apenas.
- b) I e III apenas.
- c) II e III apenas.
- d) I, II e III.

29. Considere as seguintes afirmações sobre acesso ao *PostgreSQL* em PHP:

- I. A forma nativa de estabelecer uma conexão do PHP com bancos de dados *PostgreSQL* é através da função *pg_connect()*.
- II. O uso da função *pg_close()* no final do programa, sempre que estiver aberta uma conexão com *pg_connect()* é obrigatório.
- III. O uso da função *pg_close()* no final do programa, sempre que estiver aberta uma conexão com *pg_connect()* não é obrigatório.

Estão INCORRETAS as afirmações:

- a) I e II apenas.
- b) I e III apenas.
- c) II e III apenas.
- d) I, II e III.

30. Assinale a alternativa CORRETA. No que concerne à manipulação de *cookies* em PHP, a função nativa que grava valores em um *cookie* é:

- a) `$_COOKIE`
- b) `setcookie()`
- c) `set_cookie()`
- d) `putcookie()`

31. Assinale a alternativa CORRETA. Para acessar o valor de um *cookie* utiliza-se:

- a) A função nativa `get_cookie()`
- b) A função nativa `setcookie()`
- c) A função nativa `putcookie()`
- d) O *array* superglobal `$_COOKIE`

32. Assinale a alternativa CORRETA. A função `is_numeric($x)` retornará:

- a) Valor *true* se `$x` contiver um valor não numérico.
- b) Valor *true* se `$x` contiver um valor numérico.
- c) Valor *false* se `$v` contiver um valor numérico.
- d) Valor *false* se `$v` contiver um valor não numérico.

33. Observe o código PHP a seguir:

```
<?php
$p = 12;
$t = "Brasil pentacampeão mundial";
$s = strlen($t);
if($s <= $p)
    echo "S é menor ou igual a P ($s > $p).";
else
    echo "S é maior que P ($s > $p).";
?>
```

Assinale a alternativa CORRETA que corresponde ao que será impresso na tela após a execução do programa:

- a) S é menor ou igual a P (122 > 27).
- b) S é menor ou igual a P (27 > 122).
- c) S é maior que P (27 > 1.2).
- d) S é maior que P (27 > 12).

34. Assinale a alternativa CORRETA. Em PHP podemos substituir caracteres de um *string* através da função:

- a) `strlen()`
- b) `S.replace()`
- c) `str_replace()`
- d) `strreplac()`

35. Em uma folha de estilo (CSS) a definição CORRETA para uma classe é:

- a) `#minhaclasse {}`
- b) `minhaclasse {}`
- c) `;minhaclasse {}`
- d) `.minhaclasse {}`

36. Em uma folha de estilo (CSS) a definição CORRETA para um *id* é:

- a) `+meuid {}`
- b) `#meuid {}`
- c) `=meuid {}`
- d) `.meuid {}`

37. Em uma folha de estilo (CSS) a maneira CORRETA para modificarmos a *tag body* é:

- a) `#body{}`
- b) `.body {}`
- c) `body{}`
- d) `__body{}`

38. Em uma folha de estilo (CSS) são propriedades relativas ao alinhamento de um elemento na página:

- a) *float, clear*
- b) *color, background-color*
- c) *font-family, font-size*
- d) *width, height, border*

39. A maneira CORRETA de ligarmos a folha de estilo externa "prova.css" a um documento HTML é:

- a) `<css href="prova.css" rel="stylesheet" type="text/css" />`
- b) ``
- c) `<link href="prova.css" rel="stylesheet" type="text/css" />`
- d) `<script src="prova.css" rel="stylesheet" type="text/css" />`

40. Assinale a alternativa CORRETA. A respeito de classes e identificadores (*ids*) em uma folha de estilo (CSS):

- a) Não há diferença de uso entre classes e *ids*.
- b) Uma mesma classe pode ser usada em vários objetos do DOM.
- c) Um mesmo *id* pode ser usado em vários objetos do DOM.
- d) Um *id* é o mesmo que uma *tag* HTML.