

41) Analise o código abaixo:

```
...
for (int x = 0; x < 9; ++x)
    switch(x)
    {
        case 3: System.out.print ("G");
        case 4: System.out.print ("U");
            break;
        case 5: System.out.print ("C");
        case 6:
        case 7: System.out.print ("G");
            break;
        case 8: System.out.print ("L");
            break;
        case 9: System.out.print ("A");
    }
...
```

Após ser executado, o trecho de código acima será impresso

- a) GUUCGGGL.
- b) GUUC GLA.
- c) GUCGGL.
- d) GUCGLA.

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA A)

Verifica-se que:

- A alternativa A está correta, visto que na primeira iteração do laço "for", a variável x terá o valor 1 (um), não casará com nenhuma das opções e, portanto, nada será exibido na tela. Na segunda iteração x terá o valor 2 (dois), não casará com nenhuma das opções e nada será exibido na tela. Na terceira iteração, x terá o valor 3 (três) e casará com a opção 3 (três), sendo exibido na tela a letra "G". Como não existe o comando *break* para interromper a execução, será executado o próximo comando, e será exibida a letra "U" e, então, será encontrado o comando *break* que interrompe a execução e faz o laço ir para a próxima iteração. Na quarta iteração, x terá o valor 4 (quatro) que casará com a opção 4 (quatro), será exibida a letra "U" e, então, será encontrado o comando *break* que interrompe a execução e faz o laço ir para a próxima iteração. Na quinta iteração x terá o valor 5 (cinco), casará com a opção 5 (cinco) e será exibida a letra "C" e, como não existe o comando *break* para interromper a execução, será executado o próximo comando, e será exibida a letra "G" e, então, será encontrado o comando *break* que interrompe a execução e faz o laço ir para a próxima iteração. Na sexta iteração, x terá o valor 6 (seis), casará com a opção 6 (seis), mas como não existe comando, nada será feito e irá ser executado o próximo comando, que, neste caso, irá exibir a letra "G" e, então, será encontrado o comando *break* que interrompe a execução e faz o laço ir para a próxima iteração. Na sétima iteração, x terá o valor 7 (sete) e casará com a opção 7 (sete) e será exibida a letra "G" e, então, será encontrado o comando *break* que interrompe a execução e faz o laço ir para a próxima iteração. Na oitava iteração, x terá o valor 8 (oito) e casará com a opção 8 (oito) e será exibida a letra "L", e então será encontrado o comando *break* que interrompe a execução e faz o laço ir para a próxima iteração. A nona iteração não é executada, pois x assume o valor 9 (nove), e como 9 (nove) não é menor que 9 (nove), a condição de execução do laço não é satisfeita, e o *loop* termina. Tem-se, portanto, como resultado final, a exibição da seguinte cadeia de caracteres: GUUCGGGL.
- A alternativa B está incorreta, pois tem um espaço em branco (que não deveria existir), faltam duas letras "G" e tem uma letra "A" (que não deveria existir).
- A alternativa C está incorreta, pois falta uma letra "U" e uma letra "G".
- A alternativa D está incorreta, pois falta uma letra "U", duas letras "G" e tem uma letra "A" (que não deveria existir).

Fonte: BORATTI, Isaias Camilo. **Programação orientada a objetos em Java**. Florianópolis: Visual Books, 2007. 310p. (Capítulo 5 – Desenvolvimento de Programas, páginas 133 e 139).

**42)** Considerando a declaração de variáveis abaixo utilizando a linguagem JAVA, marque a alternativa que apresenta uma atribuição não permitida.

...

```
double D;  
short S;  
float F;  
byte B;  
long L;  
int I;
```

...

- a) L = I; , L = S; , D = F; e S = B;
- b) D = B; , F = S; , I = B; e F = L;
- c) B = S; , L = B; , L = S; e I = S;
- d) D = F; , S = B; , F = I; e F = S;

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA C)

Verifica-se que:

- As alternativas A, B e D estão incorretas, pois não possuem atribuições não permitidas, não podendo ser a opção adequada.
- A alternativa C está correta, visto que o primeiro comando, "B = S;" é uma atribuição não permitida em *Java*, pois B, que é do tipo *byte*, ocupa uma quantidade de *bytes* na memória menor que S, que é do tipo *short*. Ressalta-se que as atribuições permitidas, considerando que a seta (←) representa atribuição, são:  
*double* ← *float* ← *long* ← *int* ← *short* ← *byte*

Fonte: BORATTI, Isaias Camilo. **Programação orientada a objetos em Java**. Florianópolis: Visual Books, 2007. 310p. (Capítulo 4 – Implementação de Classes, página 104).

**43)** Analise a tabela abaixo:

```
CREATE TABLE livro(  
nome VARCHAR(80) NOT NULL,  
genero VARCHAR(80) NOT NULL,  
preco FLOAT,  
vendas BIGINT  
)
```

**Onde:**

- nome: é o nome do livro;
- genero: é o gênero a que pertence o livro. (Ex: romance, ficção etc);
- preco: preço do livro;
- vendas: quantidade de livros vendidos do livro em questão.

Considerando a tabela acima, qual consulta SQL é a correta para selecionar os gêneros de livros e a média de vendas por gênero, dos gêneros que tiveram média de venda superior a 10.000 livros?

- a) SELECT genero, AVG(venda)  
FROM livro  
WHERE AVG(vendas) > 10.000  
GROUP BY genero
- b) SELECT genero, AVG(venda)  
FROM livro  
WHERE AVG(vendas) > 10.000  
HAVING AVG(vendas)
- c) SELECT genero, AVG(vendas)  
FROM livro  
GROUP BY vendas  
HAVING AVG(vendas) > 10.000
- d) SELECT genero, AVG(venda)  
FROM livro  
GROUP BY genero  
HAVING AVG(vendas) > 10.000

#### JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA D)

Verifica-se que:

- A alternativa A está incorreta, pois não possui a cláusula “HAVING” para filtrar o resultado do agrupamento; a cláusula “WHERE” não pode ser utilizada para isto, porque ela não permite realizar restrições com base nos resultados das funções agregadas.
- A alternativa B está incorreta, pois não possui a cláusula “GROUP BY” e, portanto, não tem como realizar o agrupamento por gênero para, posteriormente, aplicar a restrição de vendas superior a 10.000 livros por gênero.
- A alternativa C está incorreta, pois possui um erro de semântica, visto que a cláusula “GROUP BY vendas” estará agrupando por vendas, e não por gênero, como é pedido no enunciado da questão.
- A alternativa D está correta, visto que tem o comando “SELECT”, o campo “genero” e a função agregada “AVG” aplicada no campo “vendas”, e a origem das informações que vem da tabela “livro” especificado na parte do comando “FROM”. Tem-se a cláusula “GROUP BY” aplicada ao “genero” para poder agrupar por gênero, e a cláusula “HAVING” para poder filtrar no resultado do agrupamento apenas os gêneros que tenham média de venda maior que 10.000.

Fontes:

- COSTA, Rogério Luís de C. **SQL guia prático**. 2. ed. Rio de Janeiro: Brasport, 2006. 248p. (Capítulo 4 – Agrupamento de Dados, página 60).
- TABAGHOGHI, M. M. Saied; WILLIAMS, Hugh E. **Aprendendo MySQL**. Rio de Janeiro: Alta Books, 2007. 544 p.

**44)** Analise a tabela abaixo:

```
CREATE TABLE livro(  
  nome VARCHAR(80) NOT NULL,  
  genero VARCHAR(80) NOT NULL,  
  preco FLOAT,  
  vendas BIGINT  
)
```

**Onde:**

- nome: é o nome do livro;
- genero: é o gênero a que pertence o livro. (Ex: romance, ficção etc);
- preco: preço do livro;
- vendas: quantidade de livros vendidos do livro em questão.

Considere a tabela acima e a SQL padrão. O dono da livraria quer uma forma de garantir que não existam dois livros com o mesmo nome e o mesmo gênero simultaneamente, por considerar plágio. Mas pode existir, por exemplo, “Amor eterno”, do gênero drama, e “Amor eterno”, do gênero romance. Uma forma de garantir isso seria usando o comando:

- a) ALTER TABLE livro DROP PRIMARY KEY.
- b) ALTER TABLE livro ADD CONSTRAINT PK\_livro PRIMARY KEY(genero).
- c) ALTER TABLE livro ADD CONSTRAINT PK\_livro INDEX by\_genero(nome).
- d) ALTER TABLE livro ADD CONSTRAINT PK\_livro PRIMARY KEY(nome, genero).**

#### JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA D)

Verifica-se que:

- A alternativa A está incorreta, pois o comando tenta apagar o índice (que não existe) na tabela livro, o que não impede que sejam inseridas duas tuplas com o mesmo “nome” e “genero” na tabela livro.
- A alternativa B está incorreta, pois cria uma chave primária usando apenas o campo “genero”, o que impede apenas que sejam cadastrados gêneros idênticos, mas não impede que sejam inseridas duas tuplas com o mesmo “nome” e “genero” na tabela livro.
- A alternativa C está incorreta, pois cria um índice com o nome “by\_genero” usando o campo nome, o que não impede que sejam inseridas duas tuplas com o mesmo “nome” e “genero” na tabela livro.
- A alternativa D está correta, visto que foi adicionada uma chave primária usando os campos “nome” e “genero”, o que impede que sejam inseridos dois registros com o mesmo “nome” e “genero” na tabela livro.

Fontes:

- COSTA, Rogério Luís de C. **SQL guia prático**. 2. ed. Rio de Janeiro: Brasport, 2006. 248p.
- TABAGHOGHI, M. M. Saied; WILLIAMS, Hugh E. **Aprendendo MySQL**. Rio de Janeiro: Alta Books, 2007. 544p. (Capítulo 6 – Trabalhando com Estrutura de Banco de Dados, páginas 192 e 193).

45) A empresa de segurança X deseja fazer *backup* dos vídeos de segurança de seus clientes em um servidor que está localizado a uma distância de 40 km de sua matriz. Apesar da distância, a empresa X não quer utilizar repetidores de sinal por questão de segurança. Devido ao grande volume de dados, é necessário que a velocidade de transmissão seja superior a 1 Gbps. O local escolhido para armazenar o *backup* é uma região com elevado índice de precipitação e incidência de descargas atmosféricas. Assinale a alternativa que apresenta o meio de transmissão mais indicado para a empresa X fazer o *backup* dos vídeos.

- a) Ondas de rádio.
- b) Fios e cabos de cobre.
- c) Fibras ópticas multimodo.
- d) Fibras ópticas monomodo.

**JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA D)**

Verifica-se que:

- A alternativa A está incorreta, pois as ondas de rádio sofrem influência das precipitações atmosféricas.
- A alternativa B está incorreta, pois os fios de cobre sofrem grande interferência de campos magnéticos e ruídos eletromagnéticos, além de não alcançarem grandes distâncias sem auxílio de repetidores.
- A alternativa C está incorreta, pois, apesar das fibras ópticas multimodo serem imunes a ruídos eletromagnéticos e campos magnéticos, imunes à umidade e atingirem velocidade na ordem de Gbps, elas necessitam utilizar repetidores para alcançar longas distâncias.
- A alternativa D está correta, pois a **fibra óptica monomodo** consegue transmitir sinais a distâncias superiores a 40 km sem a necessidade de utilizar repetidores. Sua velocidade é superior a 1 Gbps, além de ser imune a ruídos eletromagnéticos e campos magnéticos (como os gerados por raios) e à umidade (como a gerada pela chuva).

Fonte: SOUSA, Lindeberg Barros de. **Redes de computadores**: guia total. São Paulo: Érica, 2009. 336p. (Capítulo 1 – Fundamentos de Comunicação de Dados, páginas 44 a 46).

46) Após a execução do código em PHP a seguir, qual será o valor da variável *\$sigma*?

```
$lista = array(1, 4, 6, 7, 3);
$sigma = 0;
foreach ($lista as $item) {
    $sigma += $item;
}
```

- a) 12.
- b) 15.
- c) 18.
- d) 21.

**JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA D)**

O trecho de código em PHP, mostrado na questão, calcula a soma dos números contidos na variável *\$lista*. A seguir, o código comentado:

<code>\$lista = array(1, 4, 6, 7, 3);</code>	Inicia a variável <i>\$lista</i> como um <i>array</i> contendo os números 1, 4, 6, 7 e 3.
<code>\$sigma = 0;</code>	Inicia a variável <i>\$sigma</i> com o valor 0.
<code>foreach (\$lista as \$item){</code>	Início do bloco <i>foreach</i> . <i>Foreach</i> executa os comandos contidos no bloco para cada item da lista.
<code>    \$sigma += \$item;</code>	Soma o valor da variável <i>\$item</i> à variável <i>\$sigma</i> e atribui o resultado à variável <i>\$sigma</i> .
<code>}</code>	Fim do bloco <i>foreach</i> .

Assim sendo, a variável *\$sigma* conterá a soma dos números 1, 4, 6, 7 e 3, cujo resultado é 21.

Fonte: SOARES, Wallace. **PHP5**: conceitos, programação e integração com banco de dados. 6. ed. São Paulo: Érica, 2010. 528 p. (Capítulo 6.4, página 92).

47) Um programador escreveu o seguinte trecho de código:

```
if (1 == 1)
    $valor = '340' + 2;
echo 'Valor: $valor';
```

O resultado desejado pelo programador, ao executar esse código, é que fosse escrita na tela a mensagem:

Valor: 342

Não foi exibida nenhuma mensagem de erro ou aviso, porém, foi mostrado um texto que não correspondeu ao desejado. Selecione a opção que explica porque o resultado foi diferente do esperado.

- a) Falta de chaves após o comando *if*.
- b) Uso do ponto e vírgula após o comando *echo*.
- c) Uso de aspas simples em torno do número 340.
- d) **Uso de aspas simples ao invés de aspas duplas no comando *echo*.**

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA D)

Acerca das assertivas apresentadas, é correto afirmar que:

- **Falta de chaves após o comando *if*.** Não é necessário o uso de chaves em blocos com apenas um comando.
- **Uso do ponto-e-vírgula após o comando *echo*.** O uso de ponto-e-vírgula após o comando é obrigatório.
- **Uso de aspas simples em torno do número 340.** As aspas não interferem na operação visto que, na execução, o valor seria automaticamente convertido para inteiro e efetuada a soma.
- No PHP, o uso de aspas duplas na definição de *strings* permite, entre outras coisas, que o nome de uma variável seja, no momento da execução do código, transformado no valor da mesma. Desta forma, **o uso de aspas simples ao invés de aspas duplas no comando *echo*** faz com que seja escrito o nome da variável em vez de seu valor.

Fonte: SOARES, Wallace. **PHP5: conceitos, programação e integração com banco de dados**. 6. ed. São Paulo: Érica, 2010. 528p. (Capítulo 3.3.2, página 51).

**48)** A respeito da integração do PHP com bancos de dados MySQL, classifique cada assertiva abaixo como verdadeira (V) ou falsa (F) e, em seguida, marque a opção com a sequência correspondente.

- ( ) É possível criar tabelas em um banco de dados MySQL a partir de um script PHP.
- ( ) Um script PHP, em um servidor, pode conectar-se a um banco MySQL em outro servidor.
- ( ) Um script PHP pode conectar-se anonimamente a um banco MySQL, em modo somente-leitura.
- ( ) A possibilidade de conexão com um banco MySQL depende da compatibilidade do navegador do usuário.

- a) V – F – F – V
- b) **V – V – F – F**
- c) F – V – V – F
- d) F – F – V – V

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA B)

É correto afirmar que:

- A primeira assertiva é verdadeira, visto que as diversas bibliotecas de integração entre PHP e MySQL fornecem funções que permitem a criação de tabelas.
- A segunda assertiva é verdadeira, visto que ao conectar-se a um banco, o *script* informa seu endereço, podendo este ser um endereço remoto.
- A terceira assertiva é falsa, pois ao conectar-se a um banco, o *script* deve informar o nome de usuário e senha para conexão.
- A quarta assertiva é falsa, pois a conexão entre um *script* PHP e um banco MySQL não é influenciada pelo navegador do usuário.

Fonte: SOARES, Wallace. **PHP5: conceitos, programação e integração com banco de dados**. 6. ed. São Paulo: Érica, 2010. 528p. (Capítulo 18, página 392).

**49)** No contexto de PHP orientado a objetos, informe se é verdadeiro (V) ou falso (F) o que se afirma abaixo. A seguir, marque a opção com a sequência correta.

- ( ) Uma classe abstrata pode conter métodos concretos (implementados).
- ( ) Os métodos de uma interface devem ser implementados em sua definição.
- ( ) Uma classe com métodos estáticos não pode ser estendida (herdada por outra classe).
- ( ) Um método estático de uma classe não pode acessar as variáveis das instâncias da classe.

- a) V – V – F – F
- b) F – F – V – V
- c) **V – F – F – V**
- d) F – V – V – F

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA C)

É correto afirmar que:

- A primeira assertiva é verdadeira, visto que a classe abstrata tem como característica não poder ser instanciada, ou

seja, não é possível criar objetos de uma classe abstrata. Não há restrição quanto à implementação de métodos numa classe abstrata.

- A segunda assertiva é falsa, pois a característica das interfaces é a impossibilidade de conter métodos concretos. Apenas métodos abstratos (sem implementação definida) podem estar contidos numa interface.
- A terceira assertiva é falsa, pois o tipo de classe que não permite herança (também chamada de extensão) é a classe final. Uma classe com métodos estáticos pode ser estendida, contanto que não seja classe final.
- A quarta assertiva é verdadeira, visto que o método estático é aquele cujo contexto é relativo à classe, e não às suas instâncias. Portanto, o método estático não tem acesso aos atributos de instâncias da classe.

Fonte: SOARES, Wallace. **PHP5: conceitos, programação e integração com banco de dados**. 6. ed. São Paulo: Érica, 2010. 528p. (Capítulo 13, página 288).

**50)** Analise o trecho de código HTML a seguir e marque a opção que contém a forma correta de acessar, num *script* PHP, o valor digitado pelo usuário no campo de texto:

```
<html >
<body>
<form name='registro' method='post' action='login.php' >
<input type='text' name='usuario' >
</form>
</body>
</html>
```

- a) **`$_POST['usuario']`**
- b) `$_FORM('usuario')`
- c) `$_POST['registro']['usuario']`
- d) `$FORM('registro')->usuario.text`

JUSTIFICATIVA DA ALTERNATIVA CORRETA: (LETRA A)

O trecho de código em HTML mostrado na questão cria um formulário, definido com o método POST e contendo um campo de texto de nome 'usuario'. Para acessar, em um *script* PHP, variáveis oriundas de um formulário HTML com método POST, é utilizado o *array* superglobal `$_POST`. O nome do campo do formulário é a chave do item no *array*. Desta forma, para acessar o campo de nome 'usuario' do formulário, usa-se a expressão `$_POST['usuario']`.

Fonte: SOARES, Wallace. **PHP5: conceitos, programação e integração com banco de dados**. 6. ed. São Paulo: Érica, 2010. 528p. (Capítulo 8.3, página 126).