

- 1) Analise o trecho de código PHP abaixo e assinale a alternativa que apresenta o valor que será impresso na tela ao fim da execução:

```
class MenorDeIdadeException extends Exception { }
try {
    $aluno = ['nome' => 'João das Neves', 'idade' => 15];
    if ($aluno['idade'] < 18) {
        throw new MenorDeIdadeException("O aluno {$aluno['nome']} é menor de idade.
Entrar em contato com o(a) responsável.");
    }
    echo 'Inscrição realizada com sucesso.';
} catch (Exception $e) { echo 'Houve um problema na inscrição do aluno.'; }
} catch (MenorDeIdadeException $e) { echo $e->getMessage(); }
```

- a) Inscrição realizada com sucesso.
b) Houve um problema na inscrição do aluno.
c) O aluno João das Neves é menor de idade. Entrar em contato com o(a) responsável.
d) O aluno João das Neves é menor de idade. Entrar em contato com o(a) responsável. Inscrição realizada com sucesso.
e) O aluno {\$aluno['nome']} é menor de idade. Entrar em contato com o(a) responsável.

- 2) Analise o trecho de código PHP abaixo e assinale a alternativa que apresenta o valor que será impresso na tela pelo comando `echo` ao fim da execução:

```
function funcao1(&$val, $chave, $aux) { $val = $aux . ' ' . $val; $index++; }
function funcao2($arr) { array_walk($arr, 'funcao1', ++$index); return $arr; }
$index = 5;
echo json_encode(funcao2(['Brasil', 'RS', 'Rio Grande']));
```

- a) ["1. Brasil","1. RS","1. Rio Grande"]
b) ["1. Brasil","2. RS","3. Rio Grande"]
c) ["6. Brasil","6. RS","6. Rio Grande"]
d) ["6. Brasil","7. RS","8. Rio Grande"]
e) [". Brasil",". RS",". Rio Grande"]

- 3) No que diz respeito aos operadores no PHP, é correto afirmar:

- I) O operador `@`, quando utilizado imediatamente antes de uma expressão, faz com que qualquer mensagem de erro gerada por esta expressão seja ignorada.
II) O operador `!` funciona como um operador lógico de negação, podendo ser utilizado apenas antes de variáveis, expressões booleanas e funções de valor diferente de `NULL`.
III) O operador ternário `??` é um operador condicional, utilizado para estruturas de `if-else` simplificadas: `[condição lógica] ?? [caso true] : [caso false]`

Assinale a alternativa que indica a(s) afirmativa(s) correta(s).

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

- 4) Analise o trecho de código PHP abaixo e assinale a alternativa que apresenta o valor que será impresso na tela ao fim da execução:

```
$var1 = true and false; $var2 = true && false;
$var3 = (true and false); $var4 = (true && false);
$vars = [$var1 or !$var2, $var2 || !$var3, $var3 or !$var4, $var4 || !$var1];
$validos = array_filter($vars, function($var) {return $var === true;});
echo count($validos);
```

- a) 0
- b) 1
- c) 2
- d) 3
- e) 4

- 5) Analise o trecho de código PHP abaixo e assinale a alternativa que melhor representa o valor que será impresso na tela ao fim da execução:

```

trait singleton {
    private static $singleton = null;
    private function __construct() { }
    public static function getInstance() {
        if (self::$singleton === null) { self::$singleton = new self(); }
        return self::$singleton;
    }
}
class A {
    use singleton;
    private function __construct() { echo "__construct class A\n"; }
}
A::getInstance();
    
```

- a) Um *PHP Parse error* informando que não existe a instrução *trait*.
- b) Um *PHP Parse error* informando que não é possível sobrescrever a função *__construct*.
- c) Um *PHP Parse error* informando que a classe deve implementar *singleton*.
- d) Nenhuma saída de tela será gerada.
- e) Uma saída de tela com o texto "*__construct class A*".

- 6) No que diz respeito às interfaces no PHP, é correto afirmar:

- I) Interfaces de objetos permitem a criação de códigos que especificam quais métodos uma classe deve implementar.
- II) Para implementar uma interface, o operador *extends* é utilizado. Ex.: *class MinhaClasse extends MinhaInterface { /* implementação */ }*
- III) Classes podem implementar apenas uma única interface;
- IV) A classe que implementa a interface precisa ter as mesmas assinaturas de método como definidas na interface. Não fazê-lo resultará em um erro fatal.

Assinale a alternativa que indica a(s) afirmativa(s) correta(s).

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

Os arquivos definidos abaixo, de um sistema operacional Ubuntu, se encontram em um diretório de caminho */home/usuario/* e devem ser considerados na solução das questões 7, 8, 9 e 10.:

/home/usuario/src/C.php

```

<?php
class C {
    public function m(string $x) { return $x; }
    public function n(int $x) { return $this->k($x + 3); }
    protected function k($v) { return $this->m(strval($v)); }
}
    
```

/home/usuario/src/D.php

```

<?php
require_once 'C.php';
    
```

```
class D extends C {
    public function k($v) {
        if (is_numeric($v)) { return $this->m(strval($v + 3));
        } else { throw new Exception('Preciso de um número!'); }
    }
}
```

/home/usuario/src/runC.php

```
<?php
require_once 'C.php';
$o = new C();
echo $o->n(3); echo $o->m('3'); echo $o->m('3' . $o->n(3));
```

/home/usuario/src/runD.php

```
<?php
require_once 'D.php';
$l = new D();
echo $l->n(3); echo $l->m('3'); echo $l->m('3' . $l->n(3));
```

/home/usuario/tests/CDTest.php

```
<?php
use PHPUnit\Framework\TestCase;
require_once __DIR__ . '/../src/C.php';
require_once __DIR__ . '/../src/D.php';

final class CDTest extends TestCase {
    public function test1(): void {
        $c = new C(); $d = new D();
        $this->assertEquals($c->m('a'), $d->m('a'));
        $this->assertNotEquals($c->m('a'), $d->m('n'));
        $this->assertFalse($c->m('1') != $d->m('1'));
        $this->assertTrue($d->m('1') == 1);
    }

    public function test2(): void {
        $c = new C(); $d = new D();
        $this->assertTrue($c->n(1) == '4');
        $this->assertEquals($c->n(1), $d->n(1) - 3);
    }

    public function test3(): void {
        $this->expectExceptionMessage('Preciso de um número!');
        $d = new D(); $d->k('a');
    }

    public function test4(): void {
        $this->expectExceptionMessage('Preciso de um número!');
        $d = new D(); $d->k('2');
    }
}
```

7) Ao executar o comando "php src/runC.php" no diretório /home/usuario/, marque a alternativa abaixo que representa a saída obtida:

- a) 336
- b) 339
- c) 369
- d) 6336
- e) 6339

8) Ao executar o comando "php src/runD.php" no diretório /home/usuario/, marque a alternativa abaixo que representa a saída obtida:

- a) 6336
- b) 6339
- c) 9339
- d) 9393
- e) PHP Fatal error: Access level to D::k() must be protected (as in class C) or weaker in /home/usuario/conc/src/D.php on line 5

9) Ao executar o comando "phpunit tests/CDTest.php" no diretório /home/usuario/, marque a alternativa abaixo que é verdadeira em relação aos resultados dos testes executados:

- a) Nenhum teste foi executado.
- b) Os testes test3 e test4 não foram executados.
- c) Os testes test3 e test4 foram executados e ambos falharam.
- d) Apenas os testes test1 e test2 foram executados com sucesso.
- e) Apenas o teste test4 falhou.

10) Utilizando a estrutura de diretórios especificada, a partir do diretório /home/usuario/, marque a alternativa abaixo que mostra quantas linhas possuem ocorrências de chamada ao método "m" nos diretórios src e tests:

- a) find ./ -name *.php | grep '>m' | wc
- b) grep -i '>m(' ./tests --include=*.*
- c) find ./ *php | grep 'm(' | wc -l
- d) sed -f '>m(' ' ' ./*php
- e) grep -R '>m(' ./ --include=*.php | wc -l

11) No que diz respeito aos comandos básicos no sistema operacional Linux, é correto afirmar:

- I) pwd, ls e chmod são comandos utilizados para obter informações sobre arquivos e diretórios do usuário.
- II) tail, cat e less são utilizados para visualizar conteúdos de arquivos.
- III) mkdir e rmdir são, respectivamente, comandos de criação e remoção de diretórios.
- IV) mv, cp e ln são comandos que podem ser utilizados para manipular diretórios e arquivos.

Assinale a alternativa que indica as afirmativas corretas.

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

12) Analise o código PHP do arquivo OrdemTest.php abaixo e assinale a alternativa com a ordem correta das funções executadas pelo PHPUnit no comando "phpunit OrdemTest.php":

```
<?php
use PHPUnit\Framework\TestCase;
final class OrdemTest extends TestCase {
    public function test1(): void {
        $this->assertTrue(1 != 1);
    }
    /**
     * @depends test1
     */
    public function test2(): void { }
    public static function setUpBeforeClass(): void { }
    public static function tearDownAfterClass(): void { }
    protected function setUp(): void { }
    protected function tearDown(): void { }
}
```

- a) setUpBeforeClass(), setUp(), test1(), test2(), tearDown(), tearDownAfterClass()
- b) setUpBeforeClass(), setUp(), test1(), tearDown(), tearDownAfterClass()
- c) setUp(), setUpBeforeClass(), test1(), test2(), tearDownAfterClass(), tearDown()
- d) test1(), test2(), setUpBeforeClass(), tearDownAfterClass(), setUp(), tearDown()
- e) test1(), setUpBeforeClass(), tearDownAfterClass(), setUp(), tearDown()

O código HTML abaixo será utilizado nas questões 13, 14, 15 e 16:

```
<div>
  <input type="text" id="todo-text" />
  <input type="button" id="btn-add" value="Adicionar" />
  <ul id="todo-lista">
    <li class="todo-item">Tarefa 1</li>
    <li class="todo-item">Tarefa 2</li>
    <li class="todo-item">Tarefa 3</li>
  </ul>
  <style>
    #todo-lista li.todo-item.concluido {
      text-decoration: line-through;
    }
  </style>
</div>
```

- 13) O código abaixo foi implementado com o intuito de adicionar o texto do input[id=todo-text] na lista ul[id=todo-lista] quando o input[btn-add] for clicado. Assinale a alternativa que apresenta os comandos que devem substituir, respectivamente, A___, B___ e C___ para que a funcionalidade desejada ocorra:

```
A___('#btn-add').B___('click', function () {
  var texto = A___('#todo-text').value.trim();
  if (texto.length > 0) {
    A___('#todo-lista').C___ += '<li class="todo-item">' + texto + '</li>';
  }
});
```

- a) document.querySelector - click - value
- b) document.querySelector - addEventListener - innerHTML
- c) window.querySelector - addEventListener - innerHTML
- d) window.querySelectorAll - click - html
- e) document.querySelectorAll - click - innerHTML

- 14) Analise o trecho de código CSS abaixo e assinale a alternativa que apresenta o valor da propriedade width de cada elemento :

```
#todo-lista { width: 200px; }
#todo-lista li { width: 100%; margin: 2px 0; padding: 10px; }
```

- a) 180px
- b) 198px
- c) 200px
- d) 204px
- e) 220px

- 15) Analise o trecho de código CSS abaixo e assinale a alternativa que apresenta o valor da propriedade background de cada elemento :

```
ul#todo-lista li.todo-item { background: green; }
ul > li.todo-item { background: yellow; }
div ul#todo-lista li.todo-item { background: blue; }
body > ul#todo-lista li.todo-item { background: salmon; }
```

```
ul li.todo-item { background: red; }
```

- a) green
- b) yellow
- c) blue
- d) salmon
- e) red

16) Uma classe de nome `.concluido` foi criada para riscar o texto de cada item de uma lista. Assinale a alternativa que apresenta o código JavaScript que implementa corretamente a funcionalidade de adicionar ou remover a classe `.concluido` no elemento `` quando clicado, mesmo se um novo item for adicionado:

- a)

```
document.querySelectorAll('#todo-lista>li').addEventListener('click', function (e){
    if (e.srcElement.classList.contains('concluido')){
        e.srcElement.classList.remove('concluido');
    } else { e.srcElement.classList.add('concluido'); } });
```
- b)

```
document.querySelector('#todo-lista>li').addEventListener('click', function (e){
    if (e.srcElement.classList.contains('concluido')){
        e.srcElement.classList.remove('concluido');
    } else { e.srcElement.classList.add('concluido'); } });
```
- c)

```
document.querySelector('#todo-lista').addEventListener('click', function (e){
    if(e.target.classList.contains('todo-item')){
        if (e.target.classList.contains('concluido')){
            e.target.classList.remove('concluido');
        } else { e.target.classList.add('concluido'); }
    } });
```
- d)

```
document.querySelectorAll('#todo-lista>li').forEach(function (el) {
    el.addEventListener('click', function (e){
        if (e.srcElement.classList.contains('concluido')){
            e.srcElement.classList.remove('concluido');
        } else { e.srcElement.classList.add('concluido'); }
    }); });
```
- e)

```
document.querySelectorAll('#todo-lista>li').forEach(function (el) {
    el.addEventListener('click', function (e){
        if (e.srcElement.class.contains('concluido')){
            e.srcElement.class.remove('concluido');
        } else { e.srcElement.class.add('concluido'); }
    }); });
```

17) Analise o trecho de código JavaScript abaixo e assinale a alternativa que apresenta o valor de saída no console:

```
(function (contador) {
    var resultado = ""; resultado += (contador++) + " ";
    setTimeout(function () { resultado += (contador++) + " "; }, 2000);
    setTimeout(function () { resultado += (contador++) + " "; }, 1000);
    console.log(resultado.trim());
})(5);
```

- a) 5
- b) 6
- c) 6 7
- d) 5 6 7
- e) 5 7 6

A estrutura de banco de dados PostgreSQL criada abaixo será utilizada nas questões 18, 19, 20 e 21:

```
CREATE SCHEMA eventos;
CREATE TABLE eventos.pessoas (id serial, nome varchar, cpf varchar);
CREATE TABLE eventos.eventos (id serial, nome varchar, data_abertura DATE);
CREATE TABLE eventos.inscricao (id_evento int, id_pessoa int, data_inscricao DATE);
INSERT INTO eventos.pessoas (id, nome, cpf) VALUES
```

```
(1, 'João', '11111111111'), (2, 'Jaime', '22222222222'), (3, 'Ariana', '33333333333');
```

```
INSERT INTO eventos.eventos (id, nome, data_abertura) VALUES  
(1, 'Evento A', '2019-12-29'), (2, 'Evento B', '2019-12-30');  
INSERT INTO eventos.inscricao (id_evento, id_pessoa, data_inscricao) VALUES  
(1, 1, '2019-12-31'), (2, 2, '2019-12-29'), (3, 1, '2019-12-29');
```

18) Qual das instruções abaixo retorna os registros que têm como data no campo `data_inscricao`, da tabela `inscricao`, o mesmo dia de execução da consulta?

- a) `SELECT * FROM eventos.inscricao WHERE inscricao.data_inscricao = NOW();`
- b) `SELECT * FROM eventos.inscricao WHERE data_inscricao = CURRENT_DATE;`
- c) `SELECT * FROM eventos.inscricao WHERE inscricao.data_inscricao = CURRENT_TIMESTAMP;`
- d) `SELECT * FROM public.inscricao WHERE eventos.inscricao.data_inscricao <= NOW();`
- e) `SELECT * FROM public.inscricao WHERE inscricao.data_inscricao <= CURRENT_DATE;`

19) Qual das instruções abaixo gera um erro quando executada?

- a) `ALTER TABLE eventos.pessoas ADD COLUMN sexo CHARACTER(1) NOT NULL DEFAULT 'M' CHECK (sexo IN ('M', 'F'));`
- b) `ALTER TABLE eventos.eventos ADD COLUMN data_encerramento DATE;`
- c) `COMMENT ON COLUMN eventos.inscricao.data_inscricao IS 'Data de solicitação da inscrição pelo candidato';`
- d) `ALTER TABLE eventos.pessoas ADD CONSTRAINT cpf_tamanho CHECK (CHAR_LENGTH(cpf) = 11);`
- e) `ALTER TABLE eventos.inscricao ADD COLUMN numero int NOT NULL;`

20) Na tentativa de remover valores inválidos do banco de dados, é desejado criar uma chave estrangeira no campo `id_evento` da tabela `inscricao`, referenciando o campo `id` da tabela `eventos`. Qual das instruções abaixo retorna a correta quantidade de inscrições com valores nulos ou inválidos na tabela `inscricao`, que precisam ser corrigidos antes da modificação?

- a) `SELECT COUNT(*) FROM eventos.eventos RIGHT JOIN eventos.inscricao ON id_evento = id WHERE id IS NULL;`
- b) `SELECT COUNT(*) FROM eventos.eventos LEFT JOIN eventos.inscricao ON id_evento = id WHERE id IS NULL;`
- c) `SELECT COUNT(*) FROM eventos.inscricao WHERE id_evento = NULL;`
- d) `SELECT COUNT(*) FROM eventos.inscricao LEFT JOIN eventos.eventos ON id_evento = id WHERE id = NULL;`
- e) `SELECT COUNT(*) FROM eventos.eventos INNER JOIN eventos.inscricao ON id_evento = id WHERE id IS NULL;`

21) As instruções abaixo foram executadas individualmente, na ordem mostrada, na estrutura de banco de dados:

- I) `ALTER TABLE eventos.inscricao ADD CONSTRAINT fkp FOREIGN KEY (id_pessoa) REFERENCES eventos.pessoas(id);`
- II) `UPDATE eventos.inscricao SET id_evento = 1 WHERE id_evento NOT IN (SELECT id FROM eventos.eventos);`
- III) `ALTER TABLE eventos.inscricao ADD CONSTRAINT fke FOREIGN KEY (id_evento) REFERENCES eventos.eventos(id);`
- IV) `ALTER TABLE eventos.inscricao ADD CONSTRAINT inscricao_idx UNIQUE (id_pessoa, id_evento);`

Assinale a alternativa que indica quais instruções retornaram mensagem de erro:

- a) Apenas I e III.
- b) Apenas II e IV.
- c) Apenas I, III, IV.

- d) I, II, III, IV.
- e) Nenhuma instrução retornou erro.

22) Qual o valor de `total`, retornado na execução da instruções abaixo?

```
SELECT SUM(t.count) AS total FROM (  
    SELECT COUNT(i.id_pessoa) FROM eventos.eventos AS e  
    INNER JOIN eventos.inscricao AS i ON i.data_inscricao >= e.data_abertura  
    GROUP BY e.nome  
    ) AS t;
```

- a) 0
- b) 2
- c) 4
- d) 6
- e) 8

23) No que diz respeito aos comandos básicos de gerenciamento de versões utilizando GIT, é correto afirmar:

- I) `git initialize` - Cria um repositório git vazio ou reinicializa um existente.
- II) `git clone <repository>` - Clona um repositório em um novo diretório.
- III) `git insert <path>` - Adiciona o conteúdo do arquivo ao índice.
- IV) `git status` - Mostra o status da árvore de trabalho.
- V) `git commit -m "Mensagem do commit"` - Cria um novo commit contendo o conteúdo atual do índice e a mensagem de registro fornecida.

Assinale a alternativa que indica as afirmativas corretas.

- a) Apenas I e II.
- b) Apenas I e III.
- c) Apenas I, III e IV
- d) Apenas II, IV e V.
- e) I, II, III, IV, V.

24) Sobre a metodologia SCRUM, marque a afirmativa verdadeira:

- a) O Scrum Master trabalha apenas para o Time de Desenvolvimento, treinando, ensinando e liderando a equipe com o objetivo de finalizar o Sprint de trabalho.
- b) O Scrum Master trabalha para o Dono do Produto (*Product Owner*), para o Time de Desenvolvimento e para a Organização. Em cada interação, seu papel é diferente.
- c) É responsabilidade do Dono do Produto (*Product Owner*) ensinar o Time *Scrum* a criar itens de *Backlog* do Produto de forma clara e concisa de acordo com a necessidade de negócio.
- d) A Organização tem o papel de liderar e treinar o Scrum Master e o Time de Desenvolvimento na adoção do Scrum.
- e) O Time de Desenvolvimento tem a responsabilidade de facilitar os Eventos Scrum em ambientes organizacionais nos quais o Scrum não é totalmente adotado e compreendido.

25) No que diz respeito à metodologia Extreming Programming, é correto afirmar:

- I) Integrar as partes do sistemas deve ser feito com a maior frequência possível.
- II) Um desenvolvedor deve ser responsável por uma parte do sistema em particular, para fazer uma melhoria em seu código outra pessoa deve pedir autorização ao responsável.
- III) Metáfora é um artifício que deve ser utilizado somente na conversa com os clientes
- IV) O design de uma funcionalidade deve ser o mais simples possível para passar nos testes criados para ela.
- V) Utilizar padrões de codificação não é recomendado pois interfere na individualidade do programador.

Assinale a alternativa que indica as afirmativas corretas.

- a) Apenas I e IV.

- b) Apenas I e V.
- c) Apenas I, II e IV.
- d) Apenas II, III e IV.
- e) I, II, III, IV, V.