

Nome : Glauber de Souza Faria

Matricula: 17213050160

AD1 - PWEB – 2020.1

QUESTÃO 1

```
<?php
function convertint($string){
    $arrayString = separa($string);
    for ($i = 0; $i < count($arrayString); $i++) {

        $arrayString[$i] = intval($arrayString[$i]);
    }
    #echo var_dump($arrayString);
    return $arrayString;
}
function separa($string){
    $string = explode( " ", $string);
    if (conta($string) == true){
        return $string;
    }
}

function conta($string){
    if (count($string) > 32) {
        return false;
    }
    else{
        return true;
    }
}

function main($string){
    if (calculoDaParidadePar($string) == 0){
        echo 'Paridade Par = Bit de paridade 0';
    }
    else{
        echo 'Paridade Impar = Bit de paridade 1';
    }
}

function calculoDaParidadePar($string){
    $bloco = convertint($string);
    $count = 0;
    foreach ($bloco as $value) {
        if ($value == 1) {
            $count = $count+1;
        }
    }
    if ($count % 2 == 0){
        return 0;
    }
}
```

```
    }  
    else{  
        return 1;  
    }
```

```
}  
#bloco de bits separados por espaços  
$bytes = "1 0 1 0 1 1";  
#Inicia a função main  
$bytes = main($bytes);  
?>
```

QUESTÃO 2

```
<?php
function ocultaEmail($entrada){
    $arrayfrase = explode(" ", $entrada);

    for ($i = 0; $i < count($arrayfrase) ; $i++) {
        if (strpos($arrayfrase[$i], '@')){
            if ($i == count($arrayfrase)-1){
                $arrayfrase[$i] = "xxx.";
            }
            else{
                $arrayfrase[$i] = "xxx,";
            }
        }
        $fraseFinal[] = $arrayfrase[$i] . " ";
    }
    #return $fraseFinal;
    imprime ($fraseFinal);
}

function imprime($entrada){
    $frase = "";
    foreach ($entrada as $value) {
        $frase = $frase . $value;
    }
    $frase = $frase;
    echo $frase;
}

$frase = "Eu, Flávio fseixas@ic.uff.br, gostaria
de solicitar o cadastro de Beltrano beltrano@hotmail.com e
Sicrano sicrano@uol.com.br";

ocultaEmail($frase);
?>
```

QUESTÃO 3

<?php

//CLASSE VIAJANTE

class Viajante{

 //NOME DO VIAJANTE

 private \$nome;

 //DIAS QUE O VIAJANTE PASSARÁ VIAJANDO

 private \$dias;

 //VALOR QUE O VIAJANTE POSUI NA VIAGEM

 private \$valor;

 //CONSTRUCT OBJECT VIAJANTE

 function __construct(\$nome,\$dias,\$valor)

 {

 //RECEBE O PARAMETRO NOME E ARMAZENA NA VARIABEL NOME

 \$this->nome = \$nome;

 //RECEBE O PARAMETRO DIAS E ARMAZENA NA VARIABEL NOME

 \$this->dias = \$dias;

 //RECEBE O PARAMETRO VALOR E ARMAZENA NA VARIABEL NOME

 \$this->valor = \$valor;

 }

 //GET NOME

 function getNome(){

 return \$this->nome;

 }

 //SET NOME

 function setNome(\$param){

```
    $this->nome = $param;  
}
```

```
//GET DIAS
```

```
function getDias(){  
    return $this->dias;  
}
```

```
function setDias($param){  
    $this->dias = $param;  
}
```

```
//GET VALOR
```

```
function getValor(){  
    return $this->valor;  
}
```

```
//SET VALOR
```

```
function setValor($param){  
    $this->valor = $param;  
}
```

```
}
```

```
//CLASE QUE UNE VIAJANTE E ITEMDESPESA
```

```
class Despesas{
```

```
//ARRAY QUE RECEBE OS OBJETOS VIAJANTE
```

```
private $viajante = array();
```

```

//ARRAY QUE RECEBE OS OBJETOS ITEMDESPESA
private $despesa = array();

//CONTADOR PARA TRABALHAR COM ARRAY VIAJANTE
private $countViajante = 0;

//CONTADOR PARA TRABALHAR COM ARRAY ITEMDESPESA
private $countItemDespesa = 0;

//CONSTRUCT OBJECT DESPESAS
function __construct($viajante)
{
    //RECEBE O OBJETO VIAJANTE E ARMAZENA NO ARRAY VIAJANTE
    $this->viajante[$this->countViajante++] = $viajante;
}

//DESTRUCT OBJECT DESPESAS
function __destruct()
{
    $this->gerarRelatorio();
}

//ADICIONA ITEM DESPESA
function adicionar($ItemDespesa){
    $this->despesa[$this->countItemDespesa++] = $ItemDespesa;
}

//gera o relatorio para impress o
function gerarRelatorio(){

    $totalDespesas = 0;
    $saldo = 0;

```

```

$diasFaltam = 0;

for($j=0; $j<= count($this->viajante)-1;$j++){
    for($i=0; $i<= count($this->despesa)-1;$i++){
        $totalDespesas = $this->despesaTotal($i,$totalDespesas);
        $saldo = $this->saldo($j,$totalDespesas);
        $diasRestantes = $this->diasRestantes($i,$j);
    }
    $this->imprimeViajantes($diasRestantes,$saldo);
    $this->imprimeDespesas($totalDespesas);
}

}

//imprime a tabela com viajantes
function imprimeViajantes($diasRestantes,$saldo){
    echo("
    <html>
    <head>
        <title>Resultado</title>
    </head>
    <body border>
        <hr></hr>
        <table border='3'>
            <tr>
                <th>Clientes</th>
                <th>Dias</th>
                <th>Valor</th>
                <th>Diaria Projetada</th>
            </tr>");
    for($j=0; $j<= count($this->viajante)-1;$j++){
        echo(

```

```

        "<tr>

            <td>".$this->viajante[$j]->getNome()."</td>

            <td align='center'>".$this->viajante[$j]->getDias()."</td>

            <td align='center'>".$this->viajante[$j]->getValor()."</td>

            <td align='center'>".round($this-
>diariaProjetada($diasRestantes,$saldo))."/Dia</td>

        </tr>");
    }
    echo("</table>

    <br/>"

    );
}

//imprime na tela a tabela com despesas
function imprimeDespesas($totalDespesas){
    echo("

    <table border='3'>

        <tr>

            <th>Tipo</th>

            <th>Dia</th>

            <th>Valor Gasto</th>

        </tr>");

    for($i=0; $i<=count($this->despesa)-1;$i++){
        echo("<tr>");
        echo("<td>".$this->despesa[$i]->getTipo()."</td>");
        echo("<td align='center'>".$this->despesa[$i]->getnumeroDia()."</td>");
        echo("<td align='center'>".$this->despesa[$i]->getvalor()."</td>");
        echo("</tr>");
    }

    echo("

    <tr>

        <th colspan='2'>Total</th>

```



```

        <th>".$totalDespesas."</th>

    </tr>

</table>

<br/>

</body>

</html>");
}

//Calcula o Saldo
function saldo($j,$totalDespesas){
    return $this->viajante[$j]->getValor() - $totalDespesas;
}

//Calcula o total das despesas
function despesaTotal($i, $totalDespesas){
    $totalDespesas += $this->despesa[$i]->getvalor();
    return $totalDespesas;
}

//calcula os dias restantes
function diasRestantes($i,$j){
    if($i == count($this->despesa)-1){
        return $this->viajante[$j]->getDias() - $this->despesa[$i]->getnumeroDia();
    }
}

//calcula a diariaProjetada
function diariaProjetada($diasRestantes, $saldo){
    return $saldo/$diasRestantes;
}

}

//CLASSE ITEM DESPESA
class ItemDespesa{

    //TIPO DE DESPESA

```

```
private $tipo;

//NUMERO DO DIA QUE A DESPESA FOI GERADA

private $numeroDia;

//VALOR GASTO NESSA DESPESA

private $valor;


//CONSTRUCT OBJECT ITEMDESPESA

function __construct($tipo,$numeroDia,$valor)
{
    //RECEBE O PARAMETRO TIPO E ARMAZENA NA VARIABEL TIPO
    $this->tipo = $tipo;

    //RECEBE O PARAMETRO NUMERODIA E ARMAZENA NA VARIABEL NUMERODIA
    $this->numeroDia = $numeroDia;

    //RECEBE O PARAMETRO VALOR E ARMAZENA NA VARIABEL VALOR
    $this->valor = $valor;
}


//GET TIPO

function getTipo(){
    return $this->tipo;
}


//SET TIPO

function setTipo($param){
    $this->tipo = $param;
}


//GET NUMERODIA

function getnumeroDia(){
    return $this->numeroDia;
```

```

    }

    //SET NUMERODIA
    function setnumeroDia($param){
        $this->numeroDia = $param;
    }

    //GET VALOR
    function getvalor(){
        return $this->valor;
    }

    //SET VALOR
    function setvalor($param){
        $this->valor = $param;
    }

}

//CLASSE ABSTRATA TIPOS DE DESPESA
abstract class TipoDespesa{
    const Deslocamento = "Deslocamento";
    const BilhetesEntrada = "Bilhetes de entrada";
    const Alimentacao = "Alimentação";
    const Compras = "Compras";
    const Outros = "Outros";
}

$viajante = new Viajante("Fulano",20,1000);
$despesas = new Despesas($viajante);
$despesas->adicionar(new ItemDespesa(TipoDespesa::Deslocamento,1,55));

?>

```