


Programação Estruturada

- O que é programação estruturada
- Exemplos de linguagens que utilizam programação estruturada
- Por que PHP?
- o PHP
- Início a programação PHP




O que é programação estruturada?

Programação estruturada é uma forma de programação de computadores que preconiza que todos os programas possíveis podem ser reduzidos a apenas três estruturas: **sequência**, **decisão** e **iteração** (repetição), desenvolvida por Michael A. Jackson no livro "*Principles of Program Design*" de 1975.




O que é programação estruturada

Tendo, na prática, sido transformada na **programação modular**, a programação estruturada orienta os programadores para a criação de estruturas simples nos programas, usando as **sub-rotinas** e as **funções**. Foi a forma dominante na criação de software anterior à programação orientada por objetos.



Exemplos de linguagens que utilizam programação estruturada

```
//COBUCLG JOB CLASS=A,MSGCLASS=A,MSGLEVEL=(1,1)
//HELOWRLD EXEC COBUCLG,PARM.COB='MAP,LIST,LET'
//COB.SYSIN DD *
001 IDENTIFICATION DIVISION.
002 PROGRAM-ID. 'HELLO'.
003 ENVIRONMENT DIVISION.
004 CONFIGURATION SECTION.
005 SOURCE-COMPUTER. IBM-360.
006 OBJECT-COMPUTER. IBM-360.
0065 SPECIAL-NAMES.
0066     CONSOLE IS CNSL.
007 DATA DIVISION.
008 WORKING-STORAGE SECTION.
009 77 HELLO-CONST PIC X(12) VALUE 'HELLO, WORLD'.
075 PROCEDURE DIVISION.
090 000-DISPLAY.
100     DISPLAY HELLO-CONST UPON CNSL.
110     STOP RUN.
//LKED.SYSLIB DD DSN=SYS1.COBLIB,DISP=SHR
//          DD DSN=SYS1.LINKLIB,DISP=SHR
//GO.SYSPRINT DD SYSOUT=A
//
```



Exemplos de linguagens que utilizam programação estruturada


```
# include <stdio.h>

struct Pessoa
{
    char nome[64]; // vetor de 64 chars para o nome
    unsigned short int idade;
    char cpf[13];
};

int main()
{
    // declaração da variável "exemplo"
    struct Pessoa exemplo = {"Fulano", 16, "00.000.000-00"};

    printf("Nome: %s\n", exemplo.nome);
    printf("Idade: %hu\n", exemplo.idade);
    printf("CPF: %s\n", exemplo.cpf);

    getchar(); // desnecessário, mas comumente utilizado em ambientes windows para
    "segurar" o terminal aberto
    return 0;
}
```



Exemplos de linguagens que utilizam programação estruturada

```
<?php
```

```
$vet01 = array();  
$vet01[] = "Sistemas operacionais";  
$vet01[] = "Compiladores";  
$vet01[] = "Bancos de dados";  
  
$vet02 = array(1, 2, 3, 4, 5);  
  
$vet03 = array( 0 => 0, 2 => 3, 10 => "item 10");  
  
for ($i = 0; $i < count($vet01); $i++) {  
    echo $vet01[$i] . "<br />";  
}
```

```
?>
```



Por que PHP?

- É uma linguagem de programação baseada em software livre, sem custos para o usuário.
- É o padrão de linguagem de **programação web** definida pela Prefeitura de Juiz de Fora.
- Possui uma comunidade de programadores cada vez maior.



O PHP

- PHP quer dizer **PHP: Hypertext Preprocessor** (PHP: Processador de Hipertexto).
- Esta linguagem nasceu pela mão de Rasmus Lerdof em 1994, como um CGI escrito em Linguagem C que inicialmente interpretava muito facilmente formulários.



O PHP

- A primeira designação dada foi de FI (Form Interpreter) porém, devido à criação de inúmeras funções pela comunidade, que se ia desenvolvendo pela internet, a linguagem teve que, em 1997, ser redenominada como PHP.



O PHP

- O PHP é uma das mais abrangentes ferramentas que o homem possui atualmente.
- Por ser uma linguagem **server-side**, ou seja roda direto do servidor e só mostra ao usuário o resultado já processado.



O PHP

- É possível a criação de uma grande variedade de coisas com o PHP, tendo em vista que esta é uma linguagem que podemos definir como inteligente, pois se não existir uma classe para uma determinada função podemos criá-la.



O PHP

- Porém, ferramentas que rodam do lado do cliente como a abertura de um pop-up, uma animação de serpentinas exibidas na tela ou um slideshow de imagens não podem ser criados por esta linguagem.



O PHP

- Se procura uma boa linguagem para criar esse tipo de ferramentas procure o **Javascript**.
- É por isso que se diz que o PHP e o Javascript são linguagens que se complementam!



O PHP

- Primeiro contato:

<http://172.20.22.210/aluno/info.php>



Início a programação PHP

- Olá mundo

```
<?php
```


```
echo "Olá mundo";
```

```
?>
```



Início a programação PHP

```
<html>
  <head>
    <title>Meu primeiro script</title>
  </head>
  <body>
    <?php
      echo "Olá mundo";
    ?>
  </body>
</html>
```



Início a programação PHP

- O uso do ponto-e-virgula

```
<?php
```

```
    echo "Olá tudo bem?";
```

```
    echo "Como você está?";
```

```
?>
```

- Não é necessário escrever em linhas diferentes os comandos, mas é recomendável para evitar confusões

```
<?php
```

```
    echo "Olá tudo bem?"; echo "Como você está?";
```

```
?>
```



Início a programação PHP

- Como o PHP é baseado no C e no C++, ele suporta a sintaxe de comentários das duas linguagens, veja abaixo.
- Com `//` comentário:

```
<?php
```

```
echo "Um comentário!"; //Comentário de um linha só
```

```
?>
```



Início a programação PHP

- Com `/*` comentário `*/`:

```
<?php
```

```
/* Isto é um comentário
```

```
de várias linhas no PHP */
```

```
echo "Outro comentário acima!";
```


```
?>
```



Início a programação PHP

- As strings passadas para a instrução echo também podem conter formatações de texto em HTML

```
<?php
echo "<h2> Título em h2 </h2>";
echo "<h3> título em h3 </h3>";
echo "<i>Em itálico</i>";
echo "<b>Em negrito</b>";
echo "<strong>Em negrito</strong>";
?>
```



Início a programação PHP

- As aspas duplas " podem sempre ser substituídas por apóstrofes/aspas simples '. Ambas as formas estão corretas.

```
<?php
```

```
echo "Na frase a seguir o nome xpto virá dentro de  
aspas: ";
```

```
echo "O personagem que eu mais gosto é o \"xpto\",  
sem dúvida";
```

```
?>
```



Início a programação PHP

- Inclusão de trechos de código

```
<?php
```

```
include ('code.php'); // Inclui e executa um  
trecho opcional de código
```

```
include 'code.php'; // Maneira alternativa,  
funciona apenas com include e require.
```

```
require ('code.php'); // O mesmo que 'include',  
porém pára a execução caso o arquivo não seja  
encontrado
```

```
require_once ('code.php'); // O mesmo que require,  
mas evita que o trecho seja incluído novamente
```

```
?>
```



Início a programação PHP

- Exemplo de uso do if e else:

```
$x=3;
```

```
if ($x==2) {
```

```
    echo "x vale 2";
```

```
} else if ($x==3) {
```

```
    echo "x vale 3";
```

```
} else {
```

```
    echo "x é diferente de 2 e de 3";
```

```
}
```



Início a programação PHP

- Exemplo de uso do switch:

```
switch ($i) {  
    case 0:  
        echo "i equals 0";  
        break;  
    case 1:  
        echo "i equals 1";  
        break;  
    case 2:  
        echo "i equals 2";  
        break;  
}
```

```
switch ($i) {  
    case "apple":  
        echo "i is apple";  
        break;  
    case "bar":  
        echo "i is bar";  
        break;  
    case "cake":  
        echo "i is cake";  
        break;  
}
```


Início a programação PHP

- Exemplo do uso do for:

```
/* exemplo 1 */
```

```
$i = 1;  
for ( ; ; ) {  
    if ($i > 10) {  
        break;  
    }  
    echo $i;  
    $i++;  
}
```

```
/* exemplo 2 */
```

```
for ($i = 1, $j = 0; $i <= 10;  
    $j += $i, print $i, $i++);
```

```
/* exemplo 3 */
```

```
for ($i = 1; $i <= 10; $i++) {  
    echo $i;  
}
```

```
/* exemplo 4 */
```

```
for ($i = 1; ; $i++) {  
    if ($i > 10) {  
        break;  
    }  
    echo $i;  
}
```

Início a programação PHP

- Exemplo do uso do while:

```
/* example 1 */
```

```
$i = 1;
```

```
while ($i <= 10) {
```

```
    echo $i++;
```

```
/* the printed value would be
```

```
$i before the increment
```

```
(post-increment) */
```

```
}
```

```
/* example 2 */
```

```
$i = 1;
```

```
while ($i <= 10):
```

```
    echo $i;
```


```
    $i++;
```

```
endwhile;
```



Início a programação PHP

- **Variáveis:**

- Começam sempre pelo símbolo \$ seguido de uma letra.
 - Podem conter símbolos numéricos (0 - 9) alfanuméricos minúsculos (a - z) e alfanuméricos maiúsculos.
 - Não podem conter espaços! Se tiverem mais do que uma palavra deverão ser interligadas por um underscore _.
 - Por exemplo: \$total_variavel.
- 

Início a programação PHP

- Alguns exemplos de variáveis:

```
<?php
$minha_variavel=4;
$minha_string="super ";
echo "$minha_string "." $minha_variavel";
?>
```

- O ponto . após a variável \$minha_string concatena as variáveis.
- Deverá escrever na tela do navegador: super 4.

Início a programação PHP

- As variáveis são no PHP representadas por um \$ seguido do nome a ela atribuído. É necessário ter em conta alguns cuidados quando criamos variáveis:

```
<?php
```

```
$var = "Bom ";
```

```
$Var = "dia!";
```

```
echo $var . $Var; // Exibe "Bom dia!"
```

```
$25arobas = "Ainda por cumprir!";
```

```
// INVÁLIDO - Nenhum nome de variável pode  
// começar por uma expressão numérica
```

```
$_25arobas = "Quase na linha!"; // VÁLIDO - Nome começa por _
```


```
?>
```



Início a programação PHP

- A inclusão de variáveis funciona de várias maneiras.

```
<?php
$carro = "Mercedes";
echo "Ele comprou um bonito $carro";           // funciona
echo "Eles compraram vários {$carro}s";       // funciona
echo "Eu faço um ${carro}";                   // funciona
?>
```



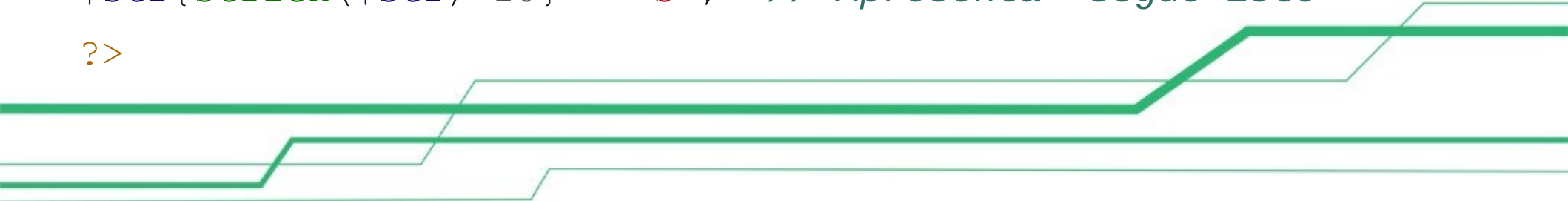
Início a programação PHP

- Podemos também modificar Strings.

```
<?php
$str = "Olha que alí há mal";
// Apresenta "Olha que alí há mal"
$str{strlen($str)-1} = "r";
// Apresenta "Olha que alí há mar"
?>
```

- ou

```
<?php
$str = "Pegue isto"; // Apresenta "Pegue isto"
$str{strlen($str)-10} = "S"; // Apresenta "Segue isto"
?>
```



Início a programação PHP

- **Manipulação de Tipos**
- No PHP não é necessário definir o tipo que queremos usar, ou seja, este é determinado pelo contexto em que é usado. Por exemplo, `$var = "string"` é uma string, já `$var = 12`, é um inteiro.



Início a programação PHP

- Para alternarmos entre os tipos usamos uma sintaxe de moldagens:

```
<?php
```

```
$foo = 5;
```

```
// $foo é um inteiro
```


```
$bar = (boolean) $foo;
```

```
// $bar é um boleano
```

```
?>
```



Início a programação PHP

- **Moldagens permitidas:**
 - *int* ou *integer*: moldar para inteiro.
 - *bool* ou *boolean*: moldar para booleano.
 - *float*, *double* ou *real*: moldar para número de ponto flutuante.
 - *string*: moldar para string.
 - *array*: moldar para array.
 - *object*: moldar para objeto.
- 


Início a programação PHP

- Exemplos de moldes:

```
<?php
$str = "Eu sou string";
$int = 12;
$num = 25/85;

$a = (boolean) $str;
$b = (string) $int;
$c = (int) $num;

echo $a . $b . $c;
?>
```



Início a programação PHP

- Arrays, exemplo:

```
<?php
$arr = array(1 => "um", 2 => "dois", 3 => TRUE);
echo $arr[1]; // Imprime "um"
echo $arr[3]; // Imprime "TRUE"
?>
```

- Aqui nos é apresentado 3 chaves (1, 2 e 3) e 3 valores ("um", "dois" e TRUE).
- Então podemos pegar uma chave de um array para imprimirmos o valor correspondente. Podemos também definir um array como vários array:

Início a programação PHP


```
<?php
$arr = array("versão" => array ("beta"    => "0.5x",
                                "alpha"   => "0.1x",
                                "release" => "0.8x",
                                "final"   => "1.x"
                                ),
            "nome" => "PHP software",
            "SO" => array("win" => "Windows",
                        "lin" => "Linux",
                        "mac" => "MacOS"
                        )
            );

// Vamos agora remover um elemento do array, visto que o nosso
// software já passou a fase alpha
unset($arr["versão"]["alpha"]);
// Também podemos apagar o array inteiro. Já não vamos
// disponibilizar o nosso software
unset($arr);
?>
```

Início a programação PHP

- Podemos especificar apenas valores num array, sendo que o interpretador irá tornar cada um dos valores com uma chave a partir do zero.

```
<?php
$arr = array(1, 45, 23, 68);
echo $arr[1]; // imprime 45
// Vamos agora apagar um valor e reindexar o nosso array:
unset($arr[1]);
$arr = array_values($arr);
// Não podemos imprimir o nosso array através de echo. Temos
// que fazer isso com print_r():
print_r($arr);
?>
```



Início a programação PHP

Nome	Exemplo	Significado
Igual	$\$a == \b	Verdadeiro se $\$a$ é igual a $\$b$
Idêntico	$\$a === \b	Verdadeiro se $\$a$ igual a $\$b$ e do mesmo tipo
Diferente	$\$a != \b	Verdadeiro se $\$a$ diferente de $\$b$
Diferente	$\$a <> \b	Verdadeiro se $\$a$ diferente de $\$b$
Não idêntico	$\$a !== \b	Verdadeiro se $\$a$ diferente de $\$b$, ou não são do mesmo tipo
Menor que	$\$a < \b	Verdadeiro se $\$a$ menor que $\$b$
Maior que	$\$a > \b	Verdadeiro se $\$a$ maior que $\$b$
Menor ou igual	$\$a <= \b	Verdadeiro se $\$a$ menor ou igual a $\$b$
Maior ou igual	$\$a >= \b	Verdadeiro se $\$a$ maior ou igual a $\$b$

Início a programação PHP

Exercícios:

<http://172.20.22.210/aluno/exe.pdf>



Início a programação PHP

Referências:

http://php.net/manual/pt_BR/

