

PROGRAMARE WEB

CURS 8- 20.04.2021

Titular: Șef. Lucr. Dr. Mat. Cărbureanu Mădălina

Copyright@Departamentul de Automatică, Calculatoare și Electronică

Universitatea Petrol-Gaze din Ploiești

LIMBAJUL PHP

- Instrucțiuni de control;
- Crearea și parcurgerea vectorilor;
- Sortarea vectorilor;
- Aplicații propuse.

INSTRUCȚIUNI DE CONTROL

Valoare, tip
nume_variabila?

- Instrucțiuni de atribuire \longrightarrow $\$nume_variabilă=expresie;$
- Instrucțiuni apeluri de funcție \longrightarrow $nume_funcție (listă\ de\ parametrii\ efectivi);$
- Instrucțiunea vidă; • •
- Instrucțiuni de decizie;
- Instrucțiuni iterative;
- Instrucțiuni de salt.

INSTRUCȚIUNI DE CONTROL

- *Instrucțiuni de decizie:*

- instrucțiunea *if* (if cu ramură vidă);
- instrucțiunea *if_else*;
- instrucțiunea *if_else_if* (scara *if_else_if*);
- instrucțiunea *switch* (decizia generalizată);
- operatorul condițional (*?:*).

Instrucțiuni de decizie

- **Instrucțiunea if (if cu ramură vidă)**

- *if (condiție) {instrucțiuni;}*

- **Instrucțiunea if_else**

- *if (condiție) { instrucțiune 1;}*
else {instrucțiune 2;}

- **Instrucțiunea if_else_if (scara if_else_if);**

- if (condiție 1)*

- instrucțiune 1;*

- else if (condiție 2)*

- instrucțiune 2;*

-*

- else if (condiție n)*

- instrucțiune n;*

- else instrucțiune n+1;*



Mod de lucru?

Instrucțiuni de decizie

- *Instrucțiunea switch (decizia generalizată)*

switch (expresie_selector)

{ case val_1: instrucțiuni_1; break;

case val_2: instrucțiuni_2; break; .

.....

case val_n: instrucțiuni_n; break;

default: instrucțiuni n+1; break;

}

Mod de lucru?

Instrucțiuni de decizie

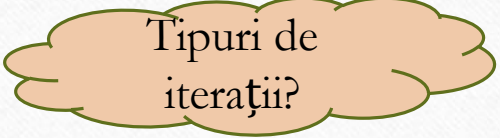
Operatorul condițional (?:)

- *expresie_1?expresie_2:expresie_3;* •

Mod de lucru?

Instrucțiuni iterative

- Instrucțiunea *while*;
- Instrucțiunea *for* ;
- Instrucțiunea *do...while*.



Tipuri de
iterații?

Instrucțiuni iterative

- Instrucțiunea *while*

while (condiție)

{ instrucțiuni; }

- Instrucțiunea *for* •

for (inițializare; condiție; actualizare)

{ instrucțiuni; }

- Instrucțiunea *do...while*

do { instrucțiuni; }

while (condiție);

for caz
particular al
instr.???

Instrucțiuni de salt

- break;
- continue.

Mod de lucru?

CREAREA ȘI PARCURGerea VECTORILOR

- Crearea unui *vector unidimensional/multidimensional* → construcția *array* ;
- Sintaxa pentru crearea unui *vector unidimensional* prin setarea explicită a valorilor:

\$nume_vector[indice_0]=valoare_1;

.....

\$nume_vector[indice_n-1]=valoare_n;


CREAREA ȘI PARCURGerea VECTORILOR

Observații:

- în limbajul PHP, *vectorii unidimensionali* sunt *vectori* simpli *indexați* sau *asociativi*;
- dacă în sintaxa pentru crearea unui *vector unidimensional* prezentată anterior, *indice* este de tip numeric atunci vectorul este unul *indexat*, iar dacă *indice* este de tip șir de caractere atunci vectorul este unul *asociativ*;
- accesul la unul dintre elementele unui vector unidimensional:

\$nume_vector[indice_i], unde indice_ i=0, ..., n-1 sau indice_i="șir_caractere";

CREAREA ȘI PARCURGEREA VECTORILOR

- Vectorii multidimensionali vectori  simpli cu mai multe dimensiuni;
- Sintaxa pentru crearea unui *vector multidimensional*:

```
$nume_vector= array (array (element_1,..., element_n),...,array(element_1,..., element_n));
```

sau

```
$nume_vector=array(componentă_1=>array("nume_cheie_1"=>"val_1",...,"nume_cheie_n"=> "val_n"),
```

.....

```
componentă_n=>array("nume_cheie_1"=>"val_1",...,"nume_cheie_n"=> "val_n"));
```

CREAREA ȘI PARCURGerea VECTORILOR

Observații:

- un vector (un *array*) ce conține alți vectori (alți *arrays*) este un vector bidimensional, iar dacă acești vectori conțin la rândul lor alți vectori, atunci vectorul este unul treidimensional și așa mai departe;
- *element_1, element_2, ..., element_n* sunt valori numerice;
- "*nume_cheie_1*",..., "*nume_cheie_n*" cât și "*val_1*", "*val_2*",..., "*val_n*", sunt șiruri de caractere (*strings*);
- accesul la unul dintre elementele unui vector multidimensional se realizează conform sintaxei:

*echo \$nume_vector[componentă_i]["nume_cheie_i"], "
", unde $i=0, \dots, n-1$.*

CREAREA ȘI PARCURGEREA VECTORILOR

- Parcurgerea *vectorilor unidimensionali* și a celor *multidimensionali* se poate realiza astfel:
 - folosind structurile iterative *for*, *while* sau *do-while*;
 - folosind funcțiile *list()*, *each()* și *foreach()*;
- Cea mai simplă formă de parcurgere este atunci când indicii vectorului sunt de tip întreg (*integer*), cât și consecutivi.

CREAREA ȘI PARCURGEREA VECTORILOR

Parcurgerea
cu while și
do-while
este
evidentă!

Parcurgerea cu for a unui *vector* *unidimensional* ($\text{strlen}(\$x)=5$)

```
$n=5;  
for($i=1;$i<=$n;$i++)  
    $x[$i]=$i;  
echo "Vectorul unidimensional este:"."<br>”;  
for($i=1;$i<=$n;$i++)  
    echo $x[$i], " ”;
```

Parcurgerea cu for a unui *vector* *multidimensional* $a(3\times 3)$

```
$n=3; $m=3; $k=1;  
for($i=1;$i<=$n;$i++)  
    for($j=1;$j<=$m;$j++)  
        $a[$i][$j]=$k++;  
echo "Vectorul multidimensional este:"."<br>”;  
for($i=1;$i<=$n;$i++)  
    { for($j=1;$j<=$m;$j++)  
        echo $a[$i][$j], " ”;  
        echo ”<br>”; }
```

CREAREA ȘI PARCURGEREA VECTORILOR

- Parcurgerea folosind funcțiile *list()*, *each()* și *foreach()*:
 - funcția *list()* permite crearea unei liste de variabile inițializate cu valorile unui vector (vectorul unidimensional $\$x$) și are efect numai asupra vectorilor indexați numeric (cu indicele de start $\$i$ egal cu 0);
 - funcția *each()* aplicată unui vector (unidimensional $\$y$, unde $\text{strlen}(\$y)=5$) returnează valoarea curentă a indicelui (a cheii), precum și valoarea memorată în componenta indicată de cheie;
 - funcția *foreach()* reprezintă o modalitate elegantă de a parcurge un vector (vectorul unidimensional $\$x$) indiferent de natura indicilor săi.

CREAREA ȘI PARCURGEREA VECTORILOR

| Parcurgere cu <i>list()</i> | Parcurgere cu <i>each()</i> | Parcurgere cu <i>foreach()</i> |
|--|---|---|
| <pre><i>\$n=3;</i> <i>for(\$i=0;\$i<=\$n-1;\$i++)</i> <i>\$x[\$i]=\$i;</i> <i>echo "Vectorul unidimensional</i> <i>este:", " ";</i> <i>for(\$i=0;\$i<=\$n-1;\$i++)</i> <i>echo \$x[\$i], " ";</i> <i>echo "Exemplu de lucru cu funcția</i> <i>list()", " ";</i> <i>list (\$a,\$b,\$c,\$d)=\$x;</i> <i>echo "Val a este:", \$a, " ";</i> <i>echo "Val b este:", \$b, " ";</i> <i>echo "Val c este:", \$c, " ";</i> <i>echo "Val d este:", \$d, " ";</i></pre> | <pre><i>echo "Exemplu de lucru cu funcția</i> <i>each()", " ";</i> <i>\$y=array(10,2,3,4,5);</i> <i>reset(\$y);</i> <i>\$sol=each(\$y);</i> <i>echo "cheia este \$sol[0], iar</i> <i>valoarea memorată este \$sol[1]";</i> <i>echo " ";</i></pre> | <pre><i>echo "exemplu de lucru cu funcția</i> <i>foreach()", " ";</i> <i>foreach(\$x as \$w)</i> <i>{ echo \$w;}</i></pre> |

Sortarea vectorilor

- Funcții predefinite:

- funcțiile *sort* (\$vector) și *rsort* (\$vector): realizează sortarea crescătoare (funcția *sort* (\$vector)), respectiv descrescătoare (funcția *rsort*(\$vector)) a valorilor elementelor vectorului unidimensional dat;
 - dezavantajul utilizării acestor funcții este dat de faptul că valorile inițiale ale indicilor (ale cheilor) se pierd, acestea fiind înlocuite cu indici numerici (valori cuprinse între 0 și n-1);
- funcțiile *asort*(\$vector) și *arsort*(\$vector): realizează sortarea crescătoare (funcția *asort*(\$vector)), respectiv descrescătoare (funcția *arsort*(\$vector)) a elementelor unui vector unidimensional;
 - avantajul utilizării acestor funcții este dat de faptul că valorile inițiale ale indicilor (cheilor) asociate valorilor vectorului sunt păstrate, spre deosebire de situația în care sunt folosite funcțiile *sort* (\$vector) și *rsort* (\$vector);

Sortarea vectorilor

- funcțiile *ksort(\$vector)* și *krsort(\$vector)*: realizează sortarea crescătoare (funcția *ksort(\$vector)*), respectiv descrescătoare (funcția *krsort(\$vector)*) a vectorului unidimensional după chei și nu după valorile acestuia, păstrând corelarea cheilor cu valorile vectorului;
- funcțiile *natsort(\$vector)* și *natcasesort(\$vector)*: realizează sortarea naturală (subșirurile de caractere formate din cifre sunt comparate ca niște numere) a elementelor unui vector unidimensional după valorile elementelor și nu după chei, cu păstrarea corelării existente între chei și valorilor vectorului;
- funcția *array_multisort(\$vector_1, \$vector_2, ..., \$vector_n)*: permite realizarea sortării unui vector multidimensional; în cazul sortării unui vector bidimensional, a sorta un astfel de vector este echivalent cu a sorta simultan mai mulți vectori unidimensionali care compun coloanele vectorului bidimensional.

Sortarea vectorilor

sort (\$vector) și rsort (\$vector)

```
<?php
echo "Script pentru sortarea unui vector folosind funcțiile
sort() și rsort()";
$materii=array("POO"=>"programare orientată pe
obiecte", "PLF"=>"programare logică și funcțională",
"LA"=>"inteligență artificială");
    foreach($materii as $indice=>$vector)
        {echo "<br>$indice=>$vector";}
    sort($materii);
echo "<br>";
echo "vectorul dupa sortarea crescătoare:", "<br>";
    foreach($materii as $indice=>$vector)
        {echo $indice, "=>", $vector, "<br>";}
    rsort($materii); echo "<br>";
echo "vectorul dupa sortarea descrescătoare:", "<br>";
    foreach($materii as $indice=>$vector)
        {echo $indice, "=>", $vector, "<br>";}?>
```

asort(\$vector) și arsort(\$vector)

```
<?php
$materii=array("POO"=>"programare orientată pe
obiecte", "PLF"=>"programare logică și funcțională",
"LA"=>"inteligență artificială");
    foreach($materii as $indice=>$vector)
        {echo "<br>$indice=>$vector";}
    asort($materii);
echo "<br>";
echo "vectorul dupa sortare crescătoare:", "<br>";
    foreach($materii as $indice=>$vector)
        {echo $indice, "=>", $vector, "<br>";}
    arsort($materii);
echo "<br>";
echo "vectorul dupa sortare descrescătoare:", "<br>";
    foreach($materii as $indice=>$vector)
        {echo $indice, "=>", $vector, "<br>";}?>
```

Sortarea vectorilor

*k*sort(\$vector) și *k*rsort(\$vector)

```
<?php
$materii=array("3"=>"POO", "2"=>"PLF",
"1"=>"LA");
    foreach($materii as $indice=>$vector)
        {echo "<br>$indice=>$vector";}
    ksort($materii);
echo "<br>";
echo "vectorul dupa sortarea crescătoare:", "<br>";
    foreach($materii as $indice=>$vector)
        {echo $indice, "=>", $vector, "<br>";}
    krsort($materii);
echo "<br>";
echo "vectorul dupa sortarea descrescătoare:", "<br>";
    foreach($materii as $indice=>$vector)
        {echo $indice, "=>", $vector, "<br>";}
?>
```

natsort(\$vector) și *natcasesort*(\$vector)

```
<?php
$materii=array("x22"=>"POO", "x21"=>"plf",
"x20"=>"LA");
    foreach($materii as $indice=>$vector)
        {echo "<br>$indice=>$vector";}
    natsort($materii);
echo "<br>";
echo "vectorul dupa sortare crescătoare cu natsort:",
"<br>";
    foreach($materii as $indice=>$vector)
        {echo $indice, "=>", $vector, "<br>";}
    natcasesort($materii);
echo "<br>";
echo "vectorul dupa sortare crescătoare cu natcasesort:",
"<br>";
    foreach($materii as $indice=>$vector)
        {echo $indice, "=>", $vector, "<br>";}?>
```

Sortarea vectorilor

array_multisort(\$vector_1, \$vector_2, ..., \$vector_n)

```
<?php
$vector1=array(6,5,4);
$vector2=array(3,2,1);
$vector3=array(9,7,8);
echo "<br>";
echo "vector1 este:";
    foreach ($vector1 as $valoare1)
        {echo "$valoare1 ";}
echo "<br>"; echo "vector2 este:";
    foreach ($vector2 as $valoare2)
        {echo "$valoare2";}
echo "<br>"; echo "vector3 este:";
    foreach ($vector3 as $valoare3)
        {echo "$valoare3";}
echo "<br>";
    array_multisort($vector1,$vector2,$vector3);
```

```
echo "afisarea vectorilor dupa sortare cu array_multisort:";
    foreach ($vector1 as $valoare1)
        {echo "$valoare1 ";}
echo "<br>"; echo "vector2 este:";
    foreach ($vector2 as $valoare2)
        {echo "$valoare2";}
echo "<br>";
echo "vector3 este:";
    foreach ($vector3 as $valoare3)
        {echo "$valoare3";}
echo "<br>";
?>
```

Aplicații propuse

- Testarea aplicațiilor prezentate în cadrul cursului;
- Testarea aplicației 2.2.6, pag. 91;
- Rezolvarea aplicației nr. 3, pag. 101;
- Rezolvarea aplicației nr. 3, pag. 102;
- **Obs:** Se va utiliza cartea “*Programare Web. Ghid teoretic și practic*”, Șef lucr. dr. mat. Cărbureanu Mădălina, Editura Universității Petrol-Gaze din Ploiești, 2020.

Spor la lucru!