

PROGRAMARE WEB

CURSURILE 10+11- 11.05.2021

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

Copyright@Departamentul de Automatică, Calculatoare și Electronică

Universitatea Petrol-Gaze din Ploiești

OBJECTIVE:

- CLASE ȘI OBIECTE ÎN PHP;
- EXEMPLU;
- MOȘTENIREA ÎN PHP;
- EXEMPLU;
- ELEMENTE DE JAVASCRIPT :
 - CARACTERISTICI;
 - TERMINOLOGIE;
 - ELEMENTE DE BAZĂ;
 - INSTRUCȚIUNI DE CONTROL.
- APLICAȚII PROPUSE.

CLASE ȘI OBIECTE ÎN PHP

??? (C++)

- Concepte de bază: clasă, obiect, moștenire;
- În limbajul PHP, sintaxa pentru declararea unei clase este următoarea:

```
<?php
class nume_clasă
    { //date membru (atribute)
        var $nume_var_1;
        .....
        var $nume_var_n;
    //metode (funcții membru)
    function nume_functie_1 (listă parametrii formali)
    { //definiția funcției}
    .....
    function nume_functie_n (listă parametrii formali)
    { //definiția funcției}
    }?>
```

CLASE ȘI OBIECTE ÎN PHP

- **Observații:**

- în limbajul PHP, datele membru pot fi inițializate doar cu valori constante;
- pentru inițializarea datelor membru cu valori ce nu sunt constante, trebuie folosit un constructor, acesta fiind o metodă a unei clase care este apelată automat în momentul creării unui nou obiect al clasei;
- pentru crearea unui obiect al cărui tip este o clasă se utilizează operatorul *new*;
- în limbajul PHP nu există destructori;
- în limbajul PHP nu este permisă *moștenirea multiplă*, ci doar *moștenirea singulară* (o clasă derivată-copil poate avea o singură clasă de bază-părinte);
- în limbajul PHP, pentru a deriva (a extinde) o clasă dintr-o altă clasă se utilizează cuvântul cheie *extends*.

CLASE ȘI OBIECTE ÎN PHP-EXEMPLU

```
<?php
class Complex
{
    var $real =0;
    var $imag =0;
    //constructor cu parametrii
    function Complex($a,$b)
    {$this->real=$a;
    $this->imag=$b;}
    function modul()
    { return sqrt($this->real * $this->real +
    $this->imag * $this->imag);}
}
```

```
$x1=3; $y1=4; $x2=1; $y2=2;
$obiect1=new complex ($x1,$y1);
$obiect2=new complex ($x2,$y2);
$obiect3=new complex (5,6);
$obiect4=new complex (3,4);
echo "Modulul obiectului 1 este:".
$obiect1->modul(), "<br>";
echo "Modulul obiectului 2 este:".
$obiect2->modul(), "<br>";
echo "Modulul obiectului 3 este".
$obiect3->modul(), "<br>";
?>
```


MOȘTENIREA ÎN PHP

- În limbajul PHP este permisă doar *moștenirea singulară*;
- Moștenirea păstrează aceleași caracteristici întâlnite în programarea orientată pe obiecte în limbajul C++, și anume:
 - reutilizarea codului clasei de bază (clasa părinte);
 - creșterea eficienței programelor prin posibilitatea de a crea noi clase (clase derivate-copil) prin adăugarea de noi caracteristici la clasele deja existente.

MOȘTENIREA ÎN PHP

- Sintaxa utilizată în derivarea unei *clase copil* (clase derivate) dintr-o clasă părinte (clasă de bază) este următoarea:

```
class clasă_bază
```

```
{//proprietăți și metode ale clasei de bază}
```

```
class clasă_derivată extends clasă_bază
```

```
{//proprietăți și metode ale clasei derivate}
```

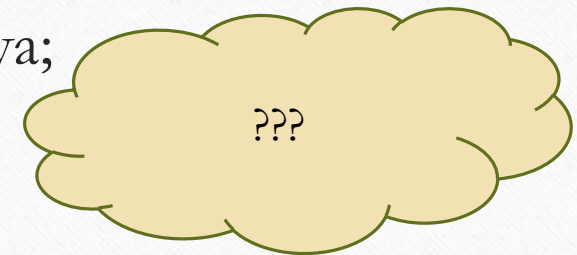
MOȘTENIREA ÎN PHP-EXEMPLU

```
<?php
class dreptunghi
{var $lung =0; var $lat =0;
//constructor cu parametrii
function dreptunghi($a,$b)
{$this->lung=$a; $this->lat=$b;}
function aria_d()
{return $this->lung * $this->lat;}
}
class patrat extends dreptunghi
{ function aria_p()
{return $this->aria_d();}
}
```

```
$x1=3; $y1=4; $x2=2; $y2=2;
$obiect1=new dreptunghi($x1,$y1);
$obiect2=new Patrat($x2,$y2);
echo "Aria dreptunghiului este:". $obiect1-
>aria_d(), "<br>";
echo "Aria pătratului este:". $obiect2-
>aria_p(), "<br>";
?>
```


ELEMENTE DE JAVASCRIPT

- JavaScript este un limbaj, respectiv tehnologie de script simplu, a cărei sintaxă este inspirată din cea a limbajului Java.
- Java și JavaScript sunt două tehnologii diferite;
- JavaScript nu este o versiune mai simplă a limbajului Java;
- Diferența esențială între Java și JavaScript



ELEMENTE DE JAVASCRIPT - CARACTERISTICI

- Limbajul JavaScript este preferat de către dezvoltatorii de site-uri Web din foarte multe considerente, precum:
 - limbaj ce poate fi interpretat de browsere;
 - limbaj ușor de învățat;
 - mărește gradul de interactivitate al paginilor Web prin introducerea câtorva linii de cod;
 - generarea de pagini Web personalizate;
 - modificarea dinamică a prezentării paginilor Web;
 - validarea conținutului unui formular creat;
 - crearea de animații personalizate;
 - executarea funcțiilor clasice ale unui limbaj de programare;
 - comunicarea cu appleturile Java.

ELEMENTE DE JAVASCRIPT - TERMINOLOGIE

- *Obiect* (substantiv) – reprezintă elementele componente (paragrafe, formulare, imagini, tabele, etc.) ale unui document HTML; o pagină Web este și aceasta tot un obiect; există 3 tipuri de obiecte:
 - Obiecte interne – furnizate de limbajul JavaScript (Array, Boolean, Date, Function, Math, Number, Object, RegExp, String, This, etc.);
 - Obiecte personalizate – create de utilizator;
 - Obiecte ale navigatorului – diferite componente ale navigatorului (standardul DOM-Document Object Model) și ale documentului HTML (button, checkbox, document, form, hidden, history, location, navigator, option, Option(), password, radio, reset, screen, select, submit, text, textarea, window);
- *Proprietate* (atribut) – descrie un obiect (length, color, align, method, action, etc.);
- *Metodă* – funcție ce definește un anumit comportament al unui obiect, obiect ce dispune de o colecție de metode predefinite precum: concat(), reverse(), sort(), max(), min(), sin(), abs(), sqrt(), blur(), click(), focus(), etc.; o funcție (metodă) este un obiect care conține cod executabil; metodele definite de către programator au următoarea sintaxă:

function nume_funcție (listă parametrii formali)

{ <! - - instrucțiuni - - > }

ELEMENTE DE JAVASCRIPT

Observații:

- apelul unei funcții se realizează prin utilizarea numelui acesteia folosind următoarea sintaxă:
nume_funcție(listă parametrii efectivi);
- în limbajul JavaScript există doar patru tipuri de date: numerice întregi (*integer*), numerice în virgulă flotantă (*float*), caracter (*string*) și boolean (*True/False*);
- spre deosebire de limbajul PHP, declararea unei variabile în limbajul JavaScript se realizează utilizând tipul universal de date *var* conform următoarei sintaxe:

var nume_variabilă;

- ca și în limbajul PHP și în limbajul JavaScript variabilele pot accepta oricând un nou tip de date, ceea ce conduce la modificarea tipului variabilei.

INSTRUCȚIUNI DE CONTROL ALE LIMBAJULUI JAVASCRIPT

- O *instrucțiune* produce o acțiune precum afișarea unui text sau modificarea proprietății unui obiect.
- O *instrucțiune* combină obiecte, proprietăți și metode.
- **Instrucțiunile de control ale limbajului JavaScript** se împart în două mari categorii, și anume:
 - *Decizia/ Selecția*: instrucțiunea *if* (if cu ramură vidă), *if...else* și *switch* (decizia generalizată);
 - *Iterația/ Repetiția/ Ciclarea*: instrucțiunea *for* (iterația cu număr cunoscut de pași), ***for...in*** (iterația care permite baleierea elementelor unei matrici sau a unui obiect), *while* (iterația cu test inițial) și *do...while* (iterația cu test final).

INSTRUCȚIUNI DE CONTROL ALE LIMBAJULUI JAVASCRIPT

Observație: Instrucțiunile de control ale limbajului JavaScript sunt în mare parte (cu excepția instrucțiunii *for...in*) similare cu cele ale limbajului C/C++.

- *Instrucțiunea if* permite execuția condiționată a unui set de instrucțiuni (dacă și numai dacă condiția este adevărată/îndeplinită). Sintaxa instrucțiunii *if* este următoare

if (condiție)
{ <!-- instrucțiuni --> }

- *Instrucțiunea if...else* permite alegerea unei alternative din două alternative posibile în urma testării valorii de adevăr a unei condiții. Sintaxa instrucțiunii *if...else* este următoarea

if(condiție)
{ <!-- instrucțiuni 1 --> }
else { <!-- instrucțiuni 2 --> }

INSTRUCȚIUNI DE CONTROL ALE LIMBAJULUI JAVASCRIPT-EXEMPLUDE UTILIZARE A INSTRUCȚIUNII *IF...ELSE*

```
<html>
<head>
<script type="text/javascript">
  n=parseInt(prompt("Introduceți numărul de
  componente ale vectorului n=",0));
  a = new Array(); var i;
  for(i=1;i<=n;i++)
    {a[i]=parseInt(prompt("Introduceți a -
    "+i+"-a componentă",0));}
  s="";
  for(i=1;i<=n;i++)
    s=s+a[i]+" "; alert("vectorul conține: "+s);
```

```
var suma=0;
var produs=1;
for(i=1;i<=n;i++)
  {if (a[i]%2==0) suma=suma+a[i];
   else produs=produs*a[i]; }
document.write("suma elementelor pare
este:",+suma+"<br>");
document.write("produsul elementelor impare
este:",+produs);
</script>
</head></body></html>
```

INSTRUCȚIUNI DE CONTROL ALE LIMBAJULUI JAVASCRIPT

- Instrucțiunea *switch* permite alegerea unei alternative din două alternative posibile. Utilizează o variabilă specială numită variabilă selector. Sintaxa instrucțiunii *switch* este următoarea:

switch (selector)

{case valoare_1: instrucțiuni_1; break;

case valoare_2: instrucțiuni_2; break;

case valoare_n: instrucțiuni_n; break;

default: instrucțiuni n+1;

}

INSTRUCȚIUNI DE CONTROL ALE LIMBAJULUI JAVASCRIPT

- Instrucțiunea *for* este utilizată pentru a executa un set de instrucțiuni cât timp condiția este îndeplinită. Sintaxa instrucțiunii *for* este următoarea:

for(inițializare; condiție; actualizare)

{<!-- - instrucțiuni -->}

- Instrucțiunea *for...in* execută aceeași instrucțiune sau aceeași secvență de instrucțiuni pentru toate proprietățile unui obiect. Sintaxa instrucțiunii *for...in* este următoarea:

for (proprietate in Obiect)

{<!-- - instrucțiuni - ->}

INSTRUCȚIUNI DE CONTROL ALE LIMBAJULUI JAVASCRIPT-EXEMPLU UTILIZARE for...in

- Scriptului JavaScript pentru afișarea tuturor proprietăților unui obiect creat.

```
</html>
<head> <title> Instrucțiunea
for...in</title> </head>
<body>
<script>
var rezultat=" ";
var obiect={disciplina:"Programare_Web:",
evaluare:"verificare,", credite:"4credite;"};
```

```
for(var proprietate in obiect)
{rezultat+=obiect[proprietate];}
document.write(rezultat);
</script>
</body>
</html>
```


INSTRUCȚIUNI DE CONTROL ALE LIMBAJULUI JAVASCRIPT

- Instrucțiunea *while* execută cât timp este îndeplinită condiția setul de instrucțiuni. Sintaxa instrucțiunii *while* este următoarea:

```
while(condiție)  
{ <!--instrucțiuni - - > }
```

- Instrucțiunea *do...while* execută un set de instrucțiuni cât timp condiția este îndeplinită. Sintaxa instrucțiunii *do...while* este următoarea:

```
do  
{ <!--instrucțiuni - -> }  
while(condiție);
```

Aplicații propuse

- Testarea aplicațiilor prezentate în cadrul cursului;
- Testarea aplicațiilor 3.2.1, 3.2.2, 3.2.3, 3.2.4, 3.2.5, 3.2.6, pag. 128-133;
- Rezolvarea aplicațiilor din cadrul lucrării de laborator nr. 1, pag. 137;
- Rezolvarea aplicațiilor din cadrul lucrării de laborator nr. 2, pag. 137;
- **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!