

Laborator 4

2020/2021

Programarea aplicațiilor web

Laborator 3

Laborator L3

- Sa se creeze un magazin simplu virtual care:
 - sa prezinte utilizatorului o lista de produse si preturi (constanta – maxim 5 produse)
 - sa preia de la acesta numarul de produse dorit
 - sa calculeze suma totala
 - sa adauge TVA 19%
 - sa prezinte un raport care sa contina:
 - total de plata
 - ora comenzii

Laborator L3 - continuare

- se creaza macar 3 pagini:
 - lista produse
 - formular comanda
 - rezultat
- forma paginilor:
 - tabel/CSS
- metoda
 - post
 - get

culoare	IMAGINE	culoare
	Continut (cu alta culoare fundal)	

Laborator – L3 - rezultat

Magazin online Firma X SRL

Lista Produse

Nr.	Produs	Pret
1	Carti	100
2	Caiete	50
3	Penare	150
4	Stilouri	125
5	Creioane	25

[Comanda](#)

Magazin online Firma X SRL

Realizati comanda

Nr.	Produs	Pret	Cantitate
1	Carti	100	<input type="text" value="1"/>
2	Caiete	50	<input type="text" value="2"/>
3	Penare	150	<input type="text" value="1"/>
4	Stilouri	125	<input type="text" value="0"/>
5	Creioane	25	<input type="text" value="0"/>

Magazin online Firma X SRL

Rezultate comanda

Pret total (fara TVA): 350

Pret total (cu TVA): 416.5

Comanda receptionata la data: 17/03/2010 ora 08:24

Utilizare LAMP

Magazin online Firma X SRL

Lista Produse

Nr.	Produs	Pret
1	Carti	100
2	Caiete	50
3	Penare	150
4	Stilouri	125
5	Creioane	25

Comanda

Magazin online Firma X SRL

Realizati comanda

Nr.	Produs	Pret	Cantitate
1	Carti	100	<input type="text" value="1"/>
2	Caiete	50	<input type="text" value="2"/>
3	Penare	150	<input type="text" value="1"/>
4	Stilouri	125	<input type="text" value="0"/>
5	Creioane	25	<input type="text" value="0"/>

Trimite

Magazin online Firma X SRL

Rezultate comanda

Pret total (fara TVA): 350

Pret total (cu TVA): 416.5

Comanda receptionata la data: 17/03/2010 ora 08:24

`<input name="x" ..`

`$_POST['x']`

`$_GET['x']`

Web Server

Apache

Server FTP

Server SSH

Includerea / controlul formei in fisierile PHP - Template

Analiza critica

- design?
 - in aplicatiile web forma este importanta
 - nu trebuie sa fie inovativa ci familiara
 - "Don't make me think!"
- ~~■ capacitatea de extindere?~~
 - ~~■ mai multe produse~~
 - ~~■ schimbare de pret~~

Template

- Sablon
- controlul simultan al formei pentru toate paginile din site
- separarea aplicatiei de forma

Lista produse

Magazin **Firma X SRL**

Magazin online Firma X SRL

Lista Produse

Nr.	Produs	Pret
1	Carti	100
2	Caiete	50
3	Penare	150
4	Stilouri	125
5	Creioane	25
Comanda		

Elemente de control

- `include()`
- `require()`
- `include_once()`
- `require_once()`

- pentru inserarea **SI** evaluarea fisierului folosit ca parametru
- folosite pentru a nu multiplica sectiunile de cod comune
- **require** opreste executia script-ului curent daca fisierul parametru **nu** este gasit
- **..._once()** verifica daca respectivul fisier a mai fost introdus si **nu** il mai introduce inca o data

Exemplu – design 2

- sectiunile repetabile pot fi mutate intr-un fisier separat si introduse cu require()
- se identifica zonele comune

```
<html>
<head>
<title>Magazin online Firma X SRL</title>
</head>
<body bgcolor="#CCFFFF">
<table width="600" border="0" align="center">
<tr><td></td></tr>
<tr><td height="600" valign="top"
bgcolor="#FFFFCC">
Continut
</td></tr>
</table>
</body>
</html>
```

Lista produse

antet.php

```
<html>
<head>
<title>Magazin online Firma X
SRL</title>
</head>
<body bgcolor="#CCFFFF"><?php
define('PRET_CARTE',100);
define('PRET_CALET',50);
define('PRET_PENAR',150);
define('PRET_STILOU',125);
define('PRET_CREION',25);
//orice cod comun PHP
?><table width="600" border="0"
align="center">
<tr><td></td></tr>
<tr><td height="600" valign="top"
bgcolor="#FFFFCC">
<h1>Magazin online Firma X SRL</h1>
```

subsol.php

```
</td></tr>
</table>
</body>
</html>
```

*.php

```
<?php require('antet.php');?>
<h2>Lista Produse</h2>
<table border="1">
...
</table>
<?php require('subsol.php');?>
```

Lista produse/template

Magazin **Firma X SRL**

Magazin online Firma X SRL

Lista Produse

Nr.	Produs	Pret
1	Carti	100
2	Caiete	50
3	Penare	150
4	Stilouri	125
5	Creioane	25

[Comanda](#)

Avantajul lucrului cu sabloane

- viteza de dezvoltare a aplicatiei
- separare clara a formei de aplicatie
- forma unitara
 - “don’t make me think”
- modificarea simultana a formei pentru toate paginile din site
- posibilitatea definirii datelor comune intr-un singur fisier
 - `define('PRET_CARTE',100);`

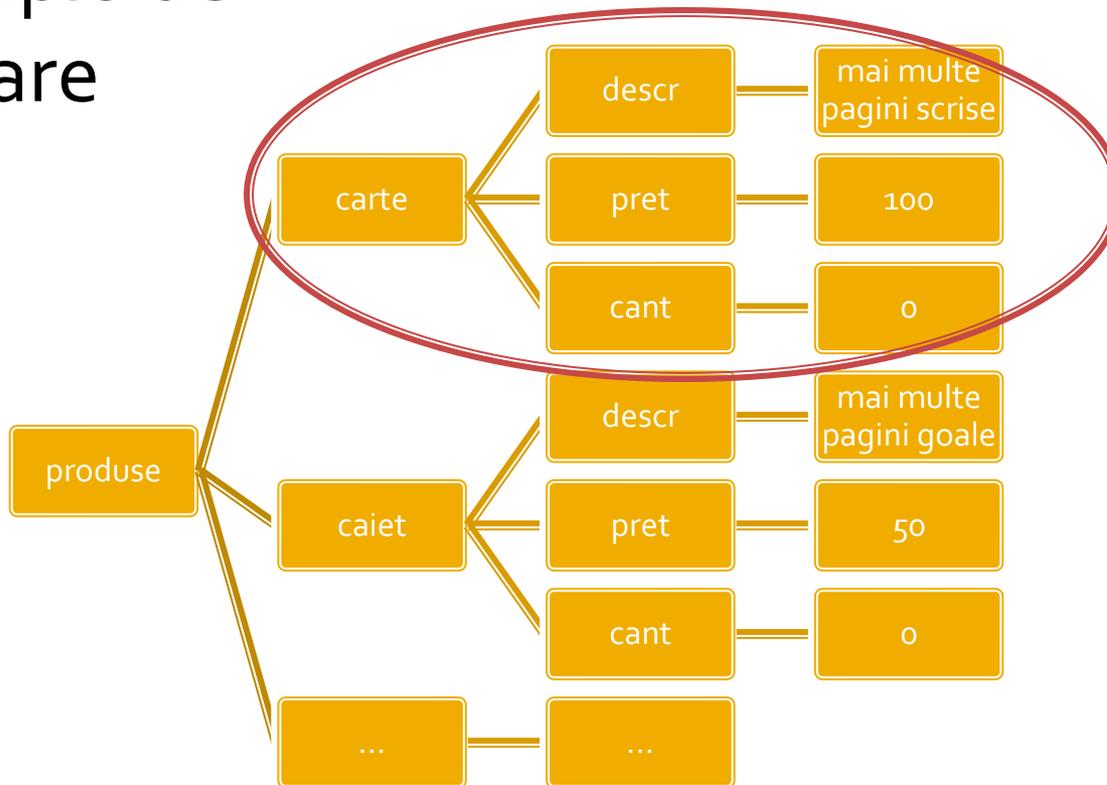
Tablouri

Laborator 4

- Sa se continue magazinul virtual:
 - pentru usurinta modificarilor ulterioare aplicatia lucreaza cu matrici
 - sa prezinte utilizatorului o lista de produse si preturi

Laborator 4 – Matrice produse

- exemplu de grupare

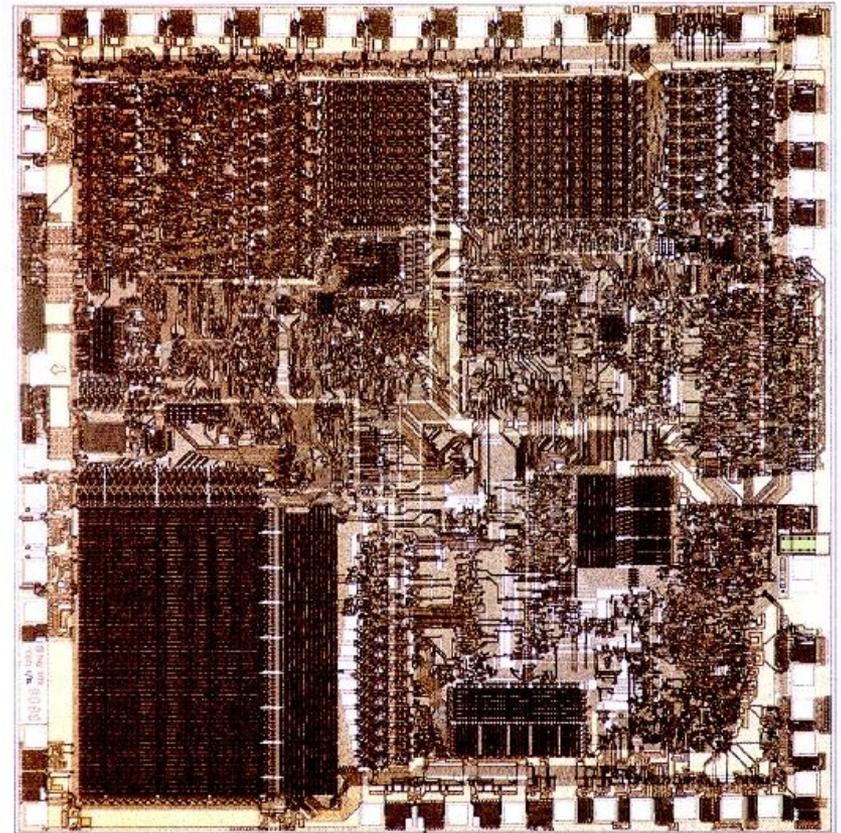


Tot Curs 2

Structuri repetitive – tablouri

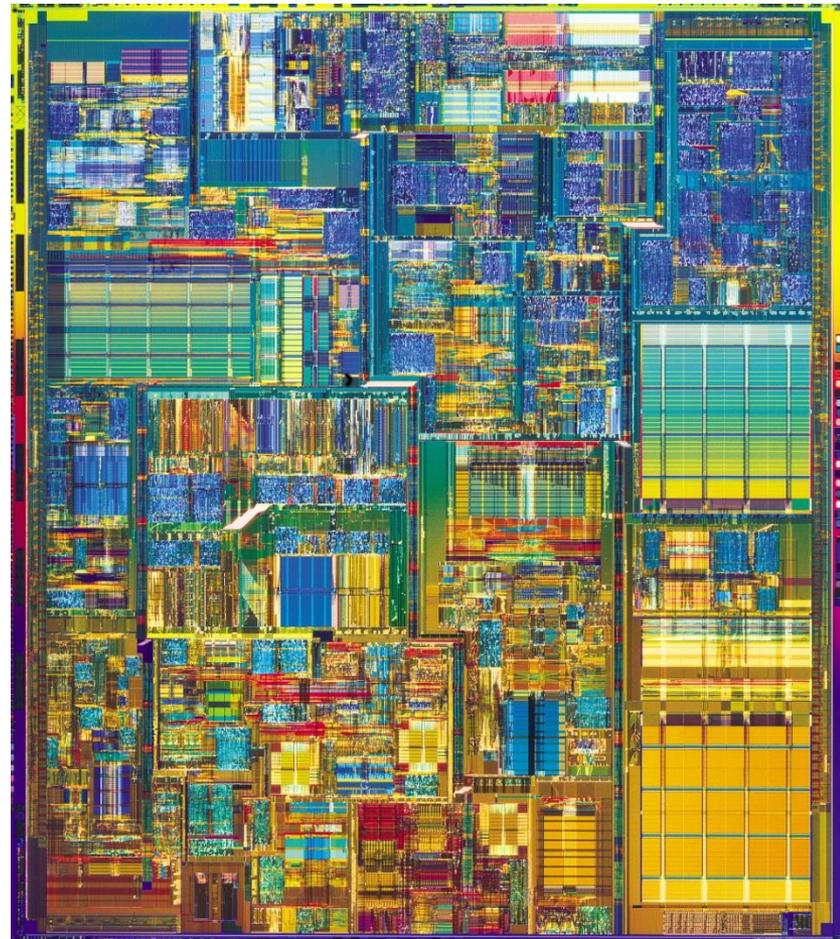
Impresionant?

- Intel® 8086
- 29.000 tranzistoare pe CPU
- 1978
- 1 MB date
- 4.7 MHz



Impresionant?

- Intel® Itanium® processors (Tukwila)
- 2009
- 2 miliarde tranzistoare pe CPU
- 16 EB date (16 G GB)
- > 3 GHz



Concepte

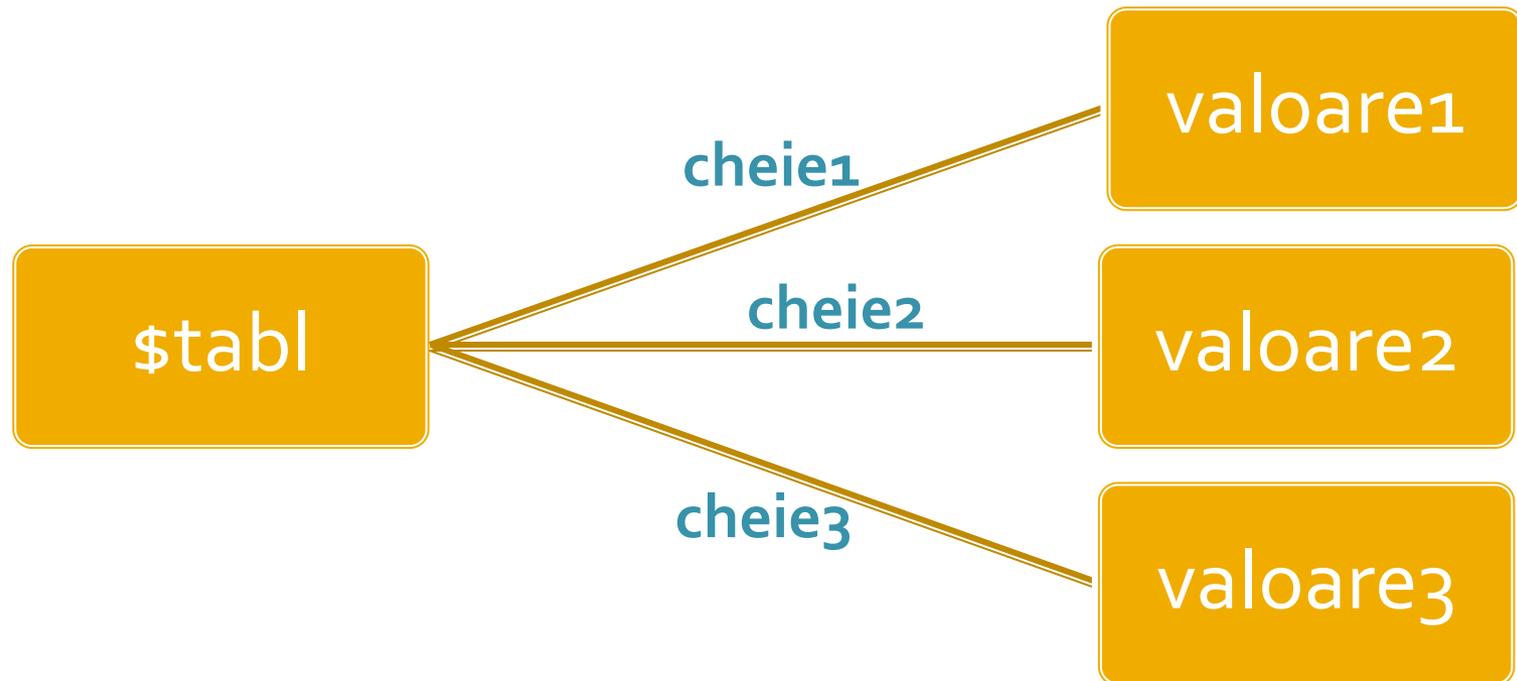
- Efectuare foarte rapida a unui numar **mic** de instructiuni, de **complexitate redusa**, repetate de un numar foarte mare de ori
- Programare: coborarea rationamentului la nivelul de **complexitate redusa**, cu obtinerea performantei prin structuri repetitive simple efectuate rapid.
- Operatii repetitive / date repetitive

Tablouri in PHP

- tabloul este tipul de variabila care asociaza **valori** unor **chei**
- spre deosebire de C, Basic, **cheile nu sunt** obligatoriu numere **intregi**, pot fi si **siruri**
- implicit cheile sunt intregi succesivi (pentru fiecare element adaugat) si primul element este 0.
- definirea unei perechi cheie / valoare
 - cheie => valoare
- definirea unui tablou
 - `$matr = array("definirea perechilor chei/valori")`
 - pereche: cheie => valoare, ...

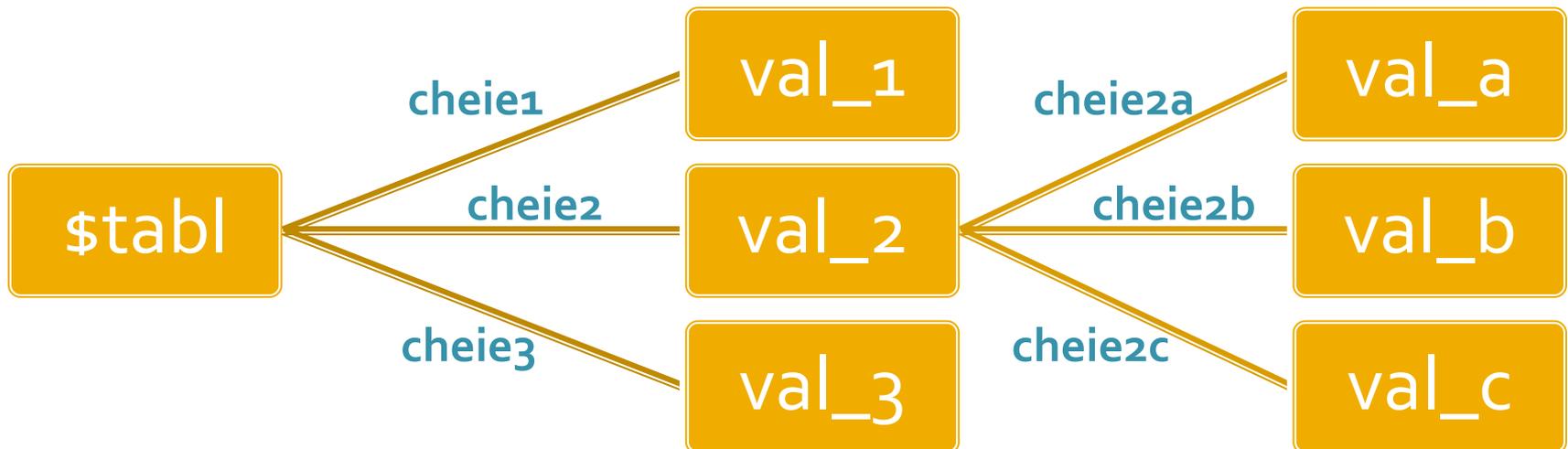
Tablou = Arbore

- `$tabl = array(cheie1 => valoarea1, cheie2 => valoarea2, cheie3 => valoarea3)`



Tablou = Arbore

- In particular, una sau mai multe dintre din valori poate fi la randul ei tablou, ducand la **ramificarea** arborelui
- `$tabl = array(cheie1 => val_1, cheie2 => array(cheie2a => val_a, cheie2b => val_b, cheie2c => val_c), cheie3 => val_3)`



Tablouri in PHP

```
$matr = array(1, 2, 3, 4, 5);
```

```
$matr[0]=1
```

```
$matr[1]=2
```

```
$matr[2]=3
```

```
$matr[3]=4
```

```
$matr[4]=5
```

```
$matr = array('a' => 1, 'b' => 2, 3, 4, 5);
```

```
$matr['a']=1
```

```
$matr['b']=2
```

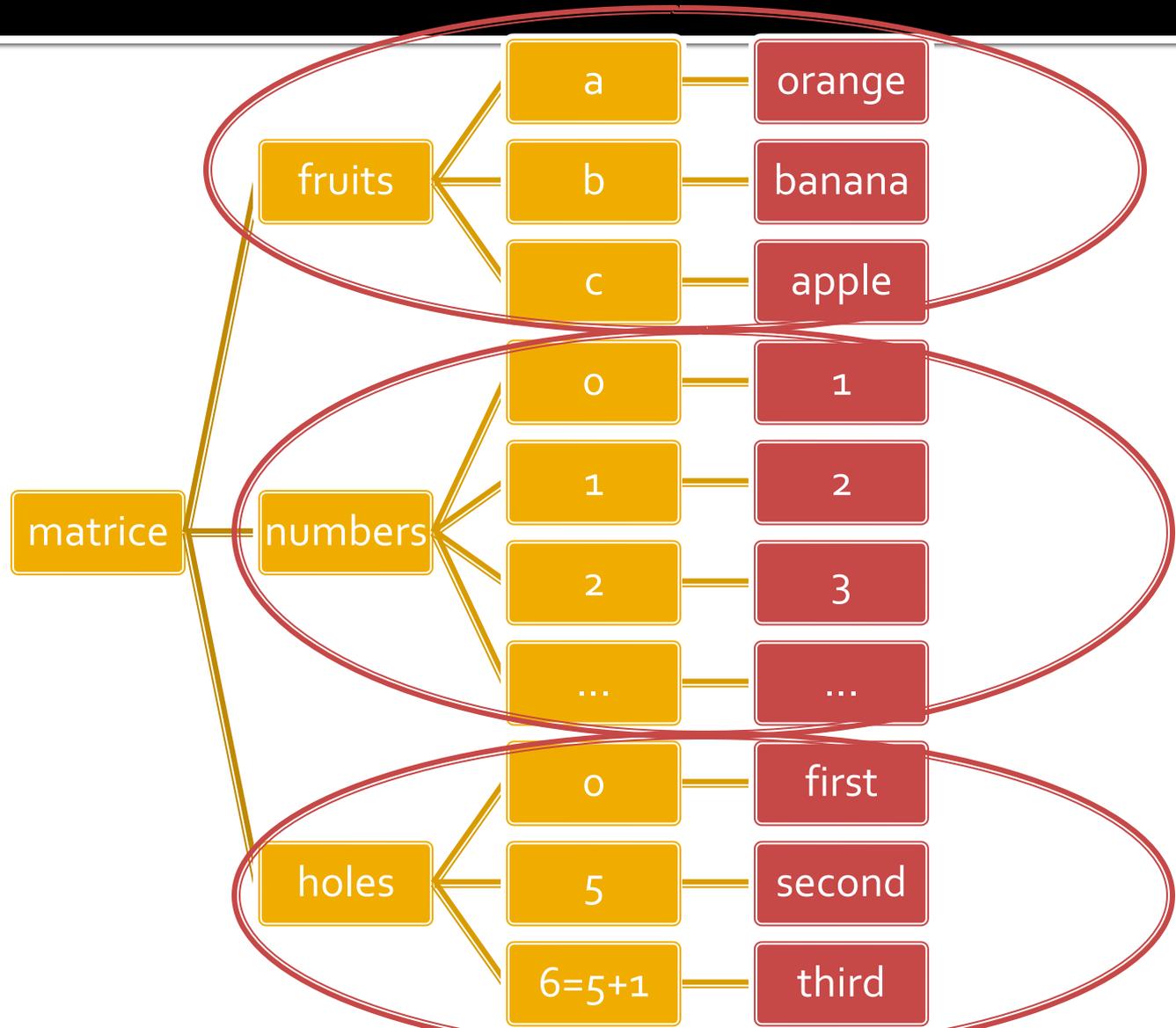
```
$matr[0]=3
```

```
$matr[1]=4
```

```
$matr[2]=5
```

```
$matrice= array (  
    "fruits" => array("a" => "orange", "b" => "banana", "c" => "apple"),  
    "numbers" => array(1, 2, 3, 4, 5, 6),  
    "holes" => array("first", 5 => "second", "third")  
);
```

Tablou = arbore



Afisarea tablourilor

```
echo "<pre>";  
print_r ($matr);  
echo "</pre>";
```

```
$matr= array (  
"fruits" =>  
array("a" => "orange", "b" => "banana", "c" => "apple",  
"ultim"),  
"numbers" =>  
array(1, 2, 3, 4, 5, 6),  
"holes" =>  
array("first", 5 => "second", "third")  
);  
echo $matr;  
echo "<pre>";  
print_r ($matr);  
echo "</pre>";
```

```
Array  
  
Array  
(  
    [fruits] => Array  
        (  
            [a] => orange  
            [b] => banana  
            [c] => apple  
            [0] => ultim  
        )  
    [numbers] => Array  
        (  
            [0] => 1  
            [1] => 2  
            [2] => 3  
            [3] => 4  
            [4] => 5  
            [5] => 6  
        )  
    [holes] => Array  
        (  
            [0] => first  
            [5] => second  
            [6] => third  
        )  
)
```

Chei

- Chei numerice
 - implicite
 - similare celorlalte limbaje de programare
 - dificil de utilizat (trebuie retinuta valoarea logica a unei anumite chei numerice)
- Chei sir
 - claritate mai mare
 - eficienta numerica mai mica
 - tablourile au un index numeric intern, implicit ascuns, accesibil prin functii :
index => cheie => valoare

Elemente de control

- `for` – util dacă la definirea tablourilor sunt folosite cheile numerice implicite (numere întregi)
- `do ... while` și `while` se pot folosi împreună cu funcții specifice caracteristice tablourilor `next()`, `prev()`, `end()`, `reset()`, `current()`, `each()`
- `foreach` - elementul de control al iteratiilor cel mai potrivit pentru chei alfanumerice

Elemente de control – foreach

- `foreach (array_expression as $key => $value) statement`
- `foreach (array_expression as $value) statement`
- iterarea prin fiecare element al tabloului
- la fiecare element variabila declarata in instructiune variabila locala `$key` ofera acces la cheia curenta iar variabila locala `$value` ofera acces la valoarea asociata
- `foreach()` lucreaza cu o **copie** a tabloului deci tabloul original nu va fi modificat prin schimbarea continutului variabilelor `$key` si `$value`

Elemente de control – foreach

```
$matr = array (  
    "fruits" => array("a" => "orange", "b" => "banana", "c" => "apple", "ultim"),  
    "numbers" => "in loc de numere",  
    "holes" => "in loc de ce era"  
);  
foreach ($matr as $scheie => $continut)  
    echo "matr[".$scheie."]=".$continut."<br />";
```

```
matr[fruits]=Array  
matr[numbers]=in loc de numere  
matr[holes]=in loc de ce era
```

Tablouri – functii utile

- `current ($matr)` – returneaza elementul indicat de indicele intern al tabloului (`~v[i]`)
- `next ($matr)` – incrementeaza indicele intern si returneaza valoarea stocata acolo (`~v[++i]`)
- `prev ($matr)` – decrementeaza indicele intern si returneaza valoarea stocata acolo (`~v[--i]`)
- `end($matr)` – muta indicele intern la ultimul element si returneaza valoarea stocata acolo (`~i=N-1;v[i]`)
- `reset($matr)` – muta indicele intern la primul element si returneaza valoarea stocata acolo (`~i=0;v[i]`)

Tablouri – functii utile

- `sort($matr)` – ordoneaza in ordine crescatoare a **valorilor** un tablou, cheile sunt sterse si recreate
 - `$fruits = array("lemon", "orange", "banana", "apple");`
`sort($fruits);`
 - `fruits[0] = apple, fruits[1] = banana, fruits[2] = lemon, fruits[3] = orange`
- `rsort($matr)` – similar, descrescator

Tablouri – functii utile

- `asort($matr)` ordoneaza in ordine crescatoare a **valorilor** un tablou, cheile sunt pastrate, inclusiv asocierea cheie => valoare
 - `$fruits = array("d" => "lemon", "a" => "orange", "b" => "banana", "c" => "apple");`
`asort($fruits);`
 - `c = apple, b = banana, d = lemon, a = orange`
- `arsort($matr)` – similar, descrescator

Tablouri – functii utile

- `ksort($matr)` ordoneaza in ordine crescatoare a **cheilor** un tablou, cheile sunt pastrate, inclusiv asocierea cheie => valoare
 - `$fruits = array("d" => "lemon", "a" => "orange", "b" => "banana", "c" => "apple");`
`ksort($fruits);`
 - a = orange, b = banana, c = apple , d = lemon
- `krsort($matr)` – similar, descrescator

Exemplu utilizare tablouri

Analiza critica

- ~~design?~~
 - ~~in aplicatiile web forma este importanta~~
 - ~~nu trebuie sa fie inovativa ci familiara~~
 - ~~"Don't make me think!"~~
- capacitatea de extindere?
 - mai multe produse
 - schimbare de pret

Exemplu

- In exemplul anterior utilizarea tablourilor va aduce urmatoarele avantaje:
 - codul va fi mai concis
 - codul va fi mai general (valabil si pentru 5 produse si pentru 1000)
 - scalabilitate crescuta (se pot adauga usor produse)

Exemplu

- fiecare produs e caracterizat de:
 - nume
 - pret
 - (eventual) descriere
 - cantitate comandata
- putem folosi unul din attribute ca si cheie (numele in exemplu)
- se poate controla (prin atributul name = "") structura variabilei globale \$_POST

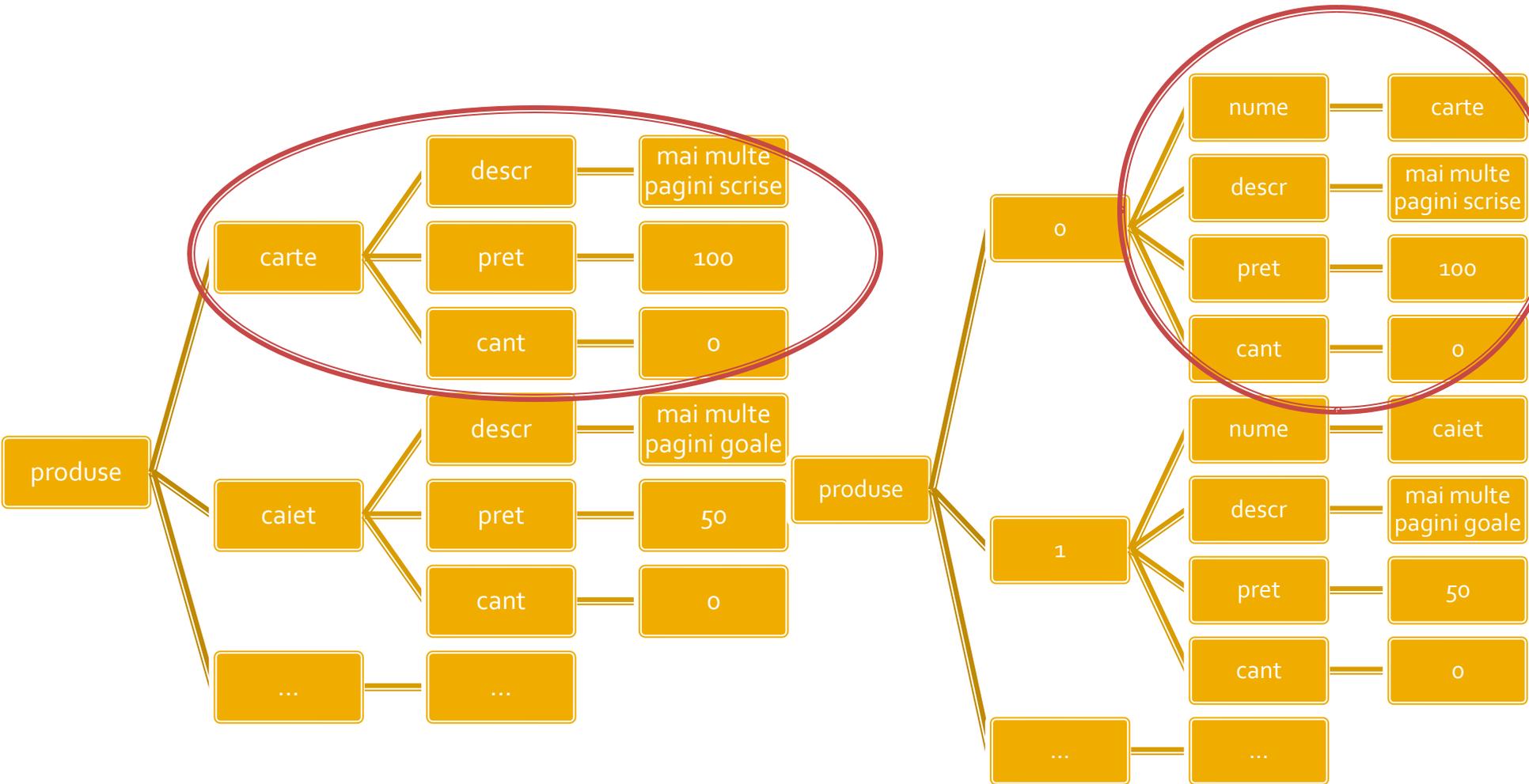
Tablou produse

- una din structurile posibile

```
$produse = array ( 'carte' => array ("descr" => "mai multe pagini scrise", "pret" => 100, "cant" => 0),  
                  'caiet' => array ("descr" => "mai multe pagini goale", "pret" => 50, "cant" => 0),  
                  'penar' => array ("descr" => "loc de depozitat instrumente", "pret" => 150, "cant" => 0),  
                  'stilou' => array ("descr" => "instrument de scris albastru", "pret" => 125, "cant" => 0),  
                  'creion' => array ("descr" => "instrument de scris gri", "pret" => 25, "cant" => 0)  
                );
```

- se urmareste obtinerea unei structuri clare
 - usor de modificat/adaugat date
 - usor de utilizat
- daca definitia se introduce in fisierul antet va fi accesibila in toate fisierele individuale

Tablou produse



Laborator – L3 – sursa 1

```
<?php
define('PRET_CARTE',100);
define('PRET_CALET',50);
define('PRET_PENAR',150);
define('PRET_STILOU',125);
define('PRET_CREION',25);
?><h1>Magazin online Firma X SRL</h1>
<h2>Realizati comanda</h2>
<form action="rezultat.php" method="post">
<table border="0">
<tr bgcolor="#cccccc"><td>Nr.</td><td width="150">Produs</td><td width="50">Pret</td><td
width="15">Cantitate</td></tr>
<tr><td>1</td><td>Carti</td><td align="center"><?php echo PRET_CARTE;?></td><td align="center"><input
name="carte_cant" type="text" value="0" size="3" maxlength="3" /></td></tr>
<tr><td>2</td><td>Caiete</td><td align="center"><?php echo PRET_CALET;?></td><td align="center"><input
name="caiet_cant" type="text" value="0" size="3" maxlength="3" /></td></tr>
<tr><td>3</td><td>Penare</td><td align="center"><?php echo PRET_PENAR;?></td><td align="center"><input
name="penar_cant" type="text" value="0" size="3" maxlength="3" /></td></tr>
<tr><td>4</td><td>Stilouri</td><td align="center"><?php echo PRET_STILOU;?></td><td align="center"><input
name="stilou_cant" type="text" value="0" size="3" maxlength="3" /></td></tr>
<tr><td>5</td><td>Creioane</td><td align="center"><?php echo PRET_CREION;?></td><td align="center"><input
name="creion_cant" type="text" value="0" size="3" maxlength="3" /></td></tr>
<tr>
<td colspan="4" align="center"><input type="submit" value="Trimite" /></td></tr>
</table>
</form>
```

Crearea listei de produse

```
<?php require('antet.php');?>
<h2>Lista Produse</h2>
<table border="1">
<tr bgcolor="#cccccc"><td>Nr.</td><td width="150">Produs</td><td width="150">Descriere</td><td
width="50">Pret</td></tr>
<?php
$index=1;
foreach ($produse as $prod => $detalii)
{
    echo "<tr><td>".$index."</td><td>".ucfirst(strtolower($prod))."</td><td>".$detalii['descr']."</td><td
align=\"center\">".$detalii['pret']."</td></tr>";
    $index++;
}
?>
<?php
$index=1;
foreach ($produse as $prod => $detalii)
{?>
<tr><td><?php echo $index;?></td><td><?php echo ucfirst(strtolower($prod));?></td><td><?php echo
$detalii['descr'];?></td><td align="center"><?php echo $detalii['pret'];?></td></tr>
<?php $index++;
} ?>
<tr><td colspan="4" align="center"><a href="formular.php">Comanda</a></td></tr></table>
<?php require('subsol.php');?>
```



Crearea listei de produse

Magazin

Firma X SRL

Magazin online Firma X SRL

Lista Produse

Nr.	Produs	Descriere	Pret
1	Carte	mai multe pagini scrise legate	100
2	Caiet	mai multe pagini goale legate	50
3	Penar	loc de depozitat instrumente de scris	150
4	Stilou	instrument de scris albastru	125
5	Creion	instrument de scris gri	25
1	Carte	mai multe pagini scrise legate	100
2	Caiet	mai multe pagini goale legate	50
3	Penar	loc de depozitat instrumente de scris	150
4	Stilou	instrument de scris albastru	125
5	Creion	instrument de scris gri	25
Comanda			

Subdivizare \$_POST

- atributul **name** in forma devine **cheie** in tabloul global \$_POST
 - `<input type="text" name="carti_cant" size="3" maxlength="3" />`
 - `$_POST['carti_cant']` contine valoarea introdusa
- realizand atributul **name** ca tablou, se obtine in \$_POST un "subtablou" (ramificare locala a arborelui) care grupeaza elementele input
 - `<input type="text" name="cant[carti]" size="3" maxlength="3" />`
 - `$_POST ['cant'] ['carti']` contine valoarea introdusa

Subdivizare \$_POST

- realizand atributul `name` ca tablou, se obtine in `$_POST` un "subtablou" (ramificare locala a arborelui) care grupeaza elementele dorite
 - `<input type="text" name="cant[carti]" size="3" maxlength="3" />`
 - `$_POST ['cant'] ['carti']` contine valoarea introdusa
- Este necesar pentru a grupa elementele similare pe care sa le prelucram la receptie cu `foreach`
- `$_POST` contine si alte elemente pe care le dorim eventual tratate separat
 - numele (`name`) si valoarea butonului "submit" apar in `$_POST` de exemplu

Crearea formei de comanda

```
<?php require('antet.php');?>
<h2>Realizati comanda</h2>
<form action="rezultat.php" method="post">
<table border="0">
<tr bgcolor="#cccccc"><td>Nr.</td><td width="150">Produs</td><td width="50">Pret</td><td
width="15">Cantitate</td></tr>
<?php
$index=1;
foreach ($produse as $prod => $detalii)
    {?>
<tr><td><?php echo $index;?></td><td><?php echo ucfirst(strtolower($prod));?></td><td
align="center"><?php echo $detalii['pret'];?></td><td><input name="<?php echo
"cant[".$prod."];?>" type="text" value="0" size="3" maxlength="3" /></td></tr>
<?php $index++;
    }?>
<tr><td colspan="4" align="center"><input type="submit" value="Trimite" /></td></tr>
</table>
</form>
<?php require('subsol.php');?>
```

Crearea listei de produse

Magazin **Firma X SRL**

Magazin online Firma X SRL

Realizati comanda

Nr.	Produs	Pret	Cantitate
1	Carte	100	<input type="text" value="0"/>
2	Caiet	50	<input type="text" value="0"/>
3	Penar	150	<input type="text" value="0"/>
4	Stilou	125	<input type="text" value="0"/>
5	Creion	25	<input type="text" value="0"/>

Prelucrarea comenzii

```
<?php require('antet.php');?>
<h2>Rezultate comanda</h2>
<p>Pret total (fara TVA):
<?php
$pret=0;
$afis="";
foreach ($_POST['cant'] as $prod => $cant)
    {
        $pret += $cant*$produse[$prod]['pret'];
        $afis .= "+".$cant."x".$produse[$prod]['pret'];
    }
echo $pret;
?>
<p>Obtinut astfel: <?php echo $afis;?></p>
<p>Pret total (cu TVA): <?php echo $pret*1.19;?></p>
<p><?php
echo "<pre>";
print_r ($_POST);
echo "</pre>";
?>
</p>
<p>Comanda receptionata la data: <?php echo date('d/m/Y')." ora ".date('H:i');?></p>
<?php require('subsol.php');?>
```

Prelucrarea comenzii

Magazin

Firma X SRL

Magazin online Firma X SRL

Rezultate comanda

Pret total (fara TVA): 600

Obtinut astfel: +2x100+2x50+2x150+0x125+0x25

Pret total (cu TVA): 714

```
Array
(
    [cant] => Array
        (
            [carte] => 2
            [caiet] => 2
            [penar] => 2
            [stilou] => 0
            [creion] => 0
        )
)
```

Comanda receptionata la data: 17/03/2010 ora 13:55

Laborator 4

Laborator 4

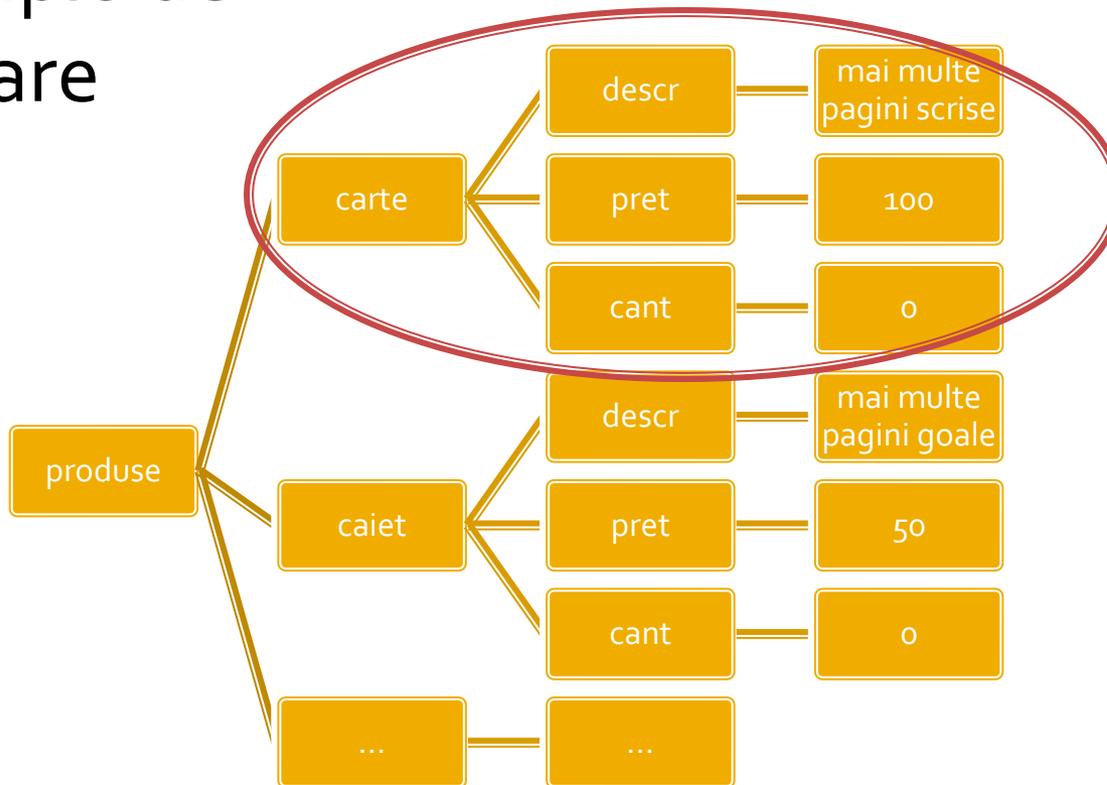
- Sa se continue magazinul virtual:
 - pentru usurinta modificarilor ulterioare aplicatia lucreaza cu matrici
 - sa prezinte utilizatorului o lista de produse si preturi
 - se preia comanda si se calculeaza suma totala

Laborator 4 – Mod de lucru

- 1. Se introduce in surse facilitatea template (S7)
- 2. Se modifica sursele pentru lucru cu matrici (S38)

Laborator 4 – Matrice produse

- exemplu de grupare



Client / Server

```
<input name="nume" ....>
```

```
echo $_POST['nume']; //ceva  
echo $_GET['nume']; //ceva  
echo $_REQUEST['nume']; //ceva
```

ceva

Trimite

get
post

Interpretor PHP primeste
\$_POST
\$_GET
\$_REQUEST

Depanare

```
echo "<pre>";  
print_r($_POST);  
echo "</pre>";
```

```
<p>temp <?php echo  
"a=";echo $a; ?> </p>
```

Contact

- Laboratorul de microunde si optoelectronica
- <http://rf-opto.etti.tuiasi.ro>
- rdamian@etti.tuiasi.ro