

3. laboratorijska vježba: PHP i DOM

Cilj vježbe

Upoznavanje s jezikom PHP i stvaranjem dinamičkih stranica weba. Upoznavanje s programskim sučeljem za rad s DOM stablom u jeziku PHP. Rad s XML dokumentom u obliku DOM, rad s podacima putem DOM sučelja, konverzija podataka iz DOM za prikaz u XHTML obliku. Pretraživanje podataka u DOM stablu.

Priprema za vježbu

Proučiti – PHP:

- osnove jezika PHP
- uključivanje PHP kôda u dokument (oznake `<?php ... ?>`)
- osnovni konstrukti jezika – podaci, operatori, upravljačke strukture, iteratori
- ugrađene funkcije (API) – rad s nizovima znakova, poljima
- preuzimanje podataka iz obrazaca
- generiranje izlaza (ispis)
- pisanje vlastitih funkcija
- rad s objektima i klasama

Proučiti – DOM:

- strukturu DOM klase
- inicijalizaciju DOM stabla iz postojećeg XML dokumenta
- dohvat elemenata iz stabla, pristup sadržaju elemenata i atributa
- inicijalizaciju XPath procesora
- filtriranje elemenata korištenjem XPath upita

Ponoviti – XPath (korišten u XSL-u):

- pristupanje elementima i atributima
- postavljanje uvjeta za pretragu (predikata)
- postavljanje složenih uvjeta – logički operatori

Zadatak za vježbu

Potrebno je **izraditi PHP skriptu** za pretraživanje strukturiranog zapisa podataka u XML dokumentu. Rad sa XML dokumentom izvodi se kroz **DOM sučelje**. **Obrazac** za unos parametara za pretraživanje, testni **XML dokument** i **tablični prikaz rezultata** pretraživanja izrađeni su u prethodnim vježbama.

Pretragu podataka u XML dokumentu moguće je izvesti na dva načina – korištenjem XPath upita ili *ručnim* provjeravanjem udovoljava li neki podatak uvjetima pretrage. Parametre za pretraživanje upisane u obrazac treba iskoristiti za oblikovanje XPath upita, odnosno provjeru smije li se neki podatak prikazati. Funkcije za **oblikovanje upita**, **provjeru uvjeta nad elementom** i sve pomoćne funkcije potrebno je definirati u **posebnoj PHP skripti** koja se uključuje u skriptu za pretraživanje.

Pretragu treba omogućiti **barem po svim elementima i atributima do druge razine** (prema opisu inačica). Pretraživanje tekstualnih podataka treba uzeti u obzir sve nizove znakova koje sadrže uvjet pretrage (a ne samo one koji su potpuno jednaki uvjetu). Pretraga bez postavljenog uvjeta treba prikazati sve podatke.

Predaja vježbe

Predaja vježbe podrazumijeva:

1. predočavanje dokumenta sa testnim podacima (XML),
2. predočavanje obrasca za pretragu podataka,
3. pretragu podataka prema proizvoljnom uvjetu, prikaz rezultata,
4. provjeru ispravnosti pretrage,
5. eventualne promjene u načinu pretrage i prikaza rezultata.

Vježba se predaje na vlastitom računalu s instaliranim PHP-om i poslužiteljem weba (preporučamo Apache) ili na fakultetskom računalu pinus.cc.fer.hr na kojem su instalirani Apache i PHP5.

Datoteke koje treba predati i prikazati su sljedećih naziva:

- pretraga.php
- funkcije.php

U ovoj će vježbi biti potrebne i druge datoteke (iz vježbe 1. i 2., npr. podaci.xml). Na poslužitelj Moodle treba postaviti ZIP arhivu sa svim datotekama potrebnima za izvođenje.

Priprema okruženja

Apache poslužitelj i PHP modul za Apache su besplatni i slobodni za korištenje. Instalacija poslužitelja Apache sa modulom za PHP za vlastito računalo dolazi u paketu XAMPP (Windows, Linux). Većina distribucija Linuxa dolazi sa paketima za instalaciju Apachea i PHP-a. Postoje i moduli za instalaciju PHP-a za druge poslužitelje (npr. Microsoft IIS), o čemu detaljnije provjerite na stranicama PHP-a.

Preporučeni način je korištenje poslužitelja instaliranog na računalo pinus.cc.fer.hr. Instalirane su verzije Apache 2.2.8 i PHP 5.2.5.

Korištenje računala pinus:

1. Pinus.cc.fer.hr je Unix računalo.
2. Na pinus se prijavljujete **korištenjem korisničkom imena i lozinke** koju koristite i za računala sa MS Windows. Za prijavu putem terminala možete koristiti PuTTY terminal i SSH vezu.
3. Za probleme s prijavljivanjem javite se CIP-u.

4. Kad se prijavite, u korisničkom kazalu (pretpostavimo da je korisničko ime ab12345) ~ab12345 stvorite poddirektorij public_html unutar kojeg ćete držati sve datoteke svojih stranica weba.

```
$> mkdir public_html
```

5. Prava pristupa na vašem korisničkom direktoriju, direktoriju public_html i svim ostalim direktorijima i datotekama koji trebaju biti dostupni putem weba trebaju biti:

- Za direktorije – korisnik: rwx, grupa: r-x, ostali: r-x (755)
- Za datoteke – korisnik:rw-, grupa: r--, ostali: r-- (644)

```
$> chmod 755 ~  
$> chmod 755 public_html  
$> chmod 644 public_html/index.html
```

6. Pristup datotekama najlakše je ostvariti programom koji podržava SCP – npr. WinSCP. WinSCP možete podesiti da koristi vanjski editor za uređivanje datoteka (isti onaj koji ste koristili u prethodnim vježbama) pa tako možete bezbolno raditi sa datotekama na pinusu.

7. Pristup public_html direktoriju kao web stranicama:

http://pinus.cc.fer.hr:4790/~korisnicko_ime/

npr. za korisnika ab12345:

<http://pinus.cc.fer.hr:4790/~ab12345/>

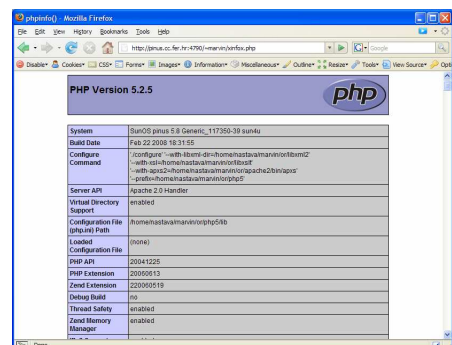
Napomena: *www poslužitelj na računalu pinus aktivan je za potrebe laboratorijskih vježbi. Još jednom, da bi vaši dokumenti bili dostupni www poslužitelju, dozvole čitanja i izvršavanja moraju biti postavljene na vašoj osnovnoj mapi, public_html mapi i na svim datotekama unutar public_html mape!*

Rad s PHP-om

Za provjera radi li PHP modul ispravno, stvorite PHP skriptu *info.php* sadržaja:

```
<?php phpinfo(); ?>
```

Pristupite li toj stranici preglednikom (<http://pinus.cc.fer.hr:4790/~ab12345/info.php>) trebali bi vidjeti nešto slično ovoj slici.



PHP skripta za pretraživanje treba se pozivati iz obrasca za postavljanje uvjeta pretrage, što možete zadati u definiciji obrasca:

```
<form action="pretraga.php" ...>
```

Podaci uneseni u obrazac vidljivi su u PHP skripti u globalnim varijablama `$_GET`, `$_POST` (u ovisnosti o tome koja metoda se koristi za prijenos podataka iz obrasca), odnosno `$_REQUEST` (neovisno o metodi).

Npr., ako u obrascu imate tekstovno polje imena 'ime', tekst upisan u to polje bit će sadržan u:

```
$_REQUEST['naslov']
```

Za provjeru postoji li neka varijabla/podatak i ima li postavljenu vrijednost možete koristiti jednu od sljedećih naredbi:

```
isset( $varijabla );  
empty( $varijabla ); // npr. if( !empty( $_REQUEST['ime'] )
```

Uključivanje vanjske datoteke (npr. datoteke s opisom funkcija) izvodi se naredbom `include`:

```
include( 'funkcije.php' );  
include_once( 'funkcije.php' );
```

Za ispis vrijednosti varijable korisne su funkcije `print_r` i `var_dump`:

```
print_r( $_REQUEST );  
var_dump( $_REQUEST );
```

Inicijalizacija DOM i učitavanje XML dokumenta kao DOM stabla:

```
$dom = new DOMDocument();  
$dom->load( 'podaci.xml' );
```

Pristupanje podelementima nekog čvora:

```
foreach( $dom->childNodes as $element )  
{  
    ...  
}
```

Pristupanje podelementima nekog čvora prema oznaci:

```
foreach( $dom->getElementsByTagName( 'element' ) as $element )  
{  
    ...  
}
```

Za atribut vrlo slično:

```
$element->getAttribute( 'atribut' );
```

Inicijalizacija XPath procesora i postavljanje upita:

```
$xp = new DOMXPath( $dom );  
$rezultat = $xp->query( "/korijen/element" );
```

Funkcija za stvaranje XPath upita treba provjeriti koji su sve parametri za pretraživanje zadani i formirati XPath upit na osnovu njih. Zbog izrade složenih upita preporučamo da elemente upita dodajete u polje:

```
if( !empty( $parametri['boja'] ))
    $upit[] = "boja='" . $parametri['boja'] . "'";
```

Jednom formirano polje pretvorite u string funkcijom *implode* koja spaja članove polja u niz znakova:

```
$xpath_upit = implode( " and ", $upit );
```

Rezultati upita pohranjeni su u objekt DOMNodeList, a pojedinim elementima može se pristupiti petljom *foreach*:

```
foreach( $rezultat as $cvor )
{
    ...
}
```

Postojeću XSL datoteku s opisom transformacije i prikaza XML dokumenta možete iskoristiti kao predložak za generiranje rezultata PHP skripte. Tako se, na primjer, iteriranje po podelementima nekog elementa izvodi petljom *foreach*:

```
<xsl:for-each select="element"> ... </xsl:for-each>

foreach( $lista as $element )
{
    ...
}
```

Dohvaćanje sadržaja trenutnog elementa XML-a izvodi se svojstvom *nodeValue*, a ime svojstvom *nodeName*:

```
<xsl:value-of select="." />

$element->nodeValue;
$element->nodeName;
```

Dohvaćanje podelementa moguće je prema imenu oznake ili ID-u, npr. za podelement *boja* koji se pojavljuje samo jednom:

```
$element->getElementsByTagName( 'boja' )->item(0)->nodeValue;
```

Napomene

Kao i u drugim jezicima, unutar iteratora (*foreach*), nije dozvoljeno mijenjanje sadržaja kolekcije po kojoj se iterira (mogući su neočekivani rezultati). Ukratko, ako iterirate po listi čvorova, nije dozvoljeno ujedno ih i brisati iz liste:

```
foreach($roditelj->getElementsByTagName( 'boja' ) as $cvor)
{
    $roditelj->removeChild( $cvor ); // nece dati zeljene rezultate
}
```

Ako je ovo potrebno, može se upotrijebiti posebno polje u koje se pohranjuju svi elementi za brisanje i naknadno se obrišu u posebnoj petlji:

```
$lista = array();

foreach($roditelj->getElementsByTagName( 'boja' ) as $cvor)
    if( treba_brisati( $cvor ))
        $lista[] = $cvor; // dodaj u listu za brisanje
foreach($lista as $brisi)
    $roditelj->removeChild( $brisi ); // stvarno obrisi
```

Ispitno gradivo vježbe

Ispitno gradivo uključuje sve navedeno u pripremi za vježbu, te detaljno razumijevanje napisanog rješenja i snalaženja u prepravcima istog.

Primjeri pitanja:

1. Što je to PHP, što DOM, a što XPath?
2. Kako se PHP kôd uključuje u HTML dokument?
3. Nabrojati neke od programskih jezika za Web aplikacije koji se izvršavaju na strani poslužitelja.
4. Što se događa kada korisnik pošalje zahtjev za stranicu koja ima ekstenziju ".php"?
5. Kako se razdvajaju HTML i PHP dijelovi u istoj tekstualnoj datoteci?
6. Koje sve postavke moraju biti prisutne da bi korisnik mogao postaviti PHP program u svoju osobnu mapu (public_html)?
7. Kako PHP programeru stavlja na uporabu ulazne parametre proslijeđene od strane obrasca?
8. Koji sve načini prijenosa parametara između obrasca i PHP skripte poznajete? Koja je razlika u mehanizmima koji se koriste?
9. Kako se struktura XML dokumenta odražava u DOM modelu?
10. Kako se pristupa pojedinim elementima DOM-om?
11. Koja je sličnost XSL oznaka za filtriranje i DOM-a kako je korišten u PHP-u?
12. Usporedite XPath i jezike za rad s podacima (SQL)?
13. Koju bi strategiju pisanja ovog primjera odabrali, kada bi ovi podaci bili u nekoj od baza podataka, umjesto u XML datoteci?

Linkovi i literatura

- W3Schools XPath Tutorial - <http://www.w3schools.com/Xpath/>
- Službene stranice PHP-a - <http://www.php.net>
- Dokumentacija za PHP – polja - <http://www.php.net/manual/en/ref.array.php>
- Dokumentacija za PHP – stringovi - <http://www.php.net/manual/en/ref.strings.php>
- Dokumentacija za PHP – DOM - <http://www.php.net/manual/en/ref.dom.php>
- XAMPP - <http://www.apachefriends.org>