Uvod u Veb i Internet tehnologije
Serversko programiranje i PHP

Milan Banković

Veb serveri

- Veb server je program koji na zahtev klijenta (tipično, veb pregledača) putem mreže (preko HTTP protokola) isporučuje traženi veb sadržaj
- Isporučeni sadržaj su po pravilu HTML strane, kao i prateći fajlovi:
  - CSS stilski listovi
  - multimedijalni fajlovi (slike, video zapisi, zvučni zapisi)
  - Javascript fajlovi (tj. kôd koji se izvršava na strani klijenta)
- Ovaj sadržaj nazivamo statičkim sadržajem, jer se on ne može prilagođavati zahtevu korisnika
  - Statički sadržaj se po pravilu čuva u obliku fajlova na disku servera i isporučuje po potrebi
  - Statički sadržaj se menja ručno, tako što osoba koja održava veb sajt ažurira fajlove na serveru
- Najpoznatiji veb serveri današnjice: Apache (slobodan softver), Nginx (slobodan softver) i Microsoft IIS (vlasnički softver)

Apache veb server

- Apache Software Foundation (`https://httpd.apache.org/`)
- Podešavanja: glavni fajl se zove http.conf ili apache.conf (može imati i dodatne fajlove koji se mogu uključiti iz glavnog fajla)
- Neke značajne konfiguracione opcije:
  - Listen [broj_porta]: port na kome se čekaju zahtevi klijenata (podrazumevano 80)
  - ServerName [domen:port]: naziv servera (DNS naziv)
  - DocumentRoot [direktorijum]: lokalna putanja na serveru na kojoj se nalazi veb sadržaj
  - DirectoryIndex [fajlovi]: podrazumevani fajlovi u direktorijumu
  - LoadModule [modul]: učitavanje modula
  - Include [putanja]: uključivanje dodatnog konfiguracionog fajla
- Opcije se mogu odnositi samo na pojedine direktorijume
  - U konfiguracionom fajlu moguće je lokalizovati dejstvo opcije
  - U samim direktorijumima se mogu nalaziti .htaccess fajlovi za redefinisanje pojedinih opcija

Dinamičke veb strane

- Dinamičke veb strane predstavljaju veb sadržaj koji se može dinamički prilagođavati zahtevu klijenta
- Tipično, klijent u sklopu HTTP zahteva šalje odgovarajući upit (engl. query) sa vrednostima parametara zahteva
- Upit je string oblika `ime1=vrednost1&ime2=vrednost2&...&imen=vrednostn`
- Upit se može zadati kao deo URL-a (iza znaka '?', kod GET zahteva), ili u telu HTTP zahteva (kod POST zahteva)
- Veb server tipično pokreće eksterni program kome predaje informacije o zahtevu klijenta
- Eksterni program obrađuje zahtev i formira HTML stranu koju prosleđuje klijentu
- Komunikacija eksternog programa sa veb serverom: CGI protokol
- Eksterni program može pristupati bazama podataka (na istom serverskom računaru ili na drugom računaru)

CGI protokol

- Common Gateway Interface (CGI): standardni interfejs za komunikaciju veb servera i eksternih programa
- Eksterni programi koji se pokreću putem CGI protokola se nazivaju CGI programi (ili CGI skriptovi)
- CGI programi se mogu pisati u bilo kom programskom jeziku (poput C-a, C++-a, Perl-a, PHP-a, ...)
- Veb server prosleđuje informacije CGI programu putem:
  - promenljivih okruženja (engl. environment variables) (npr. `QUERY_STRING`, `REMOTE_ADDR`, `REQUEST_METHOD`, `HTTP_COOKIE`, `HTTP_USER_AGENT`...)
  - standardnog ulaza (telo HTTP zahteva kod POST metoda)
- CGI program prosleđuje generisani sadržaj veb serveru putem standardnog izlaza:
  - najpre se ispisuju zaglavlja HTTP odgovora
  - nakon toga se ispisuje jedan prazan red, a zatim i HTML sadržaj
  - veb server dobijeni izlaz prosleđuje klijentu, uz izvesne modifikacije (server tipično može dodati još neka zaglavlja)

PHP jezik

- Generisanje dinamičkih veb strana pomoću uobičajenih jezika opšte namene prilično je naporan posao
- Zbog toga su nastali jezici koji su prilagođeni toj nameni
- Jedan od najpoznatijih jezika tog tipa je PHP:
  - Personal Home Page (PHP): Rasmus Lerdorf, 1994.
  - PHP Hypertext Processor (PHP): Suraski, Gutmans: 1997.
  - Zend Engine (referentna implementacija): 1999.
  - Za razvoj jezika zadužena PHP grupa
  - Zend Technologies: kompanija koja razvija Zend Engine
  - Trenutna verzija: PHP 7.3
  - `http://www.php.net/`

PHP osnove

- Dinamički tipiziran interpretirani jezik, C-olike sintakse
- Kôd jezika je ugnježden u HTML kôd
- Izvršava se na serveru i generiše dinamički HTML sadržaj koji se zatim šalje klijentu
- Tokom izvršavanja PHP programa možemo se nalaziti u dva režima:
  - PHP režim: sve između tagova `<?php` i `?>` se smatra PHP kodom i izvršava se na serveru (ispis na standardni izlaz se šalje klijentu kao deo HTML strane)
  - HTML režim: sve van tagova `<?php` i `?>` se smatra regularnim HTML kodom i prosto se ispisuje na izlaz (tj. šalje se klijentu kao HTML kôd)
- PHP interpretator može se pokretati kao eksterni program (preko CGI interfejsa) ili kao apache modul (mnogo efikasnije, samim tim uobičajeno)
- Komentari u PHP-u:
  - jednolinijski: `//` i `#`
  - višelinijski: `/* */`

Konfiguracija PHP-a

- Fajl php.ini
- Opcije su oblika: `naziv_atributa=vrednost;`
- Obično su sve opcije dobro dokumentovane u samom fajlu, kao i na PHP sajtu
- Neke od opcija se mogu redefinisati na nivou direktorijuma: .user.ini fajl
- Neke od opcija se mogu redefinisati i za svaki skript ponaosob (PHP funkcija ini_set())
- Jedna od zgodnih opcija: `short_open_tag=On;` omogućava da se za ulazak u PHP režim koristi `<?` umesto `<?php`

Prvi primer

```
<!DOCTYPE html>
<html>
 <head>
   <title>Prvi primer</title>
 </head>
 <body>
 <?php
    echo "Zdravo, ja sam PHP skript!";
  ?>
 </body>
</html>
```

Drugi primer

```
 <!DOCTYPE html>
<html>
 <head>
   <title>Drugi primer</title>
 </head>
 <body>
 <?php
  if($_GET['broj'] > 10)
  {
   ?>
     <p> Uneli ste veliki broj</p>
     <?php
  }
  else
   {
     ?>
     <p> Uneli ste mali broj</p>
    <?php
   }
  ?>
 </body>
</html>
```

Treći primer

```
 <!DOCTYPE html>
<html>
 <head>
   <title>Treci primer</title>
 </head>
 <body>
 Uneli ste broj: <?= $_GET['broj'] ?>
 </body>
</html>
```

Ispis

- Za ispis podataka koriste se funkcije:
  - echo – nema povratnu vrednost, može da ima više argumenata
  - print – ima povratnu vrednost 1, može da ima samo jedan argument
  - printf – formatirani ispis, slično kao u C-u
- echo i print nisu zaista funkcije, već jezički elementi, pa se mogu pozivati sa zagradama i bez:
  `echo` ili `echo()`
- printf je prava funkcija i može se pozivati samo sa zagradama:
  `printf()`

Tipovi podataka

- U PHP-u su podržani naredni tipovi podataka:
  - string – niske, mogu se koristiti jednostuki ili dvostruki navodnici za zapis konstanti
  - integer – podržan dekadni, oktalni i heksadekadni zapis konstanti (kao u C-u)
  - float – realni brojevi u pokretnom zarezu (tipično dvostruke tačnosti)
  - boolean – logički tip: vrednosti `true` i `false`
  - array – nizovi, zadaju se sa array(...)
  - klasni tipovi – najpre je neophodno definisati klasu objekata ključnom rečju class (po ugledu na Javu)
  - NULL – specijalni tip podataka sa samo jednom vrednošću NULL koja označava da promenljiva nema vrednost
- Funkcija var_dump() vraća tip i vrednost promenljive

PHP promenljive

- U PHP-u promenljiva počinje znakom $ za kojim sledi njen naziv
- Naziv promenljive mora počinjati slovom ili podvlakom, a svaki naredni karakter može biti slovo, podvlaka ili cifra
- Promenljive u PHP-u se ne deklarišu, već se kreiraju prilikom prve dodele vrednosti toj promenljivoj
  - tip promenljive se određuje na osnovu njene vrednosti
  - tip promenljive se može menjati tokom njenog životnog veka (dinamički tipiziran)
- Postoje tri opsega važenja promenljive:
  - *globalni* – promenljivoj deklarisanoj van funkcije može se pristupiti samo van funkcije; da bi se globalnoj promenljivoj pristupilo iz funkcije mora se navesti ključna reč global
  - *lokalni* – promenljivoj deklarisanoj unutar funkcije može se pristupiti samo iz funkcije;
  - *statički* – lokalna promenljiva koja može da nadživi izvršavanje funkcije; deklariše se korišćenjem ključne reči static

Superglobalne promenljive

- Obične globalne promenljive moraju biti lokalno deklarisane pomoću ključne reči global da bi se mogle koristiti unutar funkcije
- Za razliku od njih, superglobalne promenljive su uvek dostupne u svakoj tački programa i ne moraju se deklarisati eksplicitno
- Većina superglobalnih promenljivih su nizovskog tipa
- Neke od najznačajnijih su: `$GLOBALS`, `$_SERVER`, `$_ENV`, `$_GET`, `$_POST`, `$_COOKIE`, `$_SESSION`, `$_FILES`
- O nameni navedenih superglobalnih promenljivih će više reči biti kasnije

Konstante

- Naziv konstante ne počinje znakom $
- Konstanta se definiše naredbom: define(naziv, vrednost, case-insensitive)
- Vrednost konstante mora biti skalarnog tipa (`boolean`, `float`, `integer`, `string` ili `NULL`)
- Trećim parametrom funkcije `define` zadaje se da li naziv konstante treba da bude case-insensitive, podrazumevana vrednost je `false`
- Konstante su automatski sa globalnim opsegom važenja i mogu se koristiti u celom skriptu

Rad sa niskama

- Niska je niz karaktera;
- Karakterima se može pristupati pomoću indeksa (indeks početnog karaktera je 0): `$s[0]`
- Karakterske konstante:
  - Jednostruki navodnici: svi karakteri se doslovno tumače (osim \ i ’)
  - Dvostruki navodnici: dozvoljene su sekvence poput `\n`, `\t`, kao i supstitucija promenljivih (`"Zdravo: $ime\n"`)
- Neke od najčešće korišćenih funkcija za rad sa niskama su:
  - strlen(s) – vraća dužinu niske *s*
  - str_word_count(s) – vraća broj reči u niski *s*
  - strrev(s) – obrće nisku *s*
  - strpos(s1,s2) – traži prvo pojavljivanje niske *s*2 u niski *s*1, ako ne nađe vraća `false`
  - substr(s, pos, len) – vraća podstring počev od pozicije `pos` dužine `len`
  - str_replace(s1,s2,s) – u niski *s* svako pojavljivanje niske *s*1 menja niskom *s*2

Operatori

- Aritmetički operatori: +, -, *, /, % (ostatak pri deljenju), ** (stepena funkcija)
- Operatori dodele: =, +=, -=, *=, /=, %=
- Relacioni operatori: ==, === (jednaki i istog tipa), !=, <>, !== (nisu identični), >, <, >=, <=
- Operatori inkrementiranja/dekrementiranja: ++,−− (prefiksni i postfiksni)
- Logički operatori: and, or, xor, &&, ||, !
- Operatori nad stringovima: . (nadovezivanje), .=
- Nizovski operatori: + (unija), == (isti skup vrednosti ključ/vrednost), === (isti skup vrednosti ključ/vrednost u istom redosledu i istih tipova), !=, (<>), !==
- `instanceof` operator (kao u Javi)

Kontrolne strukture

- Sintaksa ista kao u C-u (if-else, for, while, do-while)
- U slučaju višestrukog if-else-if-a, moguće je umesto 'else if' pisati 'elseif'
- Postoji i foreach, za iteraciju kroz nizove i objekte

Funkcije

- Mogu se kreirati korisnički definisane funkcije; ne izvršavaju se prilikom učitavanja strane, već pozivom funkcije
- Osnovna sintaksa:

```
function nazivFunkcije(<lista_parametara>){
  <naredbe>
}
```

- Parametri funkcije se navode kao promenljive razdvojene zarezima
- Mogu se zadati podrazumevane vrednosti parametara na način: `$parametar = vrednost` (kao u C++-u)
- Parametri se podrazumevano prenose po vrednosti
- Za prenos po referenci, koristi se simbol & ispred naziva parametra
- Funkcija vraća vrednost naredbom: `return <izraz>;`

Nizovi

- Nizovi u PHP-u predstavljaju mape, tj. skup parova ključ/vrednost
- Ključevi mogu biti celobrojnog ili stringovskog tipa
- Vrednosti mogu biti bilo kog tipa
- Vrednosti koja je pridružena ključu se može pristupiti indeksnom sintaksom (npr. `$array[0]` ili `$array['str']`
- Ako su svi ključevi celobrojni, tada oni odgovaraju standardnim nizovima (poput onih iz C-a)
- Primeri kreiranja nizova:

```
// niz elemenata 4, 5, 'hello' (na indeksima 0, 1, 2)
$array1 = [4, 5, 'hello'];
// isto kao i malopre
$array2 = array(4, 5, 'hello');
// mapa koja sadrzi dva kljuca 'ime' i 'prezime'
$array3 = array('ime' => 'Petar', 'prezime' => 'Petrovic');
```

- Funkcijom count() može se dobiti broj elemenata niza (tj. broj parova ključ/vrednost)
- Moguće je mešati celobrojne i stringovske ključeve (mada se to uglavnom ne radi)
- Sintaksa: `$array[] = <vrednost>;` se koristi za dodavanje na kraj niza (tj. ključ će biti prvi sledeći celobrojni indeks)

Primer iteracije kroz niz

```
$array = array(4, 5, 'hello');
$n = count($array);
// prosta iteracija for petljom (kao u C-u)
for($i = 0; $i < $n; $i++)
  echo "Element niza: {$array[i]}\n";
// Iteracija foreach petljom
foreach($array as $el)
 echo "Element niza: $el\n");
// Iteracija kroz mapu (kljuc/vrednost)
$array2 = array('ime' => 'Petar', 'prezime' => 'Petrovic');
foreach($array2 as $key => $value)
  echo "Kljuc: $key, vrednost: $value\n";
// Iteracija kroz mapu uz mogucnost menjanja vrednosti
$array3 = array('rec1' => 1, 'rec2' => 5, 'rec3' => 2);
foreach($array3 as $key => &$value)
  $value++;
```

Klase

- Podrška za OOP slična kao u Javi
- Ključna reč class definiše klasu
- Uz svaki član klase stoji private, protected ili public
- Podaci članovi klase (svojstva): počinju sa $, mogu imati opcioni konstanti inicijalizator
- Funkcije članice klase (metode): funkcije unutar klase
  - podatku članu klase `$x` unutar metode pristupamo sa `$this->x`
  - metoda `f` klase se unutar druge metode klase poziva sa `$this->f()`
  - izvan klase se podacima (metodama) pristupa preko objekta (`$obj->x`, `$obj->f()`)
- Statički podaci i metode: ključna reč static
  - statičkom podatku članu klase `$x` unutar statičke metode se pristupa sa `self::$x`
  - statička metoda klase `f` se unutar druge metode poziva sa `self::f()`
  - izvan klase se statičkim podacima (metodama) pristupa preko klase (`ImeKlase::$x`, `ImeKlase::f()`)
- Konstruktor: metoda __construct(), destruktor: metoda __destruct()
- Objekti se kreiraju pomoću ključne reči new, kao i u Javi.
  - operator `new` vraća ručku na objekat na hipu (kao u Javi)

Nasleđivanje klasa, interfejsi, polimofrizam

- Nasleđivanje kao u Javi (ključna reč extends)
- Zabrana izvođenja iz klase: ključna reč final
- Pristup članu bazne klase: `parent::$x`
- Poziv metode bazne klase: `parent::f()`
  - konstruktor: `parent::__construct(args)` (mora se pozvati iz konstruktora izvedene klase ako želimo da bude izvršen)
  - destruktor: `parent::__destruct()` (mora se pozvati iz destruktora izvedene klase, ako želimo da bude izvršen)
- Interfejsi: isto kao u Javi (ključne reči interface i implements)
- Polimorfizam: isto kao u Javi (redefinisana metoda iz izvedene klase se automatski poziva)
- Apstraktne klase i metode: isto kao u Javi ključna reč abstract)

Veb interfejs PHP skripta

- PHP skript koji se izvršava u okviru veb servera dobija na raspolaganje sve informacije u vezi HTTP zahteva, klijenta, servera i radnog okruženja u obliku superglobalnih promenljivih
- Sve ove promenljive su nizovskog tipa (mape koje ključevima koji su stringovi pridružuju vrednosti)
- Upit se u rašlanjenom obliku nalazi u promenljivoj `$_GET` ili `$_POST` u zavisnosti od tipa zahteva
- Informacije o klijentu, serveru i okruženju se nalaze u promenljivoj `$_SERVER`
- Informacije o sesiji se nalaze u promenljivoj `$_SESSION`
- Informacije o kolačićima se nalaze u promenljivoj `$_COOKIE`
- `$_REQUEST`: unija `$_GET`, `$_POST` i `$_COOKIE`

PHP i MySQL

- PHP podržava rad sa više različitih SURBP sistema, među kojima je i MySQL
- U starijim verzijama PHP-a interfejs se sastojao iz niza funkcija sa prefiksom mysql_*:
  - mysql_connect(),mysql_query(), mysql_select_db(), ...
  - ovaj interfejs je konačno izbačen u verziji PHP 7
- U novijim verzijama PHP-a se koristi objektno-orijentisani interfejs zasnovan na klasi mysqli

```
$db = new mysqli('localhost', 'user', 'password', 'my_db');
$db->set_charset("utf8");
$res = $db->query("SELECT * FROM Nastavnik");
while(($row = $res->fetch_row()))
  echo "id: $row[0], ime: $row[1], prezime: $row[2]\n";
// $res->fetch_assoc() vraca asocijativni niz (mapu)
// $res->fetch_object() vraca objekat koji odgovara vrsti
```

- Za one koji ne vole klase, postoji i funkcijski alijasi:

```
$db = mysqli_connect("localhost", "user", "password", "my_db");
mysqli_set_charset($db, "utf8");
$res = mysqli_query($db, "SELECT * FROM Nastavnik");
while(($row = mysqli_fetch_row($res))
  echo "id: $row[0], ime: $row[1], prezime: $row[2]\n";
```