

Programski jezik PHP

Seminarski rad u okviru kursa
Metodologija stručnog i naučnog rada
Matematički fakultet

Dorđe Vučković, Tamara Ivanović,
Petar Simić, Stefan Stevović
djordjevuckovic996@gmail.com, tamara.d.ivanovic@gmail.com,
petar_95@yahoo.com, stefanstevovic@yahoo.com

30. april 2019

Sažetak

U ovom radu su ukratko predstavljene osnovne osobine i specifičnosti programskog jezika PHP. Ovde se možete informisati o razvojnom putu samog jezika, njegovim mogućnostima i primenama. Možete saznati koja su najpoznatija okruženja za programski jezik PHP kao i kako obaviti proces instalacije na Linux i Windows operativnim sistemima. Sam izgled PHP kôda je prikazan kroz primere koji prate ovaj rad.

Sadržaj

1	Uvod	2
2	Razvoj programskog jezika PHP	2
3	Mogućnosti programskog jezika	3
4	Osnovne osobine, podržane paradigme i koncepti	5
4.1	Implementacije PHP-a	5
4.2	Podržane paradigme	6
5	Najpoznatija okruženja	6
6	Instalacija	7
6.1	Windows OS	7
6.2	Linux OS	7
7	Primer kôda	8
8	Specifičnosti	9
8.1	Imenski prostor i forme	9
8.2	Klase PDO i DateTime	10
8.3	Kolačići i keširanje	10
9	Zaključak	11
	Literatura	11

1 Uvod

PHP (Hypertext preprocessor) je popularan skriptni jezik opšte namene. Nastao je iz potrebe njegovog stvaraoca za praćenjem poseta svog veb-sajta i tokom godina sve je popularniji kod veb-programera[8]. Kako semantika ovog jezika nije striktna, lak je za učenje, što ga čini pristupačnim programerima sa različitim nivoom znanja [21].

O popularnosti jezika govore podaci analize GitHub repozitorijuma i projekata. Tokom godina ostaje u prvih 6 jezika po upotrebi. Statistika za period 2012-2014. godine je prikazana u tabeli 1. Ispred PHP-a 2014. godine su bili jezici JavaScript, Java, Python i CSS[23].

Ovaj rad je namenjen svima koji žele da saznaju osnovne karakteristike PHP-a, šta je ono što ga izdvaja od ostalih, kao i kako da ga pokrenu na svojim računarima. Veliki broj radnih okruženja (eng. framework)[18] je podržan, ali ovde je fokus stavljen na najpoznatija. Ovaj rad se ne fokusira na sintaksu i nema za cilj da čitaoca nauči kako da programira u PHP-u, ali za to preporučujemo knjige „Learning PHP“, „PHP Cookbook“ i „Programming PHP“[18, 21, 11].

Tabela 1: Učešće PHP projekata na GitHub-u i pozicija PHP-a na listi zastupljenosti, u periodu 2012. - 2014. god.

period	PHP repozitorijumi	% zastupljenosti	pozicija
2012, II kvartal	62336	11.3	4
2012, IV kvartal	62336	8.1	4
2013, II kvartal	83169	7.6	5
2013, IV kvartal	91879	7.1	6
2014, II kvartal	118272	5.6	6
2014, IV kvartal	138771	6.3	5

2 Razvoj programskog jezika PHP

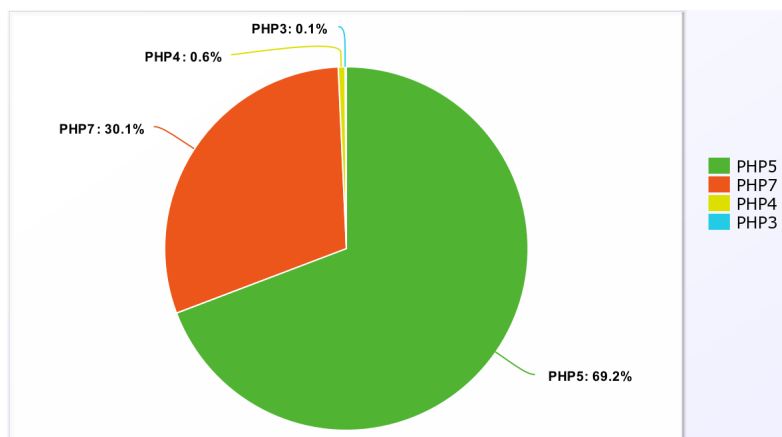
Rasmus Lerdorf je 1994. godine napravio programski jezik PHP[8], međutim danas se on značajno razlikuje od početne verzije. Za potrebe napretka svog privatnog veb-sajta i praćenja broja poseta, napisao je niz skriptova u C-u koji je nazvao „Personal Home Page Tools“, poznatije kao "**PHP Tools**". Zatim je dodao i mogućnost pristupa bazama podataka kao i kreiranje jednostavnih veb aplikacija. Sledeće godine je PHP Tools bio javno dostupan, mnogi programeri su pomogli da se uklone bagovi i da sam alat napreduje. U septembru iste godine je izbačeno PHP iz imena i alat je dobio naziv **FI**(eng. Forms Interpreter). Promenljive i sintaksa su bile slične programskom jeziku Perl, za ugnježdavanje koda u HTML fajl bilo je potrebno koristiti HTML komentare. Daljim razvojem i kompletnom izmenom kôda 1996. godine Lerdorf je objavio **PHP/FI** [7] koji je od niza alata evoluirao u programski jezik. Tokom iste godine dobio verziju 2.0.

Zeev Suraski i **Andi Gutmans** su 1997. godine redizajnirali PHP/FI jezgro i nakon toga počeli da saraduju sa Lerdorfom na pisanju novog programskog jezika. Preimenovan je u PHP(Hypertext Preprocessor), sa promenjenim značenjem jer više nije bilo ograničenja za privatnu upotrebu. Ovo je predstavljalo verziju 3.0. U PHP 3.0 je bila uključena

podrška objektno-orijentisanom programiranju, sintaksa jezika je postala konzistentna, a krajnjim korisnicima je omogućen interfejs za API-je, protokole i višestruke baze podataka. U trenutku svog najvećeg uspeha PHP 3.0 je bio instaliran na približno 10% veb servera na Internetu[8].

Andri Gutmans i Zeev Suraski su kreirali “**Zend Engine**” [14] čija je uloga bila da popravi performanse kompleksnih aplikacija, kao i modularnost osnove PHP kôda. Zend Engine je zadužen za leksičku analizu, sintaksno parsiranje i prevođenje koda u bajtkod. Takođe upravlja memorijom i sadrži sakupljač otpadaka. Manipuliše greškama i izuzecima i zadužen je za upravljanje tipovima i promenljivima. Spoj Zend Engine-a i velikog broja dodatnih karakteristika čine PHP 4.0. Ovu verziju odlikuju značajno poboljšane performanse, podrška HTTP sesijama [5] i još većem broju veb servera. Takođe, dodati su i novi jezički konstrukti.

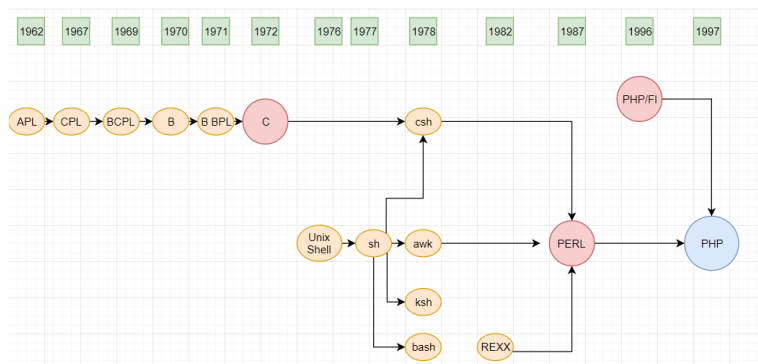
Verzija 5.0 je izašla 2004. godine, a godinu dana kasnije verzija 7.0. Ove dve verzije su najzastupljenije danas, dok prethodne verzije veoma mali broj sajtova koristi. Ovo se može videti na slici 1 koja prikazuje podatke iz 2019. godine prema podacima [22]. Takođe, na slici 2 su prikazani programski jezici koji su imali uticaj na razvoj PHP-a.



Slika 1: Procenat sajtova koji koriste određenu verziju PHP-a od ukupnog broja sajtova koji koriste PHP (mart 2019)

3 Mogućnosti programskog jezika

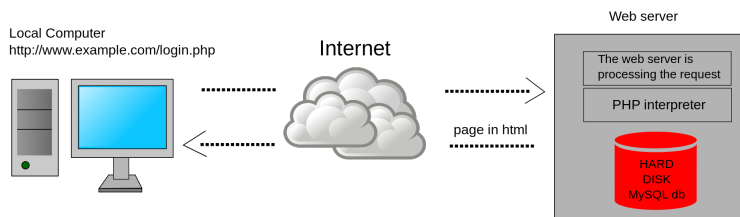
PHP je jezik koji je posebno pogodan za veb razvoj na strani servera, u većini slučajeva PHP radi na veb servisu. Može se koristiti na većini veb servera, mnogim operativnim sistemima i platformama, kao i na mnogim sistemima za upravljanje relacionim bazama podataka (RDBMS). Većina veb hosting provajdera podržava PHP za upotrebu od strane svojih klijenata. Dostupan je besplatno, a PHP grupa obezbeđuje kompletan izvorni kôd za korisnike koji će graditi, prilagoditi i proširiti ga za sopstvenu upotrebu. Svaki PHP kôd u datoteci izvršava PHP runtime[8], obično da bi kreirao dinamičan sadržaj veb stranica ili dinamičke slike koje se koriste na veb stranicama. Takođe se može koristiti za pisanje skriptova komandne linije (CLI) kao i za aplikacije grafičkog korisničkog interfejsa na strani klijenta (GUI). Interfejs komandne linije (CLI) je sredstvo za interakciju



Slika 2: Razvojno stablo PHP-a

sa kompjuterskim programom gde korisni (ili klijent) izdaje naredbe programu u obliku uzastopnih redova teksta (komandne linije). Programi sa interfejsima komandne linije su uglavnom laki za automatizaciju putem skriptovanja.

Iako je izvorno dizajniran za korišćenje dinamičkih veb stranicam PHP se sada fokusira uglavnom na pisanje skriptova na strani servera i sličan je drugim skriptnim jezicima na strani servera koji pružaju dinamički sadržaj sa veb servera klijentu. PHP je takođe privukao razvoj mnogih softverskih okvira koji pružaju gradivne blokove i strukturu dizajna za promovisanje brzog razvoja aplikacija (RAD). Brzi razvoj aplikacija je opšti termin koje se koristi za adaptivni pristup razvoju softvera i posebno je pogodan za razvoj softvera koji se pokreće zahtevima korisničkog interfejsa. Neki od njih uključuju PRADO, CakePHP, Simfoni, CodeIgniter, Laravel, Framework, Phalcon i Zend Framework, nudeći mogućnosti slične veb okvirima[15]. Kako dinamičko skriptovanje funkcioniše vidi se na slici 3.



Slika 3: Dinamička veb stranica: primer serverskog skriptiranja¹

Za specifično i naprednije korišćenje, PHP nudi dobro definisan i dokumentovan način za pisanje prilagođenih ekstenzija u C ili C++ koje dodaju funkcionalnost jeziku. Ekstenzije se mogu kompajlirati statički u PHP-U ili dinamički učitati tokom izvršavanja. Brojne ekstenzije su napisane kako bi dodale podršku za Windows API i upravljanje procesima

¹Slika preuzeta sa https://commons.wikimedia.org/wiki/File:Scheme_dynamic_page_en.svg?fbclid=IwAR1y_01iNJcHskh61pmuslMTxkVV3YZ9EERM8JYr8GSXc9vJGDiXetoVGI

na Unix zasnovanim operativnim sistemima. Osim proširivanja samog jezika u obliku dodatnih biblioteka, proširenja pružaju način za poboljšanje brzine izvršavanja tamo gde je ona kritična, a postoji i prostor za poboljšanja korišćenjem kompajliranog jezika. Kompajlirani jezik je programski jezik čija je implementacija obično kompajler (prevodioci koji generišu mašinski kod iz izvornog koda). PHP takođe nudi dobro definisane načine za ugrađivanje u druge softverske projekte. Na taj način se PHP može lako koristiti kao interni skriptni jezik za drugi projekat, a takođe obezbeđuje usko povezivanje sa specifičnim internim strukturama podataka projekta.

4 Osnovne osobine, podržane paradigme i koncepti

PHP je sistem koje je svestan interneta i njegovih ugrađenih modula za pristup FTP serverima. Protokol za prenos datoteka (FTP) je mrežni protokol koji služi za prenos datoteka između klijenta i servera na računarskoj mreži. Takođe ima ugrađene module i za pristup raznim serverima baze podataka, uključujući MySQL, Microsoft SKL Server (sistem za upravljanje relacionim bazama podataka koje je razvio Microsoft) i SKLite (koje je ugrađena baza podataka), LDAP (protokol lakog pristupa direktorijumu) servere i druge.

Neke dodatne PHP karakteristike su dostupne putem proširenja koja uključuju integraciju sa IRC-om (IRC je protokol aplikativnog sloja koji olakšava komunikaciju u obliku teksta. Proces ćaskanja radi na modelu umrežavanja klijent / server), dinamičku generaciju slika i Adobe Flash sadržaja, PHP Data Objects (PDO je objektno orijentisan interfejs za interakciju sa bazama podataka) kao sloj apstrakcije koji se koristi za pristup bazama podataka, pa čak i sintezu govora. Neke od osnovnih funkcija jezika, kao što su one koje se bave nizovima, takođe su implementirane kao proširenja[1].

Drugi projekti, kao što je **Zephir**[19], pružaju mogućnost da PHP ekstenzije budu kreirane na jeziku visokog nivoa i kompajlirane u izvorne PHP ekstenzije. Takav pristup, umesto pisanja PHP ekstenzija direktno u C, pojednostavljuje razvoj ekstenzija i skraćuje vreme potrebno za programiranje i testiranje.

4.1 Implementacije PHP-a

Originalna i najraširenija implementacija PHP-a pokreće Zend Engine i poznata je kao PHP. Da bi se razdvojila od drugih implementacija, ponekad se nezvanično zove „Zend PHP”. PHP-ov model „jedan-zahtev-za-skriptom-izvršenje” i činjenica da je Zend Engine interpreter dovodi do neefikasnosti. Kao rezultat toga, razvijeni su razni proizvodi koji pomažu poboljšanju performansi PHP-a. Kako bi se ubrzalo vreme izvršavanja i da se ne mora kompajlirati PHP izvorni kod svaki put kada se pristupi veb stranici, PHP skripte se mogu primeniti u unutrašnjem formatu PHP mašine korišćenjem operacionog keša. Operacioni keš funkcioniše tako što kešira kompajlirani oblik PHP skript u zajedničkoj memoriji kada se skripta pokrene. Jedan od operacionih keševa, Zend Opcode[14], je ugrađen u PHP od verzije 5.5. Drugi primer široko korišćenog operacionog keša je alternativni PHP keš (APC je besplatan i otvoren okvir koji

kešira izlaz PHP kompajlera u zajedničku memoriju)[8], koji je dostupan kao PECL ekstenzija.

Iako je Zend PHP još uvek najpopularnija implementacija, razvijeno je nekoliko drugih implementacija. Neko od njih su kompajleri ili podržavaju JIT[10] kompilaciju i stoga nude prednosti u odnosu na Zend PHP na račun nedostatka potpune PHP kompatibilnosti. Alternativne implementacije uključuju:

- **HHVM(HipHop Virtual Machine)[9]** - razvijen na Facebook-u i otvorenog koda, pretvara PHP kôd u bajt-kôd visokog nivoa, koji je obično poznat kao posredni jezik, koji se potom pretvara u k86-64 mašinski kôd. U vremenu izvršavanja kompajler je pravedan u vremenu (JIT), što rezultira poboljšanjem performansi do šest puta.
- **Parrot[4]** - virtuelna mašina dizajnirana da efikasno pokreće dimaničke jezike. Pipp(mali PHP programski okvir) transformiše PHP izvorni kôd u Parrot posrednu reprezentaciju, koja se zatim prevodi u Parrot-ov bajt kôd i virtuelna mašina ga izvršava.
- **Phalanger** - sastavlja PHP u bajt kôd Common Intermediate Language (CIL)
- **Kuercus** - kompajlira PHP u java bajt kôd.

4.2 Podržane paradigme

Već je rečeno da je PHP skriptni jezik opšte namene. Pored toga on podržava još mnoge koncepte. Od verzije 5.0 ima jaku podršku za **objektno-orijentisano programiranje**. To uključuje podršku za nasleđivanje, konstruktore, izuzeke, interfeke, klase i apstrakne klase. Takođe, podržava i funkcije prve klase(eng. first-class functions) što znači da funkcija može biti dodeljena promenljivoj. Korisnički definisane funkcije, kao i ugrađene funkcije mogu biti referisane od strane promenljive i pozivaju se dinamički. Podržava rekurziju, odnosno da funkcija poziva samu sebe, mada kod PHP-a je fokus na iteraciji. Funkcije se mogu prosleđivati kao argumenti tj. funkcije višeg reda i funkcija može vratiti druge funkcije. Od verzije 5.3 uvedena je i podrška za anonimne funkcije sa podrškom za zatvorenje. Verzija 5.4 je dodala mogućnost povezivanja zatvaranja sa opsegom objekta i poboljšanu podršku za callables[6] tako da se mogu koristiti naizmenično sa anonimnim funkcijama u gotovo svim slučajevima. Ove osobine opravdavaju pripadanje **funkcionalnoj paradigmi** [12].

5 Najpoznatija okruženja

Na samom početku ovog poglavlja treba da se upoznamo sa terminom **razvojno okruženje** (eng. framework). Razvojno okruženje predstavlja skup već gotovih komponenti koje čine skelet startne platforme. Ta platforma omogućava da se bude brži i efikasniji pri pisanju programa. Najbitniji faktor je to što se povećava brzina programiranja jer se ne počinje svaki put od nule već su na raspolaganju raznorazni gotovi moduli koji se mogu jednostavno koristiti. Razvojno okruženje čini rad na zahtevnim projektima jednostavnijim, omogućava pisanje čistog i ponovo upotrebljivog kôda i sugerise na moguće greške.

Postoje razna razvojna okruženja za svaki programski jezik pa tako i za PHP. Koje razvojno okruženje izabrati zavisi najviše od samog projekta. Za jezik PHP najpoznatija razvojna okruženja su Laravel[16], CodeIgniter[2] i Symfony[20].

Laravel programeri najviše koriste zbog njegove brzine, fleksibilnog rutiranja i mogućnosti upravljanja preko komandne linije. Odlikuje ga jednostavno i intuitivno korišćenje kao i čist i pregledan kôd. U potpunosti je baziran na MVC arhitekturi eng(Model View Controller).

CodeIgniter je razvojno okruženje koga odlikuje bezbednost, poseduje zaštitu od XSS i CSRF napada. Uopšte nije memorijski zahtevan jer zauzima svega par megabajta. Omogućava MVC arhitekturu, ali nije striktan po tom pitanju. Posедуje sveobuhvatnu dokumentaciju koja se dobija uz samo razvojno okruženje.

Pored sjajnih osobina poput brzine, fleksibilnosti, ponovno upotrebljivih komponenti, razvojno okruženje **Symfony** karakteriše izuzetna podrška. Osim podrške koju pruža sama firma Sensio, tvorca ovog razvojnog okruženja, veliki broj profesionalaca i firmi koristi upravo Symfony, pa je samim tim moguće dobiti odgovore na sva pitanja. Ovo razvojno okruženje podržava trenutne standarde PHP-a i pored toga omogućava korisnicima da koriste delove vlastitog softvera, bez da koriste celo razvojno okruženje.

6 Instalacija

Za kreiranje interaktivnih veb stranica i izvršavanje PHP programa neophodan je server koji podržava PHP. Prilikom samog razvoja bilo kog PHP projekta komunikacija sa tim serverom je veoma česta, stoga je izuzetno nepraktično stalno komunicirati sa nekim udaljenim serverom. Iz tog razloga se preporučuje instalacija lokalnog servera. Osim servera potrebna je i baza podataka. Naravno osim servera i baze, potrebno je instalirati i sam jezik PHP. Na novijim operativnim sistemima PHP uglavnom dolazi uz sam sistem, ali će ovde instalacija biti objašnjena od samog početka, kao da ništa od potrebnog nije instalirano.

Dakle, potrebni su sam PHP programski jezik, server i baza podataka. Postoji mogućnost da se zasebno svaka od navedenih stavki instalira i podešava, ali postoji i mnogo lakši način, a to su LAMP(linux apache mySql PHP) [13] i WAMP(windows apache mySql PHP) [3] serveri.

6.1 Windows OS

Na operativnom sistemu **Windows** sama instalacija se sastoji od preuzimanja WAMP servera sa stranice:<http://www.wampserver.com/en/>, zatim sledi pokretanje preuzetog fajla koje automatski pokreće instalaciju celog paketa. Sam proces nadalje je automatizovan i WAMP server paket stiže sa najnovijim verzijama Apache-a, MySQL-a i PHP-a. Nakon procesa instalacije biće kreirana prečica ka WampServer-u kao i folder www, obično na putanji c:\\wamp\\www. Nakon toga u folderu www potrebno je kreirati poddirektorijum koji će služiti za smeštanje PHP fajlova. Na primer, kreiran je fajl 1.php. Moguće je otvoriti ga u veb pregledaču, tako što se u meniju WampServer-a klikne na link "localhost" ili se u pregledaču ukuca <http://localhost/1.php>.

6.2 Linux OS

Kod operativnog sistema **Linux** instalacija se vrši preko terminala i zahteva instaliranje svake komponente pojedinačno, što je prikazano u 1.

```
sudo apt-get update #update sistema
```

```

2 sudo apt-get install apache2 #instalacija Apache-a
sudo apt-get install mysql-server libapache2-mod-auth-mysql php5-
mysql #instalacija MySQL-a
4 sudo mysql_install_db #instalacija MySQL servera
sudo /usr/bin/mysql_secure_installation #pokretanje skripte za
podesavanje MySQL-a
6 sudo apt-get install php5 libapache2-mod-php5 php5-mcrypt #
instalacija PHP-a

```

Listing 1: Pokretanje instalacije na Linux operativnom sistemu

Po završetku instalacije MySQL-a može biti traženo da se postavi root šifra što je i potrebno uraditi.

PHP takođe sadrži veliki broj korisnih biblioteka i modula koji se mogu dodati na server. Za izlistavanje potrebno je pokrenuti komandu “apt-cache search php5-” i biće prikazani svi moduli. Za instalaciju nekog od tih modula služi komanda “sudo apt-get install <ime_ modula>”. Nakon što je instaliran LAMP može se pokrenuti prethodno napravljeni program, npr. 1.php. Potrebno je smestiti ga u /var/www direktorijum i može se pokrenuti preko pregledača ukoliko se ukuca <http://localhost/1.php>.

7 Primer kôda

Primer 7.1 *Primer kako se pomoću PHP-a može konektovati na bazu podataka, izvršiti upit i ispitati dobijene rezultate. Cilj u ovom primeru 2 je pronaći sve studente sa zadatim imenom i prezimenom i ispitati njihov datum rođenja.*

```

<?php
2 //uspostavljanje konekcije sa bazom podataka
if(mysqli_connect_errno()){
4 die("Problem sa povezivanjem: ".mysqli_connect_error());
}
6 //deklarisanje promenljivih i učitavanje podataka
$time = $_GET['ime'];
8 $prezime = $_GET['prezime'];
//formulisanje i izvršavanje upita
10 $upit = " select datum_rodjenja from dosije where ime = '$ime'
and prezime = '$prezime' ";
//rezultat upita se smesta u promenljivu rezultat
12 $rezultat = mysqli_query($veza, $upit) or die("Problem prilikom
izvršavanja upita: ".mysqli_error($veza));
14 //racunanje broja redova rezultata (broj studenata sa zadatim
imenom i prezimenom)
$broj_pojavljivanja = mysqli_num_rows($rezultat);
16 if($broj_pojavljivanja == 0){
echo "Nema studenata koji se zovu $ime $prezime.";
18 }
else{
20 //ukoliko ima studenata ispisujemo ih
echo "<ul>";
22 for($i=0; $i<$broj_pojavljivanja; $i++){
$red = mysqli_fetch_assoc($rezultat);
24 echo "<li>";
echo $red['ime']['prezime']['datum_rodjenja'];
26 echo "</li>";
}
28 echo "</ul>";
}
30 //raskidamo konekciju sa serverom
mysqli_close($veza);
32 ?>

```

Listing 2: Primer PHP kôda

Sav PHP kôd se piše između “<?php“ i “?>“, slično kao tagovi pri pisanju html kôda. Promenljive deklariramo sa prefiksom “\$“ a naredbom “echo“ vršimo ispis. \$_GET metod smo koristili kako bismo preneli vrednosti promenljivih kroz veb stranicu do našeg kôda. Vrednosti tih promenljivih se dodaju u URL stranicu. Na primer ako bismo želeli da izlistamo sve studente koji se zovu Marko i prezivaju Marković, to bismo uradili tako što bismo u URL adresu dodali ?ime=Marko& prezime=Markovic.

8 Specifičnosti

Ono što je veoma specifično za ovaj jezik jeste to što nema striktnu sintaksu koja prati veliki broj jezika. Na primer, sledeći fragment kôda možemo zapisati na dva različita načina. Kao što je prikazano u 4 standardni pristup koristi vitičaste zagrade, dok je u alternativnom pristupu fokus na elseif grananju. Pored ovoga, PHP poseduje veliki broj specifičnosti dok su ovde izdvojene najznačajnije.



```
<?php
2 if($a > 0){
    echo "a ima pozitivnu
      vrednost";
4 } elseif($a == 0){
    echo "a je nula";
6 } else{
    echo "a ima negativnu
      vrednost";
8 }
?>
```

```
<?php
2 if($a > 0):
    echo "a ima pozitivnu
      vrednost";
4 elseif($a == 0):
    echo "a je nula";
6 else:
    echo "a ima negativnu
      vrednost";
8 endif;
?>
```

Slika 4: Primer dva fragmenta koda standardni pristup (levo) i alternativni pristup (desno).

8.1 Imenski prostor i forme

Imenski prostor (eng. namespace) [12] je uveden od verzije 5.3 i prevashodno je zamišljen radi rešavanja konflikata oko istoimenih metoda i funkcija, kako bi mogle da se koriste u okviru istog projekta. Ovo je posebno pogodno u slučaju da se radi o kompleksnim projektima, koji su raspoređeni po različitim datotekama i kada u razvoju aplikacije učestvuje grupa programera. Za deklarisanje imenskog prostora koristi se ključna reč **namespace**. Dozvoljeno je koristiti više imenskih prostora u okviru istog programa, čak i iste datoteke.

Još jedan veoma bitan koncept u radu sa PHP programima jeste rad sa formama. **Forma**[21] je zapravo deo HTML strukture za grupisanje različitih elemenata čija je funkcija prikupljanje podataka od korisnika. U zavisnosti od akcije korisnika, dobijena informacija se može proveriti kod korisnika. Takođe se može proslediti kao zahtev serveru, eventualno na ponovnu proveru i kasniju obradu. Po potrebi na kraju se može poslati neka vrsta povratne informacije korisniku. Forme se koriste, na primer, kod poručivanja hrane u piceriji preko interneta. Popunjavamo formular gde biramo vrstu pice, dodatke, adresu gde će se pica dostaviti itd. Nakon obrade i čitanja podatka iz forme, šalje se odgovor da je porudžbina primljena i da će pica ubrzo stići.

8.2 Klase PDO i DateTime

Jezik PHP se odlikuje ogromnim brojem različitih klasa koje u mnogome olakšavaju rad programerima. Jedna veoma bitna klasa je “PDO (PHP Data Object)” [15, 12] koja se koristi za povezivanje sa SQL bazama podataka. Ta klasa obezbeđuje sloj apstrakcije za skup upravljačkih programa za baze podataka kao što su MySQL, PostgreSQL i MSSQL. To znači da ma koju bazu podataka koristili, ako je PDO podržava, možemo koristiti iste funkcije za izvršavanje istih operacija nad bazom podataka. To čini kôd i samu aplikaciju portabilnom u smislu da se može koristiti u različitim bazama podataka bez nekakve modifikacije samog kôda.

Još jedna veoma korisna i zamiljiva klasa je DateTime klasa [15, 12] koja služi za reprezentaciju vremena i datuma. Pored jednostavnog definisanja i ispisa trenutnog vremena koje je predstavljeno u 3, ova klasa sadrži ogroman broj različitih funkcija za rad sa datumima poput izdvajanja dana, meseca, godine iz datuma i postavljanje vremenskih zona.

```
2 $d = new DateTime('2019-01-01T15:03:01.012345z')
  echo $d -> format('Y-m-d\TH:i:s.u') //2019-01-01T15:03:01.012345
```

Listing 3: Primer upotrebe klase DateTime

8.3 Kolačići i keširanje

Ono što se u jeziku PHP dosta koristi jesu kolačići (eng. cookies) [15, 12] koji se često koriste za identifikaciju korisnika. Kolačić je mala datoteka koju server ugrađuje na računar korisnika. Svaki put kada isti računar zahteva stranicu, ona će takođe poslati i kolačić. Pomoću PHP-a možemo kreirati i dohvatiti vrednosti kolačića. Funkcija koja postavlja vrednost kolačića je data u primeru 4.

```
2 setcookie(name, value, expire); //postavljanje kolacica
  setcookie("user", "", time() - 3600); //brisanje kolacica
```

Listing 4: Funkcija za postavljanje kolacica

Pristup disku je spor dok je pristup mreži još sporiji. Baze podataka obično koriste oba. Korišćenje lokalnog keša izbegava opterećenje mreže i pristupa disku. Kombinujući ove pristupe dobijamo sistem za memorisanje objekata distribuirane memorije (eng. memcached) [17]. Ako naša aplikacija nije distribuirana na više servera, verovatno nam neće trebati memcached. Jednostavniji pristupi keširanju su na primer serijalizovanje podataka i čuvanje u privremenoj datoteci, na primer - mogu eliminisati mnogo redundantnih podataka na svakom pojedinačnom zahtevu [12]. Primer keširanja je dat primerom 5.

```
2 <?php
  $feed = apc_fetch('news');
  if ($feed === FALSE) {
4     $feed = file_get_contents('http://example.org/news.xml');
      // Čuvanje podataka u deljenoj memoriji 5 minuta (5*60 sekundi
      )
6     apc_store('news', $feed, 300);
  }
8 // Uradi nesto sa $feed.
  ?>
```

Listing 5: Primer keširanja

9 Zaključak

Programski jezik PHP značajno olakšava veb programiranje i zbog toga je među jednim od najpopularnijih programskih jezika današnjice. Pored toga, rad sa bazama podataka je dosta jednostavan, pa to predstavlja jednu od najbitnijih olakšica u veb programiranju, korišćenjem PHP-a. Mogućnosti ovog jezika su velike, međutim kroz ovaj rad su predstavljene samo one koje ga ističu u odnosu na druge. Takođe, ovim radom smo istakli sve ono što bi programere koji se do sad nisu susreli sa PHP-om zainteresovalo da počnu da ga koriste.

Literatura

- [1] L. Atkinson. *Core PHP Programming*. Prentice Hall, 2003.
- [2] BCIT. CodeIgniter, 2019. on-line at: <https://codeigniter.com/>.
- [3] Romain Bourdon. WAMP, 2019. on-line at: <http://www.wampserver.com/en/>.
- [4] Parrot Foundation. Parrot vm, 2002-2012. on-line at: <http://parrot.org/>.
- [5] The Apache Software Foundation. HTTP session, 1999-2012. on-line at: <https://tomcat.apache.org/tomcat-5.5-doc/servletapi/javax/servlet/http/HttpSession.html>.
- [6] The PHP group. Callables, 2001-2019. on-line at: <https://www.php.net/manual/en/language.types.callable.php>.
- [7] The PHP group. PHP/FI, 2001-2019. on-line at: <https://www.php.net/manual/phpfi2.php>.
- [8] The PHP group. PHP, 2009-2019. on-line at: <https://www.php.net/manual/en/history.php.php>.
- [9] HHVM. HHVM, 2019. on-line at: <https://hhvm.com/>.
- [10] IBM. JIT compiler, 2015. on-line at: https://www.ibm.com/support/knowledgecenter/en/SSYKE2_8.0.0/com.ibm.java.vm.80.doc/docs/jit_overview.html.
- [11] R. Lerdorf and K. Tatroe. *Programming PHP*. O'Reilly Media, 2002.
- [12] Josh Lockhart. PHP Srbija, 2018. on-line at: <https://phpsrbija.github.io/php-the-right-way/>.
- [13] Canonical Ltd. and Ubuntu. Linux Apache installation, 2019. on-line at: <https://tutorials.ubuntu.com/tutorial/install-and-configure-apache#0>.
- [14] Zend Technologies Ltd. Zend Engine, 2015. on-line at: http://www.zend.com/products/zend_engine/in_depth.
- [15] P. MacIntyre. *PHP: The Good Parts*. O'Reilly Media, 2010.
- [16] TAYLOR OTWELL. Laravel, 2019. on-line at: <https://laravel.com/>.
- [17] C. Shiflett. 10 advanced php tips. 3 2009.
- [18] D. Sklar. *Learning PHP*. O'Reilly Media, Inc., 2016.
- [19] SmartBear Software. Zephyr, 2019. on-line at: <https://www.getzephyr.com/>.

- [20] Symfony. Symfony, 2019. on-line at: <https://symfony.com/>.
- [21] A. Trachtenberg and D. Sklar. *PHP Cookbook, 3rd Edition*, pages xv, 1–369. O'Reilly Media, 2014.
- [22] W3Techs. PHP stats, 2001-2019. on-line at: <https://w3techs.com/technologies/details/pl-php/all/all>.
- [23] Carlo Zapponi. Github, 2014. on-line at: <https://github.info/>.