Napredno Web programiranje

PHP

Informacije o kolegiju

- Predavanja: **Izv.prof.dr.sc. Ivica Lukić**
  ivica.lukic@ferit.hr
- Vježbe: **Izv.prof.dr.sc. Ivica Lukić**
  ivica.lukic@ferit.hr
- Konzultacije: **kontaktirati mailom radi dogovora**

OSVNOVNA LITERATURA

1. MacIntrye, Peter; Tatroe Kevin; Lerdorf Rasmus. Programiranje PHP treće izdanje. O'Reilly i IT Expert, 2015.
2. Shackelford, Adam. Beginning Amazon Web Services with Node.js. New York: Apress, 2015.
3. R. Delorme, Programming in HTML5 with Javascript and CSS3, Microsoft Press, Redmond Washington, 2014.

Dopunska literatura

1. L. Revill, jQuery 2.0 Development Cookbook, Published by Packt Publishing Ltd. Livery Place 35 Livery Street Birmingham B3 2PB, UK, 2014.
2. K. Williamson, Learning AngularJS, Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North Sebastopol, CA 95472, 2015.
3. L. Ullman, PHP Advanced and Object-Oriented Programming: Visual QuickPro Guide (3rd Edition), Peachpit Press, 1301 Sansome Street, San Francisco, CA 94111, 2012.
4. R. Nixon, Learning PHP, MySQL & JavaScript With jQuery, CSS & HTML5, O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472, 2014.
5. A. K. Pande, jQuery 2 Recipes, Apress, Apress Media LLC 233 Spring Street New York, NY 10013, 2014.
6. C. Pitt, Pro PHP MVC, Apress, Apress Media LLC 233 Spring Street New York, NY 10013, 2012.

NAČIN IZVOĐENJA NASTAVE

- Način izvođenja nastave:
  - predavanja (2 sata tjedno)
    - ukratko se izlažu OSNOVNI pojmovi i primjeri zadataka
    - za šira znanja o pojedinim temama preporuča se dostupna literatura
    - materijali dostupni na loomen stranicama
    - obvezan dolazak na 70% predavanja
  - laboratorijske vježbe (2 sata tjedno)
    - samostalan rad na računalu
    - upute za vježbe se nalaze na loomen stranicama
    - uvjet za dobivanje potpisa – odrađene SVE laboratorijske vježbe
    - Vježbe su obavezne, a izostanak se mora ispričati i nadoknaditi u prvom slobodnom terminu !

Način polaganja ispita

- Umjesto pismenog ispita izrađuje se projektni zadatak
  - Projektni zadatak izrađuje se u grupama
  - Mogu se odabrati različite tehnologije
  - Poželjno je koristiti razvojno okruženje (eng. Framework)
- Za pristup usmenom ispitu treba izraditi projektni zadatak
  - Pitanja o izradi projekta
  - Pitanja o tehnologijama obrađenim na predavanju i u samom projektu

OOP U PHP-U

- Stvaranje PHP klase

```
<?php
class osoba {
    //Stvaranje varijable
    var $ime;

    //Funkcija za postavljanje vrijednosti
    function postavi_ime($novo_ime) {
        $this->ime = $novo_ime;
    }
    //Funkcija za dohvat vrijednosti
    function dohvati_ime() {
        return $this->ime;
    }
}
?>
```

OOP u PHP-u

- Varijabla $this je ugrađena u sve objekte i ona pokazuje na trenutni objekt. Koristimo je za pristup varijablama i metodama trenutne klase
- Klase nikad nećemo stvarati u glavnoj .php datoteci, nego ćemo je spremiti u posebnu datoteku i uključiti u glavnu datoteku:

```
<html>
    <head>
        <meta charset="UTF-8">
        <title>OOP u PHP-u</title>
        <?php include("class_osoba.php"); ?>
    </head>
    <body>
    </body>
</html>
```

OOP U PHP-U

- Novu instancu klase stvaramo na sljedeći način i to upotrebom ključne riječi new
- Varijabla $Marko postaje referenca na novi objekt

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>OOP in PHP</title>
        <?php include("class_osoba.php"); ?>
    </head>
    <body>
        <?php $Marko = new osoba(); ?>
    </body>
</html>
```

OOP u PHP-u

- Dohvat podataka o objektu
- Pristup svojstvima i metodama je pomoću operatora ->

```
<body>
        <?php
          $Marko = new osoba();
          $Djuro = new osoba();
          //Postavljanje varijabli
          $Marko->postavi_ime("Marko Marić");
          $Djuro->postavi_ime("Djuro S. Zlikovski");
          //Dohvat Varijabli
          echo "Ime osobe Marko je: " . $Marko->dohvati_ime();
         echo "<br>Ime osoba Djuro je: " . $Djuro->dohvati_ime();
        ?>
    </body>
</html>
```

OOP U PHP-U

- U PHP-u također možemo koristiti konstruktor

```
class osoba {
    //Stvaranje varijable
    var $ime;
     //Konstruktor
    function __construct($ime_osobe) {
        $this->ime = $ime_osobe;
    }
    //Funkcija za postavljanje vrijednosti
    function postavi_ime($novo_ime) {
        $this->ime = $novo_ime;
    }
    //Funkcija za dohvat vrijednosti
    function dohvati_ime() {
        return $this->ime;
    }
}
```

OOP u PHP-u

- Novu osobu možemo stvoriti pomoću konstruktora koji prima jedan parametar

```
<body>
    <?php
      $Marko = new osoba();
      $Djuro = new osoba("Djuro S. Zlikovski");
      //Postavljanje varijabli
      $Marko->postavi_ime("Marko Marić");
      //Dohvat Varijabli
      echo "Ime osobe Marko je: " . $Marko->dohvati_ime();
      echo "Ime osoba Djuro je: " . $Djuro->dohvati_ime();
    ?>
</body>
```

OOP u PHP-u

- PHP koristi ovijanje (eng. Encapsulation) pomoću ključnih riječe public (nema ograničenja pristupa), private (samo unutar klase) i protected (unutar klase i klase koja ju nasljeđuje)
- Ako ništa ne navedemo pravo je public

```
<?php
class osoba {
    //Stvaranje varijable
    var $ime;
    public $visina;
    protected $oib;
    private $pin_kartice;
    //Ostatak koda....
?>
```

OOP U PHP-U

- Nasljeđivanjem se omogućuje ponovna upotrebljivost koda, a sa pravima pristupa određujemo upotrebu klase
- Klasa zaposlenik se temelji na klasi osoba i ima sva njena public i protected svojstva i metode
- Tako može koristiti metodu postavi_ime()

```
<?php
class zaposlenik extends osoba {
    function __construct($ime_zaposlenika) {
    }
}
?>
```

OOP U PHP-U

- Primjer stvaranja objekata dviju klasa
- U klasu zaposlenik koristimo metodu dohvati_ime() iz klase osoba

```
<?php
include("class_osoba.php"); ?>
<body>
    <?php
        $Mate = new osoba("Mate Mišo");
        echo "Matino ime je: " . $Mate->dohvati_ime();
        $Ana = new zaposlenik("Ana Jurić");
        echo "Zaposlenik mjeseca je: " . $Ana->dohvati_ime();
    ?>
</body>
```

OOP U PHP-U

- U naslijeđenoj klasi možemo preopteretiti metodu roditeljske klase

```
<?php
class osoba {
    //Stvaranje varijable
    var $ime;
    public $visina;
    protected $oib;
    private $pin_kartice;
    //Konstruktor
    function __construct($ime_osobe) {
        $this->ime = $ime_osobe;
    }
    //Funkcija za postavljanje vrijednosti
    function postavi_ime($novo_ime) {
        $this->ime = $novo_ime;
    }
```

OOP U PHP-U

```
//Funkcija za dohvat vrijednosti
    function dohvati_ime() {
        return $this->ime;
    }
}
class zaposlenik extends osoba {
    function __construct($ime_zaposlenika) {
    }

    protected function postavi_ime($novo_ime) {
      if ($novo_ime != "") {
          $this->ime = "Zekoslav Mrkva";
      }
    }
}
?>
```

OOP u PHP-u

- Ako u nekim slučajevima želimo pristupiti metodi iz roditeljske klase koristimo sljedeći način:

```
osoba::postavi_ime($novo_ime)
```

- Ili možemo koristiti ključnu riječ parent

```
parent::postavi_ime($novo_ime)
```

OOP u PHP-u

- Apstraktne klase i metode se koriste kada nećemo stvarati instance roditeljske klase
- Definiranjem apstraknith klasa definiramo općenito ponašanje izvedenih klasa
- Ako probamo napraviti objekt apstraktne klase dobiti ćemo pogrešku

```
<?php
abstract class Oblik {
    //Računanje površine
    abstract protected function Povrsina();
    //Računanje opsega
    abstract protected function Opseg();
}
?>
```

OOP u PHP-u

- Nasljeđujemo apstrakntu klasu

```
<?php
class Trokut extends Oblik {
    private $_stranice = array();
    private $_opseg = NULL;

    function __construct($s0 = 0, $s1 = 0, $s2 = 0) {
        //Spremamo stranice trokuta u niz
        $this->_stranice[] = $s0;
        $this->_stranice[] = $s1;
        $this->_stranice[] = $s2;

        //Izračun opsega
        $this->_opseg = array_sum($this->_stranice);
    }
```

OOP u PHP-u

```
    public function Povrsina() {
        return (SQRT(
        ($this->_opseg/2) *
        (($this->_opseg/2) - $this->_stranice[0]) *
        (($this->_opseg/2) - $this->_stranice[1]) *
        (($this->_opseg/2) - $this->_stranice[2])
        ));
    }

    public function Opseg() {
        return $this->_opseg;
    }

}
```

OOP U PHP-U

- Instanciranje objekta klase trokut

```
<!doctype html>
<html>
<body>
<?php
require('class_Oblik.php');
require('class_Trokut.php');
$stranica1 = 5;   $stranica2 = 10;   $stranica3 = 13;
// Stvaramo novi trokut
$t = new Trokut($stranica1, $stranica2, $stranica3);
echo '<p>Površina je ' . $t->Povrsina() . '</p>';
echo '<p>Opseg je ' . $t->Opseg() . '</p>';
//Brišemo objekt
unset($t);
?>
</body>
```

Interfejs u PHP-u

- U PHP-u možemo koristiti i interfejse
- Interfejse definira metode koje mora implementirati klasa koja ga koristi
- Za stvaranje interfejsa se koristi ključna riječ interface

```
<?php
interface iCrud {
    public function create($data);
    public function read();
    public function update($data);
    public function delete();
}
```

Interfejs u PHP-u

```
class User implements iCrud {
    private $_userId = NULL;
    private $_username = NULL;

    function __construct($data) {
        $this->_userId = uniqid();
        $this->_username = $data['username'];
    }

    function create($data) {
        self::__construct($data);
    }

    function read() {
        return array('userId' => $this->_userId, 'username' =>
$this->_username);
    }
```

Interfejs u PHP-u

```
    function update($data) {
        $this->_username = $data['username'];
    }

    public function delete() {
        $this->_username = NULL;
        $this->_userId = NULL;
    }
}
//Korisničko ime
$user = array('username' => 'djuro');
echo "<h2>Stvaranje novog korisnika</h2>";
$me = new User($user);
// ID korisnika
$info = $me->read();
echo "<p>ID je {$info['userId']}.</p>";
```

Interfejs u PHP-u

```
// Promijeni korisničko ime
$me->update(array('username' => 'pero'));

//Potvrdi novo korisničko ime
$info = $me->read();
echo "<p>Novo korisničko ime je {$info['username']}.</p>";

//Obriši korisnika
$me->delete();

//Obriši objekt:
unset($me);

?>
```

Svojstva (eng. Traits) u PHP-u

- Svojstva (eng. Traits) se koriste da riješe neke probleme zbog toga što se u PHP-u može naslijediti samo jedna klasa
- Ako imamo više klasa i u svakoj želimo ispisati neka svojstva kod bi morali dodati u svaku klasu
- Ako napravimo svojstvo i njega uključimo u svaku klasu možemo koristiti isti kod koji je napisan u svojstvu
- Na sljedećem slajdu je primjer svojstva koje koristimo za debugiranje neke klase neovisno o tome o kojoj se klasi radi

Svojstva (eng. Traits) u PHP-u

```php
<?php
trait Debug {
    // Ispis podataka o trenutnom objektu
    public function dumpObject() {
        //Ime klase
        $class = get_class($this);
        // Dohvat varijabli
        $attributes = get_object_vars($this);
        // Dohvat metoda
        $methods = get_class_methods($this);
        // Ispis varijabli
        echo '<h3>Varijable</h3><ul>';
        foreach ($attributes as $k => $v) {
            echo "<li>$k: $v</li>";
        }
        echo '</ul>';
```

Svojstva (eng. Traits) u PHP-u

```
        //Ispis metoda
        echo '<h3>Metode</h3><ul>';
        foreach ($methods as $v) {
            echo "<li>$v</li>";
        }
        echo '</ul>';

    }

}
?>
```

Svojstva (eng. Traits) u PHP-u

- Svojstva (eng. Traits) sada možemo upotrijebiti u klasi Trokut

```
<?php
require('class_Debug.php');
require('class_Oblik.php');
require('class_Trokut.php');

// Stvori novi objekt
$r = new Trokut(4,5,7);
// Ispis podataka:
$r->dumpObject();

?>
```

Upotreba PDO

- PDO (eng. PHP Data Object)
- Omogućava korištenje upita neovisno o SUBP
- PDO radi sa MySQL, PostgreSQL, SQLite, Oracle, Microsoft SQL Server itd.
- PHP instalacija mora imati omogućenu PDO ekstenziju
- Spoj na bazu podataka:

```
$pdo = new PDO('dsn', 'username', 'password');
```

- Username i password parametri jasni su sami po sebi

Upotreba PDO

- Parametar dsn je skraćenica za *Data Source Name*
- To je string koji sadrži sljedeće:
  - Naziv driver-a
  - Ime baze podataka
  - Opcionalno naziv hosta i port
- Za MySQL

```
$pdo = new PDO('mysql:dbname=test;host=localhost', 'username', 'password');
```

- Za SQLite

```
$pdo = new PDO('sqlite:/path/to/imebaze.sq3');
```

Upotreba PDO

- Primjer spoja na bazu i hvatanje iznimke

```
<?php
try {
    //Stvori PDO objekt
    $pdo = new PDO('mysql:dbname=test;host=localhost',
'username', 'password');

    // Obriši objekt
    unset($pdo);

} catch (PDOException $e) {
    //Hvatanje iznimke
    echo '<p>Dogodila se iznimka: ' . $e->getMessage() . '</p>';
}
?>
```

Upotreba PDO

- Primjer jednostavnog unosa u bazu

```
<?php
try {
    //Stvori PDO objekt
    $pdo = new PDO('mysql:dbname=test;host=localhost',
root', '');
    echo '<form action="dodaj_zadatak.php" method="post">
    <fieldset>
        <legend>Dodaj zadatak</legend>
        <p>Zadatak: <input name="zadatak" type="text"
size="60" maxlength="100"></p>
        <p>Roditeljski zadatak: <select
name="roditelj_id"><option value="0">Nema</option>';
```

Upotreba PDO

```
    // Upit
    $q = 'SELECT * FROM zadaci WHERE
datum_zavrsetka="0000-00-00 00:00:00" ORDER BY
datum_dodavanja ASC';
    $r = $pdo->query($q);
    // Psotavi u mod dohvata
    $r->setFetchMode(PDO::FETCH_NUM);
    // Ispis rezultata
    while ($row = $r->fetch()) {
      echo "<option value=\"$row[0]\">$row[2]</option>\n";
    }
    echo '</select></p>
    <input name="submit" type="submit" value="Dodaj
zadatak">
    </fieldset>
    </form>';
```

Upotreba PDO

```
    // Obriši objekt
    unset($pdo);

} catch (PDOException $e) {
    //Hvatanje iznimke
    echo '<p>Dogodila se iznimka: ' . $e->getMessage() . '</p>';
}
?>
```

Upotreba PDO

- Kod za dodaj_zadatak.php

```
<?php
$pdo = new PDO('mysql:dbname=test;host=localhost', 'root',
'');
 if (($_SERVER['REQUEST_METHOD'] == 'POST') &&
!empty($_POST['zadatak'])) {
        if (isset($_POST['roditelj_id']) &&
        filter_var($_POST['roditelj_id'],
FILTER_VALIDATE_INT, array('min_range' => 1)) ) {
            $roditelj_id = $_POST['roditelj_id'];
        } else {
            $roditelj_id = 0;
        }
```

Upotreba PDO

```
        // Unos u bazu podataka
        $q = 'INSERT INTO zadaci (roditelj_id, zadatak)
VALUES (:roditelj_id, :zadatak)';
        $stmt = $pdo->prepare($q);

        //Izvrši upit
        if ($stmt->execute(array(':roditelj_id' => $roditelj_id,
':zadatak' => $_POST['zadatak']))) {
            echo '<p>Zadatak je dodan!</p>';
        } else {
            echo '<p>Zadatak nije dodan!</p>';
        }

    }
?>
```

Upotreba PDO

- Kod za bazu podataka

```
CREATE TABLE `zadaci` (
  `id` int(11) NOT NULL,
  `roditelj_id` int(11) NOT NULL,
  `zadatak` varchar(1000) NOT NULL,
  `datum_dodavanja` datetime NOT NULL,
  `datum_zavrsetka` datetime NOT NULL
) ENGINE=InnoDB ;
```

Rad sa socket-om

- Socket je kanal preko kojeg dva računala mogu komunicirati
- Za otvaranje socketa u PHP-u:

```
$fp = fsockopen ($url, $port, $error_number, $error_string, $timeout);
```

- Od svih parametara jedino je URL obavezan
- Broj pogrešeke i njen string se ne predaju funkciji jer u početku nisu poznati
- Funkcija preko njih vraća pogrešku ako se dogodi
- Timeout govori koliko dugo će dugo funkcija pokušavati spajanje

Rad sa socket-om

- Primjer portova koji se koriste

Port number	Process name	Protocol used	Description
20	FTP-DATA	TCP	File transfer—data
21	FTP	TCP	File transfer—control
22	SSH	TCP	Secure Shell
23	TELNET	TCP	Telnet
25	SMTP	TCP	Simple Mail Transfer Protocol
53	DNS	TCP and UDP	Domain Name System
69	TFTP	UDP	Trivial File Transfer Protocol
80	HTTP	TCP and UDP	Hypertext Transfer Protocol
110	POP3	TCP	Post Office Protocol 3
123	NTP	TCP	Network Time Protocol
143	IMAP	TCP	Internet Message Access Protocol
443	HTTPS	TCP	Secure implementation of HTTP

Rad sa socket-om

- Server može odogovoriti sa različitim statusima

HTTP code	Meaning
200	OK
4xx	Bad request (client's fault)
5xx	Failed request (server's fault)
401	Unauthorized request
404	Resource not found
500	Internal error (bug)
503	Server overloaded

Rad sa socket-om

- Primjer validacije URL-ova

```
<?php
//Spajamo se na URL
function check_url($url) {
    //Razdvaja URL na dijelove
    $url_pieces = parse_url($url);

    //Postavljamo $path i $port
    $path = (isset($url_pieces['path'])) ? $url_pieces['path'] :  '/';
    $port = (isset($url_pieces['port'])) ? $url_pieces['port'] : 80;

    //Spajanje sa fsockopen():
    if ($fp = fsockopen($url_pieces['host'], $port, $errno,
$errstr, 30)) {
```

Rad sa socket-om

```
//Slanje zaglavlja da server ne vraća čitavu stranicu nego samo odgovor
$send = "HEAD $path HTTP/1.1\r\n";
$send .= "HOST: {$url_pieces['host']}\r\n";
$send .= "CONNECTION: Close\r\n\r\n";
fwrite($fp, $send);
//Čitanje odgovora
$data = fgets($fp, 128);
//Zatvranje spoja
fclose($fp);
//Vraćamo odgovor
list($response, $code) = explode(' ', $data);
if ($code == 200) {
    return array($code, 'good');
} else {
    return array($code, 'bad');
}
```

RAD SA SOCKET-OM

```
    } else { // Nema spoja
        return array($errstr, 'bad');
    }
}

//niz URL-ova
$urls = array(
'https://www.ferit.unios.hr/', 'https://www.unios.hr/'
);
echo '<h2>Validacija URL-ova</h2>';
// Ukidamo PHP time limit
set_time_limit(0);
```

Rad sa socket-om

```
// Validiramo svaki URL:
foreach ($urls as $url) {
    list($code, $class) = check_url($url);
    echo "<p><a href=\"$url\" target=\"_new\">$url</a>
(<span class=\"$class\">$code</span>)</p>\n";
}
?>
```

Rad sa socket-om

- Primjer sa IP geolokacijom

```
<?php
//IP Geolocation request
function show_ip_info($ip) {
    //Besplatna registracija za vlastiti api key
    $api_key = 'at_P0Eei4pKMjqB66SS1dtCI01Sttcty';
    //URL na koji se spajamo
    $api_url = 'https://geo.ipify.org/api/v1';
    $url = "{$api_url}?apiKey={$api_key}&ipAddress={$ip}";
    //Otvori konekciju
    $fp = fopen($url, 'r');
    //dohvati podatke
    $read = fgetcsv($fp);
    //Zatvori konekciju
    fclose($fp);
```

Rad sa socket-om

```
    //Ispis podatka
    echo "<p>IP Adresa: $ip<br>
    Zemlja: $read[1]<br>
    Grad: $read[3], $read[2]<br>
    Zemljopisna širina: $read[4]<br>
    Zemljopisna dužina: $read[5]</p>";
}
//Klijentova IP adresa
echo '<h2>Informacije o Vama</h2>';
show_ip_info($_SERVER['REMOTE_ADDR']);

//Informacije o nekoj stranici
$url = 'www.etfos.unios.hr';
echo "<h2>Informacije o $url</h2>";
show_ip_info(gethostbyname($url));

?>
```

Rad sa curl-om

- cURL znači client URLs
- Funkcija za pokretanje spoja je:

```
curl_init($url);
```

- Pomoću cURL-a možete pristupati stranicama, slati upite i primati odgovore
- Postoje brojne opcije koje možete koristiti
- Za više detalja pogledati literaturu
- Primjer upotrebe je na sljedećem slajdu

Rad sa curl-om

```
<h2>cURL Rezultati:</h2>
<?php
$url = 'http://localhost/service.php';
//Pokrećemo cURL spoj
$curl = curl_init($url);
//Zaustavi ako se dogodi pogreška
curl_setopt($curl, CURLOPT_FAILONERROR, 1);
//Dozvoli preusmjeravanja
curl_setopt($curl, CURLOPT_FOLLOWLOCATION, 1);
//Spremi vraćene podatke u varijablu
curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
//Postavi timeout
curl_setopt($curl, CURLOPT_TIMEOUT, 5);
//Koristi POST metodu
curl_setopt($curl, CURLOPT_POST, 1);
```

Rad sa curl-om

```
//Postavi POST podatke
curl_setopt($curl, CURLOPT_POSTFIELDS,
'name=foo&pass=bar&format=csv');

//Izvrši transakciju
$r = curl_exec($curl);

//Zatvori spoj
curl_close($curl);

//ispiši rezultate
print_r($r);

?>
```

Rad sa curl-om

- Kod za service.php

```
<?php
if (isset($_POST['format'])) {
switch ($_POST['format']) {
        case 'csv':
            $type = 'text/csv';
            break;
        case 'json':
            $type = 'application/json';
            break;
        case 'xml':
            $type = 'text/xml';
            break;
        default:
            $type = 'text/plain';
            break;
    }
```

RAD SA CURL-OM

```
//Stvori odgovor
$data = array();
$data['timestamp'] = time();
//Vrati primljene podatke
foreach ($_POST as $k => $v) {
    $data[$k] = $v;
}
//Prebaci podatke u json format
if ($type == 'application/json') {
    $output = json_encode($data);

} elseif ($type == 'text/csv') {
    //Prebaci u string
    $output = '';
    foreach ($data as $v) {
        $output .= '"' . $v . '",';
    }
```

Rad sa curl-om

```
        //Odreži zarez na kraju podataka
        $output = substr($output, 0, -1);

    } elseif ($type == 'text/plain') {
        $output = print_r($data, 1);
    }
    else { //Pogrešna upotreba
    $type = 'text/plain';
    $output = 'Servis ne koristite ispravno.';
    }
}

//Postavi zaglavlje
header("Content-Type: $type");
echo $output;
```