VILNIUSCODINGSCHOOL

PHP 4

OOP (Object-Oriented Programming)

- **OOP: programavimo stilius, kuriame susiję kintamieji ir funkcijos yra apjungiami į vieną klasę. Efektyvesnis kodas, nes geresnis kodo organizavimas, mažiau pasikartojimų.**

Klasė

- kintamųjų, funkcijų, savybių junginys
- pavadinimas rašomas iš didžiosios raidės, vienaskaitoje
- deklaruojame raktiniu žodžiu class
- klasės savybes priskiriame kintamiesiems, savybes rašome mažosiomis raidėmis, prieš jas rašome raktinį žodį public.

```php
class Car {
    // kodas
}
```

```php
class Car {
    public $gamintojas;
    public $spalva = 'raudona';
    public $atsarginisRatas = true;
}
```

Objektai

```
$bmw = new Car();
$volvo = new Car();
```

- Iš vienos klasės galima sukurti daug objektų
- Objekto sukūrimas pagal klasę vadinamas atveju (instance)
- Visiems objektams galioja klasėje nurodytos savybės ir metodai, tačiau objektai gali skirtis vienas nuo kito, nes jiems galime priskirti unikalias savybes ir metodus
- Objekto savybes pasiekiame taip:

```
echo $bmw -> spalva;
```

- Objekto savybes keičiame taip:

```
$bmw -> spalva = 'juoda';
```

Metodai

- Klasės dažnai talpina funkcijas, vadinamas metodais.
- Metodus rašome su raktiniais žodžiais public ir function

```php
class Car {
    public $gamintojas;
    public $spalva = 'raudona';
    public $atsarginisRatas = true;

    public function labas() {
        return "pyyyyp";
    }
}

$bmw = new Car();
echo $bmw -> labas();
```

$this

- Raktinis žodis $this rodo, kad naudojame klasės metodus ir savybes, ir leidžia pasiekti juos klasės apimtyje. Klasės viduje galime pasiekti ir naudoti tos klasės metodus ir savybes. Tuo atveju, tik $this rašome su $, o metodus ir savybes be.

$this

```php
class Car {
    // savybės
    public $gamintojas;
    public $spalva = 'raudona';
    public $atsarginisRatas = true;

    // metodas gali pasiekti savybes, nes naudoja $this
    public function labas() {
        return 'Pyyyyp, aš esu <i>' . $this -> gamintojas . '</i>, mano spalva: ' . $this -> spalva . '<br>';
    }
}

$bmw = new Car();
$citroen = new Car();

$bmw -> gamintojas = "BMW";
$bmw -> spalva = "juoda";

$citroen -> gamintojas = "Citroen";
$citroen -> spalva = "mėlyna";

echo $bmw -> labas();
echo $citroen -> labas();
```

Metodų ir savybių junginiai (chaining)

- Chaining’as leidžia rašyti paprastesnį kodą.
- Tarkim, kad norime apskaičiuoti kiek turime degalų automobilio bake. Šis kiekis priklausys nuo įpilto degalų kiekio ir nuvažiuoto atstumo. Sukuriame savybę bakas ir du metodus: degaluKiekis ir atstumas.

Metodų ir savybių junginiai (chaining)

```php
class Car {
    public $bakas;

    // pridedame litrus į degalų baką
    public function degaluKiekis($litrai) {
        $this -> bakas += $litrai;
        return $this;
    }

    // atimame degalų litrus iš bako, priklausomai nuo nuvažiuoto atstumo
    public function atstumas($distancija) {
        $km = $distancija;
        $litrai = $km * 0.05;
        $this -> bakas -= $litrai;
        return $this;
    }
}

$bmw = new Car();

// kiek turėsime degalų bake, jei įpylėme 10 L, o nuvažiavome 40 km?
$bakas = $bmw -> degaluKiekis(10) -> atstumas (40) -> bakas;
echo 'Bake bus likę: ' . $bakas . ' L.';
```

public ir private

- **public leidžia kodui esančiam tiek už klasės ribų, tiek klasės viduje, naudoti klasės metodus ir savybes**
- **private neleidžia kodui esančiam už klasės ribų naudotis tos klasės metodais ir savybėmis**

public ir private

```php
class Car {
    private $model;

    public function getModel() {
        return 'Šio automobilio modelis yra ' . $this -> model;
    }
}

$volvo = new Car();

// bandome pasiekti privačią savybę už kodo ribų
$volvo -> model = 'Volvo';
echo $volvo -> getModel();
```

```
Fatal error: Cannot access private property Car::$model
```

date() funkcija: php.net/manual/en/function.date.php

```php
<?php
    // savaitės diena, mėnuo, diena, metai
    echo date('l, F j, Y');
?>
```

```php
// kurią savaitės dieną bus Kalėdos?
// perkeičiame tekstinę datą į timestamp
$kalėdos2018 = strtotime('Dec 25, 2018');
echo date('l, F j, Y', $kalėdos2018);
```

DateTime klasė: php.net/manual/en/class.datetime.php

```php
$data1 = new DateTime();
$data2 = new DateTime();

$losangeles = new DateTimeZone('America/Los_Angeles');
$vilnius = new DateTimeZone('Europe/Vilnius');

$data1 -> setTimezone($losangeles);
$data2 -> setTimezone($vilnius);

echo 'Laikas Los Andžele: ' . $data1 -> format('g:ia, l, F j, Y') . '<br>';
echo 'Laikas Vilniuje: ' . $data2 -> format('g:ia, l, F j, Y') . '<br>';
```

Dažnai pasitaikančios klaidos

- **Visiškai tuščias puslapis: greičiausiai todėl, kad testuojate puslapį ne lokaliame serveryje, ir saugumo sumetimais yra nustatyta nerodyti klaidų. Failo viršuje galime laikinai pridėti <?php ini_set('display_errors', '1'); ?>**
- **Kita tuščio puslapio priežastis - sintaksės klaidos. PHP error nurodyme bus parašyta eilutė, kurioje yra klaida. Arba joje, arba virš jos gali būti, kad trūksta kabliataškio, kabutės ar skliaustelio.**

T_ENCAPSED_AND_WHITESPACE error

- **Ši klaida dažniausiai būna todėl, kad įtraukėme asociatyvinį masyvą tarp dvigubų kabučių. Sprendimas: uždėti { } iš abiejų masyvo pusių ir būtinai be tarpų.**

```php
$cars = [
    'Trucks' => 'Volvo',
    'SUV' => 'BMW',
    'Sports' => 'Ferrari'
];

echo "Populiariausi SUV automobiliai yra {$cars['SUV']}";
```

Failed to open stream: No such file or directory

- **Šios klaidos priežastis - neteisingas failo pajungimas. Jei serveris neranda failo, kurį bandome pajungti su include ar require, gausime šią klaidą. Taip nutinka, kai kelias iki mūsų failo yra neteisingai nurodytas, arba esame padarę sintaksinę klaidą rašydami failo pavadinimą.**

“Headers already sent” error

- **Tai klaida, kuri atsiranda, kai mūsų PHP faile uždarėme PHP kodą su ?> bet failo apačioje palikome vieną ar keletą tuščių eilučių. Sprendimas - arba panaikinti nereikalingas tuščias eilutes, arba panaikinti ?> PHP uždarymą (jei tame faile yra tik PHP kodas, uždarymas nėra privalomas)**

Undefined index, offset, constant, variable

- **Undefined index: viena iš dažniausiai sutinkamų priežasčių - neteisingai parašytas index asociatyviniame masyve.**
- **Offset: klaida, kuri atsiranda, kai bandome pasiekti masyvo indeksą, kuris neegzistuoja.**
- **Undefined constant: pamiršome uždėti kabutes ant index asociatyviniame masyve.**
- **Undefined variable: greičiausiai nesutampa vieno iš kintamųjų pavadinimas - tai ką deklaravome, skiriasi nuo to, ką bandome išvesti ar panaudoti savo kode. Taip pat gali būti, kad kintamasis buvo deklaruotas if/else viduje.**

Užduotis 1

- Užrašyk klasę, kuri talpintų žmogaus vardą, pavardę ir metodą pasisveikinimui.

- Sukurk du žmones su skirtingais duomenimis.

- Išvesk šiuos duomenis naršyklėje.

Užduotis 2

- **Sukurk savo funkciją, kuri naršyklėje rodytų dabartinius metus: © 2023, jei tinklalapis buvo sukurtas ankstesniais metais, turėtų rodyti tokiu formatu: © 2015-2023**

- **Pasvarstyk, kaip rasi dabartinius metus ir ar tavo funkcija turės parametrų.**

- **Naudokitės: http://php.net/manual/en/function.date.php**