



VILNIUSCODINGSCHOOL



Sesijos



- `session_start();`
- `$_SESSION['username']; $_SESSION['password'];` ir t.t.
- `$_SESSION = array();`
- `session_destroy();`
- `unset($_SESSION['username']);`
- Naudokite HTTPS

Objektai. Terminologija



- Objektiškai orientuotas programavimas – gelbsti nuo pasikartojančio kodo rašymo.
- Klasė – kiekvienas objektas sukurtas pagal klasę vadinami atvejais (**instance**).
- Duomenys susiję su objektu vadinami objekto savybėmis (**properties**)
- Funkcijos naudojamos objekto viduje vadinamos metodais (**methods**)
- Inkapsuliacija (**encapsulation**) kai tiesiogiai neprieinama prie objekto duomenų, prieinama tik per kitus viešus metodus.
- Prieinami objekto metodai vadinami objekto sąsajomis (**interface**);
- Paveldėjimas (**inheritance**) – kai reikalinga panaši klasė tačiau su papildomomis savybėmis;
 - Paveldėjimo atveju naujoji klasė turi visas savybes ir **metodus** pagrindinės klasės + savo naujus savybes ir metodus. Originalioji klasė vadinama **superclass**, o naujoji – **derived** arba **subclass**

Klasės aprašymas



- Prieš naudojant objektą reikia apsirašyti jo klasę

```
class User{  
    public $name, $password;  
  
    function save_user(){  
        echo 'Saving user ' . $this->name;  
    }  
}
```

Klasės konstruktorius/destruktorius



- Klasė gali turėti konstruktorių ir destruktorių

```
class User{  
  
    function __construct($param1, $param2){  
        // veiksmai objekto sukūrimo metu  
    }  
    public $name, $password;  
  
    function save_user(){  
        echo 'Saving user '.$this->name;  
    }  
  
    function __destruct(){  
        // veiksmai objekto naikinimo metu  
    }  
}
```

Metodai



- Aprašomi kaip ir funkcijos
- Nenaudokite _
- Turite priėjimą prie savybių per \$this

```
class User{  
    public $name, $password;  
  
    function __construct($name, $password){  
        $this->password = $password;  
    }  
  
    function get_password(){  
        return $this->password;  
    }  
}
```

Statiniai metodai



- Dažniausiai gali būti klasės su tam tikros tematikos metodų rinkiniais
- Kviečiame `Utils::ask_for_password()`;

```
Utils::increase();  
Utils::increase();
```

```
class Utils{  
    static private $counter = 0;  
  
    static function increase(){  
        self::$counter++;  
        echo "Pabandykite ".self::$counter." kartą.";  
    }  
}
```

Savybių rašymas



- Aprašomi viduje klasės
- Reikšmės gali būti priskiriamos per konstruktorių arba iš išorės. Rekomenduojama per konstruktorių, klasės viduje.
- Aprašant kintamuosius negali būti priskirti veiksmai ar funkcijos

Konstantos



- `const PI = 3.14;`

Savybių ir metodų galiojimas



- Public
- Protected
- Private

Paveldējimas



- class User
- class Subscriber extends User
- Final – apsaugo nuo perrašymo

Standartai



- <http://www.php-fig.org/psr/>

Klausimai

