Klase i objekti

Klase u php-u

- Klase su šabloni za kreiranje objekata
- Klasa se takođe smatra i korisnički definisanim tipom podatka
- Klasa se sastoji iz polja i metoda
- Podrazumevano su polja klase javna

Polja klase

```php
<?php

class Osoba
{
    public $ime;
    public $prezime;
}
//kreiranje objekta
$os1 = new Osoba();
$os1->ime = 'Marko';
$os1->prezime = 'Markovic';

?>
```

Metode klase

pubf – code snnipet

```php
class Osoba
{
    public $ime;
    public $prezime;

    public function Stampaj()
    {
        echo "Ime: {$this->ime} Prezime {$this->prezime} <br> ";
    }
}
$os1 = new Osoba();
$os1->ime = 'Marko';
$os1->prezime = 'Markovic';
$os1->Stampaj();
?>
```

Referenciranje bilo kog polja klase u metodi vrši se korišćenjem ključne reči $this
$this pokazuje na objekat koji čuva polje, tj. tekuću instancu klase

Konstruktor klase

- Konstruktor klase omogućava kreiranje i inicijalizaciju objekta klase
- Konstruktor je javna funkcija __construct()

Konstruktor klase

construct -snippet

```php
<?php
class Osoba
{
    public string $ime;
    public string $prezime;

    public function __construct(string $ime, string $prezime) {
        $this->ime = $ime;
        $this->prezime =$prezime;
    }

    public function Stampaj()
    {
        echo "Ime: {$this->ime} Prezime: {$this->prezime} <br>";
    }
}
$os1 = new Osoba('Marko', 'Markovic');
$os1->Stampaj();
?>
```

Destruktor klase

- Destruktor je javna funkcija klase
- Destruktor se izvršava kada se objekat uništava, kada se skript završava ili se izlazi iz skripta
- Destruktor je javna funkcija __destruct()

```php
<?php
class Osoba
{
    public string $ime;
    public string $prezime;

    public function __construct(string $ime, string $prezime) {
        $this->ime = $ime;
        $this->prezime =$prezime;
    }

    public function __destruct()
    {
        echo "Destruktor {$this->Stampaj()} <br>";
    }

    public function Stampaj()
    {
        echo "Ime: {$this->ime} Prezime: {$this->prezime} <br>";
    }
}
$os1 = new Osoba('Marko', 'Markovic');
//$os1->Stampaj();
?>
```

Vidljivost polja i metoda klase

- **private**
  - vidljive samo u okviru klase
- **public**
  - vidljive i van klase
- **protected**
  - vidljive samo u okviru klase u kojoj su definisane ili klase koja je nasleđuje

Modifikator private

```php
<?php
    class MojaKlasa1
    {
        private  $a = 10;
        public function prikaziA()
        {
            return $this->a;
        }
    }
    $mk = new MojaKlasa1();
    //echo $mk->a; NE MOZE
    echo $mk->prikaziA(); //MOZE
 ?>
```

Modifikator public

```php
<?php
class MojaKlasa2
{
    public $a = 10;
    var $b = 20;
}
$mk = new MojaKlasa2();
echo $mk->a; //MOZE
echo $mk->b; //moze
?>
```

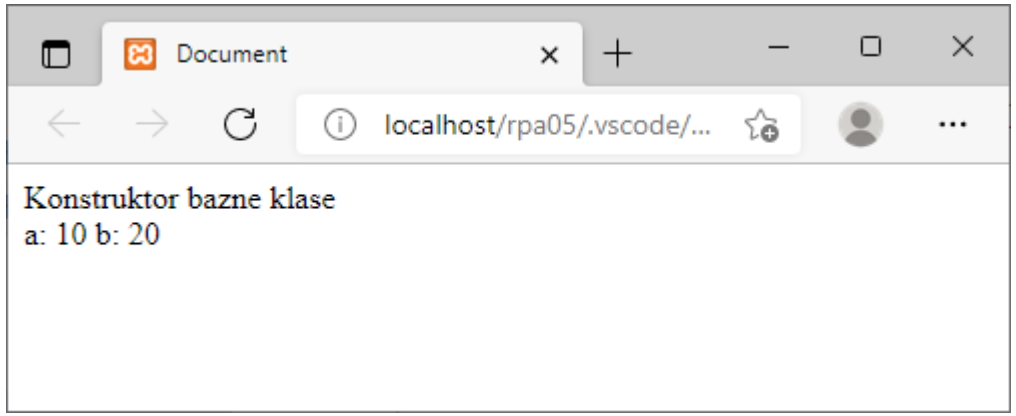
Nasleđivanje klase

- Preuzimaju se sve public i protected karakteristike roditeljske klase
- Postiže se ključnom rečju extends
- Ako se ne definiše konstruktor izvedene klase pozvaće se konstruktor bazne klase pri kreiranju objekta izvedene klase
- Konstruktor izvedene klase će prebrisati konstruktor bazne klase ako se definiše

```php
class A1
{

    protected $a;

    public function __construct()
    {
        $this->a = 10;
        echo "Konstruktor bazne klase <br>";
    }
}
```

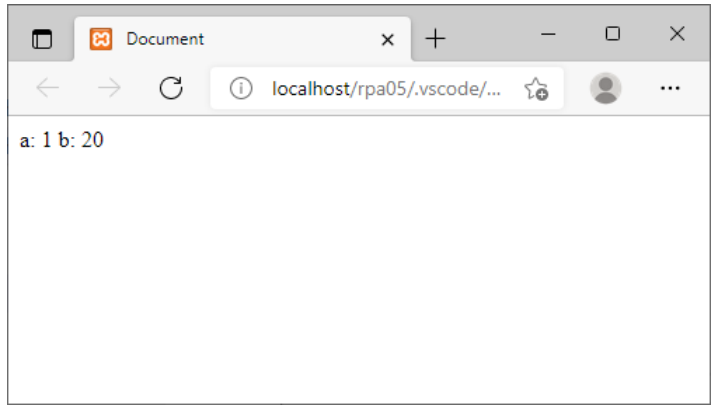
```php
class B1 extends A1
{
   public $b;

   public function Stampaj()
   {

       echo "a: {$this->a} b: {$this->b} <br>";
   }
}
```

```php
$b1 = new B1();
$b1->b = 20;
$b1->Stampaj();
```



```php
class A2
{
    protected $a =1;
    public function __construct()
    {
        $this->a = 10;
        echo "Konstruktor bazne klase <br>";
    }
}
```

```php
$b1 = new B2();
$b1->Stampaj();
```

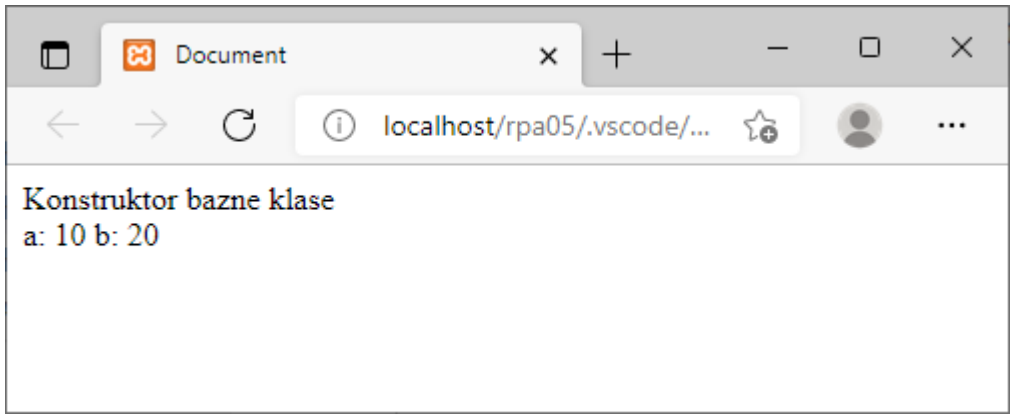



```php
class B2 extends A2
{
    public $b;

    function __construct()
    {
        $this->b = 20;
    }

    public function Stampaj()
    {
        echo "a: {$this->a} b: {$this->b} <br>";
    }
}
```

Poziv konstruktora bazne klase

```php
class A3
{
    protected $a;
    public function __construct()
    {
        $this->a = 10;
        echo "Konstruktor bazne klase <br>";
    }
}
```

```php
class B3 extends A3
{
    public $b;

    function __construct()
    {
        parent::__construct();
        $this->b = 20;
    }

    public function Stampaj()
    {

        echo "a: {$this->a} b: {$this->b} <br>";
    }
}
```



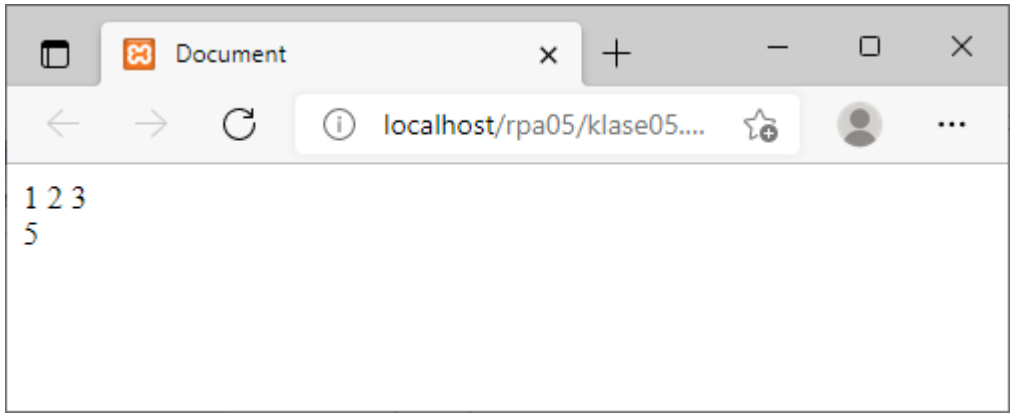

```php
$b1 = new B3();
$b1->Stampaj();
```

Prebrisavanje metode iz bazne klase

```php
class A4
   {
        public $a1;
        private $a2;
        protected $a3;
        public function __construct()
        {
            $this->a1 = 1;
            $this->a2 = 2;
            $this->a3 = 3;
        }

        public function Stampaj()
        {
            echo "{$this->a1} {$this->a2} {$this->a3}<br>";
        }
   }
```

Prebrisavanje metode iz bazne klase

```php
class B4 extends A4
  {
      public $b;

      public function __construct()
      {
          parent::__construct();
          $this->b = 5;
      }

      public function Stampaj()
      {
          parent::Stampaj();
          echo "{$this->b}";
      }
  }
  $p1 = new B4();
  $p1->Stampaj();
```

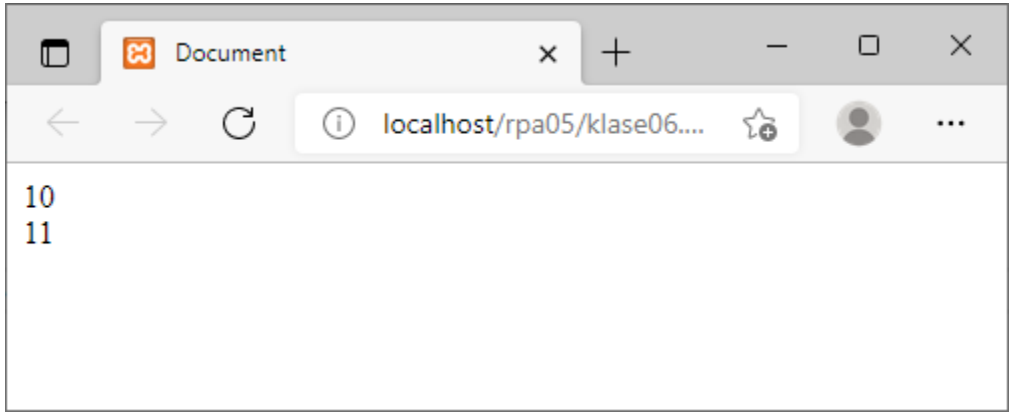


Statički članovi

- Može im se pristupiti bez instanciranja klase
- Označavaju se ključnom rečju static
- Pristupa im se uz pomoć scope resolution operatora ::
- Statičkim članovima se u metodama te klase pristupa posredstvom ključne reči self

Statički članovi

```php
<?php
class A5{
    public static $a = 10;

    public function __construct()
    {
        self::$a++;
    }
}

A5::$a = 10;
echo A5::$a . "<br>";

$a1 = new A5();
echo A5::$a . "<br>";
?>
```



Konstanta u klasi

```php
<?php
class Test{
    const pozdrav = "Pozdrav svima";
}
echo Test::pozdrav;
?>
```

Prebrisavanje toString() metode

```php
<?php
class Osoba3{
    public $ime;
    public $prezime;
    public $starost;
    public function __toString()
    {
        return "{$this->ime} {$this->prezime} {$this->starost}";
    }
}

$os1 = new Osoba3();
$os1->ime = 'Marko';
$os1->prezime = 'Markovic';
$os1->starost = 23;
echo  $os1->__toString();
?>
```

Definisanje prostora imena

namespace01.php

```php
<?php
namespace Primer1;
    class Osoba{
        public $ime;
        public $prezime;

        public function __construct($ime, $prezime)
        {
            $this->ime = $ime;
            $this->prezime = $prezime;
        }

        public function Stampaj()
        {
            echo "{$this->ime} {$this->prezime}";
        }

    }
?>
```

Referenciranje prostora imena

```php
<?php
    include "namespace01.php";
    $os = new Primer1\Osoba('Marko','Markovic');
    ?>
```