



WEB разработка

ООП Наследяване

Съдържание

Наследяване на класовете

Предефиниране на методи в класовете при наследяване

Наследяване на класовете

Class inheritance

Class inheritance

```
class Page {
    public $title;
    public $content;
    public $footer;
    public function __construct($t, $c, $f) {
        $this->$title = $t;
        $this->$content = $c;
        $this->$footer = $f;
    }
    public function render_body() {
        $str = '<h1>'.$this->$title.'</h1>';
        $str .= '<p>'.$this->$content.'</p>';
        $str .= '<p>'.$this->$footer.'</p>';

        return $str;
    }
}
```

Class inheritance - 2

Задача Създайте приложение, което

Страниците на приложението са Home, Contacts, About Us, Content

Само Home страницата има слайдер и реклама.

Всички страници имат

- header
- content
- footer

Class inheritance - 3

За да създаваме обекти от клас Home Page, притежаващи описаните свойства, трябва да копираме клас Page и в него да добавим характерните/различните за Home Page свойства.

/Home Page съдържа всички характеристики на останалите страници и в допълнение има слайдер и реклама/.

Class inheritance - 4

```
class HomePage {  
    public $title;  
    public $content;                                     <= повтарящ се код  
    public $footer;  
    public $slider;  
    public $banner;                                     <= специфични за класа HomePage свойства  
    public function __construct($t, $c, $f) {  
        $this->$title = $t;                             <= повтарящ се код  
        $this->$content = $c;  
        $this->$footer = $f;  
    }  
    public function render_body() {  
        $str = '<h1>'.$this->$title.'</h1>';  
        $str .= '<p>'.$this->$content.'</p>';             <= повтарящ се код  
        $str .= '<p>'.$this->$footer.'</p>';  
  
        return $str;  
    }  
}
```

Class inheritance - 5

Повторенията в кода се избягват като се използва наследяването на класовете или

```
class HomePage extends Page {  
    public $slider;  
    public $banner;  
}
```

```
$homepage = new HomePage();
```

```
$homepage->slider = 'Slider';  
$homepage->banner = 'Banner';
```

\$homepage притежава всички свойства и методи, характерни за обектите от клас Page плюс тези, принадлежащи на клас HomePage.

Class inheritance - 6

Класът HomePage може да има свой конструктор.

```
class HomePage extends Page {  
  
    public $slider;  
    public $banner;  
  
    public function __construct($h, $c, $f, $s, $b){  
        parent::__construct($h, $c, $f); //извикваме задължително родителския конструктор, ако  
        $this->$slider = $s; //искаме в момента на създаването на обект  
        $this->$banner = $b; //да се придаде стойност на всички свойства  
    }  
}
```

Предефиниране на методите в класа
Method overriding

Method overriding

В класът HomePage предефинирахме `__construct()`. Освен на `header`, `content`, `footer`, той задава стойности и на `slider` и `banner` свойствата.

Можем да предефинираме и други методи в класовете наследници

- В наследника създаваме едноименен метод с този в родителския
- Метода ще функционира във формата, дефиниран
 - В родителския клас, ако се извиква от негови обекти
 - В дъщерния клас, ако се извиква от негови обекти.

Method overriding - 2

```
//дефиниция в HomePage
public function render_body() {

    parent::render_body();

    $str .= '<p>.$this->slider.</p>';
    $str .= '<p>.$this->banner.</p>';

    return $str;

}
```

Дали ще извикваме родителския метод при предефинирането му в дъщерния, зависи от логиката на задачата, която решаваме.