УЕБ РАЗРАБОТКА

PHP ООП наследяване

Съдържание

- Наследяване /inheritance/
- Предефиниране /override/
- final

CLASS INHERITANCE

Class Inheritance

Наследяването е базова концепция в ООП.

Може да използваме един клас за основа на друг клас /други класове/, при което

създаваме своеобразна йерархия от класове - родителски и класове наследници.

Class Inheritance - 2

Припомнете си системата за класификация на организмите.

Вид - Род - Семейство - Разред - Клас - Тип - Царство

Всеки организъм принадлежи към един (и само един) вид, всеки вид принадлежи към един (и само един) род, и т.н. Всички живи организми са разпределени в 6 Царства.

Всеки *тип* отговаря на характеристиките на съответното *царство* и за да се различава от другите типове в това царство, притежава свои собствени черти. Аналогично/следващата степен/ всеки *клас* организми притежава чертите на съответния *тип* и за разлика от другите класове, има свои собствени специфични и т. н. всяка следваща степен надолу в йерархията на организмите притежава качествата на предходната и добавя собствени. Така описанието на всяка степен става по-конкретна от предишната...

Class Inheritance - 3

Всеки клас /наследник/ може да наследи друг клас /родител/, при което наследява всички негови свойства и методи.

Всеки клас може да наследи САМО един клас.

class Beverage extends Substances {...}

Class Inheritance - 4

parent:: се използва за обръщение към свойства/методи на класа родител.

Пример -

```
class Substances {
	private $name;
	function __construct($n){
		$this->name = $n;
	}
}

class Beverage extends Substances {
		private $alc;
		function __construct($n, $a){
		parent::__construct($n);
		$this->alc = $a;
	}
}
```

Class Inheritance - задачи

1. Дефинирайте класове за ключовите думи сгради, жилищни сгради, офис-сгради. Избройте свойства, характерни за трите ключови думи, като ги организирате в съответните класове - родителски и наследници.

2. Имате електронен магазин и трябва да дефинирате класове за потребителите, които са разпределине като администратори и купувачи. Избройте свойствата и методите /по 2/, характерни за тях, като ги организирате в съответните класове - родителски и наследници.

Извикване на наследени свойства и методи

Наследени свойства и методи

```php
<?php

class Plant
{
    public function printIInfo($string)
    {
        echo 'I am  a  ' . $string;
    }

    public function printPHP()
    {
        echo 'PHP is great.';
    }
}

class Tree extends Foo
{
    public function printDetailedInfo($string)
    {
        echo 'I am a tree, named - ' . $string . PHP_EOL;
    }
}
```

```php
$plant1 = new Plant();

$tree1 = new Tree();
$plant1->printItem('tree'); // Output: 'I am a tree'
$plant1->printPHP();       // Output: 'PHP is great'

$tree1->printDetailedInfo('The Tree');
// Output: 'I am a tree, named - The Tree'

$tree1->printPHP();       // Output: 'PHP is great'
```

Наследени свойства и методи - задачи

Задачи:

1. Създайте обекти от класовете, които дефинирахте в предходните класове.
2. Извикайте методи от класа родител върху обекти на класа наследник.

OVERRIDING
/предефиниране на методи/

overriding

Класът наследник може да предефинира поведение на метод на класа-родител.

```
1// Напитките са вещества, които хората могат да пият /по принцип/
class Beverage extends Substances {
function drink(){
        echo ‘People can drink ’.$this->name;
        }
}
2// Но ако искаме да конкретизираме, че алкохолът може да се консумира само ако си навършил 18
  г., може да презапишем drink()
class Liquor extends Beverage {
function drink(){
        echo ‘People under 18 years are not allowed to drink ’.$this->name;
        }
}
```

overriding - 2

Задачи:

1. В структурата от родителски и класове наследници, която създадохте в предишната задача - направете overriding на метод.
2. Създайте обекти и извикайте методите върху тях.

final

final

*С ключовата **final** предпазваме от евентуално предефиниране на*

- *методи в клас наследник*

final public function function_name() {...}

или

- *от наследяване на целия клас*

final class {...}