



OOP abstract methods and classes

PHP WebDevelopment 2019

Милена Томова
Vratsa Software

<https://vratsasoftware.com/>

Table of Contents

1. isA/hasA relations
2. Abstract classes and methods
3. Interfaces





Abstract classes and methods

Abstract classes and methods

We can separate the common logic - for a class and its ancestors - in a class that **will not have instances.**

Such a class begins with **abstract** keyword.

```
abstract class ClassName .....
```

The abstract class can be merely descriptive - can declare the properties and methods /only method names/.

The classes that extend the abstract class **will implement or extend** the abstract class methods.



Abstract classes and methods

```
abstract class AUser {  
    public function get_user(){  
        //method logic here  
    }  
    public function edit_user_data(){  
        //method logic here  
    }  
    public function redirect_user();  
    //the method depends on user type so every class that will extend  
    AUser has implementation for this method.  
}
```


Abstract classes and methods

```
abstract class AUser {  
    protected function get_user(){  
        //method logic here  
    }  
    public function edit_user_data(){  
        //method logic here  
    }  
    abstract function redirect_user();  
    //the method depends on user type so every class  
    that extends AUser must have implementation for this method.  
}
```

Abstract classes and methods

```
class SuperAdmin extends AUser {  
    protected function redirect_user(){  
        //logic for the SuperAdmin user to be redirected to the super-admin dashboard  
    }  
    public function edit_user_data(){  
        //logic for the SuperAdmin user to edit user`s data is different - can be  
        redefined or extended here  
    }  
}
```

- If a class **has an abstract method** it is an abstract class and its definition begins with **abstract**.
- A class can be an **abstract class** without having abstract methods.
- An abstract class is a class that by design will not have instances



Interfaces

Interfaces

It is like an abstract class having only abstract methods.

All classes that **implement** an interface must implement the methods declared in the interface.



Interfaces

Interfaces allow you to create code which specifies which methods a class must implement, without having to define how these methods are implemented.

Usually this method implementation is different in different classes.

It is possible to declare a constructor in an interface, which can be useful in some contexts.



Interfaces

One class can implement **more than one** interface.

An interface can be implemented by different, **not isA** connected classes.

Can interfaces be implemented by isA connected classes?



```
interface iInfo {  
    public function get_data( $param );  
    public function display_data();  
}
```



```
class User implements iInfo {  
    public function get_data( $param ){  
        //gets user from db  
        //returns user data from db  
    }  
  
    public function display_data(){  
        $data = $this->get_data( $this->username );  
        //display retrieved data from the db  
    }  
}
```



```
class Page implements iInfo {  
    public $content;  
    ...  
    public function __construct( $page_id ){  
        $this->content = $this->get_data( $page_id );  
        ...  
    }  
    public function get_data( $param ){  
        //gets page data from db  
        //returns data  
    }  
    public function display_data(){  
        //display $this->content  
        //display ....  
    }  
}
```

One class can implement **more than one** interface.

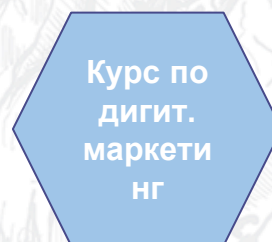
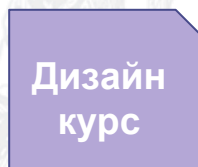
```
class User implements iInfo, iAnotherInterface {  
    public function get_data( $param ){  
        //gets user from db  
        //returns user data from db  
    }  
  
    public function display_data(){  
        $data = $this->get_data( $this->username );  
        //display retrieved data from the db  
    }  
    ...  
}
```

- The class that implements an interface must implement all the methods from the interface definition.
- If the method is declared in the interface that it will use param(s), the methods implemented in the class must be implemented with the same number of parameters.
- In an interface method access specifiers are only public.
- An interface can be extended by another interface using the **extends keyword**.

Questions?



Гнездото
Coworking



MindHub



Partners



**Telerik
Academy**



MindHub

ПРОМЯНАТА

Trainings @ Vratsa Software



- Vratsa Software – High-Quality Education, Profession and Jobs
 - www.vratsasoftware.com
- The Nest Coworking
 - www.nest.bg
- Vratsa Software @ Facebook
 - www.fb.com/VratsaSoftware
- Slack Channel
 - www.vso.slack.com

