

OOP PHP

CLASS

object

function_construct

object

```
$cat=new  
mammal("кошка");
```

property

```
echo $cat->name;
```

method

```
$cat->move (4);
```

```
<?php
class mammal
{
    public $blood, $legs;
    public function __construct($name)
    {
        $this->name = $name;
        $this->blood="теплая";
        echo "Construction class of mammal have started <br>";
    }
    public function move($legs)
    {
        if ($legs) echo "$this->name двигается на $legs ногах <br>";
        else echo "Животное плавает";
    }
}
?>
```

Destructor
function__destruct()
unset()

```
<?php
class mammal
{
    public $blood, $legs;
    public function __construct($name)
    {
        $this->name = $name;
        $this->blood="теплая";
        echo "Запущен конструктор класса mammal <br>";
    }
    public function move($legs)
    {
        if ($legs) echo "$this->name двигается на $legs ногах <br>";
        else echo "Животное плавает";
    }
    function __destruct() {
        echo "Вызван деструктор объекта <br>";
    }
}
$cat = new mammal("кошка");
echo $cat->name."<br>";
$cat-> move(4);
unset($cat);
echo "А теперь завершается работа сценария";
?>
```

Nested (enclosed) objects

```
<?php
class Room
{
    public $name;\
    function __construct($name="безымянная")
    {
        $this->name = $name;
    }
}
class House
{
    public $room;
}

$home = new House;
$home->room[] = new Room("спальня");
$home->room[] = new Room("кухня");
print($home->room[1]->name);
?>
```

Copying of objects

```
<?php
class simple_mammal
{
    public $legs;
}
$cat = new simple_mammal;
$cat -> legs = 4;
$whale = $cat;
$whale -> legs = 0;
echo $cat -> legs;
echo $whale -> legs;
?>
```

Cloning of objects

```
<?php
class mammal
{
    public $legs;
}
$cat = new mammal;
$cat -> legs = 4;
$whale = clone $cat;
$whale -> legs = 0;
echo $cat -> legs;
echo $whale -> legs;
?>
```

class inheritance

```
<?php
class mammal{}
class beast extends mammal
{
    public $fur;
    function __construct($name)
    {
        parent::__construct($name);
        echo "запущен конструктор класса beast <br>";
    }
    function move($legs)
    {
        if ($legs) echo "$this->name бегает, лазает по деревьям на ".
            $legs. " лапах <br>";
    }
    function description()
    {
        $this->fur="мягкая и пушистая";
        echo $this->name," ",$this->fur," . ";
        echo "Кровь - ", $this->blood, "<br>";
    }
}
?>
$Murka=new beast ("кошка");
$Murka-> move(4);
$Murka->description();
```


FINAL class

```
<?php
class mammal
{
    public $blood, $legs;
    public function __construct($name)
    {
        $this->name = $name;
        $this->blood="теплая";
        echo "Запущен конструктор класса mammal <br>";
    }
    public function move($legs)
    {
        if ($legs) echo "$this->name двигается на $legs ногах <br>";
        else echo "Животное плавает";
    }
    function __destruct() {
        echo "Вызван деструктор объекта <br>";
    }
}
```

```
class beast extends mammal
{
    public $fur;
    function __construct($name)
    {
        parent::__construct($name);
        echo "запущен конструктор класса beast <br>";
    }
    function move($legs)
    {
        if ($legs) echo "$this->name бегает, лазает по деревьям
        на ".
        $legs. " лапах <br>";
    }
}
```

```
function description()
{
    $this->fur="мягкая и пушистая";
    echo $this->name, " ", $this->fur, " . ";
    echo "Кровь - ", $this->blood, "<br>";
}
}
```

```
final class cat extends beast
```

```
{
    public $sound;
    function __construct($name)
    {
        parent::__construct($name);
        echo "Запущен конструктор класса cat <br>";
        $this->sound="мурр";
    }
    function speak()
    {
        echo $this->name, " говорит ", $this->sound."<br>";
    }
}
```

```
// Теперь создадим объект этого класса и вызовем его методы:
```

```
$Murka = new cat("кошка");
$Murka->move(4);
$Murka->description(); $Murka->speak();
?>
```

- `<?php`
- `define('USERNAME', "user45");`
- `define('PASSWORD', "pass45");`
- `define('DBNAME', "taxi");`
- `define('SERVER', "localhost");`
- `$link = mysqli_connect(SERVER, USERNAME, PASSWORD, DBNAME);`
- `if (!$link) {`
- `printf("Соединение установить не удалось : %s\n",`
- `mysqli_connect_error()); exit; }`

- `$query = "SELECT model, madein, reg_number FROM cars";`
- `$result=mysqli_query($link,$query);`
- `if ($result)`
- `{`
- `$rows = mysqli_num_rows($result);`
- `echo "<table >\n<tr>\n";`
- `echo "<th>Модель</th><th>Год
выпуска</th><th>Рег. номер</th></tr>\n";`
- `}`

- for (\$i=0; \$i<\$rows; \$i++)
- {
- \$r=mysqli_fetch_assoc(\$result);
- echo "<tr><td>". \$r["model"]."</td>";
- echo "<td>". \$r["madein"] ."</td>";
- echo "<td>". \$r["reg_number"]
- ."</td></tr>";
- }
- echo "</table>";
- mysqli_free_result(\$result);
- } mysqli_close(\$link); ?>

```
<?php
$mysqli = new mysqli('localhost',
'root', 'secret', 'firma');
if (mysqli_connect_errno()) {
    printf("Подключение
НЕВОЗМОЖНО: %s\n",
mysqli_connect_error());
    exit();
}
```

```
$query = "SELECT name, description  
FROM cities LIMIT 5";  
if ($result = $mysqli->query($query))  
{  
    while ($row = $result->fetch_row())  
    {  
        printf ("%s (%s)\n", $row[0],  
$row[1]);  
    }  
}
```



```
<?php
$mysqli = new mysqli('localhost',
'root', 'secret', 'firma');
if (mysqli_connect_errno()) {
    printf("Подключение
НЕВОЗМОЖНО: %s\n",
mysqli_connect_error());
    exit();
}
```

```
$stmt = $mysqli->prepare("INSERT INTO  
countries VALUES (?, ?)");  
$stmt->bind_param('ss', $name, $description);  
$name = 'Portugal';  
$description = 'The country is Spain';  
$stmt->execute();  
printf("%d Row inserted.\n",  
$stmt->affected_rows);  
$stmt->close();  
$mysqli->close();  
?>
```

Classes:

1) mysqli

2)mysqli-result

3)mysqli-stmt

Properties:

4)num_rows

5)affected_rows

Methods:

6)fetch_array(MYSQLI_ASSOC)

7)fetch_array(MYSQLI_NUM)

8)fetch_assoc()

9)fetch_row()

10)prepare()

11)query()

12)bind-param()

Objects:

`$mysqli` object of `mysqli`

`$result` object of `mysqli-result`

`$stmt` object of `mysqli-stmt`

Examples:

```
echo $result->num_rows;
```

```
$result->fetch_assoc();
```