

# Переменные названия свойств

Пусть у нас есть вот такой класс **User**:

```
1  <?php
2  class User
3  {
4      public $name;
5      public $age;
6
7      public function __construct($name, $age)
8      {
9          $this->name = $name;
10         $this->age = $age;
11     }
12 }
13
14 $user = new User('Коля', 21);
15 echo $user->name; // выведет 'Коля'
16 ?>
```

На примере этого класса мы сейчас разберем то, что названия свойств можно хранить в переменной.

К примеру, пусть у нас есть переменная **\$prop**, в которой лежит строка 'name'.

Тогда обращение **\$user->\$prop** по сути эквивалентно обращению **\$user->name**. Такое хитрое обращения к свойствам класса используется редко, но иногда бывает полезно.

Посмотрим на примере:

```
1  <?php
2  $user = new User('Коля', 21);
3
4  $prop = 'name';
5  echo $user->$prop; // выведет 'Коля'
6  ?>
```

## Задача 14.1

Сделайте класс **City** (город), в котором будут следующие свойства - **name** (название), **foundation** (дата основания), **population** (население). Создайте объект этого класса.

## Задача 14.2

Пусть дана переменная **\$props**, в которой хранится массив названий свойств класса **City**. Переберите этот массив циклом **foreach** и выведите на экран значения соответствующих свойств.

# Массив свойств

Пусть теперь дан вот такой класс **User**:



```
1  <?php
2  class User
3  {
4      public $surname; // фамилия
5      public $name; // имя
6      public $patronymic; // отчество
7
8      public function __construct($surname, $name, $pa
          tronymic)
9      {
10         $this->surname = $surname;
11         $this->name = $name;
12         $this->patronymic = $patronymic;
13     }
14 }
15 ?>
```

И пусть дан массив свойств

```
1  <?php
2  $props = ['surname', 'name', 'patronymic'];
3  ?>
```

Попробуем теперь вывести значение свойства, которое хранится в нулевом элементе массива, то есть в **\$props[0]**.

На самом деле, если просто попытаться сделать **\$user->\$props[0]** - это не будет работать:

```
1  <?php
2  $user = new User('Иванов', 'Иван', 'Иванович');
3
4  $props = ['surname', 'name', 'patronymic'];
5  echo $user->$props[0]; //!! так работать не
      будет
6  ?>
```

Для того, чтобы такое сложное имя свойства заработало, его нужно взять в фигурные скобки, вот так:

```
1  <?php
2  $user = new User('Иванов', 'Иван', 'Иванович');
3
4  $props = ['surname', 'name', 'patronymic'];
5  echo $user->{$props[0]}; // выведет 'Ива
      нов'
6  ?>
```

## Задача 14.3

Скопируйте мой код класса **User** и массив **\$props**. В моем примере на экран выводится фамилия юзера. Выведите еще и имя, и отчество.

# Ассоциативный массив

Массив, кстати, может быть и ассоциативным:

```
1  <?php
2  $user = new User('Иванов', 'Иван', 'Иванович');
3
4  $props = ['prop1' => 'surname', 'prop2' =>
      'name', 'prop3' => 'patronymic'];
5  echo $user->{$props['prop1']}; // выведет 'Ив
```

```
6 |      анов'
```

## Имя свойства из функции

Имя свойства также может быть из функции:

```
1 | <?php
2 |     function getProp()
3 |     {
4 |         return 'surname';
5 |     }
6 |
7 |     $user = new User('Иванов', 'Иван', 'Иванович');
8 |     echo $user->{getProp()}; // выведет 'Ива
9 |     нов'
```

## Имя свойства из свойства другого объекта

Имя свойства может быть даже свойством другого объекта.

Проиллюстрируем кодом. Пусть для примера дан объект **Prop**, который в свойстве value будет содержать название свойства объекта **User**:

```
1 | <?php
2 |     class Prop
3 |     {
4 |         public $value;
5 |
6 |         public function __construct($value)
7 |         {
8 |             $this->value = $value;
9 |         }
10 |     }
11 | ?>
```

Давайте выведем фамилию юзера с помощью объекта **Prop**:

```
1 | <?php
2 |     $user = new User('Иванов', 'Иван', 'Иванович');
3 |     $prop = new Prop('surname'); // будем выводить зна
4 |         чение свойства surname
5 |     echo $user->{$prop->value}; // выведет 'Ива
6 |     нов'
```

## Имя свойства из метода другого объекта

Имя свойства также может браться из метода другого объекта:

```
1 | <?php
2 |     class Prop
3 |     {
4 |         private $value;
5 |
6 |         public function __construct($value)
7 |         {
8 |             $this->value = $value;
9 |         }
10 |
11 |         public function getValue()
12 |         {
13 |             return $this->value;
```



```
14 |     }
15 | }
16 | ?>
```

Давайте выведем фамилию юзера:

```
1 | <?php
2 |     $user = new User('Иванов', 'Иван', 'Иванович');
3 |     $prop = new Prop('surname'); // будем выводить зна
      чение свойства surname
4 |
5 |     echo $user->{$prop->getValue()}; // выведет 'Ив
      анов'
6 | ?>
```

