Магический метод __toString в ООП на PHP

Методы PHP, начинающиеся с двойного подчеркивания __, называются *магическим*. Магия таких методов состоит в том, что они могут вызываться при совершении какого-то действия автоматически.

Первый магический метод, который мы с вами изучим, называется **__toString**. Он вызывается при попытке приведения экземпляра класса к строке.

Понятие *приведение к строке* лучше разберем на практическом примере.

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

```php
<?php
  class User
  {
    private $name;
    private $age;

    public function __construct($name, $age)
    {
      $this->name = $name;
      $this->age = $age;
    }

    public function getName()
    {
      return $this->name;
    }

    public function getAge()
    {
      return $this->age;
    }
  }
?>
```

Давайте создадим объект этого класса:

```php
<?php
  $user = new User('Коля', 25); // создаем объ
    ект класса
?>
```

А теперь попытаемся вывести созданный объект через **echo**:

```php
<?php
  $user = new User('Коля', 25);
  echo $user; // пытаемся вывести объект чер
    ез echo
?>
```

Попытка сделать вывод объекта через **echo** и есть преобразование к строке. В данном случае PHP выдаст ошибку, так как просто так объекты в строку не преобразуются.

Для того, чтобы убрать ошибку, мы должны в явном виде сказать PHP, что делать при попытке преобразовать объект в строку. Для этого и существует магический метод **__toString**.

Если в коде нашего класса сделать такой метод, то результат этого метода (то есть то, что он вернет через **return**) и будет строковым представлением объекта.

Пусть мы хотим, чтобы при попытке вывести объект через **echo**, выводилось имя юзера. Значит сделаем метод **__toString** и вернем в нем значение свойства **name**:

```php
<?php
  class User
  {
    private $name;
    private $age;

    public function __construct($name, $age)
    {
      $this->name = $name;
      $this->age = $age;
    }

    // Реализуем указанный метод:
    public function __toString()
    {
      return $this->name;
    }

    public function getName()
    {
      return $this->name;
    }

    public function getAge()
    {
      return $this->age;
    }
  }
?>
```

Проверим:

```php
<?php
  $user = new User('Коля', 25);
  echo $user; // выведет 'Коля' - все работает!
?>
```

Задача 53.1

Сделайте класс **User**, в котором будут следующие свойства - **name** (имя), **surname** (фамилия), **patronymic** (отчество).

Сделайте так, чтобы при выводе объекта через **echo** на экран выводилось ФИО пользователя (фамилия, имя, отчество через пробел).

Применение

Пусть у нас есть вот такой класс, с помощью которого можно добавлять элементы в массив и находить их сумму:

```php
<?php
  class Arr
  {
    private $numbers = [];

    public function add($num)
    {
      $this->numbers[] = $num;
      return $this;
    }

    public function getSum()
    {
```

```
        return array_sum($this->numbers);
      }
    }
?>
```

Давайте вспомним, как мы пользовались этим классом:

```
<?php
  $arr = new Arr;
  echo $arr->add(1)->add(2)->add(3)->get
    Sum(); // выведет 6
?>
```

Как вы видите, у нас будет цепочка методов **add**, а последним методом мы всегда должны вызвать **getSum**, чтобы получить сумму.

Давайте сделаем так, чтобы этот метод не нужно было писать, если мы выводим результат через **echo**.

Для этого нам и пригодится изученный метод **___toString**.

Есть, однако, один нюанс, мы сейчас рассмотрим.

Пусть наша реализация метода **___toString** будет такой:

```
<?php
  public function __toString()
  {
    return array_sum($this->numbers);
  }
?>
```

Данный код выдаст ошибку, так как **___toString** обязательно должен вернуть строку, а результатом **array_sum** будет число.

Исправим проблему, принудительно преобразовав результат в строку:

```
<?php
  public function __toString()
  {
    return (string) array_sum($this->numbers);
  }
?>
```

Применим изменения:

```
<?php
  class Arr
  {
    private $numbers = [];

    public function add($num)
    {
      $this->numbers[] = $num;
      return $this;
    }

    public function __toString()
    {
      return (string) array_sum($this->numbers);
    }
  }
?>
```

Проверим:

```
<?php
  $arr = new Arr;
  echo $arr->add(1)->add(2)->add(3); // выв
    едет 6
?>
```

Задача 53.2

Не подсматривая в мой код, реализуйте такой же класс **Arr**.