Контроллеры в своем MVC фреймворке на PHP

Как вы уже знаете, все контроллеры нашего фреймворка имеют метод render, который нужно вызывать для отправки данных в представление. Этот метод наши, пользовательские контроллеры наследуют от родительского класса Controller, расположенного в ядре.

Давайте сделаем этот родительский класс в файл */core/Controller.php*:

```php
<?php
  namespace Core;

  class Controller
  {
    protected function render($view, $data) {

    }
  }
?>
```

Как вы видите, метод **render** параметром принимает имя представления и данные для отображения.

Пусть этот метод возвращает объект специального класса Page, в котором будет содержаться информация о представлении данных действия контроллера.

Пусть в этом классе Page содержится имя view, данные, а также тайтл страницы и имя файла с макетом сайта:

```php
<?php
  namespace Core;

  class Page
  {
    private $layout;
    private $title;
    private $view;
    private $data;

    public function __construct($layout, $title, $view, $data)
    {
      $this->layout = $layout;
      $this->title  = $title;
      $this->view   = $view;
      $this->data   = $data;
    }

    public function __get($property)
    {
      return $this->$property;
    }
  }
?>
```

Тогда код нашего метода render будет выглядеть вот так:

```php
<?php
  namespace Core;
```

```
3
4     class Controller
5     {
6       protected $layout = 'default';
7
8       protected function render($view, $data) {
9         return new Page($this->layout, $this->tit
            le, $view, $data);
10       }
11    }
12  ?>
```

Поясню, что здесь происходит. Имя представления и данные приходят параметрами метода. Однако, в пользовательском контроллере задается еще и тайтл страницы - путем записывания свойства title. Значит, в $this->title и будет содержаться тайтл, который мы передадим конструктору класса Page.

Есть также нюансы с лейаутом. Как вы знаете, наш фреймворк использует лейаут из файла default.php. На самом деле, каждое действие может иметь и другой лэйаут. Для этого нужно в самом действии в свойство layout записать другое имя лэйаута.

Как это достигается: наш родительский контроллер имеет свойство layout, по умолчанию имеющее значение 'default.php'. Это и будет лэйаутом по умолчанию. Однако, если действие пользовательского контроллера переопределит значение свойства layout, то и лэйаут будет другим.

Задача 104.1

Скопируйте код моего класса Controller и разместите его в файле */core/Controller.php*.

Задача 104.2

Скопируйте код моего класса Page и разместите его в файле */core/Page.php*.