

УДК 004.55

Є.Ю. Катаєва, В. Ю. Шадхін, О. Л. Котінов,  
І. А. ЛитвиненкоКиївський національний університет технологій та дизайну,  
КиївДОСЛІДЖЕННЯ ТА ОПТИМІЗАЦІЯ ШВИДКОДІ  
WEB-ДОДАТКІВ

Розглянуто платформу CodeIgniter 2.1.0 для розробки Web-додатків. Наведено приклад створення Web-додатка на основі MVC моделі. Для створення додатків використовується середовище Aptana, web-сервер Apache, операційна система Linux Ubuntu 11.10 та мова програмування PHP.

**Ключові слова:** Метод MVC, CodeIgniter, розробка web-додатків

ИССЛЕДОВАНИЯ И  
ОПТИМИЗАЦИИ  
БЫСТРОДЕЙСТВИЯ WEB-  
ПРИЛОЖЕНИЙ

Рассмотрена платформа CodeIgniter 2.1.0 для разработки Web-приложений. Приведен пример создания Web-приложения на основе MVC-модели. Для создания приложений используется среда Aptana, web-сервер Apache, операционная система Linux Ubuntu 11.10 и язык программирования PHP.

RESEARCH AND OPTIMIZATION  
PERFORMANCE WEB-  
APPLICATIONS

Considered CodeIgniter 2.1.0 platform for developing Web-based applications. An example of the creation of Web-based applications based on MVC model. To create the application environment used Aptana, web-server Apache, the operating system Linux Ubuntu 11.10 and the programming language PHP.

## Вступ

На сьогодні існує багато платформ (фреймворків) для розробки web-додатків.

До списку таких платформ можна віднести: Zend Framework, CakePHP, Kohana, Symfony, Yii. Саме на основі наведених фреймворків будуються великі, високопродуктивні, масштабні web-проекти.

## Об'єкти та методи дослідження

Об'єктом дослідження є використання платформи CodeIgniter, яка працює за принципом моделі MVC. Розглянемо петтерн MVC докладніше.

MVC-модель була розроблена для того, щоб отримати більш структурований код додатка. Розшифровується MVC як "Модель-Відображення-Контролер" (Model-View-Controller). Зараз можна виділити два найбільш типових методи створення web-додатків (сайтів).

Перший метод називається "класичний", передбачає те, що в одному файлі може міститися код різних мов програмування і розмітки. На початку файлу здійснюється

починається html-розмітка сторінки. Причому всередині html-розмітки в потрібних місцях робляться вставки PHP-запит до бази даних для отримання з неї будь-якої інформації (використовується мова запитів SQL для взаємодії з базою даних). Після цього, як правило, коду, які виконують управління сайтом та є його логікою. Таким чином, маємо в одному файлі: SQL, (X)HTML і PHP.

Другий метод пов'язаний з паттерном MVC (Модель-Відображення-Контролер). Основна ідея цього підходу полягає у необхідності поділу однорідних елементів по різних файлах таким чином, щоб в одному файлі був фрагмент, написаний однією мовою, але таке трапляється вкрай рідко.

Крім ідеї рознесення різних мов в різні файли, ключовою концепцією є також поділ файлів на групи відповідно з тими функціями, які вони виконують у додатку.

Дана модель передбачає поділ всіх файлів, задіяних при розробці сайту на три групи:

1. Файли групи "модель".
2. Файли групи "відображення".
3. Файли групи "контролер".

Принцип роботи обох способів показано на рис. 1.

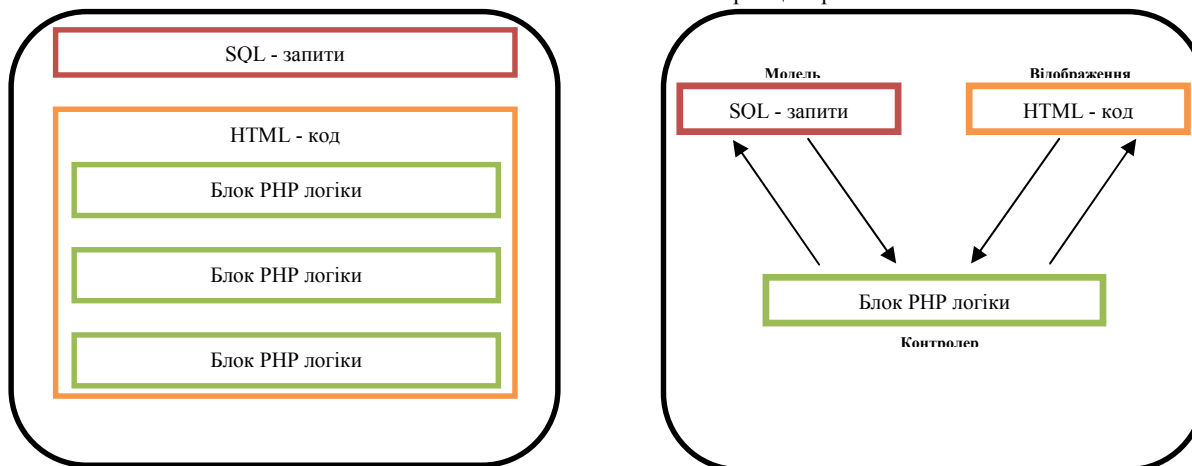


Рис. 1. Методи створення web-додатків

В межах даної схеми в моделі відбуваються операції, пов'язані з взаємодією з базою даних: вилучення даних, їх модифікація і видалення, підрахунок кількості записів у певних таблицях і т.п.

У контролері знаходиться логіка програми, тобто те, що визначає його функціонал.

Відображення ж призначено для показу кінцевому користувачеві.

Двонаправлені стрілки на рис. 1 показують те, що в парах "Модель - Контролер" і "Контролер - Відображення" існує взаємозв'язок. Розглянемо цей взаємозв'язок докладніше (рис.2).

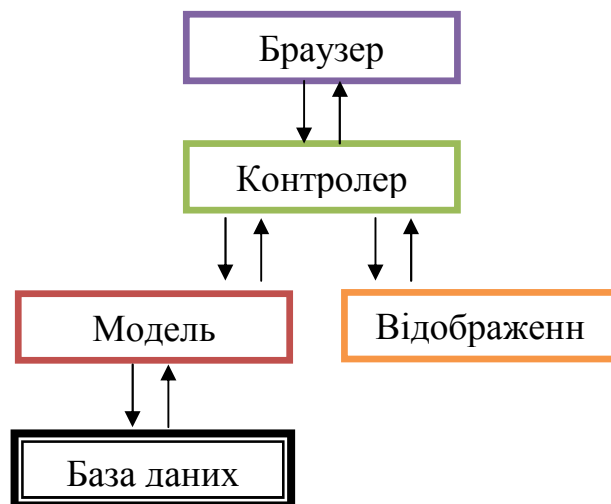


Рис. 2. Взаємодія елементів MVC

Якщо розглянути взаємодії елементів, то можна зробити опис роботи MVC:

1. Користувач вводить адресу, і браузер звертається до контролера.
2. Контролер звертається до моделі.
3. Модель звертається до бази даних (наприклад, для отримання необхідної для виведення інформації)
4. Інформація з бази потрапляє назад у модель.
5. З моделі інформація передається в контролер.
6. Контролер передає цю інформацію у відображення.
7. Відображення виводиться в браузер за допомогою контролера.

CodeIgniter – це готовий каркас для додатків, які будуть будуватися на його основі. У цей каркас включені найбільш часто використовувані бібліотеки.

Платформа є інструментарієм для тих, хто буде web-додатки на РНР. Її мета в тому, щоб дати змогу розробляти додатки швидше, ніж якщо б ви писали код з нуля, надаючи багатий набір бібліотек для часто використовуваних завдань, а також простий інтерфейс і логічну структуру для доступу

до цих бібліотек. CodeIgniter дозволяє творчо зосередитися на проектах, використовуючи мінімальний обсяг коду, необхідний для того чи іншого завдання.

Основне ядро вимагає невеликої кількості бібліотек. Це відрізняє CodeIgniter від більшості фреймворків, які вимагають значно більших ресурсів. Додаткові бібліотеки завантажуються динамічно за запитом, залежно від потреб в той чи інший момент так, що ядро залишається дуже гнучким і швидким.

Платформа складається з папок Application та System. Вона буде містити ядро і все, що з ним пов'язано. Папка Application призначена для користувача, її склад наведений на рис. 3.

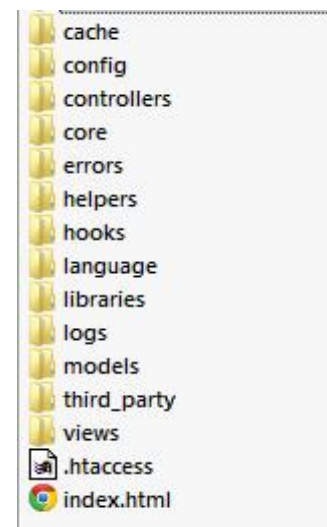


Рис. 3. Склад папки Application

## Постановка завдання

Завданням дослідження є вирішення проблеми розробки великих web-додатків, які потребують великих затрат часу, людських ресурсів, структуризації коду для зменшення обсягу часу виконання запиту такого додатка.

Для програмування додатка знадобиться багато часу і це пояснюється тим, що потрібно повністю реалізувати функціонал додатка та правильно виконати поставлену задачу, для цього знадобиться написання великої кількості бібліотек, класів, функцій тощо, тому для вчасного виконання завдання потрібна певна кількість спеціалістів, котрі будуть все це реалізовувати. Але навіть у разі успішної реалізації завдання, додаток може бути невдалим. Швидкість виконання додатка обмежена пропускнуою здатністю мережі та апаратним забезпеченням обчислювальної системи. Додаток може виконуватися тривалий час, якщо його код неструктурований. Також неструктурований код складний для розуміння людиною. В ході написання

додатка, до виконання проекту можуть бути підключені інші люди, а у випадку неструктурованого коду знадобиться багато часу для його аналізу та розуміння.

### Результати та їх обговорення

З метою демонстрації переваг платформи CodeIgniter та моделі MVC розроблено невеликий додаток, який показує ефективність використання платформи для економії обсягу часу для розробки, та структуризації коду додатка і скорочення часу його виконання.

Перший приклад є класичною моделлю розробки web-додатків. Код PHP та HTML розмітки записані в одному файлі і код є неструктурованим.

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <meta http-equiv="X-UA-Compatible"
content="IE=edge,chrome=1" />
  <title>Registration</title>
  <meta name="description" content=""
/>
  <meta name="author" content=" " />

  <meta name="viewport"
content="width=device-width; initial-
scale=1.0" />

  <link rel="shortcut icon"
href="/favicon.ico" />
  <link rel="apple-touch-icon"
href="/apple-touch-icon.png" />
</head>
<body>
  <div>
    <?php
      $login = $_POST['login'];
      $password=$_POST['user_pass'];
      $name=$_POST['name'];
      $second_name=$_POST['second_name'];
      $user_mail=$_POST['user_mail'];
      if (isset($_POST['login']))
    {
      if($login=='')
    {
      unset ($login);
      echo "Вы не ввели Login";}
    }
    if (isset($_POST['user_pass'])
      if($password=='')
    {
      unset ($password);

```

```

      echo "Вы не ввели Password";
    }
  }
  if (isset($_POST['name']))
    if($name==''){
      unset ($name);
      echo "Вы не ввели Name";
    }
  }
  if (isset($_POST['second_name'])

    if($second_name==''){
      unset ($second_name);
      echo "Вы не ввели Second_name";
    }
  }
  if (isset($_POST['user_mail'])

    if($user_mail==''){
      unset ($user_mail);
      echo "Вы не ввели User_mail";
    }
  }
  if($login!='' && $password!='' &&
$name!='' && $second_name!='' &&
$user_mail!=''){
    $dblink=mysql_connect("localhost","r
oot","wer w");
    $selected=mysql_select_db("users",
$dblink);
    if ($selected){
      $sql='INSERT INTO
`users`.`t_users`(`login`,`password`,`name
`,`second_name`,`user_mail`)
      VALUES ("'.$login.'",
"'.$password.'", "'.$name.'",
"'.$second_name.'", "'.$user_mail.'")';
    }
    else{
      die ('База данных не найдена или
отсутствует');
    }
    if (!mysql_query($sql)){
      echo 'Ошибка при добавлении данных';
    }
    else {
      echo "Вы зарегистрировались";   ?>
    }
  }
</div>
</body>
</html>

```

Цей код генерує сторінку реєстрації користувача, після заповнення полів вводу даних, дані передаються в базу даних користувачів. На рис.4,5 наведені результати роботи додатка.

Логин:	Login
Пароль:	*****
Имя:	Name
Фамилия:	SecondName
E-mail:	mymail@i.ua
<input type="checkbox"/>	Я подтверждаю правильность данных
<input type="button" value="Отправить"/>	

Рис. 4. Сторінка вводу даних

У результаті відправлення даних в таблиці бази даних буде створений запис тих даних, які ввів користувач.

Час виконання даного додатка наведено на рис. 6.



	login	password	name	second_name	user_mail
<input type="checkbox"/>  	Login	Password	Name	SecondName	mymail@i.ua

Рис. 5. Результат відправлення даних


Name Path	Method	Status Text	Type	Initiator	Size Content	Time Latency
 registration.php	POST	200 OK	text/html	Other	2.42KB 2.16KB	179ms 178ms

Рис. 6. Час виконання додатку

У разі використання платформи CodeIgniter код додатка буде виглядати зовсім по-іншому. Додаток буде складатися з таких файлів:

- контролер registration.php;
- модель mdl\_registration.php;
- відображення registration\_view.php.

Для підключення бази даних буде використовуватися файл database.php в директорії application/config/. В цій директорії будуть зберігатися всі файли для налаштування конфігурації додатка.

Файли будуть містити такий код:

```
Контролер registration.php
<?php
/**
 *
 */
class Registration extends
Ci_Controller {
function index() {
$this->load->view('registration');
$data = array(
'login'=>$this->input->post('login'),

'password'=>($this->input->post('user_pass')),

'name'=>$this->input->post('name'),
```

```
'second_name'=>$this->input->post('second_name'),

'user_mail'=>$this->input->post('user_mail'));
$this->load->model('Mdl_registration');
$this->Mdl_registration->new_User($data);
}
?>
Модель mdl_registration.php
<?php
/**
 */
class Mdl_registration extends
Ci_Model {
function new_User($data) {
$this->db->insert('t_users', $data);
}
}
?>
```

На рис. 6,7 наведені результати роботи додатка.

Логин:

Пароль:

Имя:

Фамилия:

Е-mail:

Я подтверждаю  
правильность данных

Рис. 7. Сторінка вводу даних (CodeIgniter)

У результаті відправлення даних в таблиці бази даних буде створений запис тих даних, які ввів користувач.

Час виконання цього додатку наведено на рис. 9.



	login	password	name	second_name	user_mail
<input type="checkbox"/>  	MyLogin	MyPass	MyName	MySecondName	mymail@ya.ru

Рис. 8. Результат відправлення даних (CodeIgniter)


Name Path	Method	Status Text	Type	Initiator	Size Content	Time Latency
 http://www.my-pr.ua/	POST	200 OK	text/html	Other	2.42KB 2.16KB	88ms 87ms

Рис. 9. Час виконання додатка (CodeIgniter)

Аналізуючи затрати часу на написання коду додатків, часу виконання та структуризації коду перевага надається платформі CodeIgniter, тому що використовуючи можливості, які надає нам платформа, можна написати професійний структурований код додатка, виконання якого витрачає майже в два рази менше часу у порівнянні з класичним методом програмування.

4. <http://codeigniter.com/>
5. <http://code-igniter.ru>

Стаття надійшла до редколегії 20.05.2012

**Рецензент:** д-р техн. наук, проф. В.М. Рудницький, Черкаський державний технологічний університет, Черкаси.

## Висновки

Запропоновано новий підхід до створення web-додатків з використанням моделі MVC платформи CodeIgniter. Використання методу дає змогу програмісту скоротити час створення web-сторінок за рахунок того, що кількість рядків скорочується на 25-40%, в той же час швидкодія обробки додатків збільшується більш ніж вдвічі.

## Список літератури

1. David Upton *CodeIgniter for Rapid PHP Application Development*. Packt Publishing, 2007. - 320c.
2. David Golding *Beginning CackePHP: From Novice to Professional*. APress, 2008. - 319c.
3. Thomas Myer *Professional CodeIgniter*. Wrox, 2008. - 307c.