

**Санкт-Петербургский Государственный Университет  
Математико-механический факультет**

Кафедра системного программирования

**Реализация RНР фреймворка**

Курсовая работа студентки 345 группы  
Золотухиной Алины Игоревны

Научный руководитель

Н. Н. Жуков

Санкт-Петербург

2010

## Оглавление

	стр.
Введение.....	3
1. Обзор существующих разработок .....	5
2. Реализация .....	7
3. Использование.....	10
Заключение .....	11
Список литературы .....	12

## Введение

Данная курсовая работа является частью проекта по разработке PHP-фреймворка.

В настоящее время PHP является одним из наиболее часто используемых языков программирования (по рейтингу ТЮВЕ [7] PHP стабильно занимает четвертую позицию после C, Java и C++, это в том числе означает, что PHP – первый в рейтинге среди языков веб-программирования).

Это не случайно – язык PHP имеет ряд преимуществ по сравнению с другими языками веб-программирования.

- PHP является достаточно простым языком – даже людям, впервые услышавшим о нём, будет несложно изучить его за короткое время.
- PHP похож на C и Perl, поэтому тем, кто знаком с этими языками, будет легко перейти на PHP.
- Одно из самых главных преимуществ языка PHP – это то, что он распространяется бесплатно с открытыми исходными кодами (Open Source). Такое распространение не только оказывается удобным для пользователей, но и приводит к более быстрому развитию языка, поскольку пользователи со всего мира могут его тестировать, оставлять свои отзывы, предлагать свои улучшения.
- В PHP отсутствуют скрытые издержки – необходимые функции и модули встроены изначально и, соответственно, тоже бесплатны (в отличие от, например, технологии ASP, где требуемые пакеты приходится докупать).
- Существует большое сообщество PHP-программистов, которое в трудных ситуациях может подсказать ответы на сложные вопросы.
- PHP является кросс-платформенным, он доступен для многих операционных систем (Microsoft Windows, Mac OS X, Linux, многие модификации Unix и другие), он поддерживает большинство современных веб-серверов (Apache, Microsoft Internet Information Server, Personal Web Server и многие другие), что означает свободу в выборе как операционной системы, так и вебсервера, а также лёгкий перенос кода с одной технологии на другую.
- PHP, будучи очень гибким, имеет возможность работать со многими различными типами баз данных, хотя большинство PHP-разработчиков используют MySQL (что оправдано его бесплатностью и наличием в языке PHP встроенных функция для работы с этой базой данных).

PHP-фреймворк – это программное обеспечение, облегчающее разработку проектов, использующих язык PHP. Фреймворк является абстракцией, в которой основной код, предоставляющий базовую функциональность, может быть переопределён или уточнён пользовательским кодом, предоставляющим дополнительную функциональность. Фреймворк можно назвать разновидностью библиотеки, поскольку он является допускающей повторное использование абстракцией кода, завернутой в API (Application Programming Interface), но в то же время он имеет некоторые черты, которые отличают его от библиотеки. У него существует некоторое поведение по умолчанию. Пользователь может расширять фреймворк, хотя есть некоторый неизменяемый код, который нельзя модифицировать.

Сегодня существует множество разнообразных PHP-фреймворков, и разработчику может быть сложно в них разобраться и выбрать для себя самый подходящий – на это может уйти много времени и сил перед тем, как будет выбран наилучший вариант. Цель курсовой работы заключается в сравнении существующих PHP-фреймворков и создании собственной разработки на основе проделанного сравнения. Было решено сделать фреймворк для маленьких несложных проектов, взяв за основу ключевые элементы существующих решений, но применяя наиболее подходящие для данной задачи технологии.

## 1. Обзор существующих разработок

В данной работе были рассмотрены следующие PHP-фреймворки:

- Zend
- CakePHP
- CodeIgniter
- Yii

### Zend

- Является открытым программным обеспечением под лицензией New BSD.
- Поддерживает только PHP 5.
- Фреймворк основан на идеях MVC – Model-View-Controller (Модель-Вид-Контроллер) [11]: модель, представляющая данные, отделена от представления, т.е. отображения информации, которое, в свою очередь, отделено от контроллера, отвечающего за поведение системы в целом.
- Поддерживает AJAX через JSON (JavaScript Object Notation) – текстовый формат обмена данными, аналогичный XML, но более краткий и удобочитаемый. AJAX(Asynchronous Javascript and XML) – концепция фонового обмена данных между браузером и веб-сервером в интерфейсах веб-приложений: страница полностью не перезагружается, изменяется только её часть.
- Есть возможность работы со многими базами данных, в том числе MySQL, Oracle, IBM DB2, Microsoft SQL Server, PostgreSQL, SQLite и Informix Dynamic Server.
- Реализована аутентификация, авторизация и управление сессиями.

### Недостатки

- Нет поддержки Object-Relational Mapping (ORM) – технологии программирования, которая связывает базы данных с концепциями языков ООП.

### CakePHP

- Является открытым программным обеспечением под лицензией MIT
- Поддерживает PHP 5 и PHP 4.
- Реализован MVC
- Поддерживает AJAX

### Недостатки

- Скучная документация
- Не очень удобная поддержка ORM

- Довольно медленный

### **CodeIgniter**

- Является открытым программным обеспечением под лицензией Apache/лицензией BSD
- Поддерживает PHP 5 и PHP 4.
- Реализован MVC
- Поддерживает использование Friendly URL – веб-адресов, удобных для восприятия человеком (например, адрес будет выглядеть как «frameworks/php/Basis» вместо «frameworks.php?type=php&name=Basis»)

Недостатки

- Нет поддержки ORM
- Нет поддержки Ajax

### **Yii**

- Является открытым программным обеспечением под лицензией BSD
- Поддерживает только PHP 5.
- Реализован MVC
- Использует ORM
- Очень быстрый

Недостатки:

- Неполная документация

Нашей целью была реализация удобного не требующего большого объема памяти фреймворка для быстрого создания небольших сайтов.

## 2. Реализация

### *Выбор языка*

Выбор языка производился между PHP 4 и PHP 5. Поскольку PHP 5 имеет гораздо больше возможностей (он получил некоторые черты объектно-ориентированных языков, увеличилась скорость работы стандартных PHP-функций), а популярность PHP 4 крайне мала, то было решено отказаться от поддержки PHP 4 и поддержать только PHP 5.

### *Структура MVC*

Было принято решение использовать шаблон программирования Model-View-Controller, поскольку он является предельно удобным для веб-разработки, вследствие разделения работ, связанных с дизайном, администрированием баз данных и разработкой.

### *Model - Работа с базами данных*

Было решено реализовывать поддержку баз данных с помощью ORM. Выбор производился между тремя ORM: Propel (MIT License), Doctrine (GNU Lesser General Public License) и Php-ActiveRecord (MIT License). Практически сразу же был выбран Php-ActiveRecord. Установка Propel требовала дополнительной настройки, что несовместимо с нашей целью создания простого фреймворка. Doctrine имела очень большую функциональность, большая часть которой нам бы не пригодилась, поскольку наш фреймворк создавался для несложных проектов. Кроме того, она была требовательна к объёму памяти. Что касается Php-activeRecord, то его установка сводится к простой распаковке архива, а интегрирование – к добавлению в код фреймворка нескольких строчек. Он реализует базовую функциональность и имеет небольшой вес, то есть является оптимальным для использования при создании небольших проектов.

### *View – Реализация с помощью Smarty*

Smarty (LGPL License) – это средство автоматизации создания уровня представления (уровня View). Основное его предназначение – отделение логики приложения от представления данных. Его использование позволяет менять логику приложения, не заботясь о том, что станет с представлением данных, или же наоборот, менять визуализацию данных, не изменяя логики приложения. Используя Smarty, можно также легко разделить работу дизайнеров и разработчиков – дизайнеру совсем не надо знать, какова основная логика, в то время как разработчик может не думать о том, как данные должны выглядеть на экране.

Кроме того, шаблоны могут быть написаны без использования PHP, то есть дизайнеру не требуется дополнительных знаний для выполнения своей работы.

### *Controller*

Основная логика фреймворка заключена в классе `BasisActionController`. Запрос приходит в файл `index.php`, который его перенаправляет в `BasisActionController`. Там, в методе `dispatch()`, инициализируется контроллер (`$controller`) и действие (`$action`), которые затем отдаются в метод `forward`. В методе `forward` создаётся переменная класса с именем нужного контроллера, и вызывается метод `dispatchAction`. В методе `dispatchAction` выполняется требуемое действие и происходит передача данных в `Smarty`, который затем, с помощью собственных функций, их отображает.

Также в `BasisActionController` происходит инициализация `Smarty` и связь с базой данных (с помощью `php-ActiveRecord`).

Все контроллеры, которые будет писать пользователь, должны наследоваться от класса `BasisActionController`.

### *Сессии*

Было решено реализовать поддержку аутентификации и авторизации. В классе `LoginController` реализованы методы регистрации (`registration`), входа (`login`) и выхода (`logout`). При начале работы в `index.php` командой `session_start()` запускается механизм сессий.

При регистрации(`registration()`) производится проверка входных данных (имени на непустоту, пароля на требуемую длину, совпадения пароля и повторного ввода пароля, правильного ввода цифр с картинки) и в том случае, когда всё в порядке, делается запись в базу данных и производится вход (логин).

При входе (`login()`) все переменные сессии обнуляются, и проверяется, есть ли в базе данных пользователь с введёнными именем и паролем. В случае, когда есть, сессия обновляется с новым `id`.

При выходе (`logout()`) опять же, все переменные сессии обнуляются, и удаляются куки сессии.

### *AJAX*

Для улучшения производительности и удобства пользователя было решено поддержать использование AJAX - Asynchronous Javascript and XML. Выбор используемой библиотеки делался между `Sajax` и `jQuery`. Некоторые другие библиотеки (`Projax`, `Dojo`) были



сразу определены как неподходящие в силу своей сложности и величины – нам была необходима лишь простейшая функциональность.

Sajax (BSD License) – очень простая в использовании, легко устанавливается, мало весит. То же самое можно сказать и про jQuery (MIT License и GNU GPL). Однако jQuery показалась более подходящей, поскольку является более распространённой библиотекой. Таким образом, выбор был остановлен на jQuery.

### ***Friendly URL***

Friendly URL – очень удобное свойство фреймворка, поскольку оно делает работу с проектом более понятной. В Basis есть переменная, отвечающая за то, включена ли поддержка Friendly URL, по умолчанию установленная в значение FALSE.

Если поддержка Friendly URL включена, то при обращении к BasisActionController вызывается метод getRequest(), в котором разбирается входная ссылка, разделяясь на controller, action и параметры.

### **3. Использование**

Фреймворк представляет собой папку, в которой в разных разделах хранятся как используемые библиотеки, так и написанные файлы.

Библиотеки (Smarty, php-ActiveRecord, jQuery) можно найти в папке lib, там же можно найти и BasisActionController.php и файлы для работы с Friendly URLs.

Приложение, создаваемое пользователем, целиком будет находиться в папке app. Внутри в папке controllers в ActionControllers должны располагаться контроллеры, создаваемые пользователем. Изначально там будет лежать файлы HomeController.php и LoginControllre.php, являющиеся частью маленького сайта, предназначенного для того, чтобы пользователь легче и быстрее понял возможность нашего фреймворка. Папка models предназначена для хранения моделей, шаблоны будут расположены в папке views.

## **Заключение**

Было проведено сравнение существующих PHP фреймворков. На основе проведённого сравнения был успешно разработан Basis с использованием технологий, применяемых в других фреймворках. Как и все рассмотренные технологии, Basis основывается на идеях паттерна Model-View-Controller. Реализация поддержки только PHP 5 позволяет оптимизировать работу приложения. Чтобы сделать Basis удобным для использования, была реализована поддержка ORM для упрощения работы с базами данных. Также фреймворк предоставляет поддержку AJAX для увеличения скорости работы сайтов. Конечно же, Basis не вписывается в ряд всем известных фреймворков, но целью разработки не была реализация инструмента, используемого для создания мощных сайтов. Basis создавался как инструмент для написания небольших простых сайтов, и эта цель была достигнута.

В дальнейшем планируется составить подробную документацию по разработанному фреймворку. Также хотелось бы получить какие-нибудь отзывы пользователей о сделанной работе, чтобы понять, насколько созданная нами разработка действительно полезна и необходима.

## Список литературы

1. 7 Reasons Why PHP is Better than ASP <http://www.ex-designz.net/articleread.asp?aid=159>
2. Ajax Toolkit for PHP – SAJAX <http://www.modernmethod.com/sajax/faq.phtml>
3. CakePHP : the rapid development php framework. Features  
<http://cakephp.org/pages/features>
4. CodeIgniter – Open source PHP Web application framework <http://codeigniter.com/>
5. JavaScript, Ajax and the DOM <http://www.webdirections.org/sotw10/script/>
6. PHP.SU – PHP FAQ <http://php.su/faq/?basic>
7. TIOBE Software: Tiobe Index <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>
8. Yii PHP Framework – Features <http://www.yiiframework.com/features/>
9. Zend Framework: About <http://framework.zend.com/about/overview>
10. CodeIgniter – Википедия <http://ru.wikipedia.org/wiki/CodeIgniter>
11. Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влссидес «Приемы объектно-ориентированного программирования. Паттерны проектирования». – СПб.: Питер. – 2008. – 366 с.
12. Лаура Томсон, Люк Веллинг «Разработка Web-приложений на PHP и MySQL». – СПб. : DiaSoft. – 2003. – 672 с.