

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**  
**ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**  
Геолого-географический факультет  
Кафедра динамической геологии

**РНР**

Реферат по информатике

**Выполнил:**

студент группы № 02601

Суцев Илья

студент группы № 02601

Прасолов Кирилл

«\_\_»\_\_\_\_\_2016

**Проверил:**

ассистент кафедры

Динамической геологии

\_\_\_\_\_И.А.Афонин

**Томск - 2016**

## **ОБЪЕКТ ИССЛЕДОВАНИЯ, ЦЕЛИ И ЗАДАЧИ**

**Объект исследования**— язык программирования PHP.

**Цель**—рассмотрение языка PHP, одного из лидеров среди языков применяемых для создания динамических веб-сайтов.

**Задача** — изучение истории данного языка, изучение особенностей его реализации, выявление достоинств и недостатков.

# 1.ИСТОРИЯ СОЗДАНИЯ

1. Созданное в 1994 году Расмусом Лердорфом, самое первое воплощение PHP было простым набором CGI-скриптов, написанных на языке программирования Си. Изначально используя их для отслеживания посещений своего веб-резюме, он назвал этот набор скриптов "Personal Homepages Tools" ("Инструменты для персональных домашних страниц"), но более часто упоминалось название "PHP Tools".
2. Версия PHP 3.0 подверглась значительной переработке, определившей современный облик и стиль языка программирования.
3. PHP 4.0 вышел в мае 2000 года был основан на движке Zend Engine и имел ряд нововведений.
4. Пятая версия PHP была выпущена разработчиками 13 июля в 2004 году с обновленным ядром, что существенно увеличило эффективность интерпретатора.
5. Шестая версия PHP разрабатывалась с октября 2006 года. Было сделано множество нововведений, как, например, исключение из ядра регулярных выражений POSIX и «длинных» суперглобальных массивов.
6. 3 декабря 2015 года вышла версия 7.0.0. разрабатывалась с упором на увеличение производительности и уменьшение потребления памяти. В новой версии добавлена возможность указывать тип возвращаемых из функции данных, добавлен контроль передаваемых типов для скалярных данных, а также новые операторы.

## 2.ОСОБЕННОСТИ РЕАЛИЗАЦИИ

РНР-скрипты обычно обрабатываются интерпретатором в порядке, обеспечивающем кроссплатформенность разработанного приложения:

1. лексический анализ исходного кода и генерация лексем,
2. синтаксический анализ полученных лексем,
3. генерация байт-кода
4. выполнение байт-кода интерпретатором (без создания исполняемого файла).

Для увеличения быстродействия приложений возможно использование специального ПО, так называемых акселераторов.

Важной особенностью является то, что разработчику нет необходимости заботиться о распределении и освобождении памяти. Ядро РНР реализует средства для автоматического управления памятью; вся выделенная память возвращается системе после завершения работы скрипта.

## 2.1. РАСШИРЕНИЯ

Интерпретатор состоит из ядра и подключаемых модулей, «расширений», представляющих собой динамические библиотеки. Расширения позволяют дополнить базовые возможности языка, предоставляя возможности для работы с базами данных, сокетами, динамической графикой, криптографическими библиотеками, документами формата PDF и тому подобным.

Существует огромное количество расширений, как стандартных, так и созданных сторонними компаниями и энтузиастами, однако в стандартную поставку входит лишь несколько десятков хорошо зарекомендовавших себя. Множество расширений доступно в репозитории PECL.

## 2.2.ПАРАМЕТРЫ НАСТРОЙКИ

Интерпретатор PHP имеет специальный конфигурационный файл , содержащий множество настроек, изменение которых влияет на поведение интерпретатора. Имеется возможность отключить использование ряда функций, изменить ограничения на используемую скриптом оперативную память, время выполнения, объём загружаемых файлов.

Возможно дробление большого конфигурационного файла на части. Например, широко распространена практика вынесения настроек расширений в отдельные файлы. Параметры интерпретатора могут быть переопределены в файлах конфигурации HTTP-сервера (например, .htaccess в Apache) или в самом скрипте во время выполнения при помощи команды `ini_set`.

## 2.3. РЕЖИМЫ ЗАПУСКА ИНТЕРПРЕТАТОРА(SAPI)

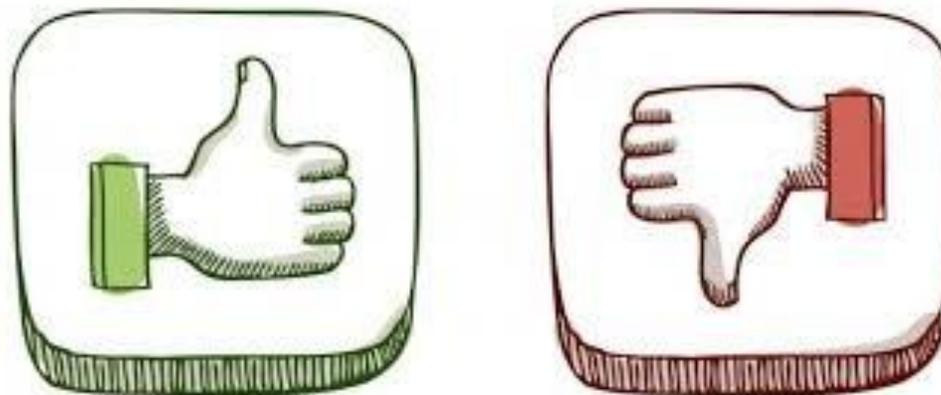
SAPI — это внешний уровень абстракции, предназначенный для встраивания интерпретатора в другие приложения и отвечает за его работу. Существует несколько основных SAPI определяющих способы запуска и использования PHP:

- В качестве модуля к веб-серверу. В этом случае интерпретатор PHP выполняется в окружении процесса веб-сервера.
- CGI SAPI. Использование CGI подразумевает запуск нового процесса для обработки каждого запроса. Сам принцип такого использования подразумевает, что интерпретатор PHP исполняет только один скрипт, после чего заканчивает свою работу.
- FPM SAPI, известный как php-fpm
- появилась возможность перезапуска пула интерпретаторов PHP без потери запросов, запуск нескольких пулов под разными пользователями, аварийный перезапуск интерпретаторов в случае проблем с ними и ещё несколько приятных дополнений.
- В качестве скрипта командной строки (CLI SAPI), являющегося исполняемым файлом, который вызывается пользователем из командной строки; скрипт выполняется в окружении вызвавшего пользователя. В этом случае возможно использование PHP для создания клиентских GUI-приложений и решения административных задач в операционных системах.

Начиная с версии PHP 5.4.0 в CLI SAPI появилась возможность запуска PHP как отдельного HTTP сервера.

### 3.ДОСТОИНСТВА И НЕДОСТАТКИ

Анализируя язык РНР, как инструмент для создания динамических веб-сайтов, можно выделить ряд преимуществ и изъянов данного языка программирования. Назовем их.



<http://itpotok.ru/obzor-osnovnih-dostoinstv-i-nedostatkov-windows-7/>

## 3.1 ДОСТОИНСТВА

Главным фактором языка РНР является практичность. РНР должен предоставить программисту средства для быстрого и эффективного решения поставленных задач. Практический характер РНР обусловлен пятью важными характеристиками: традиционностью; простотой; эффективностью; безопасностью; гибкостью.

Существует еще одна «характеристика», которая делает РНР особенно привлекательным: он распространяется бесплатно! Причем, с открытыми исходными кодами.

- Традиционность. Язык РНР будет казаться знакомым программистам, работающим в разных областях.
- Простота. Сценарий РНР может состоять из 10 000 строк или из одной строки — все зависит от специфики вашей задачи
- Эффективность. Очень важное преимущество РНР заключается в его «Движке». «Движок» РНР не является ни компилятором, ни интерпретатором. Он является транслирующим интерпретатором.
- Безопасность. РНР предоставляет в распоряжение разработчиков и администраторов гибкие и эффективные средства безопасности, которые условно делятся на две категории: средства системного уровня и средства уровня приложения.

## 3.2. НЕДОСТКИ

Конечно, язык РНР очень прост в своем использовании, но при этом он имеет и ряд существенных недостатков. С выходом последней версии, многие проблемы были решены, но не все, некоторые и по-прежнему доставляют много неудобств:

- Низкая оптимизация рекурсии. Рекурсия – это механизм, при котором функция обращается и вызывает саму себя. Хорошим примером рекурсии является сортировка. Но, к сожалению, рекурсия в РНР очень «хромает».
- Большинство модулей РНР не обеспечивают безопасность потоков. Создатели РНР заявили, что само ядро поддерживает и обеспечивает безопасность потоков, а вот дополнительные модули – нет.
- Отсутствие обратной совместимости между версиями языка. Код, созданный для более ранних версий языка, зачастую не работает или работает некорректно с более поздними версиями языка.

Надо отметить, что противоречие между обратной совместимостью и процессом развития — одна из ключевых проблем в разработке программного и аппаратного обеспечения.

## 4. ОБЛАСТИ ПРИМЕНЕНИЯ

Главная область применения РНР - написание скриптов, работающих на стороне сервера; таким образом, РНР способен выполнять все то, что выполняет любая другая программа CGI, например, обрабатывать данные форм, генерировать динамические страницы или отсылать и принимать cookies. Но РНР способен выполнять намного больше.

Существуют три основных области применения РНР:

- Создание скриптов для выполнения на стороне сервера. РНР традиционно и наиболее широко используется именно таким образом.
- Создание скриптов для выполнения в командной строке. Вы можете создать РНР-скрипт, способный запускаться без сервера или браузера.
- Создание оконных приложений, выполняющихся на стороне клиента. Возможно, РНР является не самым лучшим языком для создания подобных приложений, но, если вы очень хорошо знаете РНР и хотели бы использовать некоторые его возможности в своих клиентских приложениях, вы можете использовать РНР-GTK для создания таких приложений.

Есть еще много других интересных расширений, которые можно просмотреть как в алфавитном порядке, так и по категориям.

## 5. ДИАЛЕКТЫ

В силу популярности языка PHP и желания увеличить быстродействие основанных на нём веб-приложений, создано несколько альтернативных компиляторов, близких к PHP языку. Так в феврале 2010 года компания Facebook открыла свой компилятор PHP — HipHop (HPHP, Hyper-PHP) генерирующий код на C++, с последующей компиляцией в машинный код с помощью gcc. В июле 2013 года российская компания ВКонтакте анонсировала похожий продукт — KPHP.

В таблице (табл. 1)[4] представлен список существующих на сегодняшний момент альтернативных реализаций.

Название	Версия (примерно)	PHP	Лицензия	Результат компиляции
<u>HipHop</u>	5.4		<u>PHP License</u>	<u>C++, машинный код</u>
<u>Roadsend PHP</u>	5.3		<u>GPL/LGPL</u>	<u>C, машинный код</u>
<u>Phalanger</u>	5.5		Ms SS-PL ( <u>Shared source</u> )	<u>Microsoft IL</u>
<u>Quercus</u>	5.2		<u>GPL</u> или коммерческая	<u>JVM</u>
<u>PHC</u>	5.2		<u>BSDL</u>	<u>C, машинный код</u>
<u>Pipp</u>	5.3		<u>Artistic License</u> и <u>GNU GPL</u>	<u>Parrot</u>
<u>JPHP</u>	5.5 (частично 5.6)		<u>Apache 2.0</u>	<u>JVM</u>
<u>PHPPHP</u>	5.4		<u>MIT</u>	PHP
<u>PH7</u>	5.5		<u>Своя (Symisc Public License)</u>	<u>C, машинный код</u>
<u>HippyVM</u>	5.6		<u>MIT</u>	<u>RPython/PyPy</u>

## **ЗАКЛЮЧЕНИЕ**

Таким образом, мы можем утверждать, что язык PHP был рассмотрен, как один из лидеров среди языков применяемых для создания динамических веб сайтов. Тем самым, были достигнуты задачи по изучению истории данного языка, изучены особенности его реализации, выявлены достоинства и недостатки. Рассмотрены диалекты данного языка, и области его применения.

**СПАСИБО ЗА ВНИМАНИЕ!!**

