## 2

# Подготовка рабочей станции к разработке в Joomla!

Разработка программ для Joomla! обычно выполняется на локальной рабочей станции. С этой целью следует установить программное обеспечение веб-сервера, необходимое для выполнения Joomla на локальной машине. Кроме того, нужно организовать редактирование исходных файлов PHP и прочих файлов, составляющих приложение Joomla.

Правда, все это программное обеспечение свободно доступно для загрузки и может быть установлено во всех распространенных операционных системах, включая Windows, Mac OS X и Linux. Кроме того, имеется немало других свободно доступных средств для правки программных файлов Joomla, как будет показано далее в этой главе, где поясняется также порядок настройки рабочей станции для выполнения Joomla, создания и правки файлов, необходимых для написания прикладных программ.

Если вы уже настроили свою рабочую станцию под Joomla и разработку средствами Joomla, можете пропустить раздел "Требования к выполнению Joomla". А если вы уже пользуетесь такой интегрированной средой разработки, как Eclipse или NetBeans, то можете не читать и раздел "Рабочие инструменты".

#### Требования к запуску Joomla

Как пояснялось в главе 1, для нормальной работы Joomla требуется веб-сервер (Apache или Microsoft Internet Information Services — IIS), язык программирования PHP и база данных (чаще всего MySQL). Каждый из этих программных компонентов может быть загружен и установлен отдельно от соответствующих веб-сайтов. Но, как правило, Apache, PHP и MySQL проще установить общим пакетом. А в Mac OS X и Linux эти программные компоненты могут быть уже установлены по умолчанию.

Следует иметь в виду, что независимо от способа установки (по отдельности или общим пакетом) конечный результат остается неизменным: Apache, PHP и MySQL устанавливаются на локальной машине. Единственное отличие состоит в том, что каждый пакет может немного отличаться своей комплектацией, именами папок и отдельными параметрами настройки по умолчанию.

#### Важное примечание по поводу безопасности

В этой главе предполагается, что для разработки и тестирования создается локальная среда, а тестируемые веб-сайты не связаны со Всемирной паутиной (или просто веб). Следовательно, вопросы безопасности веб-сервера не столь актуальны, как в том случае, когда вебприложение размещается на реальном сервере. Это, разумеется, не означает, что не нужно уделять внимание защите самих программ, разрабатываемых для Joomla, от внешних атак.

Если вы работаете на удаленном сайте, доступном в веб, непременно организуйте его защиту — даже на время разработки. Это можно сделать самыми разными способами, в том числе используя файл с расширением .htaccess в Apache, требующий ввода пароля для доступа по указанному URL. В таком случае вам, возможно, потребуется консультация специалиста по безопасности веб вообще и Joomla в частности.

#### Папка DocumentRoot в Apache

Обо всех упомянутых выше программных компонентах и средах нужно знать следующее. Как пояснялось ранее, веб-сервер Арасhe представляет собой программное обеспечение связи браузера с веб-сайтом. Если веб-сервер Арасhe устанавливается локально, то URL по веб-адресу http://localhost сопоставляется с папкой на локальной рабочей станции. Как только этот URL будет введен в браузере, Арасhe произведет поиск файла index.php или index.html в папке DocumentRoot, находящейся на локальном компьютере. Если установлен пакет веб-сервера XAMPP, то папка DocumentRoot обычно находится по пути C:\xampp\htdocs, а если установлен пакет WampServer — то по пути C:\ Program Files\wamp\www.

Папку DocumentRoot можно заменить на другую, внеся соответствующие коррективы в файл httpd.conf, находящийся в папке apache/conf. Например, в пакете XAMPP этот файл находится по пути с:\xampp\apache\conf. А в пакете WampServer файл httpd.conf можно отредактировать в самом приложении, выбрав команду Config Files (Конфигурационные файлы)⇔httpd.conf.

Определившись с папкой DocumentRoot, вы можете затем выбрать место для установки файлов Joomla. Поиск подходящего места начнется с любого URL, введенного в браузере с префиксом http://localhost, т.е. с папки DocumentRoot, и следуя далее вниз по древовидной структуре папок. Так, если корневая папка для документов находится по пути c:\xampp\htdocs, то Joomla можно установить в папке c:\xampp\htdocs\ joomla\_development\joomla\_25. В этом случае URL, который нужно ввести для доступа к административной части данной установки, будет следующим: http://localhost/ joomla\_development/joomla\_25/administrator. Веб-сервер Apache будет искать файл index.php в папке joomla\_development/joomla\_25/administrator, вложенной в папку DocumentRoot. Конечно, Joomla можно было бы установить и в самой папке DocumentRoot, но тогда в ней нельзя было бы установить и все остальное. А следуя приведенному выше примеру установки, можно одновременно начать несколько проектов в отдельных подпапках, вложенных в папку DocumentRoot. Итак, с учетом всего сказанного выше, перейдем к рассмотрению различных вариантов установки требующегося программного обеспечения на разных платформах.

#### Получение своевременных инструкций

Программы и комплекты, рассматриваемые в этом разделе, постоянно обновляются новыми версиями. Со временем выходят также новые версии Windows, Linux и Mac OS X. Поэтому материал этого раздела актуален лишь на момент написания книги. А впоследствии он может устареть или оказаться неполными в силу неизбежных изменений в программном обеспечении как самого веб-сервера, так и платформы, для которой он предназначен.

По этой причине для каждого комплекта здесь приводятся ссылки, по которым нетрудно получить своевременные инструкции. Помимо этих ссылок, здесь приводится также ссылка на статью, опубликованную на веб-сайте документации по Joomla, с подробными инструкциями по установке (http://docs.joomla.org/Setting\_up\_your\_ workstation\_for\_Joomla!\_development). Эта статья обновляется по мере изменений в соответствующем программном обеспечении.

#### Платформа Windows

В различных версиях Windows (Windows 7, Windows Vista и Windows XP) программные компоненты Apache, PHP и MySQL не устанавливаются по умолчанию. Двумя наиболее распространенными стандартными пакетами для установки динамической веб-среды в Windows являются XAMPP и WAMPP.

Наиболее pacпространенным в Windows считается пакет XAMPP (http://www.apachefriends.org/en/xampp.html). Он легко устанавливается и хорошо документирован. Кроме того, он доступен в Linux и Mac OS X. Подробные инструкции по его установке в Windows приведены по адресу http://www.apachefriends.org/en/xamppwindows.html.

Еще одним стандартным для Windows является пакет WampServer. К его преимуществам относится простота перехода к другим версиям Apache, PHP и MySQL. И это весьма значительное преимущество, если, например, требуется протестировать программу в разных версиях PHP. Пакет WampServer доступен по адресу http://www.wampserver.com/ en или http://www.wampserver.com/ru. Он очень просто устанавливается в Windows и хорошо документирован на сайте по указанному выше адресу.

#### Примечание по поводу Microsoft IIS и SQL Server

Joomla может выполняться как на веб-сервере Apache, так и на веб-сервере Microsoft IIS. Но поскольку веб-сервер Apache имеет более долгую историю поддержки, а большинство веб-сайтов Joomla работают именно на нем, то разработчики в основном отдают предпочтение именно ему. По этим причинам в качестве сервера для разработки вебсайтов рекомендуется Apache. С версии 2.5 в Joomla поддерживается также база данных Microsoft SQL Server. Но в качестве основной базы данных для разработки веб-приложений в Joomla рекомендуется все же MySQL.

#### Платформа Mac OS X

Как упоминалось выше, версия пакета XAMPP имеется и для платформы Mac OS X. Ее можно загрузить по адресу http://www.apachefriends.org/en/xampp-macosx. html. В этом случае папка DocumentRoot по умолчанию находится по пути /Applications/XAMPP/htdocs/.

Имеется также пакет MAMP, предназначенный исключительно для платформы Mac OS X. Его можно загрузить по адресу http://www.mamp.info/en/mamp/index.html. В этом случае папка DocumentRoot по умолчанию находится по пути /Applications/ MAMP /htdocs/.

В Mac OS X и Linux имеется также возможность воспользоваться версиями Apache, MySQL и PHP, распространяемыми как составная часть собственной операционной системы. А поскольку Mac OS X является производной от Linux, то эти программные компоненты могут работать и в Mac OS X как ее собственные.

Если у вас нет опыта работы с этими программными компонентами, вам, вероятно, будет проще выбрать один из существующих стандартных пакетов, чем пользоваться собственными их версиями. Кроме того, стандартные пакеты ХАМРР и МАМР, как правило, содержат более актуальные версии Apache, MySQL и PHP, чем версии, входящие в комплект с операционной системой Mac OS X.

#### Важное примечание

Веб-сервер Арасhe может быть установлен по умолчанию в операционной системе Mac OS X. В этом случае он определяет порт на компьютере как "прослушивающий". Если же затем установить вторую версию Apache из пакета XAMPP или MAMP, то и он попытается воспользоваться тем же самым портом, а следовательно, не сможет работать корректно.

Во избежание подобного усложнения придется изменить один из портов. Проще всего, вероятно, изменить порт в пакете ХАМРР или МАМР на что-нибудь вроде 8080. В пакете ХАМРР для этого достаточно отредактировать файл \Applications\XAMPP\xamppfiles\etc\httpd. conf, изменив в нем строку Listen 80 на Listen 8080. А в пакете МАМР порт можно изменить на панели глобальных параметров настройки, что, в свою очередь, внесет коррективы в файл httpd.conf. И в этом случае прослушивающий порт следует изменить на Listen 8080.

#### Платформа Linux

Все, что было сказано выше относительно платформы Mac OS X, справедливо в основном и для платформы Linux. И в этом случае имеется возможность выбрать пакет XAMPP или собственные версии Apache, MySQL и PHP. Начинающим проще установить пакет XAMPP. Сведения о загрузке пакета XAMPP для платформы Linux находятся по адресу http://www.apachefriends.org/en/xampp-linux.html. По умолчанию папка DocumentRoot находится по пути /opt/lampp/htdocs/. Как пояснялось выше для платформы Mac OS X, вам, возможно, придется изменить используемый по умолчанию прослушивающий порт, если в вашей системе уже работает экземпляр веб-сервера Apache. В пакете XAMPP для этой цели придется внести соответствующие коррективы в файл /opt/lampp/etc/httpd.conf. А состоят эти коррективы в замене строки Listen 80 строкой Listen 8080.

#### Владелец файлов и папок по умолчанию

Права доступа к файлам и папкам в Linux и Mac OS X по умолчанию могут привести к сложностям при попытке установить и отредактировать файлы Joomla. Поэтому следует убедиться в том, что соответствующему пользователю Linux и Mac OS X разрешено выполнять запись в ту папку, где устанавливается Joomla и размещаются файлы Joomla.

Владелец всех файлов, созданных веб-сервером Apache, устанавливается с помощью параметров User и Group в файле httpd.conf. По умолчанию оба эти параметра настроены на "демона" (службу, работающую в фоновом режиме). А это означает, что у пользователя нет полномочий на запись в файлы и папки Joomla. Если речь идет о локальном сайте, предназначенном в основном для целей разработки и тестирования, то параметры User и Group можно настроить на зарегистрированного в системе пользователя. Благодаря этому будут получены полномочия на создание и удаление файлов и папок Joomla.

#### Рабочие инструменты

Разработчики Joomla могут пользоваться самыми разными инструментами для создания и редактирования программных файлов Joomla. Так, если задать в дискуссионной группе разработчиков программного обеспечения следующий вопрос: какие инструментальные средства они предпочитают для написания программ, то ответов на него, скорее всего, будет столько же, сколько и мнений по поводу достоинств и недостатков этих самых инструментальных средств. Аналогичное разнообразие мнений сложилось и относительно инструментальных средств, применяемых разработчиками в Joomla. Не существует единого, самого лучшего инструмента для всех и на все случаи жизни. Поэтому в последующих разделах мы постараемся как можно более объективно проанализировать достоинства и недостатки различных инструментальных средства для разработки в Joomla.

#### Интегрированная среда разработки в сравнении с текстовым редактором

При установке Joomla весь исходный код программы загружается в систему в виде простых текстовых файлов. К числу основных типов файлов в базовой установке относятся PHP (с расширением .php), XML (с расширением .xml), JavaScript (с расширением .js), CSS (с расширением .css), а также SQL (с расширением .sql). Все подобные типы определяют простые текстовые файлы, а это означает, что они могут быть отредактированы в любом текстовом редакторе. Следовательно, у вас имеется возможность поправить уже имеющиеся файлы Joomla и создать новые без специального программного обеспечения. В таком случае зачем обременять себя установкой и освоением нового программного обеспечения только для целей разработки в Joomla? Ответ на этот вопрос очевиден: производительность труда — по крайней мере, для многих разработчиков.

Назначение интегрированной среды разработки (IDE) в том и состоит, чтобы повысить производительность труда разработчиков. Такие интегрированные среды разработки, как, например, Eclipse или NetBeans, подробнее рассматриваемые далее в этой главе, предоставляют следующие функциональные возможности для упрощения процесса программирования на PHP.

- Обнаружение синтаксических ошибок в коде по мере его набора.
- Автоматическое завершение набранных частей кода на основании известного синтаксиса.
- Автоматическое отображение сведений о методах и полях редактируемого класса.
- Сравнение двух версий исходного файла (или двух разных исходных файлов) для выявления в них конкретных отличий.
- Упрощение поиска и замены во всей кодовой базе.
- Отладка программ на РНР.

Для выполнения всех перечисленных выше операций совсем не обязательно иметь IDE. Но все эти операции можно выполнять в IDE, как в едином инструментальном средстве.

Еще одно преимущество IDE состоит в том, что она сводит различные задачи разработки в единый проект. В частности, многие разработчики Joomla пользуются системой контроля версий типа Git или Subversion. К этим системам подключаются модули, работающие с IDE, что дает возможность пользоваться системой контроля версий непосредственно в IDE. Другим примером служит программа автоматического построения проектов типа Phing или Ant. И эти программы можно выполнять непосредственно в IDE, что намного упрощает труд многим разработчикам, особенно в крупных проектах.

Но некоторым применение IDE дается дорогой ценой, поскольку освоение самой среды требует дополнительного времени и труда. Кроме того, программные ошибки могут встречаться в IDE чаще, чем в текстовом редакторе, в силу ее большей сложности. И поэтому одни считают, что возникающая в связи с этим неудовлетворенность не стоит потенциального выигрыша в производительности труда. А другие легко мирятся с перспективой освоения нового программного обеспечения и возможными его недоработками, поскольку считают эти недостатки малой ценой, которую приходится платить за несомненные преимущества IDE.

Если вы собираетесь много заниматься разработкой веб-сайтов в Joomla, вам, вероятно, стоит уделить время хотя бы опробованию IDE, чтобы выяснить, насколько эта среда вам подходит. А если вы не собираетесь тратить время и силы на освоение IDE, но желаете сразу же приступить к написанию кода, то IDE вряд ли вам подойдет.

#### Варианты IDE с открытым кодом

Двумя самыми распространенными IDE для Joomla являются Eclipse и NetBeans. Обе IDE отлично справляются со своей задачей, помогая разработчику работать более продуктивно. В последующих подразделах они рассматриваются по очереди.

#### Eclipse

Несмотря на то что Eclipse давно уже стала самой распространенной IDE для разработки приложений на Java, в программировании на PHP она стала применяться относительно недавно. В 2007 году специально для разработчиков на PHP были выпущены инструментальные средства, которые так и называются: PHP Development Tools (PDT). Они представляют собой набор подключаемых модулей, адаптирующих Eclipse для работы с кодом PHP.

Ha момент написания этой книги самой последней считалась версия PDT 3.0. В рамках проекта Eclipse новая версия выпускается каждый год в июне, а вместе с ней зачастую и новая версия PDT. Пакет Eclipse можно свободно загрузить по адресу http://eclipse.org/downloads. Для разработки в Joomla обычно выбирается вариант Eclipse for PHP Developers (Eclipse для разработчиков программ на PHP) этого пакета. Подробные инструкции по установке и настройке Eclipse для разработки в Joomla можно найти на сайте документации по Joomla, перейдя по адресу http://docs.joomla.org/Setting\_up your workstation for Joomla! development.

IDE Eclipse написана на языке Java и работает на платформах Windows, Linux и Mac OS X. Это свободно доступное программное обеспечение с открытым кодом. И хотя не существует официальной статистики по поводу того, каким именно IDE отдают предпочтение разработчики Joomla, тем не менее Eclipse считается едва ли не самой распространенной IDE среди опытных разработчиков данной категории, и не в последнюю очередь потому, что она оказалась первой свободно доступной IDE для разработки на PHP.

Рассмотрим вкратце, чем же Eclipse может помочь в разработке веб-сайтов средствами Joomla. Прежде всего следует обратить внимание на ключевое понятие проекта для таких IDE, как Eclipse или NetBeans. В большинстве текстовых редакторов исходный файл правится постепенно, и самому редактору вообще ничего не известно о других файлах, свя-

занных с исходным, да это ему и не нужно. А в IDE сначала создается проект, в котором хранится группа связанных вместе файлов. Так, если вы работаете над проектом Joomla, то все приложение Joomla содержится в одном проекте. Поначалу такой подход может показаться не совсем понятным, но он вполне оправдан с точки зрения разработки, поскольку разработчикам все время приходится работать с целыми группами связанных вместе файлов.

Создавая проект РНР в Eclipse, вы получаете представление под названием PHP Explorer (Проводник по PHP), где можете просматривать весь проект и открывать любой файл для редактирования. В представлении PHP Explorer, приведенном в качестве примера на рис. 2.1, один файл открыт для редактирования в левом верхнем углу.

А теперь рассмотрим в качестве примера некоторые из упоминавшихся ранее функциональных возможностей IDE. Начнем с выделения синтаксиса. На рис. 2.2 показана программа на PHP, открытая для редактирования в Eclipse. Различные



**Рис. 2.1.** Пример представления PHP Explorer в Eclipse

синтаксические конструкции выделены разным цветом для удобства чтения ее кода. Посмотрим далее, что произойдет, если сделать ошибку в коде. Как показано на рис. 2.3, после оператора if оставлены скобки.

Как видите, слева от строки с синтаксической ошибкой появляется знак **X**, а в самой строке обнаруженные ошибки подчеркнуты волнистой линией (на экране эти признаки ошибок отображаются красным цветом). Если навести курсор на знак **X**, то появится всплывающая текстовая подсказка, как показано на рис. 2.4. Эта подсказка вкратце поясняет характер ошибки и указывает возможный способ ее устранения.

Еще одним важным свойством IDE является автоматическое завершение кода. Оно представлено в разных формах. В простейшей форме символы предоставляются попарно. Например, при наборе исходного кода PHP вместе с левой круглой скобкой практически всегда требуется и правая скобка, поэтому редактор автоматически предоставляет обе скобки. Это же относится и к фигурным скобкам, кавычкам и элементам разметки в коде HTML.

```
/**
  * Class constructor.
  * @param array $config A configuration arra
                             session name, client
  * @since 11.1
public function construct($config = array())
     jimport('joomla.utilities.utility');
     jimport('joomla.error.profiler');
     // Set the view name.
     $this-> name = $this->getName();
     // Only set the clientId if available.
     if (isset($config['clientId'])) {
         $this->_clientId = $config['clientId'];
     3
     // Enable sessions by default.
     if (!isset($config['session'])) {
         $config['session'] = true;
     3
```

Рис. 2.2. Выделение синтаксиса в редакторе исходного кода РНР, встроенном в Eclipse



Рис. 2.3. Отображение синтаксической ошибки в редакторе исходного кода РНР, встроенном в Eclipse



Рис. 2.4. Всплывающая текстовая подсказка, раскрывающая характер ошибки

Еще одна форма завершения кода имеет место в том случае, если требуется набрать имя метода. В качестве примера на рис. 2.5 набрано выражение x = this-> и затем нажата комбинация клавиш <Ctrl+Пробел>.

x = this->	
// Only set	♦ \$_clientId: integer
if (isset(\$c	
\$this->_	♦ \$_messageQueue: array
}	\$_name: array
	<ul> <li>\$input: JInput</li> </ul>
// Enable se	<ul> <li>\$requestTime: date</li> </ul>
if (!isset(\$	<ul> <li>\$scope: string</li> </ul>
\$config[	<ul> <li>\$startTime: integer</li> </ul>
}	♦ _createConfiguration(\$file) - JApplication
// Create th	♦ _createSession(\$name) - JApplication
if (class ex	checkSession() - JApplication
\$this->i	close(\$code) - JApplication
}	Press 'Ctrl+Space' to show Template Proposals



Редактору известно, что \$this — это объект типа Japplication, и поэтому он отображает все поля и методы, доступные в данном классе. Для того чтобы выбрать подходящий метод или поле, достаточно щелкнуть кнопкой мыши на его имени или воспользоваться клавишами с изображением стрелок. Как только поле или метод будут выбраны, их имя автоматически вставится в исходный текст программы. Благодаря этому ускоряется набор исходного текста программы и уменьшается вероятность совершения опечаток. По мере набора исходного текста окно поиска автоматически обновляется, чтобы отображать только те имена, которые совпадают с набранным текстом.

Кроме того, в этом окне предоставляются дополнительные сведения о каждом методе или поле. В качестве примера на рис. 2.6, *слева*, выделен метод redirect(), а на рис. 2.6, *справа*, представлена форма объявления этого метода со всеми его параметрами и краткая документация на него. Это очень удобно для поиска нужного метода и делается быстрее, чем в отдельном файле или окне.



Рис. 2.6. Список параметров и документация на выбранный метод, предоставляемые в редакторе исходного кода PHP, встроенном в Eclipse

Когда правится исходный файл, в Eclipse предоставляется список полей и методов для данного файла в отдельном представлении. Так, на рис. 2.7, *слева*, показана область правки, а на рис. 2.7, *справа*, — соответствующее схематическое представление (Outline).



Рис. 2.7. Редактор исходного кода РНР и схематическое представление класса в Eclipse

В схематическом представлении приводятся поля и методы редактируемого класса. А в схематическом представлении проекта PHP (PHP Project Outline) можно посмотреть иерархическую структуру всего проекта, как показано на рис. 2.8. В этом представлении можно найти любой класс, доступный в Joomla, и посмотреть всего его поля и методы.



Рис. 2.8. Схематическое представление проекта PHP в Eclipse

Еще одним удобным свойством такой IDE, как Eclipse, является возможность быстро сравнивать две версии исходного файла, чтобы выявить конкретные изменения. Так, на рис. 2.9 представлено сравнение текущей версии файла с предыдущей после внесения самых последних правок.



Рис. 2.9. Пример сравнения двух версий исходного файла в Eclipse

Этим свойством можно воспользоваться для сравнения различных версий одного и того же исходного файла. Их можно извлечь из местной предыстории, автоматически сохраняемой в Eclipse, или же из хранилища версии, если применяется система контроля версий.

При написании кода нередко требуется выполнить поиск в целом ряде программ или в проекте в целом, чтобы выяснить, где именно используется конкретная переменная или отдельный фрагмент кода. Для этой цели в Eclipse предоставляется очень простое и эффективное средство. В качестве примера на рис. 2.10 приведен результат поиска в проекте вхождений выражения JFactory::getSession().



Рис. 2.10. Пример поиска выражения в проекте средствами Eclipse

Ради удобства результаты поиска привязываются к редактору исходного текста. Так, если дважды щелкнуть на одной из строк кода в результатах поиска, то в редакторе исходного текста откроется целевой файл, а курсор установится на номере той строки, где был найден искомый текст. В этом представлении можно выполнять не только поиск, но и замену найденного текста.

Для многих разработчиков очень важной является также возможность отладки программы в процессе ее выполнения. Следует, однако, иметь в виду, что для начинающих программировать отладка может оказаться непростым делом, и к тому же она не является совершенно необходимой при разработке программ для Joomla. Поэтому можете пропустить остальную часть этого подраздела и перейти к чтению следующего материала.

С другой стороны, если вы знакомы в общих чертах с процессом отладки, то сможете относительно легко и быстро освоить доступный в Eclipse отладчик кода PHP. В Eclipse применяется отладчик под названием *Xdebug*, позволяющий приостанавливать выполнение программы на PHP и выполнять ее далее в пошаговом режиме отладки. В качестве примера на рис. 2.11 показана приостановленная для отладки программа, разрабатываемая для Joomla.

Image: Synt Report.       Image: Synt Report.<			
P Debug 20:       Mis Server:)       Mis Dir Dir Dir Web Dagel         If prof end lock_1700 (PHP Web Dagel       Dir	に、回る目前	eposit 📴 Git Reposi	tor 隊 Debug 🐻 PHF
For each book 120 (PPP Web Bage)	🎽 Debug 🕄 🖓 Servers) 🦉 🕪 🗉 🔳 세 🔍 👁 L은 🗟 🕱 🍸 드리 🍳	Breakpoi 😚 Expressi	(M= Variable 🛞 👘
PP Application            Name            Value              PP Application            Name            Value              PP Application            Name            Value              Book_ITM/components/concontent/view.php.View-spt: lineno 38(b content-view.php.View-spt: lineno 38(b content-view.php.View.php.View.php.View.php.View.php.View.php.Vi	Front end book_170 [PHP Web Page]		包括日
	PHP Application	lame	Value
<pre>boblaw/comparent/comest/webs/webs/webs/webs/webs/webs/webs/webs</pre>	10 <sup>10</sup> Remote Launch (suspended)	& \$ COOKIE	Array [23]
<pre>bool10//binare/joemid/weighted/or/component/weight/bev_sqt:lines/380 bool bool10//binare/joemid/weighted/weight/be/sqt:lines/apply:imid bool10//binare/joemid/weight/be/sqt:lines/apply:imid bool10//binare/joemid/weight/be/sqt:lines/apply:imid bool10//binare/joemid/weight/be/sqt:lines/apply:imid bool20//binare/joemid/weight/be/sqt:lines/apply:imid bool20//binare/joemid/weight/be/str:lines/apply:imid bool20//binare/joemid/weight/be/str:lines/apply:imid/bool20//binare/joemid/weight/be/str:lines/apply:imid/be/str:lines/apply:imid/bool20//binare/point/be/str:lines/apply:imid/be/str:lines/apply:imid/be/str:lines/apply:imid/be/str:lines/apply:imid/be/str:lines/apply:imid/be/str:lines/apply:imid/be/str:lines/apply:imid/be/str:lines/apply:imid/be/str:lines/apply:imid/be/str:lines/apply:imid/be/str:lines/apply:imid/be/str:lines/apply:imid/be/str:lines/apply:imid/be/str:lines/apply:imid/be/str:lines/apply:imid/be/str:lines/apply:imid/be/str:lines/apply:imid/be/str:lines/apply:i</pre>	book_170/components/com_content/models/article.php.ContentModelArticle->getItem :	4 S FNV	Array [0]
bool 10/components/concents/vec/addc/web/a	book_170/libraries/joomla/application/component/view.php.JView->get : lineno 348() boc		Array [0]
<pre>boblyTN/Inters/cond/application/comporent/controller/phJController-daphy:init boblyTN/inters/cond/application/comporent/dotPhJController-daphy:init boblyTN/inters/cond/application/comporent/dotPhJController-daphy:init boblyTN/inters/cond/application/comporent/dotPhJController-daphy:init boblyTN/inters/cond/application/comporent/dotPhJController-daphy:init boblyTN/inters/cond/application/comporent/dotPhJController-daphy:init boblyTN/inters/cond/application/comporent/dotPhJController-daphy:init boblyTN/inters/cond/application/tomporent/dotPhJController-daphy:init boblyTN/inters/controller/dotPJController-daphy.comcontent/dotPJLController-dap</pre>	book_170/components/com_content/views/article/view.html.php.ContentViewArticle->di	A S GET	Array [4]
<pre>bocl10/component/controller_ph_contentController_sdeply_ieners) bocl10/component/contcoller_ph_contentController_sdeply_ieners bocl10/contents/security_icontentController_sdeply_ieners bocl10/contents/security_icontentSolutions/secu</pre>	book_1/0/libraries/joomIa/application/component/controller.php.JController->display: lii	& S POST	Array [0]
<pre>bob_170/hsmers/contal/point/component/controller.phg/Controller.secute:10 bob_170/hsmers/contal/point/controller.phg/Controller.secute:10 bob_170/hsmers/contal/point/controller.phg/Component/helper:ndgr bob_170/hsmers/contal/point/controller.phg/Component/helper:ndgr bob_170/hsmers/contal/point/controller.phg/Component/helper:ndgr bob_170/hsmers/contal/point/controller.phg/Component/helper:ndgr bob_170/hsmers/contal/point/controller.phg/Component/helper:ndgr bob_170/hsmers/contal/point/controller.phg/Component/helper:ndgr bob_170/hsmers/contal/point/controller.phg/Component/helper:ndgr bob_170/hsmers/contal/point/controller.phg/Component/helper:ndgr bob_170/hsmers/contal/point/controller.phg/Component/helper:ndgr bob_170/hsmers/contal/point/controller.phg/Component/helper:ndgr bob_170/hsmers/contal/point/controller.phg/Component/helper:ndgr bob_170/hsmers/contal/point/controller.phg/Component/helper:ndgr bob_170/hsmers/contal/point/controller.phg/Component/helper:ndgr bob_170/hsmers/contal/point/controller.phg/Component/helper:ndgr bob_170/hsmers/contal/point/controller.phg/Cont</pre>	book_1/0/components/com_content/controller.php.ContentController-> display : lineno 7	A S REQUEST	Array [27]
book170/components/concontext/perspin_ence: imen 30 book170/com     book170/concontext/perspin_concent/begraph_component+degraph_com	book_170/libraries/joomla/application/component/controller.php.JController-> execute : li	A S SERVER	Array [43]
<pre>bock_TDM/harrie/gomb/application/component/helper.php/Component/helper.rhdgf bock_TDM/ndws.php(main)_simen 430 bock_TDM/ndws.php./Component/helper.rhdgf bock_TDM/ndws.php(main)_simen 430 bock_TDM/ndws.php(main)_simen 430 phplication.php</pre>	book_170/components/com_content/content.php.require_once : lineno 19() book_170/con		Array (1)
<pre>book_170/ndude/application.php.Site-&gt;dispatch: lineno 32() book_170/indude/application.php.Site&gt;&gt;dispatch: lineno 32() book_170/indude/application.php.Site&gt;&gt;dispatch: lineno 32() book_170/indude/application.php.Site&gt;&gt;dispatch: lineno 430 book_170/indude/application.php</pre>	book_170/libraries/joomla/application/component/helper.php.JComponentHelper:render		Array [11]
<pre>E book_ID0/ndec.php(main): lineno 430 book_ID0/indec.php at line 43  # pelication.php @ whickephp (Nain): lineno 430 book_ID0/indec.php at line 43 # optication.php @ whickephp (Nain): lineno 430 book_ID0/indec.php at line 43 # optication.php @ whickephp (Nain): lineno 430 book_ID0/indec.php at line 43 # optication.php @ whickephp (Nain): line 44 # optication.php @ which at line 44 # optication.php @ which at</pre>	book_1/0/includes/application.php.JSite->dispatch : lineno 191() book_1/0/includes/appli	A Success	<ul> <li>(Ininitialized)</li> </ul>
<pre>mm</pre>	book_170/index.php.{main}: lineno 43() book_170/index.php at line 43	III	
m       *         application.php       @ anticle.php S: (derty*)_join( inf: , *istegories as parent on parent.in * or parent_in *);         113       // Join on voting table         114       & gouery->palent('BOND( v. rating_pum / v.rating_count ) AS rating, v.rating_count as ri         115       & gouery->palent('BOND( v.rating_pum / v.rating_count ) AS rating, v.rating_count as ri         116       & gouery->where('a.id = ' . (int) \$pb);         117       & gouery->where('a.id = ' . (int) \$pb);         118       // Filter by start and end dates.         120       \$moutbase = \$db->Quote(Jdat->getNillDate());         121       \$moutbase = \$db->Quote(Jdatory::getDate()->toNdySQL());         122       \$gouery->where('(a.publish_up = '. \$mullDate .' OR a.publish_down >=' . \$mowtbase .');         123       #gouery->where('(a.publish_up = '. \$mullDate .' OR a.publish_down >=' . \$mowtbase .');         124       \$gouery->where('(a.publish_up = '. \$mullDate .' OR a.publish_down >=' . \$mowtbase .');         124       \$Gouery->where('Barowsr Output ) \$2 Output !> .' \$mullDate .' OR a.publish_down >=' . \$mowtbase .');         125       @consol @ Task !@ Debug Output !> Browsr Output !? Output !> .' \$mullDate .' .' .'];         126       @consol @ Task !@ Debug Output !> Browsr Output !? Output !> .'			
<pre>peptienophp  peptienophp peptienophp perturbation p</pre>	f H		
<pre>iii query-yolant uni , *_categories as parent on parent in - trpatent in , iii // Join on voting table iii \$query-select('ROND('v.rating_oum / v.rating_count ) AS rating, v.rating_count as re \$query-yolant('LET', '\$_content_rating AS v ON a.id - v.content_id'); iii \$query-ywhere('a.id - '. (int) \$p\$); iii \$query-ywhere('a.id - '. (int) \$p\$); iii \$nowhate = \$dab-youte(dba-yetBullBate()); iii \$nowhate = \$dab-youte(dba-yetBullBate()); iii \$query-ywhere('(a.publish_up = '. \$nullBate .' OR a.publish_up &lt;= '. \$nowhate .'); iii \$query-ywhere('(a.publish_up = '. \$nullBate .' OR a.publish_down &gt;= '. \$nowhate .'); iii \$query-ywhere('(a.publish_up = '. \$nullBate .' OR a.publish_down &gt;= '. \$nowhate .'); iii \$query-ywhere('(a.publish_up = '. \$nullBate .' OR a.publish_down &gt;= '. \$nowhate .'); iii \$query-ywhere('.gublish_up = '. \$nullBate .' OR a.publish_down &gt;= '. \$nowhate .'); iii \$query-ywhere('.gublish_up = '. \$nullBate .' OR a.publish_down &gt;= '. \$nowhate .'); iii \$query-ywhere('.gublish_up = '. \$nullBate .' OR a.publish_down &gt;= '. \$nowhate .'); ii \$query-ywhere('.gublish_up = '. \$nullBate .' OR a.publish_down &gt;= '. \$nowhate .'); ii \$query-ywhere('.gublish_up = '. \$nullBate .' OR a.publish_down &gt;= '. \$nowhate .'); ii \$query-ywhere('.gublish_up = '. \$nullBate .' OR a.publish_down &gt;= '. \$nowhate .'); ii \$ue statements</pre>	application.php		c
<pre>112 // Join on voting table 113 // Join on voting table 114 &amp; Equery-select('ROUND('v.rating_oum / v.rating_oumt) AS rating, v.rating_oount as re 5 &amp; Squery-select('ROUND('v.rating_oum / v.rating_oount) AS rating, v.rating_oount as re 115 &amp; Squery-select('AOUND('v.rating_bay) 116 // Files by start and and dates. 120 found the = 5db-Squere('da-yoetNulDate()); 121 &amp; SnowDate = 5db-Squere('da-yoetNulDate()); 122 &amp; Squery-swhere('(a.publish_up = '. SnulDate .'OR a.publish_up &lt;= '. SnowDate .') 123 &amp; Squery-swhere('(a.publish_up = '. SnulDate .'OR a.publish_up &lt;= '. SnowDate .') 124 &amp; Squery-swhere('(a.publish_up &lt;= '. SnulDate .'OR a.publish_up &lt;= '. SnowDate .') 125 &amp; Console &amp; Tasks &amp; Debug Output &amp; Browser Output &amp; E Outine E 1 &amp; Console &amp; Tasks &amp; Debug Output &amp; Browser Output &amp; E Outine E 1 &amp; console &amp; Tasks &amp; Debug Output &amp; Browser Output &amp; E Outine E 1 &amp; console &amp; Tasks &amp; Debug Output &amp; Browser Output &amp; E Outine E 1 &amp; console &amp; Tasks &amp; Debug Output &amp; Browser Output &amp; E Output</pre>	<pre>vquery-&gt;join( meri , *_cacegories as parene on pare</pre>	ne.iu - c.parene	//
<pre>113 // Join on voting table 114 \$query-&gt;select('RONU(V.rsting_sum / v.rating_count) AS rating, v.rating_count as re 115 \$query-&gt;select('RONU(V.rsting_sum / v.rating_count) AS rating, v.rating_count as re 116 \$query-&gt;swhere('a.id = ' . (int) \$pk); 117 \$query-&gt;swhere('a.id = ' . (int) \$pk); 118 // Filter by start and end dates. 120 \$mullbace = \$db-&gt;quote(db-&gt;qetKullbate()); 121 \$nowDate = \$db-&gt;quote(db-&gt;qetKullbate()); 122 \$query-&gt;where('(a.publish_up = '. \$mullbate .' OR a.publish_up &lt;= '. \$nowDate .') 124 \$query-&gt;where('(a.publish_up = '. \$mullbate .' OR a.publish_down &gt;= '. \$nowDate .') 124 \$Query-&gt;where('(a.publish_up = '. \$mullbate .' OR a.publish_down &gt;= '. \$nowDate .') 124 \$Query-&gt;where('(a.publish_up = '. \$mullbate .' OR a.publish_down &gt;= '. \$nowDate .') 124 \$Query-&gt;where('(a.publish_up = '. \$mullbate .' OR a.publish_down &gt;= '. \$nowDate .') 124 \$Query-&gt;where('(a.publish_up = '. \$mullbate .' OR a.publish_down &gt;= '. \$nowDate .') 124 \$Query-&gt;where('(a.publish_up = '. \$mullbate .' OR a.publish_down &gt;= '. \$nowDate .') 124 \$Query-&gt;where('(a.publish_up = '. \$mullbate .') 125 \$Query-&gt;where('.gublish_up = '. \$mullbate .') 126 \$Query-&gt;where('.gublish_up = '. \$mullbate .') 127 \$Query-&gt;where('.gublish_up = '. \$mullbate .') 128 \$Query-&gt;where('.gublish_up = '. \$mullbate .') 129 \$Query-&gt;where('.gublish_up = '. \$mullbate .') 120 \$Query-&gt;where('.gublish_up = '.) 121 \$Point _QUery _QUery _QUE \$QUE _QUER _Q</pre>	112		
114     \$query->select(*ROUND(*.rating_num / v.rating_count ) & Stating, v.rating_count as references       115     \$query->shere(*a.id = '. (int) & \$pk);       116     \$query->shere(*a.id = '. (int) & \$pk);       117     \$query->shere(*a.id = '. (int) & \$pk);       118     \$query->shere(*a.id = '. (int) & \$pk);       119     \$f. Tasks = \$ab->Quote(Tastory::getDate());       120     \$f. Tasks = \$ab->Quote(Tastory::getDate()>redNySQL());       121     \$f. Tasks = \$ab->Quote((Tastory::getDate()>redNySQL());       122     \$query->where('(a.publish, up = '. & \$nullDate . ' OR a.publish_up <= '. & \$nowDate . ')	113 // Join on voting table		
<pre>115 squery-&gt;&gt;cin('LET', 'Fontent_faing AS v ON a.id = v.content_id'); 117 118 119 119 119 120 120 121 120 121 121 121 123 124 125 125 125 125 125 125 126 127 126 127 127 127 128 129 129 129 129 129 129 129 129 129 129</pre>	114 Squery->select('ROUND( v.rating_sum / v.rating_count	) AS rating, v.m	count as re
116     \$query->where('a.id = '. (int) \$pk);       113     // Filers by start and end dates.       114     foullbare = \$db->Quote(db->qetNillbare());       120     fourDate = \$db->Quote('da->qetNillbare());       121     fourDate = \$db->Quote('da-quotNillbare());       122     fourDate = \$db->Quote('da-quotNillbare());       123     fquery->where('(a.publish_down = '. \$nullbate .' OR a.publish_down >= '. \$nowDate .')       124     fquery->where('(a.publish_down = '. \$nullbate .' OR a.publish_down >= '. \$nowDate .')       2     Console @ Tasks @ Debug Output @ Browser Output @ E Outine IS       2     @ Tasks @ Debug Output @ Browser Output @ E Outine IS	<pre>115 \$query-&gt;join('LEFT', '#content_rating AS v ON a.id</pre>	= v.content_id')	;
<pre>iii aquety-shake( all (IN) app); iii // Filter by start and end dates; iii // Snullbate = \$db-&gt;Quote(\$db-&gt;qetNullDate()); iii Snullbate - \$db-&gt;Quote(\$db-&gt;qetNullDate()); iii Snullbate - `\$db-&gt;Quote(\$db-&gt;qetNullDate - ' OR a.publish_up &lt;- '. \$nowDate - ') iii Gquery-&gt;where('(a.publish_up = '. \$nullDate . ' OR a.publish_down &gt;= '. \$nowDate - ') iii Gquery-&gt;where('iii Dubing down = '. \$nullDate . ' OR a.publish_down &gt;= '. \$nowDate . ') iii Gquery-&gt;where('iii Dubing down = '. \$nullDate . ' OR a.publish_down &gt;= '. \$nowDate . ') iii Guery-&gt;where('iii Dubing down = '. \$nullDate . ' OR a.publish_down &gt;= '. \$nowDate . ') iii Guery-&gt;where('iii Dubing down = '. \$nullDate . ' OR a.publish_down &gt;= '. \$nowDate . ') iii Guery-&gt;where('iii Dubing down = '. \$nullDate . ' OR a.publish_down &gt;= '. \$nowDate . ') iii Guery-&gt;where('iii Dubing down = '. \$nullDate . ' OR a.publish_down &gt;= '. \$nowDate . ') iii Guery-&gt;where('iii Dubing down = '. \$nullDate . ' OR a.publish_down &gt;= '. \$nowDate . ') iii Guery-&gt;where('iii Dubing down = '. \$nullDate . ' OR a.publish_down &gt;= '. \$nowDate . ') iii Guery-&gt;where('iii Gauery . `) iii Guery-&gt;where('iii Gauery . `) iii Guery . `)</pre>	110		
119     // Files by start and end dates.       120     foullbare = \$db->Quote(db->qetBullDate());       121     fnowDate = \$db->Quote(dB->qetBullDate());       122     fnowDate = \$db->Quote((Tartory::getDate()->toNtySQL());       123     fquery->where('(a.publish_up = '. \$nullDate .' OR a.publish_up <= '. \$nowDate .')	aquery-swhere(a.id = . (inc) spk);		
119 // Firle by state and makes. 120 Smillbace = 6db-3Quore(6db-3qetShillbate()); 121 SnowDate = 6db-3Quore(6db-3qetShillbate()); 122 123 Squery->Where('(a,publish_up = '. Smillbate . ' OR a.publish_up <= '. SnowDate . ') 124 Guery->Where('(a,publish_down = '. Smillbate . ' OR a.publish_down >= '. SnowDate . ') 125 Guery->Where('(a,publish_down = '. Smillbate . ' OR a.publish_down >= '. SnowDate . ') 126 Guery->Where('(a,publish_down = '. Smillbate . ' OR a.publish_down >= '. SnowDate . ') 127 Guery->Where('(a,publish_up = '. Smillbate . ' OR a.publish_down >= '. SnowDate . ') 128 Guery->Where('(a,publish_up = '. Smillbate . ' OR a.publish_down >= '. SnowDate . ') 129 Guery->Where(') = Browser Output [ St Outine S) 129 Guery->Where(') = Browser Output [ St Outine S) 129 Guery->Where(') = Browser Output [ St Outine S) 129 Guery->Where(') = Browser Output [ St Outine S) 120 Guery->Where(') = Browser Output [ St Output [ St Output ] ] = Guery = St Output [ St Output ] = St Output [ St Output ] = St Output ]	110 // Filmer by start and and dates		
111     fbowbace = 6db - Quote (Wactory: getBate() → todySQL());       122     fbowbace = 6db - Quote (Wactory: getBate() → todySQL());       123     fquery-where('(a.publish_down = '. fbullDate .' 06 a.publish_up <* '. fbowbate .');	119 // Filter by Start and End dates.		
122     Squery->where('(a.publish_up = '. \$nullDate. 'OR a.publish_up <= '. \$nowDate. ')	<pre>120 \$nowDate = \$db_&gt;Quote(#ub-&gt;getNutDate()); 121 \$nowDate = \$db_&gt;Quote(#ab-&gt;getNutDate()_&gt;toMySOT())</pre>		
123     Fquerywhere (' (a. publish_down = ' . foullDate . ' OR a. publish_up < ' . foundbate . ')	the start the start of the star		
124       \$query->where('(a.publish_down = '. \$nullDate . 'OR a.publish_down >= '. \$nowDate .          =         Console @ Tasks @ Debug Output [ >> Browser Output ] \$E Outline E?       >> E P [ P_k [ ] >> *         *       us tatements	122	.publish up <= '	. SnowDate . ')'
↓ Console ② Tasks (忌 Debug Output l ⑤ throwser Output ) 箆 Outline 記 ② □ □ 北京 図 ○ □ 北京 図 ○ □ 小市 日本 図 ○ □ □ 小市 日本 図 ○ □ □	122 123 Squery->where('(a.publish up = ', SpullDate, 'OR a	a nubligh down	>= \$nowDate
I Console (@ Tasks   ⓑ Debug Output   월 Browser Output   注: Outline ☆ 💿 👘 🖻 🍂 🗐 🐃	<pre>122 123 123 \$query-&gt;where('(a.publish_up = ' . \$nullDate . ' OR a 124 \$query-&gt;where('(a.publish down = ' . \$nullDate . ' OF</pre>	C GIDGETON GOWN	
a Console   ⊴ Tasis   46 Debug Output   49 Browser Output   at: Outline ⊠ 🔰 🖓 🖾 *********************************	122     \$query->where('(a.publish_up = '. \$nullDate .' OR a       124     \$query->where('(a.publish_down = '. \$nullDate .' OF       4     m		•
1 se statements	122 123 123 124 \$query->where('(a.publish_up = '. \$nullDate .' OR a \$query->where('(a.publish_down = '. \$nullDate .' OF m m m m m		• • • • •
	122     Squery->where('(a.publish_up = '. SnullDate .' OR       123     Squery->where('(a.publish_down = '. SnullDate .' OF           I Console ( Datas ( Pobug Output ( Datas )		, ≥ 5 \$ 5

Рис. 2.11. Отладчик, доступный в Eclipse

Слева вверху на этом рисунке показана область, называемая стеком. Ее увеличенный вид приведен на рис. 2.12. В стеке показано, каким образом выполнение программы дошло до данной точки. В рассматриваемом здесь примере отладка программы из исходного файла index.php была начата со строки кода 43 (последней в стеке). В этой строке вызывается метод JSite->dispatch, который, в свою очередь, вызывает метод компонента JComponentHelper::renderComponent и так далее до самой вершины стека, где вызывается метод ContentModelArticle->getItem.



Рис. 2.12. Пример стека из отладчика кода PHP в Eclipse

#### Важное примечание

Если вы не знаете, что такое стек, - не отчаивайтесь. Принцип его организации и действия подробно рассматривается в главе 3.

®⊜ Breakpoints ଜିନ୍ଦୁ Expre	essions 🔲= Variables 🛛 🗖
	£ + =
Name	Value
\$_COOKIE	Array [23]
\$_ENV	Array [0]
\$_FILES	Array [0]
\$_GET	Array [4]
\$_POST	Array [0]
\$_REQUEST	Array [27]
\$_SERVER	Array [43]
\$_SESSION	Array [1]
\$GLOBALS	Array [11]
\$access	<uninitialized></uninitialized>
\$archived	<uninitialized></uninitialized>
\$asset	<uninitialized></uninitialized>
\$data	<uninitialized></uninitialized>
♦ \$db	JDatabaseMySQLi
∲ Se	<uninitialized></uninitialized>
\$error	<uninitialized></uninitialized>
\$groups	<uninitialized></uninitialized>

В представлении переменных (Variables), приведенном на рис. 2.13, отображаются значения всех переменных из области действия в данной точке выполнения программы.

В Eclipse имеется также немало других удобных средств, способствующих более эффективной разработке программ для Joomla. В частности, встроенная справочная система Eclipse устанавливается вместе с этой IDE и предоставляет подробную справку о ее различных функциональных возможностях. К слабым сторонам Eclipse в отношении программирования на PHP можно отнести недостаточное количество учебного материала и пользовательской документации, хотя на сайте документации по Joomla опубликован ряд полезных статей, посвященных Eclipse.

#### PHPEclipse

PHPEclipse (http://www.phpeclipse.com) — это еще одна IDE для программирования на PHP с функциональными возможностями, аналогичными программе PDT для Eclipse. Она поддерживается вне рамок официального проекта семейства Eclipse. На момент написания этой книги активная разработка PHPEclipse, по-видимому, была приостановлена, хотя эта IDE по-прежнему доступна для загрузки.

#### **NetBeans**

Другой IDE с открытым кодом, считающейся основной для разработки программ на PHP, является NetBeans. Она свободно доступна для загрузки по адресу http:// netbeans.org/downloads/index.html. Аналогично Eclipse, IDE NetBeans была первоначально разработана для программирования на Java и теперь поддерживает разработку приложений на других языках программирования.

К главным преимуществам NetBeans относится отличная документация для разработки программ на PHP, включая и учебный видеоматериал. Именно по этой причине NetBeans является отличным выбором для тех, кто только начинает осваивать IDE.

Рис. 2.13. Представление переменных из отладчика кода РНР в Eclipse

NetBeans обладает такими же функциональными возможностями, как и Eclipse, включая выделение синтаксиса, автоматическое завершение кода, сравнение исходных файлов, глобальный поиск и отладку. Как и в Eclipse, работа в NetBeans организуется по отдельным проектам, что дает возможность без особого труда исследовать проект и находить в нем нужные файлы. В качестве примера на рис. 2.14 приведено представление проекта (Project) в NetBeans.



Рис. 2.14. Пример представления проекта в NetBeans

Для разработки программ на PHP в NetBeans поддерживается выделение синтаксиса, выдача сообщений об ошибках и автоматическое завершение кода. В качестве примера на рис. 2.15 показано, каким образом в NetBeans осуществляется выделение синтаксиса, а на рис. 2.16 — сообщается об ошибке в коде PHP. Если навести курсор на маркер ошибки, появятся дополнительные сведения об ошибке, как показано на рис. 2.17.

```
* Class constructor.
  * @param array $config A configuration array
                            session_name, clientId
  * @since 11.1
public function construct($config = array())
 {
     jimport('joomla.utilities.utility');
     jimport('joomla.error.profiler');
     // Set the view name.
     $this-> name = $this->getName();
     // Only set the clientId if available.
     if (isset($config['clientId'])) {
         $this-> clientId = $config['clientId'];
     1
     // Enable sessions by default.
     if (!isset($config['session'])) {
         $config['session'] = true;
```





Рис. 2.16. Указание на синтаксическую ошибку в NetBeans





Автоматическое завершение кода в NetBeans действует таким же образом, как и в Eclipse. В качестве примера на рис. 2.18 показано автоматическое завершение вызова метода из класса Japplication после набора выражения x =this-> и нажатия комбинации клавиш <Ctrl+Пробел>. Если набрать первую часть вызова нужного метода, то, как и в Eclipse, список методов, соответствующих набранному тексту, сужается. Если же выделить метод в исходном тексте программы, появится блок документации на этот метод, как показано на рис. 2.19.

<pre>\$x = \$this-&gt;</pre>	
🕯 integer _clientId	
// Set the array messageQueue	
<pre>\$this-&gt;_n array _name</pre>	
// orlan a JInput input	
if (inset date requestTime	
Sthis	
integer startTime	
<pre>createConfiguration(\$file)</pre>	
// Enable CreateSession(\$name)	
if (!isse OcheckSession()	
<pre>\$conf O close()</pre>	
<pre>} Oclose(\$code = 0)</pre>	
<pre> dispatch() </pre>	
// Create Odispatch (\$component = null)	
if (class) 🔘 enqueueMessage(\$msg)	
<pre>\$this @ enqueueMessage(\$msg, \$type = 'message'</pre>	)
} @ getCfg(\$varname)	
<b>+ → 即</b> 🖾	
libraries/jo	omla/a
\$_clientId	
The client identifier	
re point we need was integer	
VAR integer	
SINCE 11.1	

Рис. 2.18. Автоматическое завершение кода в NetBeans





В схематическом представлении редактируемого класса в NetBeans отображаются методы и поля этого класса. Соответствующий пример приведен на рис. 2.20. В NetBeans поддерживается также сравнение текущей и предыдущей версий исходного файла с использованием предыстории (локальной или контроля версий). Пример сравнения с предыдущей версией исходного файла приведен на рис. 2.21. Кроме того, в NetBeans допускается поиск по проекту в целом. Пример такого поиска приведен на рис. 2.22.

Projects % Files Services	index.php ∞ 🚮 application.php ∞	
⊡php joomla	Source History 🔀 💀 🕶 🗸	5-5
🛱 🖟 Source Files		
⊕… 🕦 administrator	87	
⊕… 🕦 cache	88 * Class constru	ictor.
⊕…D di	89 *	
⊕… D components	90 * Gparam arra	av Şc
🕀 🖓 images 🔻	91 *	-
application.php - Navigator 🕺 📃	92 *	
JApplication:: JObject	93 * @since 11.1	1
	94 - */	
toString():string	public function	con
🖗 _createConfiguration(string \$file):object	96 - {	-
🐌 _createSession(string \$name):\JSession	97 jimport('joo	omla.u
🔘 checkSession():\void	98 Jimport(']00	omia.e
Odse(integer \$code):\void	100 // Set the 1	view n
···· 🔘 dispatch(string \$component):\void	101 Sthis-> name	= = St
enqueueMessage(string \$msg, string \$type):\void	102	
···· 🔘 getCfg(\$varname, \$default):mixed	103 // Only set	the c
🔘 getClientId():integer	104 if (isset(\$c	config
···· 🕕 getHash(string \$seed):string	105 \$this->	clien
—	106 }	
🔘 getMenu(string \$name, array \$options):\JMenu	107	
···· 🔘 getMessageQueue():array	108 // Enable se	ession
getName():string	109 if (!isset(\$	confi

Рис. 2.20. Навигатор NetBeans, отображающий методы редактируемого класса







Рис. 2.22. Пример поиска в NetBeans

Как и в Eclipse, результаты поиска в NetBeans привязываются к редактору исходного текста. Так, если дважды щелкнуть на любой строке кода в преставлении результатов поиска, соответствующий файл откроется для редактирования, а курсор установится на месте искомого фрагмента кода. В состав NetBeans входит и встроенный отладчик, также называемый Xdebug. В качестве примера на рис. 2.23 приведена программа, приостановленная во время ceaнса отладки в NetBeans.



Рис. 2.23. Пример отладки исходного кода программы в NetBeans

В представлении переменных (Variables), доступном в NetBeans, отображаются значения всех переменных из области действия в той точке, где выполнение программы было приостановлено, как показано на рис. 2.24. А в представлении стека вызовов (Call Stack), также доступном в NetBeans, отображается содержимое стека вызовов, как показано на рис. 2.25. Как и в Eclipse, в стеке вызовов перечисляются все вызовы методов, приведшие в данную точку выполнения программы.

Varia	ables %	Call Stack	Breakpoints	Search Results	Tasks		-
Ð	△ Nam	e		Туре		Value	
éÐ.	🗉 🖵 Su	perglobals					🔺
$\sim$	⇔pk			integer			
	🗆 劒 thi	s		ContentModelArticle			
	۲	_state_set		boolean		1	
	٠	_context		string		] "com_content.article"	
	±	db		JDatabaseMySQLi			
		errors		array[0]			
		item		array[0]			
	( e	event_dean_cad	he	string		onContentCleanCache"	
	( ∲ r	name		string		"article"	
	<u>ه</u> د	ption		string		"com_content"	
	± 🎲 s	tate		JObject			

Рис. 2.24. Представление переменных при отладке в NetBeans

Variables	Call Stack %	Breakpoints	Search Results	Tasks	
Name					
compone	nts/com_conten	t/models/article.php	.ContentModelArticle-	>getItem:80	
libraries/j	oomla/applicatio	n/component/view.	php.JView->get:348		
compone	nts/com_conten	t/views/article/view	.html.php.ContentView	Article->disp	olay:36
libraries/j	oomla/applicatio	n/component/contro	oller.php.JController->	display:650	
compone	nts/com_conten	t/controller.php.Cor	ntentController->displa	y:78	
libraries/j	oomla/applicatio	n/component/contro	oller.php.JController->	execute:684	
compone	nts/com_conten	t/content.php.requi	re_once:19		
libraries/j	oomla/applicatio	n/component/helpe	r.php.JComponentHelp	er::renderCo	omponent: 156
includes/	application.php.:	)Site->dispatch: 191			
index.ph	p.{main}:43				

Рис. 2.25. Представление стека вызовов при отладке в NetBeans

#### Выбор между Eclipse и NetBeans

Обе IDE, Eclipse и NetBeans, являются полноценными средами с целым рядом полезных функций для разработки в Joomla. К тому же обе они свободно доступны. Если вы собираетесь серьезно заниматься разработкой программ на PHP для Joomla, не пожалейте времени на опробование этих IDE, чтобы выбрать из них наиболее подходящую.

Впрочем, сделать это будет на так-то просто. В связи с этим ниже приводятся некоторые соображения, на которые стоит обратить внимание, принимая окончательное решение.

- Для начинающих разрабатывать программы на PHP в NetBeans предоставляется более полно и качественно составленная документация. Кроме того, NetBeans проще настраивать и легче пользоваться отладчиком. Некоторые считают, что освоить NetBeans проще и легче, чем Eclipse, хотя первая среда настраивается в меньшей степени, чем вторая.
- В Eclipse применяются элементы управления окнами пользовательского интерфейса (так называемые виджеты), исконные для соответствующей операционной системы, поэтому внешне Eclipse выглядит как собственное приложение на каждой платформе. Так, в Windows эта среда похожа на остальные приложения для Windows, а в Mac OS X или Linux выглядит так же, как и другие приложения для этих операционных систем. С другой стороны, в NetBeans используются виджеты типа Java SWING, и поэтому данная среда выглядит одинаково на всех платформах, т.е. она не очень похожа на собственное приложение для каждой платформы. Некоторые усматривают в этом недостаток NetBeans и преимущество Eclipse.
- NetBeans имеет более скромный послужной список в качестве IDE для программирования на PHP, чем Eclipse, но быстро сравнилась с ней по своим функциональным возможностям — по крайней мере, в отношении самых простых задач разработки.
- На момент написания этой книги в NetBeans не поддерживались межплатформенные окончания строк. Так, если вы разрабатываете на платформе Windows и в ваших файлах требуются окончания строк для Linux, вам придется немного потрудиться, чтобы ввести окончания строк для Linux в файлы, созданные вами в Windows. В Eclipse (и многих других редакторах исходного текста программ) для этого достаточно настроить соответствующий параметр. Остается лишь надеяться, что такая возможность в скором времени появится и в NetBeans.

Если вы все еще не решились опробовать одну из этих сред, опробуйте обе, а затем выберите из них наиболее подходящую.

#### Коммерческие IDE

На рынке предлагается немало коммерчески доступных IDE для разработки программ на PHP. Ниже перечислены лишь некоторые из них.

- ZendStudio (http://www.zend.com/en/products/studio) распространенная коммерческая IDE для разработки программ на PHP, основанная на Eclipse.
- Aptana Studio Pro (http://www.aptana.com) другая коммерческая IDE для разработки программ на PHP, настоятельно рекомендуемая некоторыми разработчиками и также основанная на Eclipse.
- Komodo IDE (http://www.activestate.com/komodo-ide) еще одна коммерческая IDE для разработки программ на PHP.
- PhpED от компании Nusphere (http://www.nusphere.com/products/phped. htm) — эта среда предоставляет встроенный отладчик и профилировщик, а также функции развертывания готового программного продукта.

#### Текстовые редакторы

Если материал предыдущего раздела не убедил вас остановить свой выбор на IDE или хотя бы опробовать ее, то в этом разделе рассматриваются более простые инструментальные средства для разработки программ на PHP.

#### Notepad++

Notepad++ представляет собой простой в использовании, но весьма эффективный открытый редактор исходного текста, предназначенный для Windows. В нем поддерживается выделение синтаксиса и отчасти завершение кода для целого ряда языков программирования, включая и PHP. Этот редактор исходного текста можно свободно загрузить по адресу http://notepad-plus-plus.org.

В качестве примера на рис. 2.26 приведен открытый в Notepad++ исходный файл программы на PHP, написанной для Joomla. Как следует из данного примера, в Notepad++ выделение ошибок не поддерживается.

Для пользователей Mac OS X имеется простой и удобный редактор исходного текста Smultron (http://smultron.en.softonic.com/mac), обеспечивающий выделение синтаксиса в коде PHP. А для пользователей Linux имеется стандартный системный редактор gedit, также обеспечивающий выделение синтаксиса в коде PHP.

Для написания программ на PHP можно также воспользоваться стандартным текстовым редактором вроде Notepad или vi, но рассчитывать на автоматическое выделение синтаксиса в таком редакторе не приходится. Любой редактор простого текста вполне подойдет для правки всех файлов Joomla. Очевидно, что чем дальше вы отступаете от функциональных возможностей IDE, тем труднее вам будет визуально проверять набранный исходный текст программы на наличие ошибок и опечаток.

Не менее важно понять, что для правки одного и того же набора файлов Joomla можно пользоваться самыми разными программными средствами. Даже если файл находится в

проекте, определенном в IDE, он по-прежнему доступен на уровне файловой системы для редактирования за пределами IDE. Таким образом, для решения различных задач разработки можно при желании пользоваться разными программными средствами.

	application.ph	IP
19	3	*/
19	4	<pre>public static function getInstance(\$client, \$config = array(), \$pr</pre>
19	5 白	{
19	6	<pre>if (empty(self::\$instances[\$client]))</pre>
19	7 白	{
19	8	<pre>// Load the router object.</pre>
19	9	<pre>\$info = JApplicationHelper::getClientInfo(\$client, true);</pre>
20	0	
20	1	<pre>\$path = \$info-&gt;path . '/includes/application.php';</pre>
20	2	<pre>if (file_exists(\$path))</pre>
20	3 白	{
20	4	include_once \$path;
20	5	
20	6	<pre>// Create a JRouter object.</pre>
20	7	<pre>\$classname = \$prefix . ucfirst(\$client);</pre>
20	8	<pre>\$instance = new \$classname(\$config);</pre>
20	9 -	}
21	0	else
21	1 🛱	{
21	2	<pre>\$error = JError::raiseError(500, JText::sprintf('JLIB_</pre>
21	3	return \$error;
0.0	4	

Рис. 2.26. Редактор исходного текста Notepad++

#### Другие инструментальные средства

Для разработки в Joomla могут пригодиться и другие программные средства. Ниже они рассматриваются вкратце.

#### Программное обеспечение контроля версий

Это программное обеспечение считается очень важным и просто необходимым для большинства программных проектов, включая и Joomla. Программное обеспечение контроля версий автоматически отслеживает каждое изменение в кодовой базе проекта, позволяя группе разработчиков вносить изменения в кодовую базу в организованном порядке. Если же изменение, внесенное в код, вызовет какие-нибудь затруднения, оно может быть отменено с целью устранить эти затруднения. Даже если вы разрабатываете проект в одиночку, программное обеспечение контроля версий все равно принесет вам огромную пользу. Обнаружив в проекте какую-нибудь ошибку, вы можете всегда вернуться назад, чтобы выяснить, где именно вкралась ошибка. Это нередко помогает сузить поиск ошибок и недостатков в проекте.

В проектах, разрабатываемых в системе управления контентом (CMS) и на платформе Joomla, для контроля версий применяется программа Git (http://git-scm.com). Подобные проекты размещаются на веб-сайте Github (https://github.com/joomla). К числу других распространенных программ контроля версий относятся Subversion (http:// subversion.apache.org) и Mercurial (http://mercurial.selenic.com). Если вы пользуетесь IDE, то средства контроля версий обычно включаются в саму IDE. Например, в Eclipse и NetBeans поддерживаются программы GIT, Subversion и Mercurial.

### Программное обеспечение автоматического построения проектов (Ant и Phing)

Исходные файлы программ на PHP выполняются непосредственно как сценарии. Следовательно, программы на PHP не нужно компилировать, как этого требуют программы на Java или C. Разработчики, программирующие на языках, требующих компиляции или построения готовых программных продуктов, нередко пользуются программами, автоматизирующими эти процессы. К числу таких программ относятся Ant и Phing. Если Ant первоначально разработана для работы с программами на C и Java, то Phing выполняет операции построения проектов на PHP в пакетном режиме.

Если построение программ на PHP не требуется, то зачем вообще нужные такие программные средства, как Ant или Phing? Дело в том, что часто повторяющиеся пакетные задания иногда оказывается удобно автоматизировать. Так, если требуется построить расширение Joomla, то организовать работу можно, например, разделив ее на два проекта в IDE. Один проект будет содержать только код расширения и может быть связан с хранилищем для контроля версий, а другой позволит протестировать расширение в пределах установленных компонентов Joomla.

В данном примере расширение будет тестироваться в проекте тестирования, а изменения в коде — отслеживаться в проекте расширения. Возможно, было бы удобнее сначала изменить код в исходных файлах проекта тестирования, а после проверки на работоспособность скопировать файлы из проекта тестирования в проект расширения. С помощью программы Ant или Phing процесс копирования файлов из одного проекта в другой можно было бы автоматизировать. А кроме того, можно автоматизировать процесс создания архивов для упаковки расширений.

Учебный материал по применению Phing с подобной целью доступен на сайте документации на Joomla по адресу http://docs.joomla.org/Setting\_up\_your\_ workstation for extension development.

#### Программы автоматизации тестирования

В разработке программного обеспечения наметилась устойчивая тенденция к автоматизации процесса тестирования программ. Автоматические тесты могут быть написаны один раз, а затем выполнены неоднократно для обнаружения ошибок, вкравшихся в кодовую базу при внесении в нее изменений.

В проекте Joomla в настоящее время применяются две разновидности автоматического тестирования: блочное и системное. Блочное тестирование служит для тестирования платформы Joomla, а для выполнения блочных тестов — программа на PHP под названием PHPUnit. Это дает возможность писать тесты, непосредственно снабжающие данными метод в коде PHP, и сравнивать полученные результаты с предполагаемыми. Если они отличаются, выдается сообщение об ошибке.

Системное тестирование служит для проверки работоспособности системы CMS с точки зрения пользователя. С этой целью система CMS выполняется в сеансе работы

#### Резюме

с браузером под управлением тестовой программы. А в браузере выполняется последовательность команд ввода данных, включая операции с мышью и ввода с клавиатуры, имитирующие действия пользователя тестируемого программного обеспечения. Документ в окне браузера может быть автоматически проверен в системном тесте на предмет появления некоторой информации на веб-странице. Если эта информация не появляется на веб-странице, выдается сообщение об ошибке. Для создания и выполнения системных тестов в Joomla применяется программа Selenium.

Выполнение подобных автоматических тестов является важной стадией процесса paspaбoтки в Joomla. При внесении изменений в виде устраняемых программных ошибок или усовершенствований кода автоматические тесты могут быть выполнены с целью упростить проверку каких-либо нарушений, вызванных внесенными изменениями.

#### Резюме

В этой главе рассмотрены вопросы подготовки рабочей станции к написанию программ для Joomla. Сначала в ней обсуждались варианты установки программного обеспечения веб-сервера на разных платформах, включая отдельные пакеты для различных операционных систем. Затем были вкратце описаны две IDE, наиболее распространенные для программирования на PHP: Eclipse и NetBeans, а также представлены примеры, показывающие, насколько они позволяют повысить производительность труда разработчиков прикладных программ для Joomla.

Затем были вкратце рассмотрены разные варианты редакторов исходного текста для тех, кто не желает пользоваться IDE. И в конце главы было кратко описано программное обеспечение контроля версий (вроде Subversion и Mercurial), автоматического построения проектов (типа Phing и Ant), а также автоматического тестирования.

Итак, рабочую станцию можно считать подготовленной и настроенной, и теперь можно приступать непосредственно к написанию прикладных программ для Joomla.