

О техническом рецензенте



Вес Хант (Wes Hunt) был разработчиком приложений и руководителем группы по созданию пользовательского интерфейса (UX lead) с конца 90-х годов прошлого века. В настоящее время он является основателем и ведущим разработчиком в компании *Agmigent* — стартапе веб-служб в индустрии инвестиций в недвижимость. Хант использует PHP и Scala для быстрой разработки веб-приложений и REST-служб как для небольших интернет-компаний (стартапов), так и для крупных корпораций. В свободное от работы время он обучает программированию детей в Монтане через сайт CodeMontana.org (только в этом году их было более тысячи), а также занимается защитой прав программистов в группе пользователей сайта MontanaProgrammers.org.

Благодарности

Как всегда, хочу выразить признательность всем, кто работал над этим изданием книги. Также я должен вспомнить всех, кто работал над всеми предыдущими изданиями.

Некоторые из основополагающих концепций этой книги были опробованы мною на конференции в Брайтоне, где мы все собрались, чтобы познакомиться с замечательными возможностями PHP 5. Выражаю благодарность организатору конференции Энди Бадду (Andy Budd) и всему животрепещущему сообществу разработчиков Брайтона. Благодарю также Джесси Вайт-Синис (Jessey White-Cinis), которая познакомила меня на конференции с Мартином Штрейхером (Martin Streicher) из издательства Apress.

Как и прежде, сотрудники издательства Apress оказывали мне неоценимую поддержку, высказывали ценные замечания и обеспечивали всяческое содействие. Я необычайно рад работать в такой команде профессионалов.

Выражаю благодарность и свою безграничную любовь своей жене Луизе и детям Холли и Джейку за то, что они периодически отвлекали меня от сложной работы и обеспечивали разрядку.

Спасибо Стивену Метскеру (Steven Metsker) за любезное разрешение реализовать на PHP чрезвычайно упрощенную версию API синтаксического анализатора, который он представил в своей книге *Building Parsers in Java*.

Я работаю под музыку, и в первых трех изданиях этой книги я упомянул о великом диджее Джоне Пиле (John Peel), поборнике всего тайного и эклектичного. Я должен также поблагодарить радиовещателей, принявших эстафету от Джона, особенно тех, кто работает в таинственных уголках радиостанции BBC 6 Music and Dandelion Radio.

Предисловие

Когда меня впервые посетила идея написать эту книгу, объектно-ориентированные средства разработки в РНР использовали лишь избранные программисты. Однако со временем мы увидели не только безусловный рост популярности объектно-ориентированных средств языка РНР, но и развитие всего фреймворка. Безусловно, фреймворки (каркасы приложений) чрезвычайно полезны. Они составляют основу многих (на сегодняшний день, вероятно, большинства) веб-приложений. Более того, часто эти каркасы точно иллюстрируют основные подходы к проектированию программного обеспечения, рассматриваемые в этой книге.

Тем не менее здесь для разработчиков таится определенная опасность, точно так же как и при использовании любого другого полезного API. Речь идет о боязни того, что пользовательские приложения могут стать зависимыми от некоего стороннего гуру, который никак не удосужится внести исправления в созданный им каркас или по собственной прихоти внес изменения в его функционал. На самом деле подобная точка зрения часто приводит к тому, что разработчики программного обеспечения рассматривают внутреннюю структуру каркаса как некий “черный ящик”, а свою часть работы считают не более чем небольшой надстройкой, поставленной на вершину огромной и непознанной инфраструктуры.

Несмотря на то что я отношусь к злым изобретателям велосипедов, суть моих доводов состоит вовсе не в том, что я призываю вас полностью отказаться от своих любимых каркасов и начать создавать MVC-приложения “с нуля” (по крайней мере, хотя бы иногда). Наоборот, мы как разработчики должны понимать задачи, которые решает тот или иной каркас, а также разбираться в методиках, которые используются для их решения. Мы должны оценивать каждый каркас не только с точки зрения предлагаемых им функциональных возможностей, но и с точки зрения проектных решений, которые использовали их создатели, и оценить, насколько качественно они реализованы. Разумеется, что, когда нам позволяют обстоятельства, мы как разработчики должны развиваться и создавать собственные служебные и специализированные приложения, а со временем — создать собственную библиотеку повторно используемого кода.

Я надеюсь, что эта книга поможет РНР-разработчикам выработать навыки проектно-ориентированного мышления и реализовать их в собственных программных платформах и библиотеках. Здесь описаны также несколько концептуальных средств, которые пригодятся вам, когда наступит время действовать в одиночку и брать всю ответственность на себя.

Совсем недавно я потратил около года на обучение. И это то, что я настоятельно рекомендую сделать всем по целому ряду причин. Одна из них — после обучения вы сможете взглянуть на знакомый мир с другой точки зрения. После возврата к своей привычной консалтинговой деятельности я с удивлением обнаружил, что большинство моих старых клиентов и знакомых сделали решительный шаг вперед и перешли на использование системы контроля версий Git (в этом издании мне также пришлось отдать дань моде). Кроме того, почти все из них стали называть свою методологию разработки *гибкой* (*agile*), хотя трое из четверых моих новых клиентов попросили оценить их наспех созданную и по этой причине совсем не гибкую кодовую базу (codebase). В каждом проекте требуется сначала создать (или подогнать) набор модульных тестов, написать основную документацию и разработать механизм автоматизированного построения проекта. И только после этого можно браться за рефакторинг. В противном случае его результаты будут весьма плачевными. Я потратил немало усилий на то, чтобы освоить на практике все те средства

и правила, о которых говорится в последней части книги. И я надеюсь, что вы также оцените их по достоинству и что они помогут вам создавать надежные и гибкие программные системы.

От издательства

Вы, читатель этой книги, и есть главный ее критик. Мы ценим ваше мнение и хотим знать, что было сделано нами правильно, что можно было сделать лучше и что еще вы хотели бы увидеть изданным нами. Нам интересно услышать и любые другие замечания, которые вам хотелось бы высказать в наш адрес.

Мы ждем ваших комментариев и надеемся на них. Вы можете прислать нам бумажное или электронное письмо либо просто посетить наш веб-сайт и оставить свои замечания там. Одним словом, любым удобным для вас способом дайте нам знать, нравится ли вам эта книга, а также выскажите свое мнение о том, как сделать наши книги более интересными для вас.

Посылая письмо или сообщение, не забудьте указать название книги и ее авторов, а также ваш обратный адрес. Мы внимательно ознакомимся с вашим мнением и обязательно учтем его при отборе и подготовке к изданию последующих книг. Наши координаты:

E-mail: info@williamspublishing.com

WWW: <http://www.williamspublishing.com>

Адреса для писем из:

России: 127055, г. Москва, ул. Лесная, д. 43, стр. 1

Украины: 03150, Киев, а/я 152