<<PHP核心技术与最佳实践>>

图书基本信息

书名：<<PHP核心技术与最佳实践>>

13位ISBN编号：9787111401926

10位ISBN编号：7111401921

出版时间：2012-11

出版时间：机械工业出版社

作者：列旭松，陈文

页数：540

前言

前言为什么要写这本书近几年，市场上关于PHP的书已经很多了，各种培训机构也如雨后春笋般不断增加。
那为什么还要写这本书呢？
这本书存在的意义又在哪里？
这要从下面的几个问题说起。
有没有这样一本PHP教材，它不讲HTML和CSS，也不讲JavaScript基础，甚至不讲PHP语法基础？
有没有这样一本PHP教材，它不讲留言本或博客的开发，也不讲数据库的CRUD操作？
有没有这样一本PHP教材，它专注于Web开发技术的最前沿，深入浅出，适合中高级程序员的进阶和提高？
有没有这样一本PHP教材，它提倡面向对象的程序思想，提倡算法和数据结构的重要性，提倡对网络协议的深入理解，且没有大篇幅的代码，而是更多偏重于理论讲解？
有没有这样一本PHP教材，它探讨PHP的扩展开发，探讨高并发大流量的架构，深入探讨NoSQL的内部实现和细节？
以上几个问题也是我在早期PHP学习的过程中一直在寻找的答案，可是我并没有找到一本理想的PHP书籍，一本适合中高级程序员进阶的书籍。
当怀着同样问题的旭松兄找到我时，我们不禁产生一个念头：“既然现在市场上缺少一本这样的书籍，我们何不自己写一本呢？
利己利人的事值得去做。
”然后一拍即合，说做就做，现在这本书经历长达一年多的酝酿和写作过程终于完稿了。
我是在大学期间接触到PHP语言的，并马上被其简洁的语法和极高的开发效率所吸引，一头扎进PHP开发的世界中。
随着学习的深入，并经常关注PHP社区的动态，我很快意识到一些PHP社区普遍存在的问题。
比如PHP社区一直争论算法重不重要，面向对象好不好，代码质量重要还是开发速度重要的问题。
还有譬如为什么我去大型互联网公司应聘PHP程序员，却不考察我对PHP语法和函数的掌握情况，而是会问我C语言、算法、网络协议、高并发处理、MVC理论这些看似和PHP不沾边的问题。
PHP到底要怎么学，学什么，一个高级PHP程序员应该是什么样的，我想这也是很多PHP新手和工作一两年的PHP开发者的疑惑。
这本书所要解决的就是这一系列的问题。
在我看来，一本技术书籍的价值在于其对知识的提炼和与众不同的地方。
举例来说，到一个书店去看书，你最想用笔抄下来或撕下来带走的那几页，就是对你帮助最大的东西，也是你认为这本书的价值所在。
也是基于这个想法，我们思考这本书该写什么，怎么写，哪些地方对读者有帮助。
我们试图从不同的角度带领读者来看PHP，进而给这本书注入一些不一样的东西。
我们希望这是一件有意义的事。
本书适合的对象PHP爱好者；想进阶的初级PHP程序员；对PHP扩展开发感兴趣的读者；对高并发感兴趣的读者；对NoSQL应用和实现原理感兴趣的读者；从事PHP网络应用，想知道HTTP协议、Socket等更多细节的开发人员；想就职于大型互联网公司的PHP程序员；开设相关课程的大专院校的学生；公司内部培训的学员。
如何阅读本书本书一共有14章。
每章节都可以单独阅读，由于部分知识点之间存在一定的衔接，故建议按先后顺序阅读。
第1章为面向对象思想的核心概念。
本章主要讲解面向对象开发的思想，重点讲解面向对象模型的建立，以及面向对象的一些基础概念。
通过大量对比和实例，尤其是与Java的对比，力图从不同角度讲解PHP面向对象的特性，让PHP程序员看到不同的面向对象。
求同存异是本章的核心思想。

第2章为用面向对象思维写程序。
本章用简练的语言讲解了面向对象设计的五大原则，这五大原则也是理解设计模式的基础所在，帮助读者站在一个更高的角度思考面向对象。
第3章为正则表达式技巧与实战。
本章详细介绍了正则的基础语法，通过大量的示例、通俗的语言讲解正则概念，引导读者理解正则的一系列规则。
接下来，结合实际工作用安全过滤、URL重写等实例，加深对正则的应用和掌握。
最后给出正则效率优化的一些普遍技巧和替代方案，让读者对正则的使用得心应手。
第4章为PHP网络技术及应用。
本章着重介绍了HTTP协议、Socket开发、WebService、Cookie和Session使用等。
结合实战向读者阐述网络开发的核心和重点，特别是对HTTP协议的理解。
HTTP协议是Web开发的基石，也是各种面试和开发中必然遇到的知识点。
而Socket则是应用交互的桥梁，保证了有用的可扩展性。
第5章为PHP与数据库基础。
本章从不同角度分析了MySQL，介绍了PDO、MySQL优化、存储过程、事件调度机制以及MySQL安全防范等内容。
第6章为PHP模板引擎的原理与最佳实践。
本章通过实现一个简单的模板引擎，学习模板引擎的原理和使用方法，然后对比几大流行的模板引擎实现方案，简单介绍了各种实现方案的思想和优缺点，最后探讨模板引擎的意义。
第7章为PHP扩展开发。
本章的知识是本书核心内容，介绍了PHP扩展开发的几个重要知识点，如扩展框架搭建、PHP生命周期、PHP变量在内核中的实现方式、Zend引擎、内存管理等，让读者深入PHP底层，知其然也知其所以然。
第8章为缓存。
本章主要介绍了缓存的基本原理和三个衡量指标，通过几个实例加深读者对缓存的理解。
利用本章知识，读者应该能设计一个比较合理的缓存方案。
第9章为Memcached应用与内幕。
本章深入剖析了Memcached的实现和内部结构，从而使读者掌握Memcached的高级应用，对构建复杂环境的缓存层有个清晰的认识。
第10章为Redis应用与内幕。
本章重点介绍了Redis的深入应用，如事务处理、主从同步、虚拟内存等，和第9章类似，探讨了Redis的实现内幕。
合理利用Redis可以为我们解决大流量高并发的应用。
第11章为高性能网站架构。
本章探讨了高性能架构的基本出发点，重点以Handler Socket、MySQL主从复制、反向代理缓存软件Varnish和任务分发框架Gearman为例，讲述几种高性能架构中会用到的技术。
第12章为调试与测试。
科学的调试方法有助于快速找出潜在的Bug、理解复杂应用的流程、提高开发效率。
单元测试是代码质量的保障。
在这一章的最后一节介绍了使用JMeter进行压力测试的方法。
第13章为Hash算法与数据库的实现。
本章介绍了Hash算法的基本原理，用此算法实现一个简单的、基于Hash的数据库，让读者意识到算法的重要性和可操作性。
第14章为PHP编码规范。
本章介绍了PHP开发中应遵循的基本代码规范，并提出合理建议。
好的代码必然是规范的代码。
本书第1、2、3、5、6、8、12、14章由陈文撰写，第7、9、10、11、13章由列旭松撰写，第4章由两人

共同完成。
勘误和支持由于我们的水平和开发经验有限，同时计算机技术更新较快，书中难免存在不足之处，有些章节内容可能从未来的某一天开始不再适用，还望读者理解和体谅，并恳请读者批评指正。
您若对本书有什么好的建议或者对书中部分内容有疑惑，可与我们联系，我们将尽量为读者提供最满意的解答。
期待得到您的真挚反馈。
我们的联系方式如下：陈文：waitfox@qq.com列旭松：liexusong@qq.com感谢首先要感谢PHP之父Rasmus Lerdorf，是他创建了这个简单、轻松、有趣、快速而又高效的语言；其次，感谢PHP社区每一位充满活力的朋友，和你们的交流使我学到很多，本书有不少内容就来自于社区的智慧。
在这里尤其要感谢机械工业出版社华章公司的大力支持，特别是杨福川和白宇两位编辑，在一年多的时间里，因为有了你们的耐心指导、逐字逐句认真审稿和改稿才有了本书的诞生。
最后，还要感谢家人和朋友的支持。
陈文

<<PHP核心技术与最佳实践>>

内容概要

这是一本致力于为希望成为中高级PHP程序员的读者提供高效而有针对性指导的经典著作。本书系统归纳和深刻解读了PHP开发中的编程思想、底层原理、核心技术、开发技巧、编码规范和最佳实践。

全书分为5个部分：第一部分（1～2章）从不同的角度阐述了面向对象软件设计思想的核心概念、技术和原则，分析了面向对象的特性、设计模式的理念，指出了如何设计低耦合、高可扩展性的软件，等等；第二部分（3～6章）详细讲解了PHP中正则表达式的规范和使用技巧，PHP网络编程的原理、方法、技巧和一些重要的操作，PDO、数据库应用优化，数据库设计和MySQL的高级应用，PHP扩展引擎的原理与实践；第三部分（第7章）拨云见日，围绕PHP扩展开发进行了细致而深入的探讨，解析了PHP的底层实现和Zend虚拟机API，并用PHP扩展开发的实例带领读者走进PHP的底层世界，旨在让读者对PHP性能优化、底层原理进行深入的理解。
第四部分（8～11章）重点讨论了缓存的设计、Memcached的原理与实践、NoSQL数据库Redis源码分析与应用实践、高性能PHP网站的架构和设计等内容；第五部分（12～14章）详细讲解了PHP代码的调试和测试、Hash算法和数据库的实现，以及PHP的编码规范，旨在帮助读者提高开发效率，养成良好编程习惯。

作者简介

列旭松，资深PHP技术工程师，精通PHP及其相关技术，对PHP内核原理有较深入的理解，开发经验丰富。
曾自主开发了关键字匹配服务器（http://code.google.com/p/sensitive-filter-server/）和消息队列SquirrelMQ（http://code.google.com/p/squirrel-message-queue/）。
平时喜欢开发一些实用的PHP扩展，如PHP字典扩展（红黑树算法）（http://code.google.com/p/php-dict/）和PHP索引扩展（B+树算法）（http://code.google.com/p/php-mini-database/）。
精通C语言，同时对Web服务器的架构和优化、高并发服务端编程、Redis和Memcached等技术有深入的研究和认识。
活跃于PHPChina和ChinaUnix等专业社区，担任PHPChina论坛内核版块版主。

陈文，资深PHP技术工程师，精通PHP及其相关技术，尤其擅长于PHP框架开发和应用架构。他还是一位资深的Java开发工程师，具有Fortran、Scala和C++语言的开发和使用背景，在传统软件和互联网开发领域都有丰富的实战经验。
此外，他还擅长TCP/IP编程、多线程与并发程序设计、网络协议分析、数据库性能优化以及各种缓存技术，熟悉MySQL和Oracle等关系数据库产品。
现从事网络安全软件开发，以及移动SI业务开发。
对语言特性和软件设计思想有独到的见解，追求代码之美和高效率程序开发，爱好钻研底层技术，崇尚和提倡“以理论指导实践”。
尤其爱好数学，认为数学是培养和锻炼思维和逻辑能力的重要工具，对算法有一定研究。
长期活跃在PHPChina、ITeye和看雪论坛等社区，在PHPChina社区担任版主。

<<PHP核心技术与最佳实践>>

书籍目录

章节摘录

版权页： 插图： 继承并非一无是处，而组合也不是完美无缺的。
如果既要组合的灵活，又要继承的代码简洁，能做到吗？
这是可以做到的，譬如多重继承，就具有这个特性。
多重继承里一个类可以同时继承多个父类，组合两个父类的功能。
C++里就是使用的这种模型来增强继承的灵活性的，但是多重继承过于灵活，并且会带来“菱形问题”，故为其使用带来了不少困难，模型变得复杂起来，因比在大多数语言中，都放弃了多重继承这一模型。
多重继承太复杂，那么还有其他方式能比较好地解决这个问题吗？
PHP5.4引入的新的语去结构Traits就是一种很好的解决方案。
Traits的思想来源于C++和Ruby里的Mixin以及Scala里的Traits，可以方便我们实现对象的扩展，是除extend、implements外的另外一种扩展对象的疗式。
Traits既可以使单继承模式的语言获得多重继承的灵活，又可以避免多重继承带来的种件问题。
1.3.2各种语言中的多态 多态确切的含义是：同一类的对象收到相同消息时，会得到不同的结果。
而这个消息是不可预测的。
多态，顾名思义，就是多种状态，也就是多种结果。
以Java为例，由于Java是强类型语言，因此变量和函数返回值是有状态的。
比如，实现一个add函数的功能，其参数可能是两个int型整数，也可能是两个float型浮点数，而返回值可能是整型或者浮点型。
在这种情况下，add函数是有状态的，它有多种可能的运行结果。
在实际使用时，编译器会自动匹配适合的那个函数。
这属于函数重载的概念。
需要说明的是，重载并不是面向对象里的东西，和多态也不是一个概念，它属于多态的一种表现形式。
多态性是一种通过多种状态或阶段描述相同对象的编程方式。
它的真正意义在于：实际开发中，只要关心一个接口或基类的编程，而不必关心一个对象所属于的具体类。
很多地方会看到“PHP没有多态”这种说法。
事实上，不是它没有，而是它本来就是多态的。
PHP作为一门脚本语言，自身就是多态的，所以在语言这个级别上，不谈PHP的多态。
在PHP官方手册也找不到对多态的详细描述。
既然说PHP没有多态这个概念（实际上是不需要多态这个概念），那为什么又要讲多态呢？
可以看下面的例子，如代码清单1—8所示。

编辑推荐

《PHP核心技术与最佳实践》系统归纳和深刻解读PHP开发中的编程思想、底层原理、核心技术、开发技巧、编码规范和最佳实践，为PHP程序员进阶修炼提供全面而高效的指导！

<<PHP核心技术与最佳实践>>

版权说明