

PHP Objects, Patterns, and Practice **Second Edition**

深入PHP

面向对象、模式与实践 (第2版)

[美] Matt Zandstra 著
陈浩 吴孙滨 胡丹 译
陈浩 审校

- 世界级专家力作
- 完美结合面向对象思想与PHP技术
- PHP模块化编程必读



人民邮电出版社
POSTS & TELECOM PRESS

TURING

图灵程序设计丛书 Web开发系列

PHP Objects, Patterns, and Practice **Second Edition**

深入PHP

面向对象、模式与实践

(第2版)

人民邮电出版社
北京

图书在版编目 (CIP) 数据

深入PHP：面向对象、模式与实践：第2版 / (美)
赞德斯彻 (Zandstra, M.) 著；陈浩，吴孙滨，胡丹译.

— 北京：人民邮电出版社，2010.5

(图灵程序设计丛书)

书名原文：PHP Objects, Patterns, and
Practice, Second Edition

ISBN 978-7-115-22578-8

I. ①深… II. ①赞… ②陈… ③吴… ④胡… III.
①PHP语言—程序设计 IV. ①TP312

中国版本图书馆CIP数据核字(2010)第050678号

内 容 提 要

本书是 PHP 专家经典力作的最新版本。书中主要介绍了如何使用面向对象技术和设计模式编写稳定的、可维护的代码，如何使用强大的 phpDocumentor 自动文档系统创建详细的多功能文档，如何通过管理 CVS 中的代码和使用 Phing 构建系统在开发过程中得到更好的灵活性，以及如何使用 PEAR 包管理解决方案。

本书适合中高级 PHP 程序员阅读。

图灵程序设计丛书

深入PHP：面向对象、模式与实践（第2版）

- ◆ 著 [美] Matt Zandstra
- 译 陈 浩 吴孙滨 胡 丹
- 责任编辑 王军花
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
- 邮编 100061 电子函件 315@ptpress.com.cn
- 网址 <http://www.ptpress.com.cn>
- 北京顺义振华印刷厂印刷
- ◆ 开本：800×1000 1/16
- 印张：25
- 字数：622千字
- 印数：1-3 000册

2010年5月第1版

2010年5月北京第1次印刷

著作权合同登记号 图字：01-2009-0528号

ISBN 978-7-115-22578-8

定价：59.00元

读者服务热线：(010)51095186 印装质量热线：(010)67129223

反盗版热线：(010)67171154

版 权 声 明

Original English language edition, entitled *PHP Objects, Patterns, and Practice, Second Edition* by Matt Zandstra, published by Apress, 2855 Telegraph Avenue, Suite 600, Berkeley, CA 94705 USA.

Copyright © 2008 by Matt Zandstra. Simplified Chinese-language edition copyright © 2010 by Posts & Telecom Press. All rights reserved.

本书中文简体字版由Apress授权人民邮电出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

版权所有，侵权必究。

译者序

译者从接触PHP开发到现在接近十年，参加了不少大型互联网应用的开发。在此过程中，有几个问题是译者一直在关注的。

中国互联网发展如日中天，全国互联网用户数量增长很快（已达3.5亿人）。当前较为流行的站点往往每天要面对至少上百万的访问者，而且需要提供越来越复杂的服务。

作为研发者的我们，应该如何设计站点，让站点能满足当前阶段的业务需求（访问量和服务），并且保证网站有良好的架构设计，方便未来的后续开发和维护？我们应该如何解决代码的“耦合”问题，如何编写出易重用的代码，甚至达到理想化的软件层面的“美学”境界（可能是程序员的终极目标）？

如何建立一套行之有效的团队协作机制，使研发团队日常的开发标准化，提高团队沟通和合作的效率。是否有一些放之四海而皆准的准则或通用的工具可以利用？

正所谓“水涨船高”，PHP这种为互联网而生的编程语言这些年来也一直在进步。如何充分利用PHP中的新特性，使日常的开发更为高效？

读者可以从本书中找到上述问题的部分答案。因为本书不仅仅是一本PHP图书，更是一本鼓励和促进读者超越代码层次，对整个站点的“设计”进行思考的书。

条条大路通罗马。要设计出漂亮又实用的站点架构，每个人也许都有自己的体会和经验。但就PHP站点而言，从根本上不外乎“面向过程”和“面向对象”两种比较底层的开发思路。面向过程的开发方式，每个人或每个团队都有自己的开发习惯，不太容易形成系统。所以这么多年来，虽然有不少优秀的面向过程的PHP程序面世，但没能形成任何可供共享的“框架”或“思想”。而面向对象的开发方式，则积累了大量公认的模式和框架。从这个角度来说，建议读者朋友多花时间在面向对象设计上。本书就“面向对象”在PHP中的实现做了非常深入的介绍，涵盖了基础、高级知识以及设计模式的实现。

从学习角度来说，译者推荐PHP程序员沿这样的学习路线前进：PHP基础入门（语法、常用函数和扩展等）→面向对象的PHP（本书就是很好的学习资料）→网站软件架构设计（设计模式、框架等）→网站物理层次架构设计（分布式计算、存储、负载均衡、高可用性等）。在现实当中，上述这些阶段和工作其实是交织在一起的。另外，程序设计之外的很多工具 and 实践经验，也极其重要。因为个人的能力有限，个人的价值必须在团队和企业当中得到体现。本书介绍的很多工具（如版本控制工具、文档工具和应用构建工具等）都可以促进团队协作。

最后，真诚邀请读者朋友通过本书来探索PHP。不论你是刚入门的PHP开发人员，还是有多

年经验的同行，相信看完本书之后，你都会有收获。当然，一个真正的程序员，他的思想永远是超越具体语言和工具的。期待有一天，读者朋友可以在了解书中这些知识的基础上，总结出属于自己的更加优秀的开发框架甚至是设计思想，开发出在互联网上受到用户欢迎的优秀站点。

本书在PHPEye开源社区设有读者反馈版块，网址为<http://bbs.phpeye.com/forum-16-1.html>。译者将通过这个版块为读者提供勘误和答疑。

陈浩 (Haohappy)

2010年3月于上海

致 谢

如果你突然想写一本书（就像我在布莱顿一家咖啡厅喝咖啡时突然有了这个念头），可能你只想到了该写什么主题，那一刹那的冲动很容易让你忘记写书是多么不容易。我很快就重新认识到写书是非常辛苦的工作，并且不是光靠一个人单打独斗就可以完成的。在编写本书的过程中，我得到了很多帮助。

实际上，我的致谢首先要追溯到还没有本书构想的阶段。本书的萌芽来自于我在布莱顿为某个技术组织作的一次报告。该组织名为Skillswap (<http://www.skillswap.org>)，由Andy Budd管理。正是由于Andy邀请我作报告，我才开始总结PHP面向对象的相关思想。在此特别感谢Andy，我得到了很多帮助。

很偶然地，我在那次会议上遇到了Apress出版社的一位作者Jessey White-Cinis，是她牵线让我与Apress出版社的Martin Streicher编辑建立了联系，而后者则正式邀请我为Apress写一本书。谢谢Jessey和Martin，让本书的出版迈出了第一步。

Apress出版社的团队为我提供了很多帮助，即使我因为工作和家庭原因一次次地拖延交稿（我自己都觉得过分）。我特别感谢Jason Gilmore在本书先后两个版本的出版过程中对我的热情帮助、Richard Dal Porto耐心的项目管理，还有Heather Lang和Benjamin Berg忍受我总也搞不明白哪些词应该用代码体。同时也非常感谢责任编辑Tom Welsh在本书编写过程中很有见地的指导，他使我学会站在读者的角度来思考问题。

在忙于写代码和文字的时候，很容易一叶障目，不见泰山，迷失对于全书总体思路的把握。我的朋友Tolan Blundell，也是本书的技术审校，一直提醒我按着全书的计划一步步前进。感谢Tolan。

谢谢Steven Metsker，他允许我将他在*Building Parsers in Java*一书中的解析器API代码改写成简化的PHP版本放入本书。

特别谢谢我在Yahoo!的上司Fergus Sullivan，他允许我临时请假去写那些比较有难度的章节。当然，我也要感谢Yahoo!对作者比较宽松的政策，否则这样的请假是不可能被允许的。

写书期间，没有太多时间照顾家庭。在此谢谢我的妻子Louise以及我们的孩子Holly和Jake。我一直都很想念你们。

自从本书第1版发行后，我收到了很多热情洋溢和富有建设性的读者回馈。我非常抱歉没能一一回复，希望在此对你们一并表达我的谢意。你们的鼓励和支持是我写作第2版的动力。

我在写第1版时常听John Peel^①主持的节目。John 40年来一直在抨击烦乏无味粗制滥造的音乐作品，捍卫原创音乐，推崇兼收并蓄的风格。John Peel于2004年10月突然去世，令无数喜欢他的听众怅然若失。他对很多人的生活产生了巨大影响，在此表示敬意。

① 他是BBC电台资深DJ。——译者注

第1版简介

从2000年起我就开始使用PHP开发面向对象项目。当然，那时的PHP主要指PHP 4，而PHP 4对于面向对象的支持比较有限。即使如此，我发现PHP 4几乎可以满足我在面向对象开发上的所有需求，只要足够细心并遵守规范。

2003年初，我开始规划一本关于PHP 4与面向对象开发的书。这本书将用相当一部分篇幅阐述一些必需的策略、规范和变通方案，使PHP适用于面向对象开发。但这时我听说PHP 5马上就要面世，而且面世的时间将在我开始动笔写书之前。于是我放弃了那本书的写作计划，开始构思另一本书。

我开始研究PHP 5新特性的细节，并因此得到了很多启发。过去用PHP 4进行面向对象开发时让我苦恼的很多问题，现在都由于Zend Engine 2对面向对象的增强支持而化解于无形。我开始再次为写书做笔记，但这次的内容不是如何克服语言的缺陷，而是如何利用语言的新特性。

我写的书正是你手中的这本书。我尝试把读者想象成我第一次使用“对象”来开发较大项目时的样子。本书要求读者对PHP有一些基本的了解，他们要么了解PHP，要么有能力通过阅读自行掌握PHP的语法特性或功能。当然，面向对象的很多细节对于PHP来说是很新鲜的，我会在本书中完整地介绍。

我以前缺乏的是如何找对感觉去有效使用对象，还有依据什么理由在开发中作出各种选择。同时我也很想知道部署代码的最佳实践，即使用什么工具和原则来测试代码、为代码生成文档和安装代码。

这些都是本书试图阐述的。写这样一本书很有挑战性，我希望你们能有所收获。

Matt Zandstra

2004年11月于英国布莱顿

第2版简介

PHP 5已经发布3年了（恰好本书第1版也是3年前出版的）。在这3年中，面向对象的PHP开发逐渐成为了主流。PHP语言越来越成熟和稳定。同时，PHP相关的测试和集成开发工具也有了明显的进步。

在这3年中，我很高兴收到不少读者来信，他们告诉我本书很有用。有的读者赞叹说本书将混乱的对象、模式、类和组件整理得井井有条，使整个系统变得优雅。我非常高兴，因为这正是我写本书所追求的效果。

大部分计算机图书的生命是很短暂的。即使它们描述的核心概念会持续存在，一旦书上的示例代码不再适用，这些书的作用就会大受影响。这正是我编写本书第2版的原因。

我在最新版本的PHP环境（包括PHP 6）下对本书中的所有代码进行了更新和测试。同时，我也介绍了更多模式，修改了与集成工具相关的很多章节。另外，还新增了PHPUnit测试这一章。我希望第2版能够继续帮助PHP程序员使用对象和模式取得不同凡响的成果。

Matt Zandstra

2007年11月于美国旧金山

目 录

第一部分 介绍

第 1 章 PHP: 设计与管理	2
1.1 问题	2
1.2 PHP和其他语言	3
1.3 关于本书	5
1.3.1 对象	5
1.3.2 模式	5
1.3.3 实践	6
1.3.4 本书第2版中的新内容	7
1.4 小结	7

第二部分 对象

第 2 章 PHP 与对象	10
2.1 PHP对象的偶然成功	10
2.1.1 最初: PHP/FI	10
2.1.2 语法糖: PHP 3	10
2.1.3 一场静悄悄的革命: PHP 4	11
2.1.4 拥抱改变: PHP 5	12
2.2 走向未来: PHP 6	12
2.3 拥护和疑虑: 关于对象的争辩	13
2.4 小结	13
第 3 章 对象基础	14
3.1 类和对象	14
3.1.1 编写第一个类	14
3.1.2 第一个对象 (或两个)	15
3.2 设置类中的属性	15
3.3 使用方法	18
3.4 参数和类型	20
3.4.1 基本类型	20
3.4.2 获得提示: 对象类型	23

3.5 继承	24
3.5.1 继承问题	24
3.5.2 使用继承	28
3.5.3 public、private、protected: 管理类的访问	32
3.6 小结	35

第 4 章 高级特性

4.1 静态方法和属性	36
4.2 常量属性	39
4.3 抽象类	39
4.4 接口	41
4.5 错误处理	43
4.6 Final类和方法	48
4.7 使用拦截器	49
4.8 析构方法	53
4.9 使用 clone()复制对象	54
4.10 定义对象的字符串值	56
4.11 小结	57

第 5 章 对象工具

5.1 PHP和包	58
5.1.1 PHP和命名空间	60
5.1.2 包含路径	62
5.1.3 自动加载	63
5.2 类函数和对象函数	64
5.2.1 查找类	64
5.2.2 了解对象或类	65
5.2.3 了解类中的方法	66
5.2.4 了解类属性	67
5.2.5 了解继承	67

5.2.6 方法调用	68	7.4 为什么使用设计模式	102
5.3 反射API	69	7.4.1 一个设计模式定义了一个问题	102
5.3.1 入门	69	7.4.2 一个设计模式定义了一个解决 方案	102
5.3.2 开始行动	69	7.4.3 设计模式是语言无关的	102
5.3.3 检查类	71	7.4.4 模式定义了一组词汇	102
5.3.4 检查方法	72	7.4.5 模式是经过测试的	103
5.3.5 检查方法参数	74	7.4.6 模式是为协作而设计的	103
5.3.6 使用反射API	75	7.4.7 设计模式促进良好设计	103
5.4 小结	78	7.5 PHP与设计模式	103
第6章 对象与设计	79	7.6 小结	104
6.1 代码设计的定义	79	第8章 模式原则	105
6.2 面向对象设计和过程式编程	80	8.1 模式的启示	105
6.2.1 职责	83	8.2 组合与继承	106
6.2.2 内聚	83	8.2.1 问题	106
6.2.3 耦合	83	8.2.2 使用组合	108
6.2.4 正交	84	8.3 解耦	110
6.3 选择类	84	8.3.1 问题	110
6.4 多态	85	8.3.2 降低耦合	111
6.5 封装	86	8.4 针对接口编程, 而不是针对实现编程	112
6.6 忘记细节	87	8.5 变化的概念	113
6.7 四个方向标	88	8.6 父子关系	113
6.7.1 代码重复	88	8.7 模式	114
6.7.2 类知道的太多	88	8.7.1 用于生成对象的模式	114
6.7.3 万能的类	88	8.7.2 用于组织对象和类的模式	114
6.7.4 条件语句	89	8.7.3 面向任务的模式	114
6.8 UML	89	8.7.4 企业模式	114
6.8.1 类图	89	8.7.5 数据库模式	114
6.8.2 时序图	94	8.8 小结	115
6.9 小结	95	第9章 生成对象	116
第三部分 模式			
第7章 什么是设计模式? 为何使用它们	98	9.1 生成对象的问题和解决方法	116
7.1 什么是设计模式	98	9.2 单例模式	119
7.2 设计模式概览	100	9.2.1 问题	120
7.2.1 命名	100	9.2.2 实现	120
7.2.2 问题	100	9.2.3 结果	122
7.2.3 解决方案	101	9.3 工厂方法模式	122
7.2.4 效果	101	9.3.1 问题	122
7.3 《设计模式》格式	101	9.3.2 实现	125
		9.3.3 结果	126

9.4 抽象工厂模式	127	11.5 命令模式	179
9.4.1 问题	127	11.5.1 问题	179
9.4.2 实现	128	11.5.2 实现	179
9.4.3 结果	129	11.6 小结	183
9.4.4 原型模式	131	第 12 章 企业模式	184
9.4.5 问题	132	12.1 架构概述	184
9.4.6 实现	133	12.1.1 模式	184
9.5 某些模式的骗术	134	12.1.2 应用程序和层	185
9.6 小结	136	12.2 企业架构之外的基础模式	187
第 10 章 让面向对象编程更加灵活的模式	137	12.2.1 注册表	187
10.1 构造可灵活创建对象的类	137	12.2.2 实现	188
10.2 组合模式	137	12.3 表现层	196
10.2.1 问题	138	12.3.1 前端控制器	196
10.2.2 实现	139	12.3.2 应用控制器	205
10.2.3 效果	143	12.3.3 页面控制器	214
10.2.4 组合模式小结	146	12.3.4 模板视图和视图助手	218
10.3 装饰模式	146	12.4 业务逻辑层	220
10.3.1 问题	146	12.4.1 事务脚本	221
10.3.2 实现	148	12.4.2 领域模型	225
10.3.3 效果	151	12.5 小结	227
10.4 外观模式	151	第 13 章 数据库模式	228
10.4.1 问题	152	13.1 数据层	228
10.4.2 实现	153	13.2 数据映射器	228
10.4.3 效果	153	13.2.1 问题	229
10.5 小结	154	13.2.2 实现	229
第 11 章 执行及描述任务	155	13.2.3 效果	239
11.1 解释器模式	155	13.3 标识映射	239
11.1.1 问题	155	13.3.1 问题	240
11.1.2 实现	156	13.3.2 实现	240
11.1.3 解释器的问题	162	13.3.3 效果	242
11.2 策略模式	163	13.4 工作单元	243
11.2.1 问题	163	13.4.1 问题	243
11.2.2 实现	164	13.4.2 实现	243
11.3 观察者模式	167	13.4.3 效果	246
11.4 访问者模式	173	13.4.4 延迟加载	247
11.4.1 问题	174	13.4.5 问题	247
11.4.2 实现	175	13.4.6 实现	247
11.4.3 访问者模式的问题	178	13.4.7 效果	249
		13.5 领域对象工厂	249

13.5.1 问题	249	16.2 安装	290
13.5.2 实现	249	16.3 生成文档	290
13.5.3 效果	250	16.4 DocBlock注释	292
13.6 标识对象	252	16.5 类的文档	293
13.6.1 问题	252	16.6 文件的文档	294
13.6.2 实现	252	16.7 属性的文档	295
13.6.3 效果	256	16.8 方法的文档	296
13.7 选择工厂和更新工厂模式	257	16.9 在文档中创建链接	297
13.7.1 问题	257	16.10 小结	300
13.7.2 实现	257	第17章 使用CVS进行版本控制	301
13.7.3 效果	259	17.1 为什么要使用版本控制	301
13.8 数据映射器中剩下些什么	260	17.2 获得CVS	302
13.9 小结	262	17.3 配置CVS代码库	303
第四部分 实践		17.3.1 创建代码库	303
第14章 良好和糟糕的实践	264	17.3.2 配置客户端	304
14.1 超越代码	264	17.3.3 在多台机器上运行CVS	304
14.2 借一个轮子	264	17.4 开始项目	305
14.3 合作愉快	266	17.5 更新和提交	307
14.4 为你的代码插上双翼	267	17.6 增加和删除文件及目录	309
14.5 文档	267	17.6.1 添加文件	309
14.6 测试	268	17.6.2 删除文件	310
14.7 小结	269	17.6.3 添加目录	310
第15章 PEAR	270	17.6.4 添加二进制文件	311
15.1 什么是PEAR	270	17.6.5 删除目录	311
15.2 安装PEAR包	271	17.7 标记和导出项目	312
15.3 使用PEAR包	273	17.7.1 标记项目	313
15.4 创建自己的PEAR包	278	17.7.2 通过标签获取项目代码	313
15.4.1 package.xml	278	17.7.3 导出项目	314
15.4.2 package.xml的组成元素	278	17.8 创建项目分支	314
15.4.3 contents元素	279	17.9 小结	317
15.4.4 依赖	282	第18章 使用PHPUnit进行测试	318
15.4.5 使用phprelease进行灵活的 自定义安装	283	18.1 功能测试与单元测试	318
15.4.6 发布包	284	18.2 手工测试	319
15.4.7 创建自己的PEAR频道	284	18.3 引入PHPUnit	321
15.5 小结	288	18.3.1 创建测试用例	321
第16章 用phpDocumentor生成文档	289	18.3.2 断言方法	322
16.1 为什么要使用文档	289	18.3.3 测试异常	323
		18.3.4 运行测试套件	324
		18.3.5 约束	326

18.3.6 模拟与桩	328	20.1.2 封装和委托	355
18.3.7 失败是成功之母	331	20.1.3 解耦	355
18.4 警告	333	20.1.4 复用性	356
18.5 小结	335	20.1.5 美学	356
第 19 章 用 Phing 实现项目的自动构建	336	20.2 模式	356
19.1 什么是Phing	336	20.2.1 模式给我们带来了什么	357
19.2 获取和安装Phing	337	20.2.2 模式和设计原则	357
19.3 编写build文档	337	20.3 实践	359
19.3.1 目标	339	20.3.1 测试	359
19.3.2 属性	340	20.3.2 文档	359
19.3.3 类型	345	20.3.3 版本控制	360
19.3.4 任务	348	20.3.4 自动构建	360
19.4 小结	352	20.3.5 我们还遗漏了什么	360
第五部分 结论		20.4 小结	361
第 20 章 对象、模式与实践	354	第六部分 附录	
20.1 对象	354	附录 A 参考文献	364
20.1.1 选择	355	附录 B 简单的解析器	366

Part 1

第一部分

介 绍

本 部 分 内 容

■ 第1章 PHP：设计与管理

2004年初PHP 5发布，它带来的新特性中最重要的是加强了对面向对象编程（OOP）的支持。它使PHP社区开始对“对象”和“设计”产生很大兴趣。事实上，这一趋势从PHP 4支持面向对象编程就已经开始了，而PHP 5使这个过程向前跨跃了一大步。

本章介绍对象编码技术可以满足我们的一些需求。我们将十分简要地总结在Java中模式的发展和相关的实践，然后看看在PHP中类似的过程。

我们还会在本章简单介绍全书涉及的主题。

本章包括以下内容。

- 灾难的演变：项目变糟糕。
- 设计与PHP：面向对象设计技术如何深入影响了PHP社区。
- 本书：对象、模式与实践。

1.1 问题

我们所面对的问题在于PHP太易于使用了！它吸引你马上试验你的想法，并给出令人满意的结果。你可能会直接在Web页面上书写大量的代码，因为PHP本身就有这样的功能；你也可能会在文件中添加工具函数（如数据库访问代码），然后把文件放到页面中去，不知不觉中写出来的Web应用程序也可以正常运行。

你将会陶醉于这条毁灭之路。当然，你并不会意识到这一点，因为你的站点看起来是那么奇妙。站点运行良好，客户也很开心，站点的用户也愿意为之付费。

然而当开始一个新的开发阶段时，再去看看以前的代码，问题就出现了。现在你有了一个更大的开发团队、更多的用户和更充足的预算。在没有任何预警的情况下，事态开始恶化，就像你的项目中中了毒一样。

新加入的程序员开始费力地理解你的代码，虽然对你而言是个几分钟就能搞定的事，即便代码有些绕。她将会花费超出你想象的时间来成为一个合格的团队成员。

要做一个简单的改变，你发现必须更新20多个页面。这样原本估计只要一天就可完成的工作，实际上却用了3天。

一个程序员保存了他对某个文件的修改，而实际上你在较早时候曾对这个文件做了大量修改，于是你修改过的内容全部作废。这个损失在三天内都没有发现，因为你有自己本地的副本，