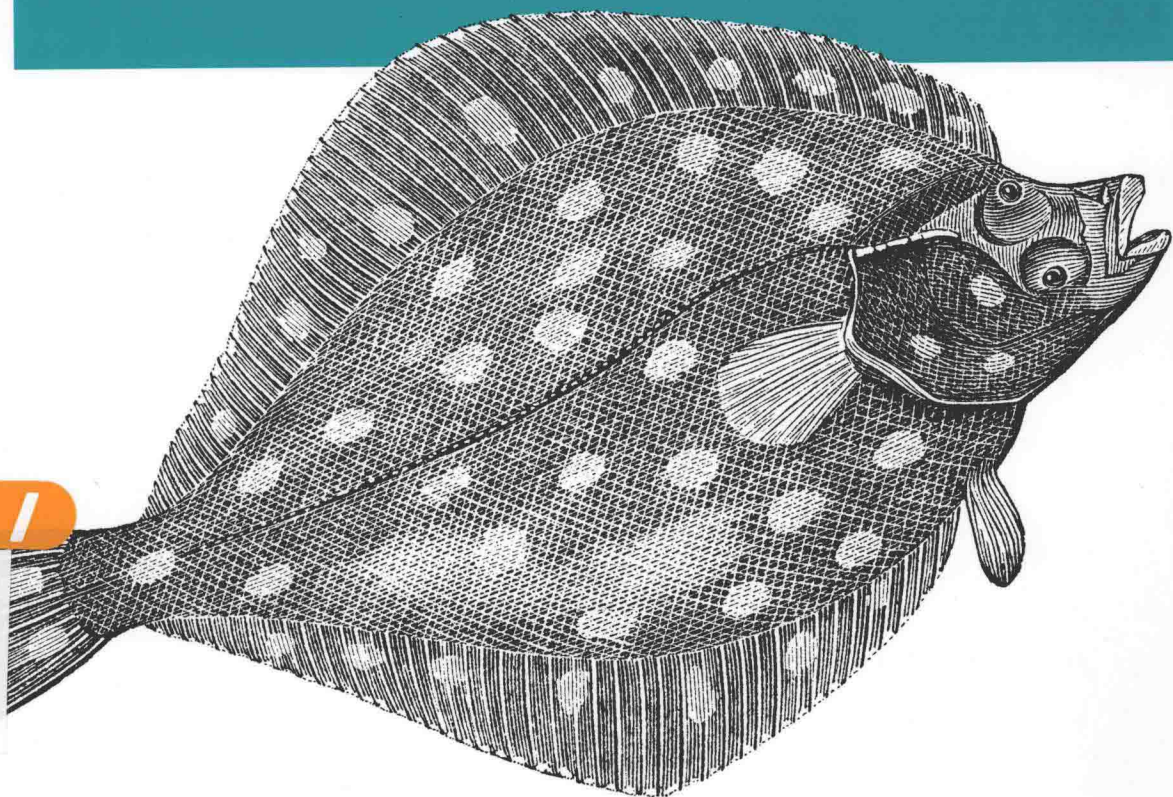


Learning PHP Design Patterns

Learning

PHP设计模式



O'REILLY®

中国电力出版社

William Sanders 著

苏金国 王宇飞 等译

Learning PHP设计模式

William Sanders 著
苏金国 王宇飞 等译

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo

O'Reilly Media, Inc. 授权中国电力出版社出版

中国电力出版社

图书在版编目 (CIP) 数据

Learning PHP设计模式/ (美) 桑德 (Sanders, W.) 著; 苏金国等译. —北京: 中国电力出版社, 2014.2

书名原文: Learning PHP Design Patterns

ISBN 978-7-5123-5272-8

I. ①L… II. ①桑… ②苏… III. ①PHP语言—程序设计 IV. ①TP312

中国版本图书馆CIP数据核字 (2013) 第285791号

北京市版权局著作权合同登记

图字: 01-2013-6930号

©2013 by O'Reilly Media, Inc.

Simplified Chinese Edition, jointly published by O'Reilly Media, Inc. and China Electric Power Press, 2012.
Authorized translation of the English edition, 2013, O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由O'Reilly Media, Inc. 出版2013。

简体中文版由中国电力出版社出版2013。英文原版的翻译得到O'Reilly Media, Inc.的授权。此简体中文版的出版和销售得到出版权和销售权的所有者——O'Reilly Media, Inc.的许可。

版权所有。未得书面许可, 本书的任何部分和全部不得以任何形式重制。

- 封面设计/ Karen Montgomery, 张健
出版发行/ 中国电力出版社 (<http://www.cepp.sgcc.com.cn>)
地 址/ 北京市东城区北京站西街19号 (邮政编码100005)
经 销/ 全国新华书店
印 刷/ 北京丰源印刷厂
开 本/ 787毫米×980毫米 16开本 21印张 385千字
版 次/ 2014年2月第一版 2014年2月第一次印刷
印 数/ 0001—3000册
定 价/ 58.00元 (册)

敬告读者

本书封底贴有防伪标签, 刮开涂层可查询真伪
本书如有印装质量问题, 我社发行部负责退换

版权专有 翻印必究

Learning PHP设计模式

O'Reilly Media, Inc.介绍

O'Reilly Media通过图书、杂志、在线服务、调查研究和会议等方式传播创新知识。自1978年开始，O'Reilly一直都是前沿发展的见证者和推动者。超级极客们正在开创着未来，而我们关注真正重要的技术趋势——通过放大那些“细微的信号”来刺激社会对新科技的应用。作为技术社区中活跃的参与者，O'Reilly的发展充满了对创新的倡导、创造和发扬光大。

O'Reilly为软件开发人员带来革命性的“动物书”；创建第一个商业网站（GNN）；组织了影响深远的开放源代码峰会，以至于开源软件运动以此命名；创立了Make杂志，从而成为DIY革命的主要先锋；公司一如既往地通过多种形式缔结信息与人的纽带。O'Reilly的会议和峰会集聚了众多超级极客和高瞻远瞩的商业领袖，共同描绘出开创新产业的革命性思想。作为技术人士获取信息的选择，O'Reilly现在还将先锋专家的知识传递给普通的计算机用户。无论是通过书籍出版，在线服务或者面授课程，每一项O'Reilly的产品都反映了公司不可动摇的理念——信息是激发创新的力量。

业界评论

“O'Reilly Radar博客有口皆碑。”

——Wired

“O'Reilly凭借一系列（真希望当初我也想到了）非凡想法建立了数百万美元的业务。”

——Business 2.0

“O'Reilly Conference是聚集关键思想领袖的绝对典范。”

——CRN

“一本O'Reilly的书就代表一个有用、有前途、需要学习的主题。”

——Irish Times

“Tim是位特立独行的商人，他不光放眼于最长远、最广阔的视野并且切实地按照Yogi Berra的建议去做了：‘如果你在路上遇到岔路口，走小路（岔路）。’回顾过去Tim似乎每一次都选择了小路，而且有几次都是一闪即逝的机会，尽管大路也不错。”

——Linux Journal

谨以此书纪念我的父亲William B. Sanders (1917–2012)。

目录

前言	1
----------	---

第1部分 轻松掌握设计模式基础

第1章 PHP与面向对象编程	13
----------------------	----

1.1 中级和高级编程初探	13
1.2 为什么采用面向对象编程	14
1.2.1 解决问题更容易	14
1.2.2 模块化	15
1.3 类与对象	16
1.3.1 单一职责原则	16
1.3.2 PHP中的构造函数	17
1.4 客户类作为请求者	17
1.5 速度如何	21
1.5.1 开发和修改速度	21
1.5.2 团队速度	21
1.6 顺序和过程式编程有什么问题	22
1.6.1 顺序编程	22
1.6.2 过程编程	22
1.6.3 即时回报还是长期回报	23

第2章 OOP基本概念	25
2.1 抽象	25
2.1.1 抽象类	26
2.1.2 抽象属性和方法	27
2.1.3 接口	29
2.1.4 接口和常量	31
2.1.5 类型提示：类似数据类型	32
2.2 封装	34
2.2.1 日常生活中的封装	35
2.2.2 通过可见性保护封装	35
2.2.3 获取方法和设置方法	38
2.3 继承	39
2.4 多态	41
2.4.1 一个名字，多个实现	43
2.4.2 设计模式中的内建多态性	44
2.5 慢慢来	44
第3章 基本设计模式概念	46
3.1 MVC实现编程松耦合和重新聚焦	46
3.2 设计模式基本原则	48
3.2.1 第一个设计模式原则	49
3.2.2 代码提示中使用接口数据类型	50
3.2.3 抽象类及其接口	51
3.2.4 第二个设计模式原则	53
3.2.5 使用客户的基本组合	54
3.2.6 委托：IS-A和HAS-A的差别	58
3.3 设计模式作为备忘录	58
3.4 选择设计模式	59
3.4.1 是什么导致了重新设计	60
3.4.2 什么会变化	60
3.4.3 设计模式与框架有什么区别	61

第4章 结合使用设计模式和UML	62
4.1 为什么是统一建模语言 (UML)	62
4.2 类图	63
4.3 参与者符号	64
4.4 关系说明	66
4.4.1 相识关系	66
4.4.2 聚合关系	68
4.4.3 继承和实现关系	70
4.4.4 创建关系	72
4.4.5 多重关系	73
4.5 对象图	73
4.6 交互图	74
4.7 面向对象编程中图和记法的作用	75
4.8 UML工具	75
4.9 其他UML	76

第2部分 创建型设计模式

第5章 工厂方法设计模式	81
5.1 什么是工厂方法模式	81
5.2 何时使用工厂方法	82
5.3 最简单的例子	83
5.3.1 工厂的工作	83
5.3.2 客户	86
5.4 适应类的修改	86
5.4.1 增加图像元素	86
5.4.2 调整产品	87
5.4.3 修改文本产品	89
5.4.4 修改图像产品	90
5.4.5 增加新产品和参数化请求	91
5.4.6 一个工厂多个产品	92
5.4.7 新工厂	92

5.4.8 新产品	93
5.4.9 有参数的客户	95
5.4.10 辅助类	96
5.4.11 文件图	97
5.4.12 产品改变：接口不变	97
第6章 原型设计模式.....	100
6.1 原型设计模式	100
6.2 何时使用原型模式	101
6.3 克隆函数	102
6.3.1 克隆不会启动构造函数	103
6.3.2 构造函数不要做具体工作	104
6.4 最简单的原型例子	105
6.5 为原型模式增加OOP	108
6.5.1 现代企业组织	109
6.5.2 接口中的封装	109
6.5.3 接口实现	111
6.5.4 组织客户	113
6.5.5 完成修改，增加特性	115
6.5.6 动态对象实例化	116
6.6 PHP世界中的原型	118
第3部分 结构型设计模式	
第7章 适配器模式.....	123
7.1 什么是适配器模式	123
7.2 何时使用适配器模式	124
7.3 使用继承的适配器模式	126
7.4 使用组合的适配器模式	130
7.4.1 从桌面环境转向移动环境	131
7.4.2 适配器和变化	139

第8章 装饰器设计模式 140

8.1 什么是装饰器模式	140
8.2 何时使用装饰器模式	141
8.3 最简单的装饰器例子	142
8.3.1 Component接口	142
8.3.2 Decorator接口	143
8.3.3 具体组件	144
8.3.4 具体装饰器	145
8.3.5 客户	146
8.4 关于包装器	148
8.4.1 包装器包装基本类型	148
8.4.2 PHP中的内置包装器	148
8.4.3 设计模式包装器	149
8.5 包装多个组件的装饰器	150
8.5.1 多个具体组件	150
8.5.2 包含多个状态和值的具体装饰器	151
8.5.3 开发人员约会服务	151
8.6 HTML用户界面	157
8.6.1 Client类传递HTML数据	161
8.6.2 从变量名到对象实例	162
8.6.3 增加装饰	162

第4部分 行为型设计模式

第9章 模板方法模式 167

9.1 什么是模板方法模式	167
9.2 何时使用模板方法	168
9.3 最简单的例子：对图像和图题使用模板方法模式	169
9.3.1 抽象类	169
9.3.2 具体类	170
9.4 客户	170
9.5 好莱坞原则	171

9.6 结合其他设计模式使用模板方法模式	173
9.6.1 客户工作负担减轻	174
9.6.2 模板方法参与者	174
9.7 工厂方法参与者	175
9.8 模板方法设计模式中的钩子	178
9.8.1 建立钩子	180
9.8.2 实现钩子	181
9.8.3 客户以及捕获钩子	182
9.9 短小精悍的模板方法模式	183
第10章 状态设计模式	185
10.1 什么是状态模式	185
10.2 何时使用状态模式	186
10.3 状态机	187
10.4 开灯关灯：最简单的状态设计模式	188
10.4.1 情境为王	188
10.4.2 状态	191
10.4.3 客户通过上下文做出请求	193
10.5 增加状态	194
10.5.1 改变接口	194
10.5.2 改变状态	195
10.5.3 更新Context类	197
10.5.4 更新客户	199
10.6 导航工具：更多选择和单元格	199
10.6.1 建立一个矩阵状态图	200
10.6.2 建立接口	201
10.6.3 上下文	202
10.6.4 状态	204
10.6.5 客户选择一条路径	210
10.7 状态模式与PHP	212

第5部分 MySQL和PHP设计模式

第11章 通用类负责连接，代理模式保证安全 215

11.1 一个简单的MySQL接口和类	215
11.1.1 重要的接口	216
11.1.2 通用MySQL连接类和静态变量	216
11.1.3 简单客户	218
11.2 保护代理完成登录	219
11.2.1 建立登录注册	220
11.2.2 实现登录代理	224
11.3 代理和真实世界安全	230

第12章 策略设计模式的灵活性 232

12.1 封装算法	232
12.1.1 区分策略和状态设计模式	233
12.1.2 请不要加条件语句	234
12.1.3 算法族	234
12.2 最简单的策略模式	235
12.2.1 客户和触发器脚本	235
12.2.2 Context类和Strategy接口	239
12.2.3 具体策略	240
12.3 增加数据安全性 and 参数化算法来扩展策略模式	243
12.3.1 数据安全性辅助类	243
12.3.2 为算法方法增加参数	246
12.3.3 调查表	246
12.3.4 数据输入模块	248
12.3.5 客户请求帮助	253
12.3.6 Context类重要的小改变	254
12.3.7 具体策略	255
12.4 灵活的策略模式	260

第13章 职责链设计模式	262
13.1 推卸责任	262
13.2 MySQL咨询台中的职责链	264
13.2.1 构建和加载响应表	264
13.2.2 咨询台职责链	269
13.3 自动职责链和工厂方法	274
13.3.1 职责链和日期驱动请求	275
13.3.2 工厂方法完成任务	278
13.4 易于更新	284
第14章 利用观察者模式构建多设备CMS.....	285
14.1 内置观察者接口	285
14.2 何时使用观察者模式	286
14.3 使用SPL实现观察者模式	287
14.3.1 SplSubject	288
14.3.2 SplObserver	289
14.3.3 SplObjectStorage	289
14.3.4 SPL具体主题	289
14.3.5 SPL具体观察者	291
14.3.6 SPL客户	291
14.4 自由的PHP和观察者模式	293
14.4.1 抽象Subject类和ConcreteSubject实现	293
14.4.2 观察者和多个具体观察者	294
14.4.3 客户	296
14.5 建立一个简单CMS	298
14.5.1 CMS工具	298
14.5.2 多个设备观察者	303
14.6 用OOP方式思考	315

前言

随着PHP日益成为很多程序员首选的服务器端程序，将专业技术和编程结构相结合已经势在必行。设计模式（Design patterns）这一概念借用自Christopher Alexander的《The Timeless Way of Building》（牛津大学出版社）一书，是指对给定上下文（环境）中某个经常出现的问题得出的一种一般性可重用的解决方案。在日常的开发工作中，PHP程序员总会在某种软件开发环境中遇到“经常出现”的问题，PHP设计模式就是针对这些“经常出现”的PHP编程问题提出的一组解决方案。简单地说，PHP设计模式是用来处理专业软件开发现实问题的工具。

它们并不是具体的库或模板，而是可以用来解决问题的更为一般性的结构。我总喜欢把设计模式想成是建立循环结构。需要处理某种迭代时就会使用循环。没错，当然也可以用其他方法来处理迭代，不过循环确实是一个非常灵活的工具，可以大大节省开发过程的时间。（你是愿意编写10000次相同的代码行还是更愿意使用循环呢？相比之下，使用循环实在是简洁得多）！

另外，如果说循环是迭代的一种“罐装解决方案”，可以认为设计模式也是一种“罐装解决方案”。在PHP中可以采用多种不同方式使用循环结构，如for语句、while语句以及其他类似的结构。同样，设计模式也可以采用多种不同的方式实现，这取决于所要解决的问题的本质。

不过，之所以采用设计模式，最重要的原因在于，这些模式可以为复杂的问题提供解决方案。随着开发的程序变得越来越庞大，毋庸置疑，它们也会变得越来越复杂。在面向对象编程（object-oriented programming, OOP）环境中，这种复杂性会有所降低，因为你处理的是封装的模块，而在顺序或过程式编程中，做出任何改变都有可能导致程序像一摞纸牌一样坍塌。设计模式不仅为一般性编程问题提供了解决方案，而且通过提供对

象之间的松耦合还支持对复杂的大程序做出修改。所以如果要进行修改，并不需要从头开始重新编程，即使是很庞大、很复杂的程序，也完全可以只增加必要的改动，而其他一切均保持不变。

另外，设计模式的宗旨就是重用。毕竟，程序员一直在重用同样的算法。那么为什么不按照同样的思路重用更大的结构呢？一方面，框架和模板可以支持重用，不过它们往往过于特定。这就引入了另一种做法：可以重用结合设计模式的PHP程序，特别是在庞大的复杂程序中。由于结合设计模式的程序可以很容易地做出修改，因此对相同类型的特定问题重用这些程序也很容易。减少开发时间和资源不仅可以节省成本，还可以更好地为你的客户服务。客户将得到能充分满足其功能需求的良构程序，另外开发人员也可以基于不太可能崩溃的坚实基础很容易地完成修改（毕竟，顾客的需求总是在不断改变！）。

本书面向对象

在某种程度上，所有优秀的程序员都会意识到需要跳出顺序和过程式编程的束缚，而转向下一种更合理的编程方式，这就是面向对象编程。要转向OOP，需要从观念上有所转变：并不是把编程看做是一系列语句，而应当看做是对象之间的一种交互和通信。设计模式的基础便是OOP，也就是说，OOP原则将转换为可重用代码模式。这些正是很多专业程序员使用的工具。由于编程设计模式的开发需要学术领域和商业领域的精诚合作，这些概念并不仅限于针对某一个问题，不过同时也确实可以用来处理实际问题。《PHP设计模式学习指南》面向的是专业的程序员，他们希望进一步节省其开发和再开发时间，并为客户提供高质量的代码。

特别需要指出，这本书对那些对编程充满热情的人也非常适用。他们可能认为整晚编程都不算什么，因为他们认为这很有意思，上床睡觉只是为了醒来后可以开始另一个程序。如果编程总是全新的，每天都会有新的发现或bug，开发人员就必须全力拼搏，需要用复杂而新颖的方式充分运用大脑，这种体验有点像悟禅。如果你有这种体验，应该懂得我的意思。这没有办法说清楚，也不好解释。（我自己甚至都无法解释清楚，我也不知道为什么会对编程中遇到的挑战和激励如此着迷。）

设计模式对理解力提出了更高的挑战，这本书并不适合那些对PHP和编程都很陌生的人。如果你刚开始学习PHP，可以看一看Robin Nixon的《Learning PHP, MySQL, JavaScript, and CSS, 2nd Edition》（O'Reilly），然后再来考虑PHP设计模式。另外，这本书（或所有关于设计模式的高深的书）无法承诺你能很快、很容易地掌握设计模式。这种学习是一个漫长的旅程，最明智的建议就是学着享受这个旅程。这需要花许多时间和功夫。

本书假设

这本书假设你已经知道如何用PHP编程，而且希望你的编程水平能更上几个台阶。实际上，这里假设你已经是一个很好的PHP程序员，而且用过MySQL，知道如何开发HTML页面和使用CSS。这里还假设你知道学习PHP设计模式绝不可能一蹴而就。设计模式的学习就像是一个渐近完成的蜕变。

本书内容

这本书分为5大部分。

第1部分是对OOP的一个复习/介绍：

第1章介绍面向对象编程（object-oriented programming, OOP），以及如何利用模块化更容易地处理复杂的编程问题。

第2章讨论OOP中的一些基本概念，如抽象、封装、继承和多态，以及实现这些概念的PHP结构。

第3章继续讨论设计模式中的基本概念、设计模式类别以及如何选择特定的模式来处理特定的问题。

第4章介绍统一建模语言（Unified Modeling Language, UML），并解释这本书将如何应用UML。

第2部分介绍创建型设计模式：

第5章介绍工厂方法（Factory Method）模式，其目的是创建对象，这种模式属于类设计模式。这一章提供的例子包括动态创建可以显示图片、正文体和标题正文的页面。

第6章介绍如何使用原型（Prototype）模式，其目的也是创建对象，这种模式属于对象设计模式。如果创建一个对象作为原型，然后通过克隆来创建更多实例以节省开销，这种情况就可以使用原型模式。

第3部分解释结构型设计模式：

第7章介绍如何使用类和对象适配器（Adapter）模式。这一章给出的例子展示了如何对现有的结构进行修改，从而允许开发人员增加新的功能。

第8章解释如何利用装饰器（Decorator）模式改变原有的对象而不会对更大的程序带来破坏。你会看到如何用约会网站上的不同首选项装饰男性和女性约会对象。

第4部分介绍行为型设计模式：

第9章介绍如何使用模板方法（Template Method）模式——这也是最容易创建和使