

Pro PHP Programming

PHP编程实战

[加拿大] Peter MacIntyre
Brian Danchilla 著
[美] Mladen Gogala
王爱华 姚军强 译

- PHP核心缔造者Suraski作序推荐
- 实例丰富，操作性强

TURING 图灵程序设计丛书

Pro PHP Programming

PHP编程实战

[加拿大] Peter MacIntyre
Brian Danchilla 著
[美] Mladen Gogala
王爱华 姚军强 译

人民邮电出版社
北京

图书在版编目 (C I P) 数据

PHP编程实战 / (加) 麦金太尔 (MacIntyre, P.),
(加) 丹奇利亚 (Danchilla, B.), (美) 贡加拉
(Gogala, M.) 著; 王爱华, 姚军强译. — 北京: 人民
邮电出版社, 2014.10

(图灵程序设计丛书)

ISBN 978-7-115-36658-0

I. ①P… II. ①麦… ②丹… ③贡… ④王… ⑤姚…
III. ①PHP语言—程序设计 IV. ①TP312

中国版本图书馆CIP数据核字(2014)第177741号

内 容 提 要

本书主要包括: PHP 5.3 面向对象编程、PHP 的一些新特性、如何编写基于 SQL 和 NoSQL 的数据库交互程序、如何使用流行的 PHP 相关工具、如何编写社交媒体应用程序, 并介绍了 NoSQL 数据存储以及为云生态编写 PHP 程序。

本书适合中高级 PHP 程序员以及想学习 PHP 的有一定基础的程序员。

-
- ◆ 著 [加拿大] Peter MacIntyre Brian Danchilla
[美] Mladen Gogala
译 王爱华 姚军强
责任编辑 朱 巍
执行编辑 李岩俨
责任印制 焦志炜
- ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
三河市海波印务有限公司印刷
- ◆ 开本: 800×1000 1/16
印张: 22.75
字数: 538千字 2014年10月第1版
印数: 1-3 000册 2014年10月河北第1次印刷
- 著作权合同登记号 图字: 01-2011-7471号



定价: 69.00元

读者服务热线: (010)51095186转600 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京崇工商广字第 0021 号

版权声明

Original English language edition, entitled *Pro PHP Programming* by Peter MacIntyre, Brian Danchilla, Mladen Gogala, published by Apress, 2855 Telegraph Avenue, Suite 600, Berkeley, CA 94705 USA.

Copyright © 2011 by Peter MacIntyre, Brian Danchilla and Mladen Gogala. Simplified Chinese-language edition copyright © 2014 by Posts & Telecom Press. All rights reserved.

本书中文简体字版由Apress L.P.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。

鉴于我别的书献给了妻子和孩子，所以我想把这本书献给 PHP 社区上所有那些保持这个语言与时俱进、健全和不断壮大的人。同时献给开源社区及其理想和观念，希望这种精神可以一直持续下去！

——Peter

献给我的父亲和母亲。

——Brian

献给我钟爱的妻子和儿子。

——Mladen

序

PHP 没有什么高贵的出身,而是始于一个黑客项目——试图以一种容易和快乐的方式来开发网站,没人料到它会像今天这样流行。

多年以来,人们曾从多个角度衡量 PHP 的受欢迎度,比如使用 PHP 部署的网站数量、亚马逊上 PHP 书籍的销量、使用 PHP 的杰出公司的数量、基于 PHP 的项目数量、创立社团的规模,等等。

不过,衡量 PHP 的受欢迎程度还有另一种方式,一种不怎么“科学”的角度。

2008 年,我和妻子 Anya 去布宜诺斯艾利斯度蜜月。经过了长途飞行后,我们抵达了这个陌生的国家,这里充满了新面孔和我们从未见过的事物。我们住在名为 Noster Bayres 的小旅馆中,在填完旅店登记表后,招待员问我是否是 Suraski,“就是那个 PHP 小子”,想象一下我有多吃惊吧。原来他当时在用 PHP 为圣特尔莫的邻居开发网络社区。

虽然之前所有的衡量角度都无可辩驳地证明了 PHP 的极致功能、重要性和受欢迎程度,但对我而言,却是在世界另一端的那个小旅馆中发生的插曲让我肯定了这些观点。如果连旅馆的接待员都在用 PHP,那它当然是主流技术。

大约 3 年后,任何 Web 开发人员都必须具备高级 PHP 的编程技巧,而且可以说,随着 Web 以及基于 HTTP 的通信的爆炸式增长,所有开发人员都必须有这种本领。本书通过更多先进的现代 PHP 开发理念为你指路,包括面向对象、移动应用开发,以及对云计算支持很重要的可伸缩数据源等。相信读者在本书中获得的知识有助于充分利用 PHP 的先进功能,并且使它成为前行中重要的工具。祝你 PHP 之旅愉快!

Zeev Suraski, PHP 核心缔造者之一, Zend 公司 CTO

致 谢

我要感谢 Apress 的 Frank Pohlmann 和 Jessica Belanger，是他们指导了本书的写作并将其献给 PHP 社区。由于我已为各大出版社编写了多本 PHP 书籍，所以起初接到这个额外的写作任务时，我还颇不情愿，但在 Frank 的鼓励和劝说下，我还是把这个任务给接了下来。感谢他的鼓舞以及为我提供的这次良机。在此过程中，我们也成为了好朋友。在此项目中，我所收获的这份友情要比单纯编写更多 PHP 代码显得更有意义。

技术编辑 Thomas Meyer 和加工编辑 Tracy Brown 也做了大量的工作，对此，请允许我脱帽向你们致敬。

感谢我的合著者们！在这次绝妙的写作历程中，我从你们每个人那里学到了很多东西，在各方面都有所进步。与来自不同国家、背景与经历的作者一道工作，正是一种令人愉悦与成长的体验。

Peter MacIntyre

衷心感谢我的爱侣 Tressa，她在我编写本书时一直在支持和鼓励我。衷心感谢我的父母、哥哥 Robert 和姐姐 Karen，他们一直都非常信任我和我所做的事——即使他们根本不懂得我所做的。我还要衷心感谢我的合著者 Peter 和 Mladen，以及整个 Apress 团队。

Brian Danchilla

多年来，我从和我一起工作过的人们身上学到很多，我万分感谢他们，特别是来自 Video Monitoring Services 的同事 Vinod Mummidi 和 Arik Itkis。我们在一起的时候，经常会不停地讨论 PHP 概念。我的经理 Gerry Louw 也非常支持并协助我的工作。我也衷心感谢我的协作编辑，Apress 的 Jessica Belanger。她的热情和专业指导让我动容，没她就没有本书。另外，还要感谢我的合著者 Peter MacIntyre 和 Brian Danchilla。在本书写作过程中，他们孜孜不倦地校对草稿，并对本书的重点内容建议颇多。我也衷心感谢 Tom Welsh 和 Tim Hawkins，他们为本书付出了很多精力，并提供了很多好的想法，其中有很多已经被采纳进本书中。最后，我要感谢两位至关重要的人，我的妻子 Beba 和儿子 Marko。他们在本书编写过程中一直给予我无限的关爱、支持与耐心。

Mladen Gogala

PHP简介

欢迎阅读另一本关于伟大编程语言 PHP 的书。这本书的独特之处，在于它着眼于最高端知识以及更高级、更前沿的主题。我们竭尽所能使它与现代快速发展的互联网世界保持同步。我们衷心地希望本书的读者能从中级水平成长为更高水平的 PHP 工程师。

PHP的起源

PHP 起始于 Rasmus Lerdorf 先生领导和设计的一个项目。1995 年 6 月，他发布了 1.0 版的个人主页工具包（Personal Home Page Tools——PHP 最初的名字）。起初，PHP 是一套很小的函数集，用来自动创建和维护当时新兴互联网的简单主页。从那时起，PHP 开始迅速发展。PHP 是从一开始就开源的开发编程语言之一。Lerdorf 非常有远见，他看到了这个工具和语言的必要性和潜力，它将随着互联网社区一起发展，甚至前程更为远大。

PHP是什么

那么，究竟什么是 PHP？就目前的版本来说，它用起来怎么样？简单来说，PHP 只不过是 HTML 标记生成器。如果看一下 PHP 生成网页的源代码，你只会看到 HTML 标签，也许还有一些 JavaScript 脚本，但是没有纯 PHP 代码。当然，这样观察一门语言过于简单，如今，大约有 35%~59%（数据来源不同）的 Web 开发活动在使用 PHP。无论你采信哪个数据，在今天的市场上，PHP 都是最受欢迎的 Web 开发语言。

别看我使用了“在市场上”这个词，你必须明白 PHP 是免费的。没错，免费的！它是一款开源产品，因此，现实中它没有实际的市场。所以，在没有任何实体及个人领导和监督的产品中，它表现得十分抢眼，非常流行，使用极为广泛。

注意 要想了解更多开源信息，请阅读 Eric S. Raymond 的著作《大教堂与集市》，看他所做的关于开源产品（集市）和闭源产品（教堂）的比较。你可以在 www.catb.org/~esr/writings/cathedral-bazaar/ 上找到相关信息。

实际上，Zend 公司（zend.com）是 PHP 领域的领导者，创建了很多辅助产品支持和增强 PHP，Zend 也是 PHP 发展方向的指导者。这是因为，Zend 公司的两位创始人——Zeev Suraski 和 Andi

Gutmans——从 PHP 第 3 版开始就接管了 PHP 的研发。

PHP 的语言结构非常开放、宽容，它是类型松散的（当然还有更多特性）。这就意味着变量不必定义数据类型，不像一些其他语言，必须在使用前先定义变量类型。相反，它会根据运行时变量的内容来查询数据，试图确定变量的数据类型。例如一个变量`$information`，在代码文件执行时可能有很多不同的值。这点在某些方面可能是缺点，因为在代码运行时数据发生了改变就会造成无效代码段，这段代码本想接收一个整数变量，却接收了一个字符串。

PHP 也可以用面向对象的编程（OOP）设计方式来写程序类、属性和方法；继承、多态和封装也都是 PHP 语言所支持的特性。这就增加了代码的健壮性、可重用性和代码整体上的易用性。当然，在技术领域，面向对象的编程方法已经发展了很长时间，并且集成于 PHP 里也已经很多年了。

PHP 另一个有价值的特征是 PHP 进程，它可以在命令行运行（Linux 或 Windows），因此可以用来做无人看管（CRON）的计划脚本。这提高了它的灵活性，因为当程序员在服务器环境中工作时，就不必学习另一种语言来完成不同的任务了。你可以用相同的语言来开发网页，也可用它来管理文件系统（如果你这样选择）。

PHP 还有很多亮点，至少它是一门非常开放的语言。PHP 可以用来做很多事情，不仅仅局限于 Web 开发。通过合适的连接库把它和数据库连接起来，你就可以创建一个动态网页，甚至一个网络应用了。把它和另外的库（例如 `tcpdf` 库）组合起来，你就可以动态地创建 Adobe PDF 文档了。这仅仅是两个例子，本书将讲解很多类似的附加函数库，敬请关注！

本书概览

那么，通过这本书，我们希望为读者，为程序员提供什么？我们将尽最大的努力，使书中内容保持最新、最前沿，以便你能够使用 PHP 最新的功能以及 PHP 集成模块。我们没有花太多的时间来讲解 PHP 语言的简单主题，比如：什么是变量或如何写 `for/next` 循环。

因此，我们希望你能成为一名高级 PHP 程序员，本书内容甚至可以用于备考 Zend 认证工程师。以下是各章的简要介绍。

第 1 章：面向对象

本章将为你讲解许多概念和代码示例，这些概念和示例将在本书后面详细讲解。我们先介绍一些面向对象的概念以及如何在 PHP 中实现它们，然后学习一些更深入的问题。在学习其他章节前，你应真正理解本章的内容。

第 2 章：异常和引用

本章继续讲面向对象的概念，并深入学习 `try/catch` 异常编码。这是在 PHP 代码中处理异常的一种更优雅的方式，一旦熟练掌握，你就有了一个相当强大的处理错误的方法。接下来将讨论引用编码，以及这对使用的类和函数有什么影响。

第 3 章：PHP 在移动设备上的应用

这个世界越来越离不开移动设备，总有更小却更强大的设备发布。Apple、RIM、HTC 和很

多其他厂商都在努力，试图在这个有利可图的市场中提高其份额。但这些设备需要可用的应用，因此，本章我们将讲 PHP 为迎合移动开发而成长和改进的方面。

第 4 章：PHP 在社交媒体上的应用

由于 PHP 的大力协助，社交媒体也取得了急速扩张。例如，最为领先的 Facebook 是用 PHP 开发的。还有许多其他网站，如 Flickr、雅虎的部分模块，甚至很多博客应用主要都依赖于 PHP。在这一章中，我们研究一些接口，这些接口用来与上述社交网站集成。

第 5 章：前沿

本章将要研究一些最新的功能以及如何 Web 项目中使用它们。

第 6 章：表单设计与管理

在本章，我们将多花点时间来研究如何实现数据录入表单的设计和管理的功能和技术。处理输入的数据，并响应返回坏数据（例如无效的日期格式），以及如何优雅地将数据存入 Web 系统。

第 7、8 章：数据库集成

当然，目前 Web 开发的主要方面之一就是数据存贮和显示能力，这些数据源于同一个数据源。在这两章中，我们将研究许多不同的数据操纵方式。从小范围使用的数据库，像各种各样的 NoSQL，到 MySQLi 这样的铁杆数据库引擎，以及我们收集的其他工具，例如 PDO 和 Sphinx 的技术。

第 9 章：数据库集成——Oracle

当涉及超级大型数据集时，我们会把 PHP 和 Oracle 连接起来一起使用。在本章中，我们将研究它们之间的关系以及它们之间是如何“联合”的。

第 10 章：函数库

正如我们前面提到的，PHP 非常开放，能与其他库一起工作。在本章中，我们将学习一些非常流行和高级的函数库。本章将通过使用很少的函数库来实现动态生成 PDF 表单，使用 RSS 聚合，生成专业的 E-mail 以及集成 Google 地图。

第 11 章：安全

当然，如果不讨论 Web 安全的最新技术，本书将是一本不完整的书。本章介绍这个大主题。我们将研究目前最安全的加密算法 SHA-1。本章还设计其他保护数据的主题，像保护从 Web 系统外输入到 Web 系统的数据。

第 12 章：使用 Zend Studio 进行敏捷开发

本章是一个与 PHP 擦边的主题，它不单是一个 PHP 主题。在这里，我们要讨论一下如何使用一个最流行的 PHP 集成开发环境 Zend Studio for Eclipse。使用 Zend Studio，我们将研究一组开发者如何通过敏捷的方式（你听说过极限编程吗）一起工作。我们将讨论 SVN、Bugzilla 和 MyLyn 配合使用，以便使团队开发在许多方面都非常高效。

第 13 章：重构、单元测试和持续集成

本章实际上是前一章的延伸，更多地讲了如何使 PHP 开发更加敏捷的内容。重构和单元测试是本章的重点，你将学习如何把这两点更好地用在编码项目中。

第 14 章：XML 和 PHP

使用 XML 无疑成为多年来的主流，因此它也第一次成为流行语。在本章，我们将讨论如何

使用 SimpleXML 从外部源取得 XML。此外，我们从自己的系统内生成 XML 数据，供他人使用。

第 15 章：JSON 和 Ajax

同样，通过以下两方面我们从纯 PHP 又前进了一小步：使用 JSON 库；如何使 JSON 库和 Ajax 配合，使 Web 应用响应更加快速。

第 16 章：结束语

在最后一章，我们来看看那些没有包含在此书中的 PHP 资源，比如大量的 Web 可用资源、杂志和会议，这些资源都可以扩展知识面，让你更深入地理解这门伟大语言和 PHP 社区。

PHP 的未来

这是一个很难写的话题。由于 PHP 是一款真正的开源产品，所以很难预测社区在近期和遥远的将来何去何从。但是凭心而论，我对这个社区很有信心。多年来，我一直是一名 PHP 程序员，从未看到这个集体有何方向性错误。我知道生活中的移动领域将继续增长和快速扩展，PHP 已经采取措施来完全迎合这一事实。在不久的将来会发生什么？也许电话、智能手机与数据的互操作性将集成融合在一起；也许是会深入扩展到语音识别技术和网络应用——谁知道呢？依据我的经验，可以确定的是，到目前为止，PHP 以及其支持社区指引技术世界的脉搏，他们不会让我们失望。展望 PHP 的未来是一件令人欣慰的事，它就像一轮冉冉升起的太阳，预示着更加美好的未来。

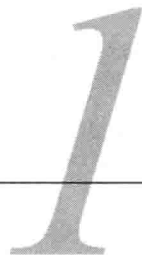
目 录

第 1 章 面向对象.....1	3.5.1 Android 开发.....47
1.1 类.....1	3.5.2 支持 PHP 的 Adobe Flash Builder.....47
1.2 继承和重载.....3	3.6 QR 码.....48
1.3 各种“魔术方法”.....7	3.7 总结.....49
1.3.1 __get 和 __set 方法.....7	第 4 章 PHP 在社交媒体中的应用.....50
1.3.2 __isset 方法.....8	4.1 OAuth.....50
1.3.3 __call 方法.....9	4.2 Twitter.....51
1.3.4 __toString()方法.....9	4.2.1 公有的搜索 API.....51
1.4 复制、克隆和比较对象.....10	4.2.2 私有 REST API.....52
1.5 接口、迭代器和抽象类.....12	4.2.3 使用 Twitter OAuth 连接网站 的登录.....63
1.6 类的作用域与静态成员.....16	4.2.4 更多的 API 方法和示例.....67
1.7 总结.....18	4.3 Facebook.....70
第 2 章 异常和引用.....19	4.3.1 为退出 Facebook 添加链接.....75
2.1 异常.....19	4.3.2 请求额外的权限.....76
2.2 引用.....24	4.3.3 图像 API.....77
2.3 总结.....27	4.4 总结.....79
第 3 章 PHP 在移动设备上的应用.....28	第 5 章 前沿.....80
3.1 移动设备的差异性.....28	5.1 命名空间.....80
3.2 检测设备.....29	5.1.1 命名空间和自动加载.....83
3.2.1 用户代理.....29	5.1.2 命名空间小结.....83
3.2.2 内置的 PHP 支持.....29	5.2 匿名函数（闭包）.....84
3.3 侦测移动性能.....32	5.3 Nowdoc.....85
3.3.1 WURFL 系统.....32	5.4 局部 goto 语句.....87
3.3.2 Tera-WURFL.....40	5.5 标准 PHP 库.....88
3.4 渲染工具.....44	5.6 Phar 扩展.....91
3.4.1 WALL.....44	5.7 总结.....94
3.4.2 图像调整.....45	
3.4.3 响应性 CSS.....46	
3.5 模拟器和 SDK.....47	

第 6 章 表单设计与管理	95	10.4 电子邮件和短信	205
6.1 数据验证	95	10.5 gChartPHP: Google 图表 API 包	208
6.2 上传文件/图片	101	10.6 总结	211
6.3 图像转换和缩略图	102	第 11 章 安全	212
6.4 正则表达式	104	11.1 永不信任数据	212
6.5 多语言集成	107	11.1.1 register_globals	212
6.6 总结	108	11.1.2 白名单和黑名单	213
第 7 章 数据库集成 I	109	11.1.3 表单数据	214
7.1 MongoDB 简介	110	11.1.4 \$_COOKIES、\$_SESSION 和 \$_SERVER	215
7.1.1 MongoDB 查询	114	11.1.5 Ajax 请求	216
7.1.2 更新 MongoDB	118	11.2 常见的攻击	216
7.1.3 MongoDB 中的聚合	120	11.2.1 同源策略	216
7.1.4 MongoDB 总结	123	11.2.2 跨站脚本	217
7.2 CouchDB 简介	123	11.2.3 跨站请求伪造	219
7.2.1 使用 Futon	124	11.3 会话	220
7.2.2 CouchDB 总结	129	11.4 预防 SQL 注入	221
7.3 SQLite 简介	130	11.5 过滤器扩展	222
7.4 总结	139	11.6 php.ini 和服务器设置	225
第 8 章 数据库集成 II	140	11.6.1 服务器环境	225
8.1 MySQLi 扩展简介	140	11.6.2 强化 php.ini	226
8.2 PDO 简介	147	11.7 密码算法	227
8.3 ADOdb 简介	151	11.8 总结	228
8.4 使用 Sphinx 全文检索	155	第 12 章 使用 Zend Studio for Eclipse、 Bugzilla、Mylyn 和 Subversion 进行敏捷开发	229
8.5 总结	163	12.1 敏捷开发的原则	229
第 9 章 数据库集成 III	164	12.2 项目的敏捷开发	230
9.1 Oracle 关系型数据库简介	164	12.3 Bugzilla 简介	231
9.2 基础知识: 连接和执行 SQL	166	12.4 Eclipse 下的 Mylyn	233
9.3 数组接口	169	12.5 Eclipse 中组合 Bugzilla 和 Mylyn	235
9.4 PL/SQL 存储过程及游标	173	12.6 延伸功能的优点	239
9.5 使用 LOB 类型	176	12.7 总结	240
9.6 再探数据库连接: 连接池	180	第 13 章 重构、单元测试和持续集成	241
9.7 数据库和 PHP 中的字符集	183	13.1 重构	241
9.8 总结	184	13.1.1 小的重构	242
第 10 章 函数库	185	13.1.2 更大的遗留代码的示例	245
10.1 SimplePie	186	13.2 单元测试	259
10.2 TCPDF	189		
10.3 集成谷歌地图	201		

13.3 持续集成.....	274	15.2.1 传统的 Web 模型.....	311
13.3.1 持续集成服务器.....	275	15.2.2 Ajax Web 模型.....	312
13.3.2 版本控制.....	275	15.2.3 异步事件与同步事件.....	313
13.3.3 静态分析.....	276	15.2.4 XMLHttpRequest 对象.....	314
13.3.4 构建自动化.....	277	15.2.5 使用 XMLHttpRequest.....	316
13.3.5 Jenkins 服务器的安装.....	278	15.2.6 高级的 JavaScript API.....	321
13.4 总结.....	281	15.2.7 jQuery 的例子.....	321
第 14 章 XML.....	282	15.2.8 使用 Ajax 向 PHP 脚本发送 数据.....	326
14.1 XML 入门.....	282	15.3 一个简单的图形程序.....	328
14.2 模式 (schema).....	283	15.4 总结.....	336
14.3 SimpleXML.....	284	第 16 章 结束语.....	337
14.3.1 从字符串解析 XML.....	284	16.1 资源.....	337
14.3.2 从文件解析 XML.....	285	16.1.1 www.php.net.....	337
14.3.3 命名空间.....	290	16.1.2 www.zend.com.....	338
14.3.4 RSS.....	292	16.1.3 devzone.zend.com.....	338
14.3.5 使用 SimpleXML 生成 XML.....	294	16.1.4 PHP Architect Magazine: www.phparch.com.....	339
14.4 DOMDocument.....	299	16.2 会议.....	339
14.5 XMLReader 和 XMLWriter.....	301	16.3 PHP 认证.....	340
14.6 总结.....	303	16.4 总结.....	341
第 15 章 JSON 和 Ajax.....	304	附录 正则表达式.....	342
15.1 JSON.....	305		
15.2 Ajax.....	311		

面向对象



本章的主要目的是介绍面向对象的基本概念。这实际上是在问：“PHP是面向对象的吗？”最简单的回答是，PHP考虑到了用户数据类型的定义和分层组织。这本书介绍了PHP面向对象方面一些新元素。从第4版开始，PHP发生了非常巨大的变化，具备了基本的面向对象的功能。例如，在PHP 4中不能定义方法和成员的可见性，而在PHP 5.3中已经增加了命名空间。

在本章中，我们将介绍类、继承、对象的创建和接口定义，还将介绍一些迭代器这样的基础内容。好了，开始吧。

1.1 类

类是简单的用户定义类型。在面向对象的语言中，类就是创建类的对象或实例（功能副本）的模版；类描述了其所属的任何对象的共同特点；类的目的是封装对象的定义和行为，对最终用户隐藏它的具体实现，并使最终用户按照文档说明和期望的方式来使用类对象。封装使程序更小，更易于管理，这是因为对象已经包含了处理它们所需的逻辑。类还有自动加载的功能，这样就可以把脚本分成更小、更易于管理的代码块。

在看一个简单的PHP类之前，我们先来介绍一些术语。

- 类成员或类属性：是一个变量，是类的数据部分。
- 类方法：在类中定义的函数。

现在我们要定义一个在二维平面中的点类，使用了笛卡儿坐标定义（如代码清单1-1所示）。由于设计这个类纯粹是为了教学目的，而且它有几个严重的缺陷，所以不建议你把它作为代码开发基础。

代码清单1-1 二维平面

```
<?php
class Point {
    public $x;
    public $y;

    function __construct($x,$y) {
        $this->x=$x;
        $this->y=$y;
    }
}
```

```
function get_x() {
    return($this->x);
}
function get_y() {
    return($this->y);
}
function dist($p) {
    return(sqrt( pow($this->x-$p->get_x(),2)+
                pow($this->y-$p->get_y(),2)));
}
} // 结束类的定义
$p1=new Point(2,3);
$p2=new Point(3,4);
echo $p1->dist($p2), "\n";
$p2->x=5;
echo $p1->dist($p2), "\n";
?>
```

这个类并不简单，有相当多的地方需要分析和修改。首先，正如前面所说的那样，这个类描述了平面里的一个点，并通过笛卡尔坐标 x 和 y 来定义。这里用到了关键词`public`，随后我们会讲解它。还有一个构造方法`__construct`，当使用`new`操作符在内存中创建`Point`类的一个新对象（或实例）时，它就会被调用。换句话说，当执行`$p1=new Point(2,3)`这一行时，方法`__construct`就会被自动引用和执行，并且类名后面圆括号中的参数，会被传递给`__construct`方法使用。

`__construct`方法中引用了`$this`变量。`$this`变量指向的是面向对象的类的实例本身。它始终指向当前对象这个焦点，等价于面向对象中的“我”。在几乎所有的面向对象的语言中，都有这个变量的变体，在一些语言中甚至叫做“`self`”。

类的构造函数就是对给定的类对象进行初始化（或实例化）时执行的方法。在这个代码示例中，它用来给坐标赋初值。坐标（变量 x 和 y ）是`Point`类的成员。此外，还定义了其他几个方法——两个`get`方法和一个`dist`方法，`dist`方法用来计算两点之间的距离。

接下来我们来看看关键字`public`。声明为`public`的成员，允许全局访问成员变量的数据。在我们的脚本中，有一行是`$p2->x=5;`，这说明一个点的 x 坐标被直接操作了。像这样的访问是无法控制的，即使在最简单的情况下也是相当糟糕的。好的实践是通过`get`和`set`方法来读或写类的成员，这样就可以控制类的成员。换句话说，使用`get`和`set`方法，就可以控制数据成员的值。对`public`成员来说，`get`和`set`函数是多余的，因为可以直接设置成员变量，就像`$p2->x=5`。不管怎样，使用`public`成员，成员变量的值是不可控的。可以为每一个成员直接编写`set`和`get`函数，但是PHP提供了所谓的“魔术方法”，用来取代必须为每一个成员写两个函数的方法。

使用`private`和`protected`关键词，可以更好地保护类的成员变量。这两个关键字的确切含义将在下一节中解释。值得注意的是，`public`是默认的可见性。如果没有指定成员或方法的可见性，它的默认属性就是`public`。写做

```
class C {
    $member;
    function method() {...}
    ...
}
```


完全等价于写成：

```
class C {
    public $member;
    public function method() {...}
    ...
}
```

与public的类成员相比，private的类成员或方法只能在相同类的方法中可见。只要不是类的方法就无法访问类的任何私有成员，也包括任何其他的私有方法。如果类成员\$x和\$y前面的关键词public被替换成关键词private，并且试图直接访问的话，结果将会是：

```
PHP Fatal error: Cannot access private property Point::$x in script2.1 on line 25
```

换句话说，在第25行，我们简单的小技巧\$p2->x=5将不再起作用。而构造函数没有任何问题，包括get_x()和get_y()同样没有任何问题，因为它们都是类的成员。这是好事，因为这样将不能直接操纵类对象；用类来实现它们本不应该实现的行为，有可能从根本上改变它们的行为。总之，类是自足的，它就像一个可以控制入口的高速公路——有受限的入口和出口坡道。

现在共有和私有成员已经讲明白了，但是保护成员和方法又如何理解呢？保护成员和方法只能被成员从属的类的方法访问，以及把它们从属的类做为基类的子类的方法所访问。本书将在接下来的部分深入讲解它们。

1.2 继承和重载

正如本章开头所述，类可以通过分层的方式来组织。分层结构可以通过继承建立。为了演示继承，我们开发一个叫employee的类。公司的雇员中有一些是经理，所以经理类将会继承自一般雇员类。使用继承被认为更加专业化。因此，事不宜迟，来看看这个类（如代码清单1-2所示）。

代码清单1-2 employee类示例

```
<?php
class employee {
    protected $name;
    protected $sal;
    function __construct($name, $sal = 100) {
        $this->name = $name;
        $this->sal = $sal;
    }
    function give_raise($amount) {
        $this->sal+= $amount;
        printf("Employee %s got raise of %d dollars\n", $this->name, $amount);
        printf("New salary is %d dollars\n", $this->sal);
    }
    function __destruct() {
        printf("Good bye, cruel world: EMPLOYEE:%s\n", $this->name);
    }
}
```