



Python 3 Object-oriented Programming (Second Edition)

# Python 3

# 面向对象编程 (第2版)

[加] Dusty Phillips 著  
孙雨生 译



中国工信出版集团



电子工业出版社  
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY  
http://www.phei.com.cn



Python 3 Object-oriented Programming (Second Edition)

# Python 3

## 面向对象编程 (第2版)

---

[加] Dusty Phillips 著  
孙雨生 译

电子工业出版社  
Publishing House of Electronics Industry  
北京·BEIJING



## 内 容 简 介

本书主要介绍如何使用 Python 3 进行面向对象编程。第 1~4 章介绍面向对象这一编程范式的基本准则，以及 Python 是如何运用这些准则实现面向对象编程的；第 5~8 章介绍如何利用 Python 中的内置函数快速、简单地实现面向对象编程；第 9~11 章介绍许多面向对象编程中常用的设计模式，以及如何使用符合 Python 语言习惯的方式来实现这些设计模式；最后，第 12、13 章介绍 Python 3 中与并发编程相关的主题。

本书中的每一章都会包含一节案例学习的内容，通过一个实践相关的案例将本章介绍的主要内容以及前面章节中介绍过的内容串联起来。除此之外，每一章最后的练习旨在指导你利用本章学习到的知识，使用面向对象编程改善以往项目中的代码，并进一步掌握如何在合适的时机使用 Python 3 进行面向对象编程。

Copyright © 2015 Packt Publishing. First published in the English language under the title ‘Python 3 Object-oriented Programming, Second Edition (9781784398781)’.

本书简体中文版专有出版权由 Packt Publishing 授予电子工业出版社。未经许可，不得以任何方式复制或抄袭本书的任何部分。专有出版权受法律保护。

版权贸易合同登记号 图字：01-2015-7448

### 图书在版编目 (CIP) 数据

Python 3 面向对象编程：第 2 版 / (加) 达斯帝·菲利普斯 (Dusty Phillips) 著；孙雨生译. —北京：电子工业出版社，2018.6

书名原文：Python 3 Object-oriented Programming, Second Edition

ISBN 978-7-121-34136-6

I. ①P… II. ①达… ②孙… III. ①软件工具—程序设计 IV. ①TP311.561

中国版本图书馆 CIP 数据核字 (2018) 第 087133 号

策划编辑：付 睿

责任编辑：李云静

印 刷：三河市鑫金马印装有限公司

装 订：三河市鑫金马印装有限公司

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱 邮编：100036

开 本：787×980 1/16 印张：27.5 字数：549 千字

版 次：2015 年 7 月第 1 版

2018 年 6 月第 2 版

印 次：2018 年 6 月第 1 次印刷

定 价：99.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：(010) 88254888, 88258888。

质量投诉请发邮件至 [zltz@phei.com.cn](mailto:zltz@phei.com.cn)，盗版侵权举报请发邮件至 [dbqq@phei.com.cn](mailto:dbqq@phei.com.cn)。

本书咨询联系方式：010-51260888-819, [faq@phei.com.cn](mailto:faq@phei.com.cn)。



## 关于作者

**Dusty Philips** 是一位来自加拿大的软件开发者和作者，现居于华盛顿州的西雅图市。作者在过去 15 年间活跃于开源社区并主要以 Python 作为开发语言。他是著名的 Puget Sound Programming Python 会议组的共同创始人，如果你在附近区域，欢迎前往参加相关会议。

《Python 3 面向对象编程》由 Packt 出版社出版，这是他的第一本书。他同时也创作了书籍《用 Kivy 创建应用》(O'Reilly)，介绍关于 Python 的移动开发库。同时其发布了 *Hacking Happy*，介绍技术人员的精神健康之旅。他曾在本书第 1 版出版之后不久因为自杀倾向而入院治疗，并自此成为精神健康的公开倡导者。



# 关于审校人员

**AMahdy AbdElAziz** 是拥有超过 8 年经验的软件工程师，用过多种语言和框架。最近 5 年关注 Android 和移动开发，包括跨平台工具以及 Android 内核，例如为嵌入式设备创建自定义的 ROM 和 AOSP。

他目前正在信息技术研究所教授 Python。你可以通过访问 <http://www.amahdy.net> 找到更多关于他的信息。

**Grigoriy Beziuk** 是 Crowdage 基金的前首席信息官，在本书写作时他是一名独立的软件开发者。他拥有众多编程语言和技术的经验，包括不同环境下、不同版本的 Python，涵盖纯科学应用和现代生产环境网络开发条件下的应用。

---

我想要感谢我的妈妈，Larisa Beziuk，是她赋予了我生命；感谢所有让我的人生变得更加有趣的老师和朋友；感谢所有现在和过去我爱的人……为了所有的一切。

---

**Krishna Bharadwaj** 是 SMERGERS (<https://www.smergers.com/>) 的联合创始人，这是一个区块链创业公司，帮助中小企业从投资者和不同的金融机构募集资金。他过去曾就职于一些早期阶段的创业公司，例如 BlockBeacon (圣莫尼卡) 和 PricePoint (加利福尼亚)，另外还有一些大型组织，例如美国国家仪器、Bangalore、Google (纽约)。Krishna 在大学期间接触到 Python 和 FOSS，并在个人项目和专业工作中大量应用它们。出于对教学的热爱，他游历了诸多大学，并在任何有机会的时候指导研讨会。

他获得了加州大学洛杉矶分校的计算机科学硕士学位，以及班加罗尔 BMS 工程学院的信息科学与工程学士学位。可以通过他的邮箱 [krishna@krishnabharadwaj.info](mailto:krishna@krishnabharadwaj.info) 或个人网站 <http://www.krishnabharadwaj.info/> 联系他。

**Justin Cano** 其最近毕业于加州大学里弗赛德分校，获得了计算机工程学士学位。目前作为软件工程师在硅谷工作，并希望以后可以加入 Google 或 Apple 这种更大的技术公司。

他从六年级开始接触编程，使用 HTML 和 CSS 创建了一些小型的网站。他在加州大学



里弗赛德分校的第一学年开始学习计算机科学理论和 C++，并在第三学年开始学习 Python。

Justin 承认最初并没有马上被 Python 所吸引，因为 C++ 和 Python 的抽象还是非常不同的。直到参加编程竞赛和挑战时他才开始对 Python 产生兴趣，主要是因为他感觉 Python 语法的优雅和可读性可以帮助他更快、更自然地将想法转换成代码。现在他经常使用 Python，在使用领域专用语言之前先用 Python 来创建软件应用的原型。

---

我要感谢作者花时间写了这本书，我从中获取了很多关于 Python 语言和设计模式的很有价值的思考及信息。这本书加深了我对 Python 的理解，我相信我现在是一名更有经验的 Python 开发者了。

---

Anthony Petitbois 是一名拥有 13 年电子游戏行业开发与操作专业经验的在线架构师，并且拥有超过 20 年的软件开发经验。他对新技术富有激情并且喜欢用创造性的方法解决复杂问题。

他在业余时间学习新语言和新平台，以及玩电子游戏，他于 2009 年从法国移居国外，现在他和他的家人住在美丽的加拿大不列颠哥伦比亚地区。

Claudio Rodriguez 其最初的工作是为通用电气开发 PLCs，但是他的主要目标永远都是通过研究和开发将梦想变成现实。这促使他从自动化工程转向软件工程以及软件的结构化方法——面向对象设计。在他读硕士期间，不得不学会从合适的地方寻找资源，并与书籍、研究文献和研讨会成为朋友。最终，他开始致力于控制电弧炉系统的工作，但是客户的需求让他去学习了更多的控制技术。他热爱复杂的 AI 技术，并总是被文献、书籍以及测试用的计算机所环绕，不过他依然通过为客户提供美妙而充满活力的应用来与现实世界保持联系。



# 译者序

Python 是一门多范式编程语言，这意味着你可以用它进行面向对象编程，也可以选择使用面向过程编程的方式，甚至可以尝试函数式编程。而 Python 最令人着迷之处在于，它拥有极大灵活性的同时，使用的是十分简单、优雅的语法，因此甚至被称为“可执行的伪代码”。本书主要围绕 Python 面向对象编程相关的概念与方法，同时也介绍了一些 Python 3 的新增功能、特性。众所周知，Python 3 的升级、推广在 Python 开发者中一直存在较多争议，批判者的声音主要集中在其向后不兼容这一点。不过译者认为 Python 3 是非常值得开发者花时间去学习、升级的，Python 3 中的字符串默认采用 UTF-8 编码，这无疑给中文环境的开发者带来了极大的便捷；此外，采用新的异步编程模型也使得 Python 在服务器开发领域更具竞争力。幸运的是，本书对于这几点都通过单独章节分别进行了详细介绍，无论读者是刚刚接触 Python 的初学者，还是打算将 Python 从之前的版本升级到最新的版本，本书都将很有帮助。



## 第 2 版序

我需要忏悔，在我写本书第 1 版的时候，我根本不知道自己在做什么。我认为我了解 Python 并且知道该怎么写。很快我就发现这是错的。幸运的是，通过本书的写作我已经完全适应了这两点。

我担心大家可能会不喜欢《Python 3 面向对象编程》这本书，所以我两年没有参加 PyCon。直到收获了几十条正面的反馈，我才重获信心，终于参加了 2012 年在圣克拉拉举办的 PyCon。我很快发现每个人都听说过我或我的书，完全是我庸人自扰！

书写完之后我也很怕重新阅读它。在收到诸多赞誉之前，它一直严实地放在我的书架上，直到我需要引用其中的内容回答读者的提问。在准备本书第 2 版的时候，我最终不得不面对我的心魔。令我惊喜和开心的是，我发现 5 年前自己所写的这本书既准确又令人愉快，一如很多读者的评价所提到的。

初次阅读之后不久，我收到了来自 Amazon 上的第一条负面评价。如果我在完成本书后立即看到这条评价，那将是毁灭性的。幸运的是，4 年来的正面评价以及我自己对写作上的自信，让我可以忽略辛辣的批评并将剩下的部分作为建设性的反馈。事实上，读者提出的很多瑕疵都是本书最初出版时 Python 还未实现的特征。《Python 3 面向对象编程》出版已经有些年头了，显然已经到了需要升级的时候。升级的结果现在已经在你的手里（或在你的电子书阅读器上）。

我经常纳闷，为什么技术类书籍的作者要如此详细地描述不同版本书籍之间的区别。我的意思是，有多少人会在读了第 1 版之后还读第 2 版呢？就像软件的版本一样，你可以笃定最新的版本是最好的，但你不会真的想要关心它的历史。不过，这本书占据了我过去一年大部分的时间，所以我不得不提一下这本书已经变得多好了。

其实第 1 版组织得稍微有些混乱，很多章节结束之后直接跳到下一章，有几个关键位置上的主题之间存在跳跃，甚至更糟糕的是，彼此不相关。现在，讨论设计模式之前的 2 章被重新调整并分为 3 章，这样更加顺畅地切换到下一个主题。



我同时也彻底删除了关于 Python 3 第三方库的一整章内容。在 Python 3 和这本书都还是新鲜事物的时候，这一章的存在还算合理。只有几个库移植到 Python 3，对它们进行详尽的讨论是合理的。不过，我没办法深入探讨这些主题的细节，坦白地说，针对每一个主题我都可以再写一整本书。

最后，我添加了全新的一章，关于并发的内容。关于这一章我纠结了一阵，现在可以坦白承认它与面向对象编程并没有直接的关系。然而，和单元测试那一章一样，我认为理解并发是所有语言中不可或缺的部分，特别是对于 Python 生态系统中的面向对象编程。当然，如果你不同意大可跳过这些章节（或者等你改变主意之后再次阅读它们）。

请尽情享受这本书以及你的面向对象编程之旅！

Dusty Phillips



# 前言

本书介绍了面向对象范式的相关术语，通过循序渐进的例子聚焦面向对象设计。本书介绍从简单且是面向对象编程工具箱中最有用的继承关系，到异常处理和设计模式的内容，以一种面向对象的方式来看待面向对象的概念。

在这一过程中，我们将会学习整合 Python 编程语言中面向对象和非面向对象的方面，学习操作文件和字符串的复杂性，并强调（正如 Python 3 所做的）二进制和文本数据的区别。

我们将会发现单元测试的乐趣，会用到两个单元测试框架。最后，我们会探索 Python 众多的并发范式，学习如何让不同对象在同一时间一起运行。

## 本书包含哪些内容

本书可以粗略地分为 4 个主要部分。前 4 章我们将会深入探讨面向对象编程的准则以及 Python 是如何运用它们的。第 5~8 章我们将会通过学习这些准则是如何应用到 Python 内置函数中的，来探讨一些 Python 特有的应用。第 9~11 章是关于设计模式的内容。最后两章探讨与 Python 编程相关同时可能很有趣的主题。

第 1 章，面向对象设计，介绍了重要的面向对象概念。主要关于一些相关术语，如抽象、类、封装、继承。我们也简单地介绍了用来建模类和对象的 UML。

第 2 章，Python 对象，讨论类和对象以及如何在 Python 中使用它们。我们将会学习 Python 对象的属性和行为，以及如何将类组织到包和模块中去。最后，我们将会学习如何保护我们的数据。

第 3 章，对象相似时，帮助我们更加深入地探讨继承关系。包括多重继承以及如何扩展内置类型。这一章也介绍了多态和鸭子类型在 Python 中是如何运作的。



第 4 章, 异常捕获, 介绍异常和异常处理。我们将会学习如何创建我们自己的异常, 以及如何利用异常控制程序流程。

第 5 章, 何时使用面向对象编程, 介绍如何创建和使用对象。我们将会看到如何用属性封装数据以及限制数据的访问。这一章也讨论了 DRY 准则以及如何避免代码重复。

第 6 章, Python 数据结构, 介绍 Python 内置类的面向对象特征。我们将会学习元组、字典、列表以及集合, 还有几个更高级的容器类型。我们也将学习如何扩展这些标准对象。

第 7 章, Python 面向对象的捷径, 正如标题所言, 这一章介绍使用 Python 节省时间的方法。我们将会学习很多有用的内置函数, 例如用默认参数实现方法重载。我们也将看到函数本身也是对象的特性以及如何利用这一特性。

第 8 章, 字符串与序列化, 介绍字符串、文件以及格式化。我们将会讨论字符串、字节以及字节数组之间的区别, 还有将序列化文本、对象和二进制数据转化为几种规范表示的多种不同方式。

第 9 章, 迭代器模式, 介绍了设计模式的概念, 以及迭代器模式的 Python 图形化实现。我们将会学习列表、集合以及字典的推导形式。我们也将理解生成器与协程。

第 10 章, Python 设计模式 I, 介绍了几种设计模式, 包括装饰器模式、观察者模式、策略模式、状态模式、单例模式以及模板模式。每一种设计模式都通过 Python 实现的相关的例子程序进行讨论。

第 11 章, Python 设计模式 II, 专注讨论更多的设计模式, 如适配器模式、门面模式、享元模式、命令模式、抽象模式以及组合模式。用更多的例子说明 Python 习惯用法与规范实现之间的区别。

第 12 章, 测试面向对象程序, 以介绍为何对 Python 应用进行测试如此重要为开始。强调测试驱动开发, 并介绍两种不同的测试工具 `unittest` 和 `py.test`。最后, 讨论模拟测试对象和代码覆盖率相关内容。

第 13 章, 并发, 这一章是关于 Python 所支持 (以及缺乏) 的并发模式的速成教程。讨论了线程、多进程、`Future` 以及新的 `AsyncIO` 库。

每一章都包含相关的例子和一个案例学习, 案例学习会将本章内容整合到一个可运行的 (可能不完整的) 程序中。



## 学习本书你需要用到什么

本书中所有的例子都是基于 Python 3 解释器的。确保你用的不是 Python 2.7 或更早的版本。在写本书时，Python 3.4 是最新的版本。大部分例子也可以在较早版本的 Python 3 中运行，但是为了最大限度地减少可能给你带来的失望情绪，最好用最新版本。

所有的例子都可以运行在任何支持 Python 的操作系统上。如果不能，请作为错误信息提交给我。

有些例子需要连接互联网。你可能会想要有一个这样的课外研究和调试。

除此之外，本书中有些例子依赖于不使用 Python 的第三方库。在用到它们的时候会进行介绍，因此你并不需要提前安装。但是为了保持完整性，下面列出会用到的所有库：

- pip
- requests
- pillow
- bitarray

## 本书的目标读者

本书尤其针对那些没有面向对象编程经验的人，同时也假设你已经拥有了基本的 Python 技能。你将会深入学习面向对象准则。对于将 Python 用作“胶水”语言并希望提升编程技巧的系统管理员来说，本书也特别有用。

如果你熟悉用其他语言进行面向对象编程，那么本书将会帮助你用符合 Python 语言习惯的方式将你的这些知识应用到 Python 生态系统中。

## 约定

本书使用不同的文本格式来区别不同的信息。下面是一些格式的例子，以及它们的含义。

文本中的代码、数据库表名、目录名、文件名、文件后缀名、路径名、URL、用户输入以及 Twitter 用户名以如下方式展示：“我们从字典中查询类，并存储到名为 `PropertyClass`



的变量中。”

一个代码块可以设定为如下所示的样式：

```
def add_property(self):
    property_type = get_valid_input(
        "What type of property? ",
        ("house", "apartment")).lower()
    payment_type = get_valid_input(
        "What payment type? ",
        ("purchase", "rental")).lower()
```

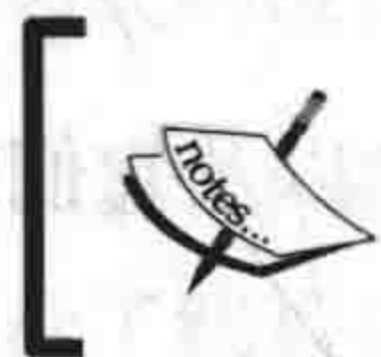
当我们想要让你特别关注代码块中的特定部分时，相关的行或项将会加粗显示：

```
def add_property(self):
    property_type = get_valid_input(
        "What type of property? ",
        ("house", "apartment")).lower()
    payment_type = get_valid_input(
        "What payment type? ",
        ("purchase", "rental")).lower()
```

任何命令行的输入或输出都将会被设定为如下格式：

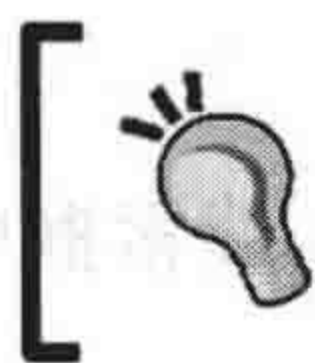
```
>>> c1 = Contact("John A", "johna@example.net")
>>> c2 = Contact("John B", "johnb@example.net")
>>> c3 = Contact("Jenna C", "jennac@example.net")
>>> [c.name for c in Contact.all_contacts.search('John')]
['John A', 'John B']
```

新的术语或重要单词将会以黑体呈现。你在屏幕、菜单或对话框中看到的单词，将会以这种形式出现：“将出现参数不够的错误，和我们前面忘记 self 参数一样。”



警告或重要注意将出现在这里。





提示和技巧将出现在这里。

## 读者反馈

欢迎读者反馈意见。让我们知道你是如何看待本书的——有什么是你喜欢或不喜欢的。读者反馈对我们非常重要，因为可以帮助我们开发对你们更有用的内容。

可以通过 [feedback@packtpub.com](mailto:feedback@packtpub.com) 发送一般性的反馈，只要在邮件主题中提及本书书名即可。

如果有哪个你擅长的领域，你想要写作或贡献一本书，请访问 [www.packtpub.com/authors](http://www.packtpub.com/authors) 查看我们的作者指南。

## 客户支持

现在你已经是 Packt 尊贵的读者了，我们提供如下一些资源可以帮助你收获更多。

### 下载示例代码

你可以从 <http://www.packtpub.com> 上你的账户中下载所有你购买的 Packt 出版的书籍中的示例代码。如果你从别处购买了本书，你可以访问 <http://www.packtpub.com/support> 并注册，然后直接通过邮件获取这些文件。

### 勘误

尽管我们已经尽可能保证内容的准确性，但仍然难以避免存在错误。如果你在我们的书中发现了错误——不管是文字错误还是代码错误——如果你能够告诉我们，我们将会非常感激。这样一来，你也可以帮助其他读者避免很多困惑，并且帮助我们在后续版本中不断改进。如果你发现任何错误，请访问 <http://www.packtpub.com/submit-errata>，选择对应的书，单击**勘误提交表**链接，并输入详细的错误信息。一旦确认了你的勘误，你提交的内容将会被接受，这一勘误信息将会上传到我们的网站，或者添加到对应书籍的勘



误章节列表中。

访问 <https://www.packtpub.com/books/content/support> 并在搜索框中输入书名，可以查看之前提交的勘误。所需的信息将会出现在勘误章节中。

## 版权

流传于互联网上的内容，版权是所有媒体正在面临的问题。在 Packt，我们严格保护版权与许可。如果你在网看到任何关于我们书籍的非法副本，请立即向我们提供地址或网站名，我们将进行追踪维权。

请通过 [copyright@packtpub.com](mailto:copyright@packtpub.com) 向我们提供可疑盗版材料的链接。

感谢你帮助我们保护我们的作者以及我们向你提供有价值内容的能力。

## 问题

如果对本书有任何问题，你可以通过 [questions@packtpub.com](mailto:questions@packtpub.com) 与我们取得联系，我们将会倾尽全力帮你解决。

---

轻松注册成为博文视点社区 ([www.broadview.com.cn](http://www.broadview.com.cn)) 用户，扫码直达本书页面。

- **下载资源**：本书提供示例代码及资源文件，均可在“[下载资源](#)”处下载。
- **提交勘误**：您对书中内容的修改意见可在“[提交勘误](#)”处提交，若被采纳，将获赠博文视点社区积分（在您购买电子书时，积分可用来抵扣相应金额）。
- **交流互动**：在页面下方“[读者评论](#)”处留下您的疑问或观点，与我们和其他读者一同学习交流。

页面入口：<http://www.broadview.com.cn/34136>





## 博文视点诚邀精锐作者加盟

十载耕耘奠定专业地位

以书为证彰显卓越品质

《C++Primer (中文版) (第5版)》、《淘宝技术这十年》、《代码大全》、《Windows内核情景分析》、《加密与解密》、《编程之美》、《VC++深入详解》、《SEO实战密码》、《PPT演义》……

“圣经”级图书光耀夺目,被无数读者朋友奉为案头手册传世经典。

潘爱民、毛德操、张亚勤、张宏江、咎辉Zac、李刚、曹江华……

“明星”级作者济济一堂,他们的名字熠熠生辉,与IT业的蓬勃发展紧密相连。

十年的开拓、探索和励精图治,成就博古通今、文圆质方、视角独特、点石成金之计算机图书的风向标杆:博文视点。

“凤翱翔于千仞兮,非梧不栖”,博文视点欢迎更多才华横溢、锐意创新的作者朋友加盟,与大师并列于IT专业出版之巔。

### 英雄帖

江湖风云起,代有才人出。  
IT界群雄并起,逐鹿中原。  
博文视点诚邀天下技术英豪加入,  
指点江山,激扬文字  
传播信息技术,分享IT心得

### · 专业的作者服务 ·

博文视点自成立以来一直专注于IT专业技术图书的出版,拥有丰富的与技术图书作者合作的经验,并参照IT技术图书的特点,打造了一支高效运转、富有服务意识的编辑出版团队。我们始终坚持:

**善待作者**——我们会把出版流程整理得清晰简明,为作者提供优厚的稿酬服务,解除作者的顾虑,安心写作,展现出最好的作品。

**尊重作者**——我们尊重每一位作者的技术能力和生活习惯,并会参照作者实际的工作、生活节奏,量身制定写作计划,确保合作顺利进行。

**提升作者**——我们打造精品图书,更要打造知名作者。博文视点致力于通过图书提升作者的个人品牌和技术影响力,为作者的事业开拓带来更多的机会。



### 联系我们

博文视点官网: <http://www.broadview.com.cn>

CSDN官方博客: <http://blog.csdn.net/broadview2006/>

投稿电话: 010-51260888 88254368

投稿邮箱: [jsj@phei.com.cn](mailto:jsj@phei.com.cn)



@博文视点Broadview



微信公众账号 博文视点Broadview





# 目录

|                              |    |
|------------------------------|----|
| <b>第 1 章 面向对象设计</b> .....    | 1  |
| 面向对象 .....                   | 1  |
| 对象和类 .....                   | 3  |
| 指定属性和行为 .....                | 5  |
| 数据描述对象 .....                 | 5  |
| 行为就是动作 .....                 | 7  |
| 隐藏细节并创建公共接口 .....            | 8  |
| 组合 .....                     | 11 |
| 继承 .....                     | 13 |
| 继承提供抽象 .....                 | 15 |
| 多重继承 .....                   | 16 |
| 案例学习 .....                   | 16 |
| 练习 .....                     | 24 |
| 总结 .....                     | 24 |
| <b>第 2 章 Python 对象</b> ..... | 26 |
| 创建 Python 类 .....            | 26 |
| 添加属性 .....                   | 28 |
| 让它做点什么 .....                 | 29 |
| 初始化对象 .....                  | 32 |
| 自我解释 .....                   | 34 |
| 模块和包 .....                   | 36 |
| 组织模块 .....                   | 38 |
| 组织模块内容 .....                 | 41 |
| 谁可以访问我的数据 .....              | 44 |