Предисловие

Язык программирования Python чрезвычайно популярен и используется для различных приложений. Язык Python предназначен для того, чтобы относительно легко создавать небольшие программы. Чтобы создать более сложное программное обеспечение, нам нужно приобрести ряд важных навыков программирования и проектирования программного обеспечения.

В этой книге описан **объектно-ориентированный** подход к созданию программ на Python. Он вводит терминологию объектно-ориентированного программирования, демонстрируя проектирование программного обеспечения и программирование на Python на пошаговых примерах. В нем описывается, как использовать наследование и композицию для создания программного обеспечения из отдельных элементов. В нем показано, как использовать встроенные исключения и структуры данных Python, а также элементы стандартной библиотеки Python. Ряд распространенных шаблонов проектирования описан с подробными примерами.

В этой книге рассказывается о том, как писать автоматизированные тесты, чтобы подтвердить, что наше программное обеспечение работает. В нем также показано, как использовать различные библиотеки параллелизма, доступные как часть Python; это позволяет нам писать программное обеспечение, которое может использовать несколько ядер и несколько процессоров в современном компьютере. Расширенное тематическое исследование охватывает простой пример машинного обучения, показывающий ряд альтернативных решений умеренно сложной проблемы.

Для кого предназначена эта книга

Эта книга предназначена для людей, которые являются новичками в объектно-ориентированном программировании на Python. Он предполагает базовые навыки Python. Для читателей с опытом работы в другом объектно-ориентированном языке программирования эта книга раскроет многие отличительные особенности подхода Python.

Из-за использования Python для науки о данных и анализа данных, эта книга затрагивает связанные с математикой и статистикой концепции. Некоторые знания в этих областях могут помочь сделать применение концепций более конкретным.

О чем эта книга

Эта книга разделена на четыре общих раздела. В первых шести главах представлены основные принципы и концепции объектно-ориентированного программирования и то, как они реализуются в Python. В следующих трех главах подробно рассматриваются встроенные функции Python через призму объектно-ориентированного программирования. *В главах 10, 11* и *12* рассматривается ряд распространенных шаблонов проектирования и то, как их можно обрабатывать в Python. В заключительном разделе рассматриваются две дополнительные темы: тестирование и параллелизм.

*Глава 1 «Объектно-ориентированное проектирование»* знакомит с основными концепциями, лежащими в основе объектно-ориентированного проектирования. Это обеспечивает дорожную карту через идеи состояния и поведения, атрибутов и методов, а также того, как объекты группируются в классы. В этой главе также рассматривается инкапсуляция, наследование и композиция. В тематическом исследовании для этой главы представлена задача машинного обучения, которая представляет собой реализацию классификатора k-ближайших соседей (k-NN).

*Глава 2, Объекты в Python*, показывает, как определения классов работают в Python. Это будет включать аннотации типов, называемые подсказками типов, определения классов, модули и пакеты. Мы поговорим о практических соображениях по определению и инкапсуляции классов. В тематическом исследовании начнут реализовываться некоторые классы для классификатора k-NN.

*В главе 3 «Когда объекты похожи»* рассматривается, как классы связаны друг с другом. Это будет включать в себя то, как использовать наследование и множественное наследование. Рассмотрим концепцию полиморфизма среди классов в классовой иерархии. В тематическом исследовании будут рассмотрены альтернативные конструкции для вычислений расстояния, используемых для поиска ближайших соседей.

*В главе 4 «Ожидание неожиданного»* подробно рассматриваются исключения Python и обработка исключений. Рассмотрим встроенную иерархию исключений. Мы также рассмотрим, как уникальные исключения могут быть определены для отражения уникальной проблемной области или приложения. В этом практическом примере мы будем применять исключения для проверки данных.

*Глава 5 «Когда использовать объектно-ориентированное программирование»* более глубоко погружается в методы проектирования. В этой главе будет рассмотрено, как атрибуты могут быть реализованы с помощью свойств Python. Также мы рассмотрим общую концепцию менеджера для работы с коллекциями объектов. В тематическом исследовании эти идеи будут применены для расширения реализации классификатора k-NN.

*Глава 6 «Абстрактные базовые классы и перегрузка операторов»* представляет собой глубокое погружение в идею абстракции и в то, как Python поддерживает абстрактные базовые классы. Это будет включать в себя сравнение типизации уток с более формальными методами определения протокола. Он будет включать в себя методы перегрузки встроенных операторов Python. В нем также будут рассмотрены метаклассы и то, как их можно использовать для изменения построения классов. В тематическом исследовании будут пересмотрены некоторые из существующих классов, чтобы показать, как абстракция должна использоваться осторожно, чтобы привести к упрощению дизайна.

*В главе 7 «Структуры данных Python»* рассматривается ряд встроенных коллекций Python. В этой главе рассматриваются кортежи, словари, списки и наборы. В нем также рассматривается, как классы данных и именованные кортежи могут упростить проектирование, предоставляя ряд общих функций класса. В тематическом исследовании будут пересмотрены некоторые более ранние определения классов для использования этих новых методов.

*В главе 8 «Пересечение объектно-ориентированного и функционального программирования»* рассматриваются конструкции Python, которые не являются просто определениями классов. Хотя весь Python является объектно-ориентированным, определения функций позволяют нам создавать вызываемые объекты без беспорядка в определении класса. Мы также рассмотрим конструкцию контекстного менеджера Python и оператор with. В тематическом исследовании мы рассмотрим альтернативные проекты, которые позволяют избежать некоторого беспорядка в классе.

*В главе 9 «Строки, сериализация и пути к файлам»* рассматривается способ сериализации объектов в виде строк и способы синтаксического анализа строк для создания объектов. Мы рассмотрим несколько физических форматов, включая Pickle, JSON и CSV. В тематическом исследовании будет рассмотрен вопрос о том, как выборочные данные загружаются и обрабатываются классификатором k-NN.

*В главе 10 «Шаблон итератора»* описывается вездесущая концепция итерации в Python. Все встроенные коллекции являются итерируемыми, и этот шаблон проектирования занимает центральное место во многих аспектах работы Python. Мы также рассмотрим понимание Python и функции генератора. В тематическом исследовании будут рассмотрены некоторые более ранние проекты с использованием выражений генератора и перечислены понимания для разбиения образцов для тестирования и обучения.

*В главе 11 «Общие шаблоны проектирования»* рассматривается некоторый общий объектно-ориентированный дизайн. Это будет включать в себя шаблоны проектирования Decorator, Observer, Strategy, Command, State и Singleton.

*В главе 12 «Расширенные шаблоны проектирования»* рассматриваются некоторые более продвинутые объектно-ориентированные проекты. Это будет включать в себя шаблоны адаптера, фасада, flyweight, абстрактной фабрики, композита и шаблона.

*В главе 13 «Тестирование объектно-ориентированных программ»* показано, как использовать unittest и pytest для предоставления автоматизированного набора модульных тестов для приложения Python. Здесь будут рассмотрены некоторые более продвинутые методы тестирования, такие как использование макетов объектов для изоляции тестируемого блока. В тематическом исследовании будет показано, как создавать тестовые случаи для дистанционных вычислений, описанных в *главе 3*.

*В главе 14 «Параллелизм»* рассматривается, как мы можем использовать многоядерные и многопроцессорные компьютерные системы для быстрого выполнения вычислений и написания программного обеспечения, реагирующего на внешние события. Мы рассмотрим потоки и многопроцессорную обработку, а также модуль асинхронизации Python. В тематическом исследовании будет показано, как использовать эти методы для настройки гиперпараметров на модели k-NN.

Чтобы получить максимальную отдачу от этой книги

Все примеры были протестированы с Python 3.9.5. Инструмент ***mypy*** версии 0.812 использовался для подтверждения согласованности подсказок типа.

Некоторые из примеров зависят от подключения к Интернету для сбора данных. Эти взаимодействия с веб-сайтами обычно включают небольшие загрузки.

Некоторые из примеров включают пакеты, которые не являются частью встроенной стандартной библиотеки Python. В соответствующих главах мы отмечаем пакеты и предоставляем инструкции по установке. Все эти дополнительные пакеты находятся в индексе пакетов Python на https://pypi.org.

Загрузка файлов примера кода

Пакет кодов для книги также размещен на GitHub по адресу https://github.com/PacktPublishing/Python-Object-Oriented-Programming---4th-edition.

У нас также есть другие пакеты кодов из нашего богатого каталога книг и видео, доступных по адресу https://github.com/PacktPublishing/. Проверьте их!

Загрузка цветных изображений

Мы также предоставляем PDF-файл, содержащий цветные изображения скриншотов / диаграмм, используемых в этой книге. Вы можете скачать его здесь:

https://static.packt-cdn.com/downloads/9781801077262\_ColorImages.pdf

Используемые соглашения

Существует ряд текстовых соглашений, используемых в этой книге.

CodeInText: указывает кодовые слова в тексте, имена таблиц базы данных, имена папок, имена файлов, расширения файлов, пути, фиктивные URL-адреса, пользовательский ввод и дескрипторы Twitter. Например: «Вы можете подтвердить, что Python запущен, импортировав модуль антигравитации в командной строке >>>» (antigravity module at the >>> prompt).

Блок кода задается следующим образом:

class Fizz:

def member(self, v: int) -> bool:

return v % 5 == 0

Когда мы хотим привлечь ваше внимание к определенной части блока кода, соответствующие строки или элементы выделены жирным шрифтом:

class Fizz:

**def member(self, v: int) -> bool:**

return v % 5 == 0

Любой ввод или вывод командной строки записывается следующим образом:

python -m pip install tox

**Полужирный:** указывает новый термин, важное слово или слова, которые отображаются на экране, например в меню или диалоговых окнах. Например: «Формально объект представляет собой коллекцию **данных (data)** и связанного с ними **поведения (behaviors)**».

…

Свяжитесь с нами

Обратная связь от наших читателей всегда приветствуется.

**Общая обратная связь:** отправьте feedback@packtpub.com по электронной почте и укажите название книги в теме вашего сообщения. Если у вас есть вопросы о каком-либо аспекте этой книги, пожалуйста, напишите нам по адресу questions@packtpub.com.

**Опечатки:** Несмотря на то, что мы позаботились о том, чтобы обеспечить точность нашего контента, ошибки случаются. Если вы нашли ошибку в этой книге, мы будем признательны, если вы сообщите нам об этом. Пожалуйста, посетите http://www.packtpub.com/submit-errata, выберите свою книгу, нажмите на ссылку Форма отправки опечаток и введите детали.

**Пиратство:** Если вы столкнетесь с какими-либо незаконными копиями наших работ в любой форме в Интернете, мы будем признательны, если вы предоставите нам адрес местоположения или название веб-сайта. Пожалуйста, свяжитесь с нами по copyright@packtpub.com со ссылкой на материал.

**Если вы заинтересованы в том, чтобы стать автором:** Если есть тема, в которой у вас есть опыт, и вы заинтересованы в написании или внесении вклада в книгу, пожалуйста, посетите http://authors.packtpub.com.

Поделитесь своими мыслями

Once you've read *Python Object-Oriented Programming, Fourth Edition*, we'd love to hear your thoughts! Please click here to go straight to the Amazon review page for this book and share your feedback.

После того, как вы прочитали Объектно-ориентированное программирование на Python, четвертое издание, мы будем рады услышать ваши мысли! Пожалуйста, [нажмите здесь, чтобы перейти прямо на страницу обзора Amazon для этой книги](https://www.amazon.com/ap/signin?openid.pape.max_auth_age=3600&openid.return_to=https%3A%2F%2Fwww.amazon.com%2Freview%2Fcreate-review%2F%3Fasin%3D1801077266&openid.identity=http%3A%2F%2Fspecs.openid.net%2Fauth%2F2.0%2Fidentifier_select&openid.assoc_handle=amzn_scarface_mobile_us&openid.mode=checkid_setup&language=en_US&openid.claimed_id=http%3A%2F%2Fspecs.openid.net%2Fauth%2F2.0%2Fidentifier_select&pageId=login&openid.ns=http%3A%2F%2Fspecs.openid.net%2Fauth%2F2.0) и поделиться своим отзывом.

Ваш отзыв важен для нас и технического сообщества и поможет нам убедиться, что мы предоставляем контент отличного качества.