

УДК 004.42
ББК 32.973.26-018.2
К17

Object-Oriented Python: Master OOP by Building Games and GUIs
Irv Kalb

Copyright © 2022 by Irv Kalb. Title of English-language original: Object-Oriented Python: Master OOP by Building Games and GUIs, ISBN 9781718502062, published by No Starch Press Inc. 2458th Street, San Francisco, California United States 94103. The Russian-language edition. Copyright © 2024 by Eksmo Publishing House under license by No Starch Press Inc. All rights reserved.

Кальб, Ирв.

К17 **Объектно-ориентированное программирование с помощью Python / Ирв Кальб ; [перевод с английского М. А. Райтмана]. — Москва : Эксмо, 2024. — 512 с. — (Мировой компьютерный бестселлер).**

ISBN 978-5-04-186627-3

Объектно-ориентированное программирование (ООП) — это метод, основанный на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определенного класса, а классы образуют иерархию наследования, что позволяет по-другому думать о вычислительных задачах и решать их с возможностью многократного использования. «Объектно-ориентированное программирование с помощью Python» предназначено для программистов среднего уровня и представляет собой практическое руководство, которое глубоко изучает основные принципы ООП и показывает, как использовать инкапсуляцию, полиморфизм и наследование для написания игр и приложений с использованием Python.

Книга начинается с рассказа о ключевых проблемах, присущих процедурному программированию, затем вы познакомитесь с основами создания классов и объектов в Python.

Затем вы научитесь создавать графические интерфейсы с помощью pygame, благодаря чему вы сможете писать интерактивные игры и приложения с виджетами графического пользовательского интерфейса (GUI), анимацией, различными сценами и многообразной игровой логикой.

УДК 004.42
ББК 32.973.26-018.2

ISBN 978-5-04-186627-3

© Райтман М.А., перевод на русский язык, 2024
© Оформление. ООО «Издательство «Эксмо», 2024

Моей замечательной жене Дорин.

Ты клей, который держит нашу семью вместе.

Много лет назад я сказал: «Я делаю», но я имел в виду: «Я сделаю».

ОБ АВТОРЕ

Ирв Кальб – профессор в UCSC Silicon Valley Extension и Университете Кремниевой долины (ранее Политехнический колледж Когсвелла), где он преподает вводные курсы программирования и курсы объектно-ориентированного программирования на языке Python. Ирв имеет степени бакалавра и магистра в области компьютерных наук, более 30 лет занимается объектно-ориентированным программированием на различных языках и более 10 лет преподает. У него десятилетний опыт разработки программного обеспечения с акцентом на образовательное ПО. Как Furry Pants Productions он и его жена создали и выпустили два обучающих диска с персонажем – далматинцем Дарби в главной роли. Ирв также является автором *Learn to Program with Python 3: A Step-by-Step Guide to Programming* («Учимся программировать на Python 3. Пошаговое руководство по программированию»).

Ирв активно участвовал в раннем развитии спорта Ultimate Frisbee®. Он возглавил создание многих версий официального сборника правил и стал соавтором и издателем первой книги об этом виде спорта – *Ultimate: Fundamentals of the Sport* («Ultimate: Основы спорта»).

О ТЕХНИЧЕСКОМ АВТОРЕ

Монте Давидовф — независимый консультант по разработке программного обеспечения. Его области специализации включают DevOps и Linux. Монте программирует на Python уже более 20 лет. Он использовал Python для разработки разнообразного программного обеспечения, включая критически важные для бизнеса приложения и встроенное ПО.

КРАТКОЕ СОДЕРЖАНИЕ

Об авторе	6
О техническом авторе	7
Благодарности	17
Введение	19
Часть I. Введение в объектно-ориентированное программирование	29
1. Процедурные примеры Python	31
2. Моделирование физических объектов с помощью объектно-ориентированного программирования	55
3. Мысленные модели объектов и значение self	89
4. Управление несколькими объектами	101
Часть II. Графические пользовательские интерфейсы с pygame	139
5. Введение в pygame	141
6. Объектно-ориентированный pygame	183
7. Виджеты pygame GUI	211
Часть III. Инкапсуляция, полиморфизм и наследование	233
8. Инкапсуляция	235
9. Полиморфизм	263
10. Наследование	297
11. Управление памятью, используемой объектами	335
Часть IV. Использование ООП в разработке игр	363
12. Карточные игры	365
13. Таймеры	381

14. Анимация	399
15. Сцены	419
16. Полноценная игра: Dodger	457
17. Шаблоны проектирования и резюме	491
Предметный указатель	503

ПОДРОБНОЕ СОДЕРЖАНИЕ

Об авторе	6
О техническом авторе	7
Благодарности	17
Введение	19
Для кого эта книга?	20
Версия(-и) Python и установка	20
Как я объясняю ООП?	22
Структура книги	23
Среды разработки	26
Виджеты и примеры игр	26

Часть I. Введение в объектно-ориентированное программирование

29

1. Процедурные примеры Python	31
Карточная игра «Больше-меньше»	32
Представление данных	32
Реализация	33
Повторное использование кода	35
Моделирование банковского счета	36
Анализ необходимых операций и данных	36
Реализация 1. Одна учетная запись без функций	37
Реализация 2. Одна учетная запись с функциями	39
Реализация 3. Два счета	42
Реализация 4. Несколько счетов с использованием списков	44
Реализация 5. Список словарей учетных записей	47
Общие проблемы с процедурной реализацией	50
Объектно-ориентированное решение — первый взгляд на класс	51
Выводы	52
2. Моделирование физических объектов с помощью объектно-ориентированного программирования	55
Построение программных моделей физических объектов	56
Состояние и поведение: пример выключателя освещения	56
Введение в классы и объекты	58
Классы, объекты и экземпляры	60
Написание класса на Python	61

Область видимости и переменные экземпляра	63
Различия между функциями и методами	64
Создание объекта из класса	66
Вызов методов объекта	66
Создание нескольких экземпляров из одного класса	67
Типы данных Python реализованы как классы	69
Определение объекта	70
Создание несколько более сложного класса	70
Представление более сложного физического объекта как класса	73
Передача аргументов методу	79
Несколько экземпляров	81
Параметры инициализации	83
Использование классов	85
ООП как решение	86
Выводы	86
3. Мысленные модели объектов и значение self	89
Повторный обзор класса DimmerSwitch	90
Высокоуровневая мысленная модель № 1	91
Более глубокая мысленная модель № 2	92
В чем смысл слова «self»?	95
Выводы	99
4. Управление несколькими объектами	101
Класс банковского счета	101
Импорт кода класса	105
Создание тестового кода	107
Создание нескольких учетных записей	107
Несколько объектов учетной записи в списке	110
Несколько объектов с уникальными идентификаторами	112
Создание интерактивного меню	115
Создание объекта диспетчера объектов	118
Создание объекта диспетчера объектов	120
Основной код, создающий объект диспетчера объектов	123
Лучшая обработка ошибок с исключениями	125
try и except	125
Инструкция raise и пользовательские исключения	126
Использование исключений в нашей банковской программе	128
Класс счета с исключениями	128
Оптимизированный класс банка	130
Основной код, обрабатывающий исключения	132
Вызов одного и того же метода для списка объектов	134
Интерфейс по сравнению с реализацией	136
Выводы	137

Часть II. Графические пользовательские интерфейсы с pygame

139

5. Введение в pygame	141
Устанавливаем pygame	142
Детали окон	143
Система координат окна	144
Цвета пикселей	147

Программы, управляемые событиями	148
Используем Pygame	150
Создаем пустое окно	151
Рисуем изображение	155
Обнаруживаем щелчок мыши	158
Обрабатываем клавиатуру	161
Создаем анимацию, основанную на местоположении	167
Используем rect pygame	169
Воспроизводим звуки	173
Воспроизводим звуковые эффекты	173
Воспроизведение фоновой музыки	175
Рисуем фигуры	176
Справка по примитивным фигурам	179
Выводы	181
6. Объектно-ориентированный pygame	183
Создаем заставку мяча с помощью Pygame	183
Создаем класс Ball	184
Используем класс Ball	186
Создаем много объектов Ball	188
Создаем много, много объектов Ball	190
Создаем многократно используемую объектно-ориентированную кнопку	190
Создаем класс кнопки	191
Основной код, использующий SimpleButton	194
Создаем программу с несколькими кнопками	196
Создаем многократно используемое отображение текста	197
Шаги для отображения текста	198
Создаем класс SimpleText	199
Демоверсия Ball с SimpleText и SimpleButton	201
Сравнение интерфейса и реализации	203
Обратные вызовы	204
Создаем обратный вызов	205
Используем обратный вызов с SimpleButton	206
Выводы	209
7. Виджеты pygame GUI	211
Передаем аргументы функции или методу	212
Позиционные параметры и параметры ключевых слов	213
Дополнительные примечания к параметрам ключевых слов	214
Используем None в качестве значения по умолчанию	216
Выбираем ключевые слова и значения по умолчанию	217
Значения по умолчанию в виджетах GUI	218
Пакет pygame.widgets	218
Установка	219
Общий подход к разработке	220
Добавляем изображение	221
Добавляем кнопки, флажки и переключатели	222
Вывод и ввод текста	226
Другие классы pygame.widgets	229
pygame.widgets в примере программы	231
Важность последовательного API	231
Выводы	231

8. Инкапсуляция	235
Инкапсуляция с помощью функций	236
Инкапсуляция с помощью объектов	237
Объекты владеют своими данными	237
Интерпретации инкапсуляции	238
Прямой доступ и почему его следует избегать	238
Строгая интерпретация с помощью геттеров и сеттеров	244
Безопасный прямой доступ	246
Делаем переменные экземпляра более закрытыми	247
Неявно закрытый	247
Более явно закрытый	248
Декораторы и @property	249
Инкапсуляция в классах <code>pygame</code>	254
История из реального мира	255
Абстракция	257
Выводы	260
9. Полиморфизм	263
Отправляем сообщения объектам реального мира	264
Классический пример полиморфизма в программировании	265
Пример, использующий фигуры <code>pygame</code>	266
Класс квадратной формы	267
Класс круглой и треугольной формы	268
Основная программа, создающая фигуры	272
Расширяем шаблон	274
<code>pygame</code> проявляет полиморфизм	275
Полиморфизм для операторов	276
Магические методы	277
Магические методы оператора сравнения	278
Магические методы в классе <code>Rectangle</code>	280
Использование магических методов основной программой	282
Магические методы математических операторов	284
Векторный пример	285
Создаем строковое представление значений в объекте	288
Класс <code>Fraction</code> с магическими методами	291
Выводы	295
10. Наследование	297
Наследование в объектно-ориентированном программировании	298
Реализуем наследование	300
Пример работника и менеджера	301
Базовый класс: работник	301
Подкласс: менеджер	302
Тестовый код	305
Представление клиента о подклассе	306
Примеры наследования из реального мира	307
<code>InputNumber</code>	308
<code>DisplayMoney</code>	311
Пример использования	314
Наследование нескольких классов от одного базового класса	317

Абстрактные классы и методы	323
Как pygame применяет наследование	327
Иерархия классов	329
Сложность программирования с наследованием	331
Выводы	333
11. Управление памятью, используемой объектами	335
Жизненный цикл объекта	335
Подсчет ссылок	336
Сбор мусора	343
Переменные класса	343
Константы переменных класса	344
Переменные класса для подсчета	345
Собираем все воедино: пример программы «Шары»	347
Модуль констант	349
Код основной программы	350
Менеджер шаров	353
Класс шаров и объекты	356
Управляем памятью: слоты	359
Выводы	362

Часть IV. Использование ООП в разработке игр 363

12. Карточные игры	365
Класс Card	366
Класс Deck	369
Игра «Больше-меньше»	371
Основная программа	372
Объект Game	373
Тестируем с помощью <code>__name__</code>	377
Другие карточные игры	379
Колода для блек-джека	379
Игры с необычными колодами	379
Выводы	380
13. Таймеры	381
Демонстрационная программа таймера	382
Три подхода к реализации таймеров	383
Подсчет фреймов	383
Таймер событий	384
Создаем таймер путем вычисления прошедшего времени	386
Устанавливаем pygame helpers	388
Класс Timer	389
Отображаем время	392
CountUpTimer	393
CountDownTimer	397
Выводы	398
14. Анимация	399
Создаем классы анимации	400
Класс SimpleAnimation	400
Класс SimpleSpriteSheetAnimation	405
Объединяем два класса	410

Классы анимации в pygame	411
Класс Animation	412
Класс SpriteSheetAnimation	413
Общий базовый класс: PygAnimation	415
Пример программы анимации	416
Выводы	418
15. Сцены	419
Концепция конечного автомата	420
Пример pygame с конечным автоматом	423
Демоверсия программы, использующая менеджер сцен	430
Основная программа	432
Создаем сцены	433
Типичная сцена	437
Игра «Камень-ножницы-бумага», использующая сцены	439
Взаимодействие между сценами	444
Запрашиваем информацию у целевой сцены	445
Отправляем информацию целевой сцене	446
Отправляем информацию всем сценам	446
Проверяем взаимодействие между сценами	447
Реализация менеджера сцен	447
Метод run()	449
Основные методы	451
Взаимодействие между сценами	453
Выводы	454
16. Полноценная игра: Dodger	457
Модальные диалоговые окна	458
Диалоговые окна Yes/No и Предупреждения	458
Диалоговые окна с ответом	462
Создаем полноценную игру: Dodger	465
Обзор игры	465
Реализация	466
Дополнения к игре	488
Выводы	489
17. Шаблоны проектирования и резюме	491
Модель Представление Контроллер	492
Пример отображения файла	492
Пример статистического отображения	493
Преимущества шаблона MVC	499
Резюме	500
Предметный указатель	503

БЛАГОДАРНОСТИ

Я хотел бы поблагодарить людей, которые помогли создать эту книгу:

- Эла Свейгарта — за то, что приучил меня к `pygame` (особенно за его код `Pygbutton`), и за то, что позволил мне использовать концепцию его игры `Dodger`;
- Монте Давидовфа, который помог мне получить исходный код и документацию к нему для правильной сборки с помощью `GitHub`, `Sphinx` и `ReadTheDocs`. Этот человек творил чудеса, используя бесчисленное множество инструментов, чтобы управляться с файлами;
- Монте Давидовфа (да, это все тот же парень) — за то, что оказался выдающимся техническим рецензентом. Монте дал отличные технические и авторские предложения по всей книге, и многие примеры кода стали более `Pythonic` и более ООП-ориентированными именно после его комментариев;
- Тепа Сатъя Кхиейу, который проделал потрясающую работу, нарисовав все оригинальные схемы для этой книги.
Я не художник (я даже не играю ни одного на ТВ). Теп смог взять мои примитивные карандашные наброски и превратить их в четкие, последовательные произведения искусства;

- Харрисона Янга, Кевина Лая и Эмили Эллис — за их вклад в художественное оформление некоторых игр;
- ранних рецензентов: Илью Кацюка, Джейми Калб, Гергану Анджелову и Джо Лангмюра, которые нашли и устранили много опечаток и внесли отличные предложения по исправлениям и уточнениям;
- всех редакторов, которые работали над этой книгой: Лиз Чедвик (редактор по развитию), Рейчел Хед (редактор) и Кейт Камински (производственный редактор). Все они внесли огромный вклад, задавая вопросы и часто переписывая и реорганизуя некоторые из моих объяснений. Они также были чрезвычайно полезны в расставлении и удалении запятых [нужна ли она здесь?] и удлинении моих предложений, как здесь, чтобы убедиться, что точка встречается там, где нужна (ОК, я остановлюсь!). Я боюсь, что никогда не пойму, где следует использовать «который», а не «что», или как расставлять запятые и тире, но я рад, что они знают! Спасибо также Морин Форис (верстальщице) за ее ценный вклад в готовый продукт;
- всех студентов, которые побывали на моих занятиях в течение многих лет в UCSC Silicon Valley Extension и в Университете Кремниевой долины (ранее Политехнический колледж Когсвелла). Их отзывы, предложения, улыбки, недовольство, моменты озарения, разочарование, понимающие кивки и даже большие пальцы вверх (в классах Zoom в эпоху ковида) были чрезвычайно полезны в формировании содержания этой книги и моего общего стиля преподавания;
- наконец, мою семью, которая поддерживала меня в длительном процессе написания, тестирования, редактирования, переписывания, редактирования, отладки, редактирования, переписывания, редактирования (и так далее) этой книги и связанного с ней кода. Без вас я бы не справился. Я не был уверен, достаточно ли у нас книг в библиотеке, поэтому написал еще одну!