

Краткое содержание

Глава 1. Простые задачи	25
Глава 2. Задачи поиска	48
Глава 3. Задачи с ограничениями.....	80
Глава 4. Графовые задачи.....	99
Глава 5. Генетические алгоритмы	128
Глава 6. Кластеризация методом k-средних.....	149
Глава 7. Простейшие нейронные сети	167
Глава 8. Составительный поиск	197
Глава 9. Другие задачи	220
Приложение А. Глоссарий.....	236
Приложение Б. Дополнительные ресурсы.....	242
Приложение В. Коротко об аннотациях типов.....	247

Оглавление

Благодарности	12
Об этой книге	14
Торговые марки.....	14
Форум этой книги	14
Об авторе	15
Об иллюстрации на обложке	16
От издательства	18
Введение	19
Почему именно Python.....	20
Что такое классическая задача программирования.....	20
Какие задачи представлены в этой книге	21
Для кого эта книга	22
Версии Python, хранилище исходного кода и аннотации типов	22
Никакой графики и пользовательских интерфейсов — только стандартная библиотека.....	23
Книги этой серии.....	24
Глава 1. Простые задачи	25
1.1. Ряд Фибоначчи.....	25
1.1.1. Первый вариант рекурсии.....	25
1.1.2. Использование базовых случаев	27

1.1.3. Спасение — в мемоизации	29
1.1.4. Автоматическая мемоизация	30
1.1.5. Будьте проще, Фибоначчи!	30
1.1.6. Генерация чисел Фибоначчи с помощью генератора	31
1.2. Простейшее сжатие	32
1.3. Невскрываемое шифрование	37
1.3.1. Получение данных в заданной последовательности	38
1.3.2. Шифрование и дешифрование	39
1.4. Вычисление числа π	41
1.5. Ханойские башни	42
1.5.1. Моделирование башен	42
1.5.2. Решение задачи о ханойских башнях	44
1.6. Реальные приложения	46
1.7. Упражнения	47
Глава 2. Задачи поиска	48
2.1. Поиск ДНК	48
2.1.1. Хранение ДНК	48
2.1.2. Линейный поиск	50
2.1.3. Бинарный поиск	51
2.1.4. Параметризованный пример	54
2.2. Прохождение лабиринта	56
2.2.1. Создание случайного лабиринта	57
2.2.2. Мелкие детали лабиринта	58
2.2.3. Поиск в глубину	59
2.2.4. Поиск в ширину	63
2.2.5. Поиск по алгоритму A^*	67
2.3. Миссионеры и людоеды	73
2.3.1. Представление задачи	74
2.3.2. Решение	76
2.4. Реальные приложения	78
2.5. Упражнения	78
Глава 3. Задачи с ограничениями	80
3.1. Построение структуры для задачи с ограничениями	81
3.2. Задача раскраски карты Австралии	85
3.3. Задача восьми ферзей	88
3.4. Поиск слова	91

3.5. SEND + MORE = MONEY.....	94
3.6. Размещение элементов на печатной плате	96
3.7. Реальные приложения.....	97
3.8. Упражнения	98

Глава 4. Графовые задачи..... 99

4.1. Карта как граф.....	99
4.2. Построение графовой структуры	102
4.2.1. Работа с Edge и Graph.....	106
4.3. Поиск кратчайшего пути.....	108
4.3.1. Пересмотр алгоритма поиска в ширину.....	108
4.4. Минимизация затрат на построение сети.....	110
4.4.1. Работа с весами	110
4.4.2. Поиск минимального связующего дерева	114
4.5. Поиск кратчайших путей во взвешенном графе.....	121
4.5.1. Алгоритм Дейкстры.....	121
4.6. Реальные приложения.....	126
4.7. Упражнения	127

Глава 5. Генетические алгоритмы 128 |

5.1. Немного биологической теории.....	128
5.2. Обобщенный генетический алгоритм.....	130
5.3. Примитивный тест.....	137
5.4. SEND + MORE = MONEY, улучшенный вариант.....	140
5.5. Оптимизация сжатия списка.....	143
5.6. Проблемы генетических алгоритмов.....	146
5.7. Реальные приложения.....	147
5.8. Упражнения	148

Глава 6. Кластеризация методом k-средних..... 149

6.1. Предварительные сведения.....	150
6.2. Алгоритм кластеризации k-средних	152
6.3. Кластеризация губернаторов по возрасту и долготе штата.....	158
6.4. Кластеризация альбомов Майкла Джексона по длительности	163
6.5. Проблемы и расширения кластеризации методом k-средних.....	165
6.6. Реальные приложения.....	166
6.7. Упражнения	166

Глава 7. Простейшие нейронные сети	167
7.1. В основе — биология?	168
7.2. Искусственные нейронные сети	170
7.2.1. Нейроны	170
7.2.2. Слои	171
7.2.3. Обратное распространение	172
7.2.4. Ситуация в целом	176
7.3. Предварительные замечания	176
7.3.1. Скалярное произведение	177
7.3.2. Функция активации	177
7.4. Построение сети	179
7.4.1. Реализация нейронов	179
7.4.2. Реализация слоев	180
7.4.3. Реализация сети	182
7.5. Задачи классификации	185
7.5.1. Нормализация данных	186
7.5.2. Классический набор данных радужной оболочки	187
7.5.3. Классификация вина	190
7.6. Повышение скорости работы нейронной сети	192
7.7. Проблемы и расширения нейронных сетей	193
7.8. Реальные приложения	195
7.9. Упражнения	196
Глава 8. Состязательный поиск	197
8.1. Основные компоненты настольной игры	197
8.2. Крестики-нолики	199
8.2.1. Управление состоянием игры в крестики-нолики	200
8.2.2. Минимакс	203
8.2.3. Тестирование минимакса для игры в крестики-нолики	206
8.2.4. Разработка ИИ для игры в крестики-нолики	208
8.3. Connect Four	209
8.3.1. Подключите четыре игровых автомата	209
8.3.2. ИИ для Connect Four	214
8.3.3. Улучшение минимакса с помощью альфа-бета-отсечения	215
8.4. Другие улучшения минимакса	217
8.5. Реальные приложения	218
8.6. Упражнения	218

Глава 9. Другие задачи	220
9.1. Задача о рюкзаке	220
9.2. Задача коммивояжера	226
9.2.1. Наивный подход	226
9.2.2. Переходим на следующий уровень.....	230
9.3. Мнемоника для телефонных номеров.....	231
9.4. Реальные приложения.....	234
9.5. Упражнения	234
Приложение А. Глоссарий.....	236
Приложение Б. Дополнительные ресурсы.....	242
Б.1. Python	242
Б.2. Алгоритмы и структуры данных.....	243
Б.3. Искусственный интеллект.....	244
Б.4. Функциональное программирование.....	245
Б.5. Полезные проекты с открытым исходным кодом для машинного обучения	245
Приложение В. Коротко об аннотациях типов.....	247
В.1. Что такое аннотации типов	247
В.2. Как выглядят аннотации типа.....	248
В.3. Почему полезны аннотации типов	249
В.4. Каковы недостатки аннотаций типов	251
В.5. Источники дополнительной информации.....	252

Посвящается моей бабушке Эрминии Антос,
которая всю жизнь училась сама и учила других.

Благодарности

Я благодарю всех сотрудников издательства Manning, которые помогали в создании этой книги: Шерил Вейсман, Дирдре Хайам, Кэти Теннант, Дотти Марсико, Дженет Вейл, Барбару Мирецки, Александра Драгосавлевича, Мэри Пирджис и Марию Тюдор.

Я благодарю специалиста по отбору новых книг Брайана Сойера, который предложил нам переключиться на Python после того, как я закончил книгу о Swift. Спасибо редактору по развитию Дженнифер Стаут за неизменно доброжелательное отношение. Благодарю научного редактора Фрэнсис Буонтемпо — она внимательно изучила каждую главу и на каждом шагу давала подробные полезные комментарии. Я благодарю литературного редактора Энди Кэрролла, чье исключительное внимание к деталям как в книге о Swift, так и в этой позволило отловить несколько допущенных мной ошибок. Спасибо также корректору Хуану Руфесу.

Эту книгу рецензировали Эл Кринкер, Эл Пезевски, Алан Богусевич, Брайан Канада, Крейг Хендерсон, Дэниэл Кенни-Юнг, Эдмонд Сисей, Ева Барановска, Гэри Барнхарт, Джефф Кларк, Джеймс Уотсон, Джефффри Лим, Иенс Кристиан, Бредал Мэдсен, Хуан Хименес, Хуан Руфес, Мэтт Лемке, Майур Патил, Майкл Брайт, Роберто Касадей, Сэм Зейдел, Торстен Вебер, Том Джеффрис и Уилл Лопес. Спасибо всем, кто высказал конструктивную и конкретную критику во время создания этого издания. Ваши отзывы были учтены.

Я благодарю свою семью, друзей и коллег, которые вдохновили меня взяться за написание этой книги сразу же после публикации *Classic Computer Science Problems in Swift*. Я благодарю своих подписчиков в «Твиттере» и в других сетях за то, что поддерживали меня и помогали в работе над книгой — в большом и малом. Огромное спасибо моей жене Ребекке Копец и маме Сильвии Копец, которые всегда поддерживают все мои проекты.

Мы создали эту книгу за довольно короткое время. Основная ее часть была написана летом 2018 года. Я ценю, что издательство Manning согласилось сократить свой обычно гораздо более длительный технологический процесс, чтобы я мог работать в удобном для себя режиме. Я знаю, что это затронуло всю команду, поскольку всего за несколько месяцев мы провели три этапа проверок на разных уровнях с привлечением множества разных людей. Большинство читателей были бы поражены тем, сколько разных видов проверок проходит техническая книга в обычном издательстве и какое количество людей принимает участие в ее рецензировании и пересмотре. Я благодарю всех работников издательства, всех официальных рецензентов и всех, кто участвовал в процессе подготовки издания.

Наконец, и это самое главное, я благодарю своих читателей за покупку книги. Я думаю, что в мире, полном унылых электронных учебников, важно поддерживать выпуск бумажных книг, позволяющих автору зазвучать в полный голос. Онлайн-учебники могут быть отличным ресурсом, но благодаря вашему спросу полноформатные, проверенные и тщательно сверстанные бумажные книги по-прежнему занимают свое место в обучении программированию.

Об этой книге

Торговые марки

В издании упомянуты различные торговые марки. Символ торговой марки при каждом появлении фирменного названия не ставится, так как эти названия используются только в целях издания и в интересах владельца товарного знака, без намерения посягнуть на товарный знак. Слово Python является зарегистрированным товарным знаком Python Software Foundation. Connect Four — торговая марка Hasbro, Inc.

Форум этой книги

Покупка книги включает бесплатный доступ к частному веб-форуму, организованному издательством Manning Publications, где можно оставлять комментарии о книге, задавать технические вопросы, а также получать помощь от автора и других пользователей. Чтобы получить доступ к форуму, перейдите на <https://www.manning.com/books/classic-computerscience-problems-in-Python>. Узнать больше о других форумах на сайте издательства Manning и познакомиться с правилами вы сможете на странице <https://forums.manning.com/forums/about>.

Издательство Manning обязуется предоставить своим читателям место встречи, на которой может состояться содержательный диалог между отдельными читателями и между читателями и автором. Однако со стороны автора отсутствуют какие-либо обязательства уделять форуму внимание в каком-то определенном объеме — его присутствие на форуме остается добровольным (и неоплачиваемым). Мы предлагаем задавать автору неожиданные вопросы, чтобы его интерес не угасал! Форум и архивы предыдущих обсуждений будут доступны на сайте издательства, пока книга находится в печати.

Об авторе



Дэвид Копец — старший преподаватель на кафедре компьютерных наук и инноваций в колледже Шамплейн в Берлингтоне, штат Вермонт. Он опытный разработчик программного обеспечения и автор книг *Classic Computer Science Problems in Swift* (Manning, 2018) и *Dart for Absolute Beginners* (Apress, 2014). Дэвид получил степень бакалавра экономики и степень магистра компьютерных наук в Дартмутском колледже. Вы можете связаться с ним в «Твиттере» по имени @davekopec.

Об иллюстрации на обложке

Иллюстрация на обложке называется «Одежда китайского бонзы или священника»¹ и позаимствована из книги «Коллекция платья разных народов, старинного и современного»², изданного в Лондоне в 1757–1772 годах. Как указано на титульном листе, это гравюры, выполненные на медных пластинах и раскрашенные гуммиарабиком.

Томаса Джеффериса (1719–1771) называли географом короля Георга III. Он был английским картографом, ведущим поставщиком карт своего времени. Он гравировал и печатал карты для правительственных и других государственных учреждений, выпускал широкий спектр коммерческих карт и атласов, особенно Северной Америки. В ходе работы интересовался особенностями одежды населения тех земель, которые обследовал и нанес на карту. Зарисовки костюмов блестяще представлены в этом издании. В конце XVIII века увлечение далекими землями и путешествия ради удовольствия были относительно новым явлением, и коллекции, подобные этой, были популярны, позволяя как туристам, так и тем, кто путешествует, не вставая с кресла, познакомиться с жителями других стран.

Разнообразие иллюстраций в книгах Джеффериса — яркое свидетельство уникальности и оригинальности народов мира в то время. С тех пор тенденции в одежде сильно изменились, а региональные и национальные различия, которые

¹ Habit of a Bonza or Priest in China.

² A Collection of the Dresses of Different Nations, Ancient and Modern.

были такими значимыми 200 лет назад, постепенно сошли на нет. В наши дни часто сложно отличить друг от друга жителей разных континентов. Оптимисты могут сказать, что взамен культурному и визуальному многообразию мы получили более насыщенную и интересную личную жизнь (или по крайней мере ее интеллектуальную и техническую стороны).

В то время как большинство компьютерных изданий мало чем отличаются друг от друга, компания Manning выбирает для своих книг обложки, показывающие богатое региональное разнообразие, которое Джефферис воплотил в своих иллюстрациях два столетия назад. Это ода находчивости и инициативности современной компьютерной индустрии.

От издательства

Ваши замечания, предложения, вопросы отправляйте по адресу comp@piter.com (издательство «Питер», компьютерная редакция).

Мы будем рады узнать ваше мнение!

На веб-сайте издательства www.piter.com вы найдете подробную информацию о наших книгах.

Введение

Благодарю вас за покупку книги «Классические задачи Computer Science на языке Python». Python — один из самых популярных языков программирования, и программистами на Python становятся специалисты из абсолютно разных областей. У некоторых из них есть специальное техническое образование, другие изучают Python на досуге, третьи используют его в профессиональной среде, но не для разработки программного обеспечения. Задачи, описанные в этой книге, среднего уровня. С их помощью опытные программисты смогут освежить свои знания в области информатики, изучив некоторые расширенные возможности языка. Программисты-самоучки смогут быстрее обучиться программированию, рассматривая решение классических задач на выбранном языке — Python. Эта книга охватывает такое разнообразие методов решения задач, что каждый найдет в ней что-то для себя.

Это издание не является введением в Python. На эту тему есть множество других превосходных книг, выпущенных как в Manning, так и в других издательствах¹. В данной книге предполагается, что вы уже программист среднего или высокого уровня на Python. Для примеров использовалась версия Python 3.7, но вам не нужно знать все тонкости последней версии языка. Фактически примеры были подобраны так, чтобы они послужили учебным материалом, помогающим читателям достичь высокого уровня мастерства. Однако эта книга не подходит для читателей, совершенно не знакомых с Python.

¹ Если вы только начинаете знакомство с Python, то, возможно, предварительно захотите прочитать книгу Наоми Седер *The Quick Python Book* (Naomi Ceder, 3rd ed. Manning, 2018).

Почему именно Python

Python используется в разных сферах деятельности: в анализе и обработке данных, кинопроизводстве, компьютерном обучении, управлении в сфере информационных технологий и многих других. В сущности, нет области применения программирования, в которой бы он не задействовался (за исключением, может быть, разработки ядер операционных систем). Python любят за гибкость, красивый и лаконичный синтаксис, объектно-ориентированную чистоту и кипучую деятельность сообщества программистов. Активное сообщество важно — это означает, что Python приветствует новичков и в нем предусмотрена обширная система доступных библиотек для разработчиков.

Из-за всего этого Python иногда рассматривается как язык для начинающих, и такая характеристика, вероятно, верна. Большинство людей согласятся с тем, что Python легче изучать, чем, например, C++, и сообщество этого языка действительно более дружелюбно относится к новичкам. В результате многие осваивают Python, потому что он доступен, и начинают писать на нем программы, которые желательно получить довольно быстро. Но эти люди, возможно, никогда не учились информатике (computer science) и не знают о существовании эффективных методов решения задач. Если вы один из тех программистов, которые знают Python, но не знакомы с информатикой, то эта книга для вас.

Другие изучают Python как второй, третий, четвертый или пятый язык после долгой работы в области создания программного обеспечения. Для них встреча с задачами, которые уже приходилось решать на другом языке, поможет ускорить изучение Python. А эта книга может стать хорошим способом освежить знания перед собеседованием или раскрыть отдельные методы решения задач, которые они раньше не предполагали использовать в своей работе. Я бы посоветовал таким читателям просмотреть оглавление, чтобы увидеть, есть ли в этой книге темы, которые их интересуют.

Что такое классическая задача программирования

Есть мнение, что компьютеры в информатике — то же самое, что телескопы в астрономии. Если это так, то, возможно, язык программирования подобен объективу телескопа. В любом случае термин «классические задачи программирования» здесь означает «задачи программирования, которые обычно преподают в курсе информатики для студентов».

Существуют задачи программирования, которые предлагают решить начинающим программистам и которые уже стали достаточно распространенными, чтобы считаться классическими, как в аудитории во время экзамена на степень бакалавра (в области информатики, разработки программного обеспечения и т. п.), так и в рамках учебника среднего уровня по программированию (например, в книге

начального уровня по искусственному интеллекту или алгоритмам). В данном издании вы найдете набор именно таких задач.

Задачи варьируются от тривиальных, которые решаются написанием нескольких строк кода, до сложных, требующих построения систем на протяжении нескольких глав. Одни относятся к области искусственного интеллекта, другие требуют простого здравого смысла. Некоторые из них практичны, другие — причудливы.

Какие задачи представлены в этой книге

Глава 1 познакомит вас с методами решения задач, которые, вероятно, известны большинству читателей. Такие вещи, как рекурсия, запоминание и манипулирование битами, являются важными строительными блоками других методов, которые будут рассмотрены в последующих главах.

За этим плавным вступлением идет глава 2, посвященная задачам поиска. Тема поиска столь обширна, что под ее заголовком, вероятно, можно поместить большинство задач этой книги. В главе 2 представлены наиболее важные алгоритмы поиска, включая бинарный поиск, поиск в глубину, поиск в ширину и A^* . Эти алгоритмы будут часто использоваться в остальной части книги.

В главе 3 мы построим структуру для решения широкого круга задач, которые могут быть абстрактно определены переменными ограниченными областями изменения (limited domains). Это такая классика, как задача восьми ферзей, задача о раскрашивании карты Австралии и криптоарифметика $SEND + MORE = MONEY$.

В главе 4 исследуется мир графовых алгоритмов, применение которых оказывается удивительно широким для непосвященных. В этой главе вы построите графовую структуру данных и с ее помощью решите несколько классических задач оптимизации.

В главе 5 исследуются генетические алгоритмы — менее детерминистская методика, чем большинство описанных в этой книге. Она иногда позволяет решить задачи, с которыми традиционные алгоритмы не способны справиться за разумное время.

Глава 6 посвящена кластеризации *k-средних* и, пожалуй, является самой алгоритмически специфичной главой в этой книге. Данный метод кластеризации прост в реализации, легок для понимания и широко применим.

Глава 7 призвана объяснить, что такое нейронная сеть, и дать читателю представление о том, как выглядит простейшая нейронная сеть. Здесь не ставится цель всесторонне раскрыть эту захватывающую развивающуюся область. В этой главе вы построите нейронную сеть на чисто теоретической основе, не используя внешние библиотеки, чтобы по-настоящему прочувствовать, как работает нейронная сеть.

Глава 8 посвящена состязательному поиску в идеальных информационных играх для двух игроков. Вы изучите алгоритм поиска, известный как минимакс,

который можно применять для разработки искусственного противника, способного хорошо играть в такие игры, как шахматы, шашки и Connect Four.

Глава 9 охватывает интересные и забавные задачи, которые не совсем вписываются в остальные главы этой книги.

Для кого эта книга

Эта книга предназначена для программистов среднего и высокого уровня. Опытные специалисты, которые хотят углубить свое знание Python, найдут здесь задачи, приятно знакомые со времен обучения информатике или программированию. Программисты среднего уровня познакомятся с этими классическими задачами на выбранном языке — Python. Для разработчиков, которые готовятся к собеседованию по программированию, издание, скорее всего, станет ценным подготовительным материалом.

Кроме профессиональных программистов, эту книгу могут счесть полезной студенты, обучающиеся по программам бакалавриата по информатике и интересующиеся Python. Она не претендует на роль строгого введения в структуры данных и алгоритмы. *Это не учебник по структурам данных и алгоритмам.* Вы не найдете на ее страницах доказательств теорем или обильного использования нотаций O большого ($big-O$). Напротив, эта книга позиционируется как доступное практическое руководство по методам решения задач, которые должны стать конечным продуктом изучения структуры данных, алгоритмов и классов искусственного интеллекта.

Подчеркну еще раз: предполагается, что читателям знакомы синтаксис и семантика Python. Читатель с нулевым опытом программирования едва ли извлечет пользу из этой книги, а программисту с нулевым опытом в Python наверняка будет трудно. Другими словами, «Классические задачи Computer Science на языке Python» — это книга для программистов, работающих на Python, и студентов, изучающих информатику.

Версии Python, хранилище исходного кода и аннотации типов

Исходный код в этой книге написан в соответствии с версией 3.7 языка Python. В нем используются функции, которые стали доступными только в Python 3.7, поэтому часть кода не будет работать на более ранних версиях Python. Вместо того чтобы пытаться заставить примеры работать в более ранней версии, просто загрузите последнюю версию Python, прежде чем начинать работу с книгой.

В этой книге применяется только стандартная библиотека Python (за небольшим исключением — в главе 2 установлен модуль `typing_extensions`), поэтому весь код этой книги должен работать на любой платформе, где поддерживается Python (macOS, Windows, GNU/Linux и т. п.). Код был протестирован только на CPython

(основной интерпретатор Python, доступный на python.org), хотя, скорее всего, бо́льшая его часть будет работать в любой версии другого интерпретатора Python, совместимой с Python 3.7.

В книге не объясняется, как использовать инструменты Python, такие как редакторы, IDE, отладчики и Python REPL. Исходный код книги доступен в Интернете в репозитории GitHub по адресу <https://github.com/davecom/ClassicComputerScienceProblemsInPython>. Исходный код разделен на папки по главам. Имена исходных файлов вы увидите в тексте глав в заголовках листингов кода. Эти исходные файлы находятся в соответствующих папках хранилища. Для того чтобы запустить выполнение задачи, просто введите `python3 имяфайла.py` или `python имяфайла.py` в зависимости от того, как на вашем компьютере называется интерпретатор Python 3.

Во всех листингах кода в этой книге задействуются аннотации (иногда говорят подсказки) типов Python, также известные как сигнатуры типов. Это относительно новая функция языка Python, и она может смутить программистов Python, которые никогда не встречали ее раньше. Аннотации типов применяются по следующим причинам.

1. Они дают ясное представление о типах переменных, параметрах функций и возвращаемых функциями значений.
2. Вследствие п. 1 они в некотором смысле автоматически документируют код. Вместо того чтобы искать возвращаемый тип функции в комментарии или строке документации, можно просто посмотреть на ее сигнатуру.
3. Они позволяют проверять корректность кода. Одно из популярных средств проверки типов в Python — `mypy`.

Не всем нравятся аннотации типов, и решение использовать их в книге было, честно говоря, рискованным. Я надеюсь, что они скорее помогут, чем помешают. Писать программу на Python с аннотациями типов немного дольше, но при чтении кода это обеспечивает бо́льшую ясность. Интересно отметить, что аннотации типов не влияют на фактическое выполнение кода в интерпретаторе Python. Вы можете удалить их из любого кода в этой книге, и он все равно должен работать. Если вы никогда прежде не встречали аннотации типов и чувствуете, что вам нужно более подробно с ними познакомиться, прежде чем углубиться в книгу, обратитесь к приложению В, в котором содержится ускоренный курс по аннотациям типов.

Никакой графики и пользовательских интерфейсов — только стандартная библиотека

В этой книге нет примеров, которые формировали бы графический вывод или использовали графический пользовательский интерфейс (GUI). Почему? Цель ее состоит в том, чтобы решить поставленные задачи с помощью максимально кратких и удобочитаемых методов. Зачастую вывод графики мешает или делает решения

значительно более сложными, чем нужно, чтобы проиллюстрировать рассматриваемые методику или алгоритм.

Кроме того, если не применять какую-либо платформу с графическим интерфейсом, то весь код, представленный в этой книге, становится в высшей степени переносимым. Он может так же легко работать на дистрибутиве Python, встроенном в Linux, как и на компьютере под управлением Windows. Кроме того, было принято сознательное решение использовать пакеты из стандартной библиотеки Python вместо любых внешних библиотек, как это делается в большинстве книг по углубленному изучению Python. Почему? Потому что цель этой книги состоит в том, чтобы научить читателя методам решения задач на основе базовых принципов, а не способом «установить решение командой `pip install`». Я надеюсь, что благодаря необходимости решать каждую задачу с нуля вы начнете понимать, что происходит внутри популярных библиотек. Как минимум, работая в этой книге только со стандартной библиотекой, мы получим лучше переносимый и более простой в применении код.

Это не означает, что в некоторых случаях графические решения более наглядны для реализации выбранного алгоритма, чем текстовые решения. Просто они не являются темой этой книги, и их использование лишь усложнило бы ее.

Книги этой серии

Это вторая книга из серии «Классические задачи Computer Science», опубликованной издательством Manning. Первой была *Classic Computer Science Problems in Swift*, вышедшая в 2018 году. В каждой книге этой серии мы стремимся представить видение конкретного языка в свете решения одних и тех же (в основном) задач программирования.

Если вам понравится эта книга и вы решите изучить другой язык, описанный в одной из книг серии, переход от одной к другой может оказаться простым способом улучшить знания этого языка. В настоящее время серия охватывает только Swift и Python. Первые две книги я написал сам, потому что имею значительный опыт работы с обоими языками, но мы уже обсуждаем создание новых книг серии в соавторстве с людьми, которые являются экспертами в других языках. Если вам понравится эта книга, я призываю вас следить за выходом других изданий серии. Для получения дополнительной информации о них посетите сайт <https://classicproblems.com/>.