

Оглавление

Предисловие	24
Благодарности	26
О книге	27
Для кого эта книга	27
Структура книги	27
О коде в книге	29
Форум LiveBook.....	30
Другие источники информации	30
Об авторе	31
Иллюстрация на обложке.....	32
От издательства	33
Глава 1. Разработка стратегии прагматичного обучения.....	34
1.1. О пользе прагматичного подхода	35
1.1.1. Написание удобочитаемого кода Python.....	36
1.1.2. Думайте о опроверждаемости еще до написания кода.....	36
1.2. Что Python делает хорошо — или не хуже, чем другие языки	38
1.3. Чего Python не делает или что делает недостаточно хорошо	40
1.4. О чем вы узнаете из книги	41
1.4.1. Ориентация на предметно-независимые знания	41

1.4.2. Решение задач посредством синтеза	43
1.4.3. Изучение навыков в контексте.....	43
Итоги.....	44

Часть 1

Использование встроенных моделей данных

Глава 2. Обработка и форматирование строк 46

2.1. Как использовать f-строки для интерполяции и форматирования.....	47
2.1.1. Форматирование строк до появления f-строк	47
2.1.2. Использование f-строк для интерполяции переменных	49
2.1.3. Использование f-строк для интерполяции выражений.....	50
2.1.4. Применение спецификаторов для форматирования f-строк.....	52
2.1.5. Обсуждение.....	57
2.1.6. Задача	58
2.2. Как преобразовать строки для получения представляемых данных	58
2.2.1. Проверка строк на представление алфавитно-цифровых значений	59
2.2.2. Преобразование строк в числа.....	60
2.2.3. Вычисление строк для получения представляемых данных	62
2.2.4. Обсуждение.....	64
2.2.5. Задача	65
2.3. Как объединять и разбивать строки	65
2.3.1. Объединение строк с пробельными символами.....	66
2.3.2. Объединение строк без ограничителей.....	67
2.3.3. Разбиение строк для создания списка	69
2.3.4. Обсуждение.....	71
2.3.5. Задача	71
2.4. Какие возможности предоставляют регулярные выражения	71
2.4.1. Работа с регулярными выражениями в Python.....	72
2.4.2. Определение шаблона в виде необработанной строки	74
2.4.3. Основы синтаксиса регулярных выражений	75
2.4.4. Анализ совпадений.....	79
2.4.5. Часто используемые методы	81
2.4.6. Обсуждение.....	82
2.4.7. Задача	82

2.5. Как использовать регулярные выражения для обработки текста.....	82
2.5.1. Создание шаблона для поиска совпадений.....	83
2.5.2. Извлечение данных из совпадений.....	84
2.5.3. Использование именованных групп для обработки текста	85
2.5.4. Обсуждение.....	86
2.5.5. Задача.....	87
Итоги.....	87
Глава 3. Встроенные контейнеры данных.....	88
3.1. Как выбрать между списком и кортежем	89
3.1.1. Кортежи для неизменяемых данных, списки — для изменяемых данных	89
3.1.2. Кортежи для разнородных, списки для однородных данных.....	91
3.1.3. Обсуждение.....	92
3.1.4. Задача.....	93
3.2. Как сортировать списки сложных данных с помощью специализированных функций.....	93
3.2.1. Сортировка списков в порядке по умолчанию	94
3.2.2. Использование встроенной функции как ключа сортировки.....	94
3.2.3. Использование нестандартных функций для более сложных задач сортировки.....	95
3.2.4. Обсуждение.....	97
3.2.5. Задача.....	97
3.3. Как построить облегченную модель данных с использованием именованных кортежей	97
3.3.1. Альтернативные модели данных.....	98
3.3.2. Создание именованных кортежей для хранения данных	99
3.3.3. Обсуждение.....	102
3.3.4. Задача.....	102
3.4. Как обращаться к ключам, значениям и элементам словарей.....	102
3.4.1. Прямое использование объектов динамических представлений (keys, values и items)	103
3.4.2. Будьте осторожны с исключением KeyError.....	105
3.4.3. Предотвращение KeyError с предварительной проверкой: непитонический способ	105
3.4.4. Использование метода get для обращения к элементу словаря.....	106

10 Оглавление

3.4.5. Побочный эффект метода <code>setdefault</code>	107
3.4.6. Обсуждение.....	108
3.4.7. Задача.....	108
3.5. Когда использовать словари и множества вместо списков и кортежей.....	109
3.5.1. Преимущества постоянной эффективности поиска.....	109
3.5.2. Хешируемость и хеширование.....	112
3.5.3. Обсуждение.....	115
3.5.4. Задача.....	116
3.6. Как использовать операции над множествами для проверки отношений между списками.....	116
3.6.1. Проверка вхождения всех элементов в другой список.....	116
3.6.2. Проверка вхождения любого элемента списка в другой список.....	118
3.6.3. Работа с несколькими объектами <code>set</code>	120
3.6.4. Обсуждение.....	122
3.6.5. Задача.....	122
Итоги.....	122

Глава 4. Работа с последовательностями 124

4.1. Как получать подпоследовательности и выполнять над ними операции с помощью срезов.....	125
4.1.1. Использование всех возможностей слайсинга.....	125
4.1.2. Различия между срезами и интервалами.....	128
4.1.3. Использование именованных объектов <code>slice</code> для обработки данных последовательностей.....	129
4.1.4. Операции с элементами списков с применением слайсинга.....	130
4.1.5. Обсуждение.....	132
4.1.6. Задача.....	132
4.2. Как использовать положительные и отрицательные индексы для получения элементов.....	132
4.2.1. Положительное индексирование от начала списка.....	133
4.2.2. Отрицательное индексирование от конца списка.....	134
4.2.3. Объединение положительных и отрицательных индексов.....	135
4.2.4. Обсуждение.....	135
4.2.5. Задача.....	135

4.3. Как искать элементы последовательности по критерию	136
4.3.1. Проверка вхождения элемента.....	136
4.3.2. Использование метода <code>index</code> для поиска элемента	137
4.3.3. Поиск подстроки в строке.....	138
4.3.4. Поиск экземпляра пользовательского класса в списке	139
4.3.5. Обсуждение.....	140
4.3.6. Задача.....	140
4.4. Как распаковать последовательности.....	141
4.4.1. Распаковка коротких последовательностей с однозначным соответствием.....	141
4.4.2. Выборка смежных элементов с использованием выражений со звездочкой	142
4.4.3. Пометка нежелательных элементов подчеркиваниями	144
4.4.4. Обсуждение.....	145
4.4.5. Задача.....	145
4.5. Когда следует выбрать другие модели данных, помимо списков и кортежей.....	146
4.5.1. Использование множеств в ситуациях с проверкой принадлежности	146
4.5.2. Деки для обеспечения принципа FIFO.....	146
4.5.3. Обработка многомерных данных средствами <code>NumPy</code> и <code>Pandas</code>	149
4.5.4. Обсуждение.....	150
4.5.5. Задача.....	150
Итоги.....	151

Глава 5. Итерируемые объекты и перебор 152

5.1. Как создавать распространенные контейнеры данных с использованием итерируемых объектов	153
5.1.1. Итерируемые объекты и итераторы	154
5.1.2. Проверка итерируемости	156
5.1.3. Использование итерируемых объектов для создания встроенных контейнеров данных	158
5.1.4. Обсуждение.....	160
5.1.5. Задача.....	161

12 Оглавление

5.2. Что такое включения списков, словарей и множеств	161
5.2.1. Создание списков из итерируемых объектов с использованием списковых включений.....	161
5.2.2. Создание словарей из итерируемых объектов с использованием словарных включений.....	163
5.2.3. Создание множеств из итерируемых объектов с использованием включений множеств.....	164
5.2.4. Применение фильтрующего условия.....	165
5.2.5. Встроенные циклы for	166
5.2.6. Обсуждение.....	167
5.2.7. Задача.....	168
5.3. Как улучшить циклы for с использованием встроенных функций	168
5.3.1. Перечисление элементов и enumerate	169
5.3.2. Обратная перестановка элементов функцией reversed.....	170
5.3.3. Соединение итерируемых объектов с использованием zip	171
5.3.4. Сцепление нескольких итерируемых объектов функцией chain	173
5.3.5. Фильтрация итерируемого объекта функцией filter	175
5.3.6. Обсуждение.....	175
5.3.7. Задача.....	176
5.4. Как использовать необязательные инструкции в циклах for и while.....	177
5.4.1. Выход из циклов с помощью инструкции break	179
5.4.2. Пропуск итерации с помощью инструкции continue	180
5.4.3. Использование else в циклах for и while	183
5.4.4. Обсуждение.....	186
5.4.5. Задача.....	187
Итоги.....	187

Часть 2

Определение функций

Глава 6. Определение дружественных к пользователю функций ... 190

6.1. Как определить аргументы по умолчанию для упрощения вызова функций	191
6.1.1. Вызов функций с аргументами по умолчанию.....	191
6.1.2. Определение функций с аргументами по умолчанию	192

6.1.3. Предотвращение проблем с назначением аргументов по умолчанию для изменяемых параметров.....	195
6.1.4. Обсуждение.....	198
6.1.5. Задача.....	198
6.2. Как определить и использовать возвращаемое значение при вызовах функций.....	198
6.2.1. Явное и неявное возвращение значения.....	199
6.2.2. Определение функций, возвращающих 0, 1 или несколько значений.....	200
6.2.3. Использование нескольких значений, возвращаемых при вызове функции.....	203
6.2.4. Обсуждение.....	204
6.2.5. Задача.....	204
6.3. Как использовать аннотации типов для написания понятных функций.....	205
6.3.1. Определение аннотаций типов для переменных.....	205
6.3.2. Использование аннотаций типов в определениях функций.....	207
6.3.3. Нетривиальное применение аннотаций типов в определениях функций.....	208
6.3.4. Обсуждение.....	212
6.3.5. Задача.....	212
6.4. Как повысить гибкость функции при помощи *args и **kwargs.....	212
6.4.1. Позиционные и ключевые аргументы.....	213
6.4.2. Получение переменного количества позиционных аргументов.....	215
6.4.3. Получение переменного количества ключевых аргументов.....	217
6.4.4. Обсуждение.....	218
6.4.5. Задача.....	219
6.5. Как правильно написать doc-строку для функции.....	219
6.5.1. Базовая структура doc-строки функции.....	220
6.5.2. Указание действия функции в кратком описании.....	222
6.5.3. Документирование параметров и возвращаемого значения.....	222
6.5.4. Определение возможных исключений.....	223
6.5.5. Обсуждение.....	224
6.5.6. Задача.....	225
Итоги.....	225

Глава 7. Продвинутое использование функций 227

7.1. Как определяются лямбда-функции	228
7.1.1. Создание лямбда-функций.....	228
7.1.2. Использование лямбда-функций для выполнения небольших операций.....	229
7.1.3. Потенциальные проблемы при использовании лямбда-функций.....	230
7.1.4. Обсуждение.....	233
7.1.5. Задача.....	233
7.2. К каким последствиям приводит использование функций как объектов.....	233
7.2.1. Хранение функций в контейнерах данных	234
7.2.2. Передача функций в аргументах функций высшего порядка ...	235
7.2.3. Функции как возвращаемые значения.....	237
7.2.4. Обсуждение.....	238
7.2.5. Задача.....	239
7.3. Как проверить быстрдействие функций с декораторами.....	239
7.3.1. Декорирование функции для вывода ее быстрдействия	240
7.3.2. Строение функции-декоратора	242
7.3.3. Упаковка для передачи метаданных декорируемой функции...	246
7.3.4. Обсуждение.....	248
7.3.5. Задача.....	248
7.4. Как использовать функции-генераторы в качестве поставщика данных, эффективно расходующего память.....	249
7.4.1. Создание генератора для получения квадратов.....	250
7.4.2. Использование генераторов для экономии памяти	252
7.4.3. Использование генераторных выражений.....	253
7.4.4. Обсуждение.....	254
7.4.5. Задача.....	254
7.5. Как создать частичные функции для упрощения вызова функций....	255
7.5.1. «Локализация» общих функций для упрощения вызовов.....	256
7.5.2. Создание частичной функции для локализации функции	256
7.5.3. Обсуждение.....	257
7.5.4. Задача.....	257
Итоги.....	258

Часть 3

Определение классов

Глава 8. Определение пользовательских классов	260
8.1. Как определить метод инициализации для класса	261
8.1.1. Загадочный self: первый параметр <code>__init__</code>	261
8.1.2. Правильный выбор аргументов в <code>__init__</code>	265
8.1.3. Назначение всех атрибутов в <code>__init__</code>	267
8.1.4. Определение атрибутов класса за пределами метода <code>__init__</code>	270
8.1.5. Обсуждение.....	271
8.1.6. Задача.....	271
8.2. Как определяются методы экземпляров, статические методы и методы классов.....	271
8.2.1. Определение методов экземпляров для выполнения операций с отдельными экземплярами	271
8.2.2. Определение статических методов для вспомогательной функциональности	273
8.2.3. Определение методов класса для обращения к атрибутам уровня класса.....	274
8.2.4. Обсуждение.....	275
8.2.5. Задача.....	276
8.3. Как организовать более точное управление доступом в классе.....	276
8.3.1. Создание защищенных методов с префиксом <code>_</code>	277
8.3.2. Создание приватных методов с префиксом <code>__</code>	279
8.3.3. Создание атрибутов, доступных только для чтения (read-only).....	281
8.3.4. Проверка целостности данных с использованием set-метода (сеттера).....	283
8.3.5. Обсуждение.....	284
8.3.6. Задача.....	285
8.4. Как настроить строковое представление класса	285
8.4.1. Переопределение <code>__str__</code> для вывода содержательной информации об экземпляре.....	285
8.4.2. Переопределение <code>__repr__</code> для получения информации экземпляра.....	287
8.4.3. Различия между <code>__str__</code> и <code>__repr__</code>	288

16 Оглавление

8.4.4. Обсуждение.....	290
8.4.5. Задача.....	290
8.5. Для чего и как создаются надклассы и подклассы.....	290
8.5.1. Когда уместно использовать подклассы.....	291
8.5.2. Автоматическое наследование атрибутов и методов надкласса.....	293
8.5.3. Переопределение методов надкласса для реализации нестандартного поведения.....	294
8.5.4. Создание ограниченных методов в надклассе.....	296
8.5.5. Обсуждение.....	297
8.5.6. Задача.....	298
Итоги.....	298

Глава 9. Продвинутое использование классов 300

9.1. Как создавать перечисления.....	301
9.1.1. Перечисления без обычных классов.....	301
9.1.2. Создание класса перечислений.....	303
9.1.3. Использование перечислений.....	304
9.1.4. Определение методов для класса перечислений.....	305
9.1.5. Обсуждение.....	307
9.1.6. Задача.....	307
9.2. Как использовать классы данных для устранения шаблонного кода....	307
9.2.1. Создание класса данных с декоратором dataclass.....	308
9.2.2. Определение значений по умолчанию для полей.....	309
9.2.3. Определение неизменяемых классов данных.....	311
9.2.4. Создание подкласса для существующего класса данных.....	312
9.2.5. Обсуждение.....	313
9.2.6. Задача.....	313
9.3. Как подготовить и обработать данные JSON.....	314
9.3.1. Структура данных JSON.....	315
9.3.2. Соответствие типов данных JSON и Python.....	316
9.3.3. Десериализация строк JSON.....	316
9.3.4. Сериализация данных Python в формат JSON.....	319
9.3.5. Обсуждение.....	321
9.3.6. Задача.....	321

9.4. Как создать отложенные атрибуты для улучшения быстродействия	322
9.4.1. Сценарий использования	322
9.4.2. Переопределение специального метода <code>__getattr__</code> для реализации отложенных атрибутов.....	323
9.4.3. Реализация свойства в виде отложенного атрибута.....	325
9.4.4. Обсуждение.....	327
9.4.5. Задача.....	327
9.5. Как определить классы с четким разделением ответственности	328
9.5.1. Анализ класса	328
9.5.2. Создание дополнительных классов для разделения ответственности.....	331
9.5.3. Связывание взаимодействующих классов	332
9.5.4. Обсуждение.....	335
9.5.5. Задача	335
Итоги.....	335

Часть 4

Операции с объектами и файлами

Глава 10. Основы работы с объектами 338

10.1. Как проверить тип переменной для повышения гибкости кода	339
10.1.1. Проверка типа объекта функцией <code>type</code>	340
10.1.2. Проверка типа объекта с использованием <code>isinstance</code>	341
10.1.3. Обобщенная проверка типа объекта	343
10.1.4. Обсуждение	345
10.1.5. Задача.....	345
10.2. Как выглядит жизненный цикл объекта.....	345
10.2.1. Инстанцирование объекта.....	346
10.2.2. Активность в пространствах имен	347
10.2.3. Подсчет ссылок.....	348
10.2.4. Уничтожение объекта.....	350
10.2.5. Обсуждение	351
10.2.6. Задача.....	352
10.3. Как скопировать объект.....	352
10.3.1. Создание (поверхностной) копии.....	353

18 Оглавление

10.3.2. Потенциальные проблемы с поверхностным копированием..	354
10.3.3. Создание глубокой копии.....	357
10.3.4. Обсуждение	358
10.3.5. Задача.....	358
10.4. Как обратиться к переменной и изменить ее в другой области видимости.....	359
10.4.1. Обращение к произвольной переменной: правило LEGB при поиске имен	359
10.4.2. Изменение глобальной переменной в локальной области видимости	362
10.4.3. Изменение охватывающей переменной.....	364
10.4.4. Обсуждение	365
10.4.5. Задача.....	365
10.5. Что такое вызываемость и что она означает.....	365
10.5.1. Различия между классами и функциями	366
10.5.2. Снова о функции высшего порядка map	367
10.5.3. Использование вызываемых объектов в аргументе key.....	368
10.5.4. Создание декораторов в форме классов	369
10.5.5. Обсуждение	370
10.5.6. Задача.....	371
Итоги.....	371

Глава 11. Работа с файлами 373

11.1. Как выполнять чтение и запись в файлы через управление контекстом.....	374
11.1.1. Открытие и закрытие файлов: менеджер контекста	374
11.1.2. Разные способы чтения данных из файла.....	376
11.1.3. Разные способы записи данных в файл	379
11.1.4. Обсуждение	383
11.1.5. Задача.....	383
11.2. Как работать с табличными файлами данных.....	383
11.2.1. Чтение CSV-файлов	383
11.2.2. Чтение CSV-файла с заголовком	385
11.2.3. Запись данных в CSV-файл	386
11.2.4. Обсуждение	388
11.2.5. Задача.....	388

11.3. Как сохранять данные в файлах с использованием консервации	389
11.3.1. Консервация объектов для сохранения данных	389
11.3.2. Восстановление данных посредством расконсервации.....	390
11.3.3. Достоинства и недостатки консервации.....	393
11.3.4. Обсуждение	395
11.3.5. Задача.....	395
11.4. Как управлять файлами на своем компьютере.....	396
11.4.1. Создание каталогов и файлов.....	396
11.4.2. Получение списка файлов по шаблону	397
11.4.3. Перемещение файлов в другой каталог	398
11.4.4. Копирование файлов в другой каталог	400
11.4.5. Удаление файлов.....	401
11.4.6. Обсуждение	401
11.4.7. Задача.....	401
11.5. Как получить метаданные файла	402
11.5.1. Получение информации, относящейся к имени файла.....	402
11.5.2. Получение информации о размере файла и времени изменения	404
11.5.3. Обсуждение	405
11.5.4. Задача.....	405
Итоги.....	406

Часть 5

Защита кодовой базы

Глава 12. Ведение журнала и обработка исключений 408

12.1. Как следить за работой программы посредством логирования.....	409
12.1.1. Создание регистратора для логирования событий приложения.....	409
12.1.2. Использование файлов для хранения событий приложения .	411
12.1.3. Добавление нескольких обработчиков.....	413
12.1.4. Обсуждение	414
12.1.5. Задача.....	414
12.2. Как правильно хранить журнальные записи	415
12.2.1. Классификация событий приложения по уровням.....	415

12.2.2. Назначение уровня для обработчика.....	417
12.2.3. Форматы для обработчика	418
12.2.4. Обсуждение	420
12.2.5. Задача.....	420
12.3. Как обрабатывать исключения	421
12.3.1. Обработка исключений в блоке try..except.....	421
12.3.2. Указание исключений в секции except.....	424
12.3.3. Обработка нескольких исключений.....	425
12.3.4. Вывод расширенной информации об исключении.....	427
12.3.5. Обсуждение	427
12.3.6. Задача.....	428
12.4. Как использовать секции else и finally при обработке исключений.....	428
12.4.1. Использование else для продолжения логики секции try	429
12.4.2. Завершение обработки исключений в секции finally.....	430
12.4.3. Обсуждение	432
12.4.4. Задача.....	432
12.5. Как выдавать содержательные исключения с пользовательскими классами исключений	433
12.5.1. Выдача исключений с пользовательским сообщением.....	434
12.5.2. Встроенные классы исключений предпочтительны.....	435
12.5.3. Определение пользовательских классов исключений.....	437
12.5.4. Обсуждение	439
12.5.5. Задача.....	440
Итоги.....	440

Глава 13. Отладка и тестирование 442

13.1. Как выявить проблемы с помощью трассировки	443
13.1.1. Как генерируется трассировка	444
13.1.2. Анализ трассировки при выполнении кода в консоли	446
13.1.3. Анализ трассировки при выполнении скриптов	446
13.1.4. Последний вызов в трассировке.....	448
13.1.5. Обсуждение	449
13.1.6. Задача.....	449
13.2. Как провести отладку программы в интерактивном режиме.....	449
13.2.1. Активизация отладчика в точке останова.....	450

13.2.2. Построчное выполнение кода.....	451
13.2.3. Выполнение с заходом в функции.....	453
13.2.4. Анализ важных переменных.....	455
13.2.5. Обсуждение.....	456
13.2.6. Задача.....	456
13.3. Как тестировать функции автоматически.....	456
13.3.1. Принципы тестирования функций.....	457
13.3.2. Создание подкласса TestCase для тестирования функций.....	458
13.3.3. Подготовка теста.....	461
13.3.4. Обсуждение.....	462
13.3.5. Задача.....	462
13.4. Как провести автоматическое тестирование класса.....	463
13.4.1. Создание подкласса TestCase для тестирования класса.....	463
13.4.2. Реакция на сбои в тестах.....	464
13.4.3. Обсуждение.....	465
13.4.4. Задача.....	466
Итоги.....	466

Часть 6

Построение веб-приложения

Глава 14. Завершение реального проекта.....	468
14.1. Как использовать для проекта виртуальную среду.....	469
14.1.1. Причины использования виртуальных сред.....	469
14.1.2. Создание виртуальной среды для каждого проекта.....	470
14.1.3. Установка пакетов в виртуальной среде.....	471
14.1.4. Использование виртуальных сред в Visual Studio Code.....	472
14.1.5. Обсуждение.....	473
14.1.6. Задача.....	474
14.2. Как построить модели данных для проекта.....	474
14.2.1. Выявление бизнес-целей.....	475
14.2.2. Создание вспомогательных классов и функций.....	476
14.2.3. Создание класса Task.....	477
14.2.4. Обсуждение.....	484
14.2.5. Задача.....	484

22 Оглавление

14.3. Как использовать в приложении базу данных SQLite	484
14.3.1. Создание базы данных	484
14.3.2. Чтение записей из базы данных.....	486
14.3.3. Сохранение записей в базе данных.....	488
14.3.4. Обновление записи в базе данных.....	489
14.3.5. Удаление записи из базы данных	490
14.3.6. Обсуждение	490
14.3.7. Задача.....	491
14.4. Как построить веб-приложение для взаимодействия с клиентом (фронтенд)	491
14.4.1. Основные возможности streamlit.....	491
14.4.2. Интерфейс приложения.....	493
14.4.3. Отслеживание действий пользователя в течение сеанса.....	495
14.4.4. Настройка боковой панели.....	498
14.4.5. Вывод задач.....	501
14.4.6. Вывод подробной информации о задаче	503
14.4.7. Создание новой задачи	505
14.4.8. Организация проекта	506
14.4.9. Запуск приложения	507
14.4.10. Обсуждение	508
14.4.11. Задача	508
Итоги.....	508
Решения задач	510