

УДК 004.438Python

ББК 32.973.22

Ш58

Дональд Р. Шихи

Ш58 Структуры данных в Python: начальный курс / пер. с англ. А. В. Снастина. – М.: ДМК Пресс, 2022. – 186 с.: ил.

ISBN 978-5-93700-110-8

В книге рассматриваются основополагающие вопросы, относящиеся к структурам данных в языке программирования Python. Теоретические концепции и абстрактные понятия подкрепляются простыми примерами. По мере изучения основ вводятся такие темы, как стратегии решения задач, продвинутое использование языка Python, принципы объектно-ориентированного проектирования и методологии тестирования. Подробно рассматриваются структуры данных, встроенные в язык Python, а также абстрактные типы данных (АТД): стеки, очереди, связанные списки, деревья, графы и др.

Книга предназначена для всех, кто изучает язык программирования Python и предполагает активно использовать как встроенные структуры данных, так и собственные реализации АТД.

УДК 004.438Python

ББК 32.973.22

Все права защищены. Любая часть этой книги не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами без письменного разрешения владельцев авторских прав.

Оглавление

Глава 1. Предисловие	9
Глава 2. Основы языка программирования Python	10
2.1. Последовательность, выбор и итерация.....	10
2.2. Выражения и вычисление	11
2.3. Переменные, типы и состояние	11
2.4. Наборы данных	14
2.4.1. Строки (str)	14
2.4.2. Списки (list).....	14
2.4.3. Кортежи (tuple)	15
2.4.4. Словари (dict)	16
2.4.5. Множества (set).....	17
2.5. Некоторые общие правила работы с наборами данных	18
2.6. Итерации по наборам данных	18
2.7. Другие формы управления потоком выполнения	19
2.8. Модули и импортирование	21
Глава 3. Объектно-ориентированное программирование.....	24
3.1. Простой пример	25
3.2. Инкапсуляция и открытый (общедоступный) интерфейс класса	28
3.3. Наследование и отношение «является» (is a)	29
3.4. Утиная типизация	31
3.5. Композиция и отношения «содержит» (has a)	32
Глава 4. Тестирование.....	34
4.1. Написание тестов.....	34
4.2. Модульное тестирование с использованием unittest	35
4.3. Разработка через тестирование	36
4.4. Что необходимо тестировать	37
4.5. Тестирование и объектно-ориентированное проектирование	38
Глава 5. Анализ во время выполнения.....	39
5.1. Измерение времени выполнения (тайминг) программ.....	40
5.2. Пример: сложение первых k чисел	44
5.3. Моделирование времени выполнения программы.....	46
5.3.1. Операции со списком	47
5.3.2. Операции со словарем	47
5.3.3. Операции с множеством	48

5.4. Асимптотический анализ и порядок роста	48
5.5. Сосредоточимся на самом худшем случае	49
5.6. О-большое	49
5.7. Самые важные свойства использования О-большого	50
5.8. Практическое использование О-большого и общие функции	50
5.9. Основания логарифмов	51
5.10. Практические примеры	51
Глава 6. Стеки и очереди.....	53
6.1. Абстрактные типы данных	53
6.2. Абстрактный тип данных «стек»	54
6.3. Абстрактный тип данных «очередь».....	55
6.4. Обработка ошибок	57
Глава 7. Деки и связные списки	59
7.1. Абстрактный тип данных «дек»	59
7.2. Связные списки	60
7.3. Реализация очереди с помощью класса LinkedList	61
7.4. Хранение длины	63
7.5. Тестирование на основании АТД.....	64
7.6. Основные уроки	68
7.7. Шаблоны проектирования: шаблон «обертка».....	68
Глава 8. Двусвязные списки	70
8.1. Объединение двусвязных списков	72
Глава 9. Рекурсия.....	74
9.1. Рекурсия и индукция	75
9.2. Некоторые основные правила	75
9.3. Стек вызовов функций	76
9.4. Последовательность Фибоначчи.....	77
9.5. Алгоритм Евклида.....	78
Глава 10. Динамическое программирование.....	80
10.1. Жадный алгоритм	80
10.2. Рекурсивный алгоритм	81
10.3. Версия с мемоизацией.....	81
10.4. Алгоритм динамического программирования	82
10.5. Еще один пример	83
Глава 11. Двоичный поиск.....	85
11.1. Абстрактный тип данных «упорядоченный список».....	87

Глава 12. Сортировка	89
12.1. Алгоритмы сортировки, выполняемые за квадратичное время	89
12.2. Сортировка в Python	93
Глава 13. Сортировка методом «разделяй и властвуй»	95
13.1. Сортировка слиянием	96
13.1.1. Анализ	97
13.1.2. Итераторы слияния	98
13.2. Быстрая сортировка	100
Глава 14. Выбор	104
14.1. Алгоритм quickselect	105
14.2. Анализ	106
14.3. В последний раз без рекурсии	107
14.4. Резюме стратегии «разделяй и властвуй»	107
14.5. Замечание о дерандомизации	108
Глава 15. Отображения и хеш-таблицы	109
15.1. Абстрактный тип данных «отображение»	109
15.2. Минимальная реализация	110
15.3. Расширенный абстрактный тип данных «отображение»	111
15.4. Это слишком медленно	113
15.4.1. Сколько контейнеров мы должны использовать?	114
15.4.2. Двойное хеширование	116
15.5. Вынос общих частей в суперкласс	116
Глава 16. Деревья	120
16.1. Еще несколько определений	121
16.2. Деревья с точки зрения рекурсии	121
16.3. Абстрактный тип данных дерево	123
16.4. Реализация	124
16.5. Обход дерева	126
16.6. Если хотите немного развлечься...	127
16.6.1. Есть одно «но»	128
16.6.2. Уровень за уровнем	128
Глава 17. Деревья двоичного поиска	130
17.1. Абстрактный тип данных «упорядоченное отображение»	130
17.2. Определение и свойства дерева двоичного поиска	130
17.3. Минимальная реализация	131
17.3.1. Метод floor	134
17.3.2. Итерация	135
17.4. Удаление	135

Глава 18. Сбалансированные деревья двоичного поиска	138
18.1. Реализация класса BSTMapping	139
18.1.1. Совместимость снизу вверх шаблонов «фабрика»	140
18.2. Взвешенные сбалансированные деревья	140
18.3. Сбалансированные по высоте деревья (АВЛ-деревья)	142
18.4. Косые деревья	144
Глава 19. Очереди с приоритетами	146
19.1. Абстрактный тип данных «очередь с приоритетами»	146
19.2. Использование списка	146
19.3. Кучи	149
19.4. Хранение дерева в списке	150
19.5. Создание кучи с нуля, _heapify	151
19.6. Значимость и изменение приоритетов	152
19.7. Итеративный проход по очереди с приоритетами	154
19.8. Пирамидальная сортировка	155
Глава 20. Графы	156
20.1. Абстрактный тип данных граф	157
20.2. Реализация класса EdgeSetGraph	157
20.3. Реализация класса AdjacencySetGraph	158
20.4. Пути и связность	160
Глава 21. Поиск в графах	163
21.1. Поиск в глубину	164
21.2. Исключение рекурсии	165
21.3. Поиск в ширину	166
21.4. Взвешенные графы и кратчайшие пути	167
21.5. Алгоритм Прима для минимальных остовных деревьев	169
21.6. Оптимизация поиска по первому наилучшему (приоритетному) совпадению	170
Глава 22. (Непересекающиеся) множества	172
22.1. Абстрактный тип данных «непересекающиеся множества»	172
22.2. Простая реализация	173
22.3. Сжатие пути	174
22.4. Слияние по высоте	175
22.5. Слияние по весу	176
22.6. Объединение эвристик	176
22.7. Алгоритм Краскала	177
Предметный указатель	179