

Spis treści

O autorze 9

O recenzencie 10

Przedmowa 9

Rozdział 1. Zrozumieć programowanie funkcyjne 17

Paradygmat programowania 18

Podział paradygmatu proceduralnego 19

Korzystanie z paradygmatu funkcyjnego 20

Korzystanie z funkcyjnych hybryd 22

Tworzenie obiektu 23

Stos żółwi 24

Klasyczny przykład programowania funkcyjnego 25

Eksploracyjna analiza danych 28

Podsumowanie 29

Rozdział 2. Podstawowe pojęcia programowania funkcyjnego 31

Funkcje pierwszej klasy 32

Czyste funkcje 32

Funkcje wyższego rzędu 33

Dane niemutowalne 34

Wartościowanie ścisłe i nieścisłe 36

Rekurencja zamiast jawnego stanu pętli 37

Funkcyjne systemy typów 41

Znajome terytorium 41

Pojęcia zaawansowane 42

Podsumowanie 43

Rozdział 3. Funkcje, iteratory i generatory 45

Pisanie czystych funkcji 46

Funkcje jako obiekty pierwszej klasy 48

Korzystanie z łańcuchów znaków 49

Używanie krotek i krotek nazwanych 50

Korzystanie z wyrażeń generatorowych 52

Odkrywanie ograniczeń generatorów 54

Łączenie wyrażeń generatorowych 56

Czyszczenie surowych danych za pomocą funkcji generatorowych 56

Korzystanie z list, słowników i zbiorów 58

Korzystanie z mapowań stanowych 61

Wykorzystanie modułu bisect do tworzenia mapowania 63

Używanie stanowych zbiorów 64

Podsumowanie 65

Rozdział 4. Praca z kolekcjami 67

Przegląd rodzajów funkcji 68

Praca z obiektami iterowalnymi 68

Parsowanie pliku XML 69

Parsowanie pliku na wyższym poziomie 71

Tworzenie par elementów z sekwencji 73

Jawne użycie funkcji iter() 76

Rozszerzanie prostej pętli 77

Stosowanie wyrażeń generatorowych do funkcji skalarnych 80

Wykorzystanie funkcji any() i all() jako redukcji 81

Używanie funkcji `len()` i `sum()` 83

Używanie `sum` i zliczeń w obliczeniach statystycznych 84

Korzystanie z funkcji `zip()` do tworzenia struktury i spłaszczania sekwencji 87

Rozpakowywanie spakowanej sekwencji 88

Spłaszczanie sekwencji 89

Nadawanie struktury płaskim sekwencjom 90

Tworzenie struktury płaskich sekwencji - podejście alternatywne 92

Wykorzystanie funkcji `reverse()` do zmiany kolejności elementów 93

Wykorzystanie funkcji `enumerate()` w celu uwzględnienia numeru porządkowego 94

Podsumowanie 94

Rozdział 5. Funkcje wyższego rzędu 97

Wykorzystanie funkcji `max()` i `min()` do wyszukiwania ekstremów 98

Korzystanie z formatu wyrażen `lambda` w Pythonie 101

Wyrażenia `lambda` i rachunek `lambda` 103

Korzystanie z funkcji `map()` w celu zastosowania funkcji do kolekcji 103

Wykorzystanie wyrażen `lambda` i funkcji `map()` 104

Użycie funkcji `map()` w odniesieniu do wielu sekwencji 105

Wykorzystanie funkcji `filter()` do przekazywania lub odrzucania danych 107

Użycie funkcji `filter()` do identyfikacji wartości odstających 108

Funkcja `iter()` z wartością "strażnika" 109

Wykorzystanie funkcji `sorted()` do porządkowania danych 110

Pisanie funkcji wyższego rzędu 111

Pisanie mapowań i filtrów wyższego rzędu 112

Rozpakowywanie danych podczas mapowania 113

Opakowywanie dodatkowych danych podczas mapowania 115

Spłaszczanie danych podczas mapowania 116

Strukturyzacja danych podczas filtrowania 118

Pisanie funkcji generatorowych 119

Budowanie funkcji wyższego rzędu z wykorzystaniem obiektów wywoływalnych 121

Zapewnienie dobrego projektu funkcyjnego 123

Przegląd wybranych wzorców projektowych 124

Podsumowanie 125

Rozdział 6. Rekurencje i redukcje 127

Proste rekurencje numeryczne 128

Implementacja optymalizacji ogonowej 129

Pozostawienie rekurencji bez zmian 130

Obsługa trudnego przypadku optymalizacji ogonowej 131

Przetwarzanie kolekcji za pomocą rekurencji 132

Optymalizacja ogonowa dla kolekcji 133

Redukcje i składanie kolekcji z wielu elementów w jeden element 134

Redukcja grupowania - z wielu elementów do mniejszej liczby 136

Budowanie mapowania za pomocą metody Counter 136

Budowanie mapowania przez sortowanie 137

Grupowanie lub podział danych według wartości klucza 139

Pisanie bardziej ogólnych redukcji grupujących 142

Pisanie redukcji wyższego rzędu 143

Pisanie parserów plików 144

Podsumowanie 150

Rozdział 7. Dodatkowe techniki przetwarzania krotek 153

Używanie krotek do zbierania danych 154

Używanie krotek nazwanych do zbierania danych 156

Budowanie nazwanych krotek za pomocą konstruktorów funkcyjnych 159

Unikanie stanowych klas dzięki wykorzystaniu rodzin krotek 160

Przypisywanie rang statystycznych 163

Opakowanie zamiast zmiany stanu 165

Wielokrotne opakowanie zamiast zmian stanu 166

Obliczanie korelacji rangowej Spearmana 167

Polimorfizm i dopasowywanie typów ze wzorcami 169

Podsumowanie 174

Rozdział 8. Moduł itertools 175

Praca z iteratorami nieskończonymi 176

Liczenie za pomocą count() 176

Zliczanie z wykorzystaniem argumentów zmiennoprzecinkowych 177

Wielokrotne iterowanie cyklu za pomocą funkcji cycle() 179

Powtarzanie pojedynczej wartości za pomocą funkcji repeat() 181

Używanie iteratorów skończonych 182

Przypisywanie liczb za pomocą funkcji enumerate() 182

Obliczanie sum narastających za pomocą funkcji accumulate() 185

Łączenie iteratorów za pomocą funkcji chain() 186

Podział iteratora na partycje za pomocą funkcji groupby() 187

Scalanie obiektów iterowalnych za pomocą funkcji zip_longest() i zip() 188

Filtrowanie z wykorzystaniem funkcji compress() 189

Zbieranie podzbiorów za pomocą funkcji islice() 190

Filtrowanie stanowe z wykorzystaniem funkcji dropwhile() i takewhile() 191

Dwa podejścia do filtrowania za pomocą funkcji filterfalse() i filter() 192

Zastosowanie funkcji do danych z wykorzystaniem funkcji starmap() i map() 193

Klonowanie iteratorów za pomocą funkcji tee() 194

Receptury modułu itertools 195

Podsumowanie 197

Rozdział 9. Dodatkowe techniki itertools 199

Wyliczanie iloczynu kartezjańskiego 200

Redukowanie iloczynu 200

Obliczanie odległości 202

Uzyskanie wszystkich pikseli i wszystkich kolorów 204

Analiza wydajności 205

Przeformowanie problemu 207

Łączenie dwóch transformacji 207

Permutacje zbioru wartości 209

Generowanie wszystkich kombinacji 210

Receptury 212

Podsumowanie 213

Rozdział 10. Moduł functools 215

Narzędzia przetwarzania funkcji 216

Memoizacja wcześniejszych wyników za pomocą dekoratora lru_cache 216

Definiowanie klas z dekoratorem total_ordering 218

Definiowanie klas liczbowych 221

Stosowanie argumentów częściowych za pomocą funkcji partial() 222

Redukcja zbiorów danych za pomocą funkcji reduce() 223

Łączenie funkcji map() i reduce() 224

Korzystanie z funkcji reduce() i partial() 226

Użycie funkcji map() i reduce() do oczyszczania surowych danych 226

Korzystanie z funkcji reduce() i partial() 227

Podsumowanie 230

Rozdział 11. Techniki projektowania dekoratorów 231

Dekoratory jako funkcje wyższego rzędu 231

Korzystanie z funkcji `update_wrapper()` z modułu `functools` 235

Zagadnienia przekrojowe 236

Funkcje złożone 236

Wstępne przetwarzanie nieprawidłowych danych 238

Dekoratory z parametrami 239

Implementacja bardziej złożonych dekoratorów 242

Kwestie złożonego projektu 243

Podsumowanie 246

Rozdział 12. Moduły `multiprocessing` i `threading` 247

Programowanie funkcyjne a współbieżność 248

Co naprawdę oznacza współbieżność? 248

Warunki brzegowe 249

Współdzielenie zasobów za pomocą procesów lub wątków 249

Jak uzyskać największe korzyści? 250

Korzystanie z pul wieloprocessowych i zadań 251

Przetwarzanie wielu dużych plików 252

Parsowanie plików logu - pobieranie wierszy 253

Parsowanie wierszy logu do postaci obiektów `namedtuple` 254

Parsowanie dodatkowych pól obiektu `Access` 256

Filtrowanie szczegółów dostępu 259

Analiza szczegółów dostępu 261

Pełny proces analizy 262

Korzystanie z puli wieloprocessowej w celu przetwarzania równoległego 263

Korzystanie z funkcji `apply()` do wykonywania pojedynczych żądań 265

Korzystanie z funkcji `map_async()`, `starmap_async()` i `apply_async()` 265

Bardziej złożone architektury przetwarzania wieloprocessowego 266

Korzystanie z modułu `concurrent.futures` 267

Korzystanie z pul wątków modułu `concurrent.futures` 267

Korzystanie z modułów `threading` i `queue` 268

Projektowanie współbieżnego przetwarzania 268

Podsumowanie 270

Rozdział 13. Wyrażenia warunkowe i moduł `operator` 271

Ocena wyrażeń warunkowych 272

Wykorzystywanie nieściśłych reguł słownikowych 273

Filtrowanie wyrażeń warunkowych zwracających `True` 274

Wyszukiwanie pasującego wzorca 275

Używanie modułu `operator` zamiast wyrażeń `lambda` 276

Pobieranie wartości nazwanych atrybutów podczas korzystania z funkcji wyższego rzędu 278

Wykorzystanie funkcji `starmap` z operatorami 279

Redukcje z wykorzystaniem funkcji modułu `operator` 281

Podsumowanie 282

Rozdział 14. Biblioteka `PyMonad` 283

Pobieranie i instalacja modułu `pymonad` 284

Kompozycja funkcyjna i rozwijanie funkcji 284

Korzystanie z rozwijanych funkcji wyższego rzędu 286

Rozwijanie funkcji w trudny sposób 288

Kompozycja funkcyjna i operator `*` z biblioteki `pymonad` 288

Funktory zwykłe i aplikatywne 290

 Korzystanie z leniwego funktora List() 291

Funkcja bind() i operator >> 294

Implementacja symulacji za pomocą monad 295

Dodatkowe własności biblioteki pymonad 298

Podsumowanie 299

Rozdział 15. Podejście funkcyjne do usług sieciowych 301

Model HTTP żądanie-odpowiedź 302

 Wstrzykiwanie stanu za pomocą plików cookie 303

 Serwer o projekcie funkcyjnym 304

 Szczegóły widoku funkcyjnego 304

 Zagnieżdżanie usług 305

Standard WSGI 306

 Zgłaszanie wyjątków podczas przetwarzania WSGI 309

 Praktyczne aplikacje WSGI 310

Definiowanie usług sieciowych jako funkcji 311

 Tworzenie aplikacji WSGI 312

 Pobieranie surowych danych 314

 Stosowanie filtra 315

 Serializowanie wyników 316

 Serializacja danych w formatach JSON lub CSV 317

 Serializacja danych do formatu XML 318

 Serializacja danych do formatu HTML 319

Monitorowanie użycia 320

Podsumowanie 322

Rozdział 16. Optymalizacje i ulepszenia 323

Memoizacja i buforowanie 324

Specjalizacja memoizacji 325

Ogonowe optymalizacje rekurencji 327

Optymalizacja pamięci 328

Optymalizacja dokładności 329

Redukcja dokładności w zależności od wymagań odbiorców 329

Studium przypadku - podejmowanie decyzji na podstawie testu zgodności chi-kwadrat 330

Filtrowanie i redukcja surowych danych z wykorzystaniem obiektu Counter 332

Odczyt podsumowanych danych 333

Obliczanie sum za pomocą obiektu Counter 334

Obliczanie prawdopodobieństw na podstawie obiektów Counter 335

Obliczanie oczekiwanych wartości i wyświetlanie tabeli krzyżowej 337

Obliczanie wartości chi-kwadrat 339

Obliczanie progu wartości chi-kwadrat 339

Obliczanie niekompletnej funkcji gamma 340

Obliczanie kompletnej funkcji gamma 343

Obliczanie szans na losową dystrybucję 344

Funkcyjne wzorce projektowe 346

Podsumowanie 348

Skorowidz 349