

Spis treści

O autorach	11
O recenzencie	12
Wstęp	13
Rozdział 1. Projektowanie obiektowe	19
Wprowadzenie do obiektowości	20
Obiekty i klasy	22
Określanie atrybutów i zachowań	24
Dane opisują stan obiektu	25
Zachowania są akcjami	27
Ukrywanie szczegółów i tworzenie interfejsów publicznych	28
Kompozycja	31
Dziedziczenie	34
Dziedziczenie zapewnia abstrakcję	36
Wielokrotne dziedziczenie	37
Studium przypadku	38
Wprowadzenie i omówienie problemu	40
Widok kontekstu	42
Widok logiczny	44
Widok procesu	46
Widok programistyczny	48
Widok fizyczny	50
Wnioski	51
Przypomnij sobie	52
Ćwiczenia	52
Podsumowanie	53

Rozdział 2. Obiekty w Pythonie	54
Prezentacja odpowiedzi typów	54
Sprawdzanie typów	56
Tworzenie klas w Pythonie	59
Dodawanie atrybutów	61
Zapewnianie możliwości działania	62
Inicjalizacja obiektów	65
Podpowiedzi typów i wartości domyślne	67
Podawanie wyjaśnień w napisach dokumentujących	68
Moduły i pakiety	71
Organizowanie modułów	74
Organizowanie kodu w moduły	78
Kto ma dostęp do moich danych?	82
Biblioteki innych twórców	83
Studium przypadku	86
Widok logiczny	86
Próbki i ich stan	88
Zmiany stanu próbek	89
Odpowiedzialności klasy	93
Klasa TrainingData	95
Przypomnij sobie	97
Ćwiczenia	98
Podsumowanie	99
Rozdział 3. Kiedy obiekty są do siebie podobne	100
Proste dziedziczenie	101
Rozszerzanie typów wbudowanych	103
Przesłanianie i super	106
Wielokrotne dziedziczenie	108
Problematiczny diament	111
Różne zestawy argumentów	118
Polimorfizm	121
Studium przypadku	124
Widok logiczny	125
Jeszcze jedna odległość	130
Przypomnij sobie	132
Ćwiczenia	132
Podsumowanie	133
Rozdział 4. Oczekując nieoczekiwanego	134
Zgłaszanie wyjątków	135
Zgłaszanie wyjątku	137
Efekty wyjątków	139
Obsługa wyjątków	141
Hierarchia wyjątków	147
Definiowanie własnych wyjątków	148
Wyjątki nie są wyjątkowe	150

Studium przypadku	154
Widok kontekstu	155
Widok przetwarzania	156
Co może pójść źle?	158
Nieprawidłowe zachowanie	158
Tworzenie próbek na podstawie danych z plików CSV	159
Walidacja wartości wyliczeniowych	163
Odczyt plików CSV	165
Nie powtarzaj się	167
Przypomnij sobie	168
Ćwiczenia	168
Podsumowanie	170
Rozdział 5. Kiedy korzystać z programowania obiektowego	171
Traktujmy obiekty jako obiekty	172
Dodawanie zachowań do klas danych przy wykorzystaniu właściwości	177
Wszystko o właściwościach	181
Dekoratory — inny sposób tworzenia właściwości	183
Określanie, kiedy należy używać właściwości	184
Obiekty menedżerów	187
Usuwanie powtórzeń	192
W praktyce	194
Studium przypadku	197
Walidacja danych wejściowych	198
Dzielenie próbek wejściowych	200
Hierarchia klas próbek	201
Wyliczenie purpose	203
Właściwości ustawiające	205
Powtarzające się instrukcje if	206
Przypomnij sobie	206
Ćwiczenia	207
Podsumowanie	208
Rozdział 6. Abstrakcyjne klasy bazowe i przeciążanie operatorów	209
Tworzenie abstrakcyjnej klasy bazowej	211
Abstrakcyjne klasy bazowe kolekcji	214
Abstrakcyjne klasy bazowe i odpowiedzi typów	215
Moduł collections.abc	217
Tworzenie własnych abstrakcyjnych klas bazowych	223
Wyjaśniamy magię	227
Przeciążanie operatorów	229
Rozszerzanie klas wbudowanych	234
Metaklasy	237
Studium przypadku	243
Rozszerzanie klasy listy w celu utworzenia dwóch podlist	244
Podział poprzez tasowanie	246
Dzielenie próbek metodą inkrementalną	248
Przypomnij sobie	250
Ćwiczenia	251
Podsumowanie	253

Rozdział 7. Struktury danych w Pythonie	254
Puste obiekty	254
Kroki i krotki nazwane	256
Krotki nazwane i <code>typing.NamedTuple</code>	259
Klasy danych	262
Słowniki	265
Przypadki stosowania słowników	270
Stosowanie <code>defaultdict</code>	272
Listy	276
Sortowanie list	278
Zbiory	284
Trzy typy kolejek	288
Studium przypadku	292
Model logiczny	292
Niezmienne klasy danych	295
Klasy <code>NamedTuple</code>	298
Wniosek	301
Przypomnij sobie	301
Ćwiczenia	302
Podsumowanie	303
Rozdział 8. Łączenie programowania obiektowego i funkcyjnego	304
Wbudowane funkcje Pythona	305
Funkcja <code>len()</code>	305
Funkcja <code>reversed()</code>	306
Funkcja <code>enumerate()</code>	307
Alternatywa dla przeciążania metod	309
Domyślne wartości parametrów	311
Zmienne listy argumentów	313
Rozpakowywanie argumentów	319
Funkcje są także obiektami	321
Obiekty funkcji i funkcje zwrotne	323
Stosowanie funkcji do modyfikowania klas	328
Obiekty wywoływalne	330
Plikowe operacje wejścia-wyjścia	331
Działanie w kontekście	335
Studium przypadku	339
Ogólna postać sposobu przetwarzania	340
Rozdzielanie danych	341
Ponowne przemyślenie problemu klasyfikacji	342
Funkcja <code>partition()</code>	345
Podział danych w jednym przejściu	346
Przypomnij sobie	348
Ćwiczenia	349
Podsumowanie	350

Rozdział 9. Łańcuchy, serializacja i ścieżki do plików	352
Łańcuchy znaków	353
Operacje na łańcuchach znaków	354
Formatowanie łańcuchów znaków	357
Łańcuchy znaków są zapisywane w Unicode	366
Wyrażenia regularne	372
Dopasowywanie wzorców	374
Parsowanie informacji przy użyciu wyrażeń regularnych	383
Ścieżki dostępu do plików	386
Serializacja obiektów	390
Dostosowywanie działania modułu pickle	393
Serializacja danych w formacie JSON	395
Studium przypadku	398
Konstrukcja formatu CSV	399
Wczytywanie danych CSV w formie słowników	400
Wczytywanie danych CSV w formie listy	403
Serializacja danych JSON	404
Format JSON z danymi rozdzielanymi znakami nowego wiersza	406
Walidacja danych JSON	407
Przypomnij sobie	409
Ćwiczenia	420
Podsumowanie	412
Rozdział 10. Wzorzec Iterator	413
Krótko o wzorcach projektowych	413
Iteratory	414
Protokół iteratorów	415
Listy składane	417
Wyrażenia list składanych	418
Wyrażenia zbiorów i słowników składanych	420
Wyrażenia generatorów	422
Funkcje generatorów	424
Zwracanie elementów z innego iteratora	428
Stosy generatorów	430
Studium przypadku	434
Zarys konstruowania zbiorów	434
Wiele podziałów	436
Testowanie	440
Niezbędny algorytm k-NN	441
Algorytm k-NN korzystający z modułu bisect	442
Algorytm k-NN korzystający z modułu heapq	443
Wniosek	444
Przypomnij sobie	446
Ćwiczenia	447
Podsumowanie	448

Rozdział 11. Często stosowane wzorce projektowe	449
Wzorec Dekorator	450
Przykład wzorca Dekorator	451
Dekoratory w Pythonie	458
Wzorec Obserwator	461
Przykład wzorca Obserwator	463
Wzorec Strategia	466
Przykład wzorca Strategia	467
Wzorec Strategia w Pythonie	471
Wzorec Polecenie	472
Przykład wzorca Polecenie	473
Wzorec Stan	476
Przykład wzorca Stan	477
Stan a Strategia	485
Wzorec Singleton	485
Implementacja wzorca Singleton	486
Studium przypadku	490
Przypomnij sobie	497
Ćwiczenia	498
Podsumowanie	499
Rozdział 12. Zaawansowane wzorce projektowe	500
Wzorec Adapter	501
Przykład wzorca Adapter	502
Wzorec Fasada	506
Przykład wzorca Fasada	507
Wzorec Piórko	510
Przykład implementacji wzorca Piórko w Pythonie	512
Przechowywanie w buforze wielu komunikatów	519
Optymalizacja pamięci przy użyciu atrybutu <code>__slots__</code>	520
Wzorec Fabryka abstrakcyjna	521
Przykład wzorca Fabryka abstrakcyjna	522
Fabryki abstrakcyjne w Pythonie	528
Wzorec Kompozyt	529
Przykład wzorca Kompozyt	531
Wzorec Metoda szablonowa	536
Przykład wzorca Metoda szablonowa	536
Studium przypadku	541
Przypomnij sobie	544
Ćwiczenia	545
Podsumowanie	547
Rozdział 13. Testowanie oprogramowania obiektowego	548
Po co testować?	549
Programowanie na podstawie testów	550
Cele testowania	551
Wzorce testowania	552

Testowanie przy użyciu frameworka unittest	554
Wykonywanie testów jednostkowych przy użyciu pakietu pytest	556
Funkcje setup i teardown pakietu pytest	559
Przygotowania i porządki przy użyciu konfiguracji początkowych	561
Bardziej wyszukane konfiguracje początkowe	566
Pomijanie testów narzędzia pytest	572
Imitowanie obiektów przy użyciu atrap	574
Dodatkowe techniki korygowania	578
Obiekt sentinel	581
Ile testów wystarczy?	582
Testowanie a programowanie	586
Studium przypadku	587
Testy jednostkowe klas obliczających odległości	588
Testy jednostkowe klasy Hyperparameter	593
Przypomnij sobie	596
Ćwiczenia	598
Podsumowanie	599
 Rozdział 14. Współbieżność	 600
Podstawowe informacje o przetwarzaniu współbieżnym	601
Wątki	603
Wiele problemów związanych z wątkami	605
Wieloprocusowość	607
Pule procesów	609
Kolejki	613
Problemy związane z wieloprocusowością	617
Moduł concurrent.futures	618
Moduł AsyncIO	623
AsyncIO w działaniu	624
Czytanie kodu AsyncIO	626
AsyncIO w rozwiązaniach sieciowych	627
Prezentacja aplikacji zapisującej wpisy w dzienniku	634
Klienty AsyncIO	637
Rozwiązanie problemu uczujących filozofów	640
Studium przypadku	643
Przypomnij sobie	648
Ćwiczenia	649
Podsumowanie	651
 Skorowidz	 652