

Spis treści

Wstęp	15
Do Czytelnika	15
O książce	17
Konwencje typograficzne	19
Przykłady kodu	20
Podziękowania	21
Rozdział 1. Wprowadzenie do Javy	23
1.1. Java jako platforma programistyczna	23
1.2. Słowa klucze białej księgi Javy	24
1.2.1. Prostota	25
1.2.2. Obiektowość	25
1.2.3. Sieciowość	26
1.2.4. Niezawodność	26
1.2.5. Bezpieczeństwo	26
1.2.6. Niezależność od architektury	27
1.2.7. Przenośność	28
1.2.8. Interpretacja	28
1.2.9. Wysoka wydajność	29
1.2.10. Wielowątkowość	29
1.2.11. Dynamiczność	30
1.3. Aplety Javy i internet	30
1.4. Krótka historia Javy	31
1.5. Główne nieporozumienia dotyczące Javy	35
Rozdział 2. Środowisko programistyczne Javy	38
2.1. Instalacja oprogramowania Java Development Kit	38
2.1.1. Pobieranie pakietu JDK	38
2.1.2. Instalacja pakietu JDK	39
2.1.3. Instalacja plików źródłowych i dokumentacji	41
2.2. Używanie narzędzi wiersza poleceń	42
2.3. Praca w zintegrowanym środowisku programistycznym	46
2.4. JShell	49

Rozdział 3. Podstawowe elementy języka Java	53
3.1. Prosty program w Javie	53
3.2. Komentarze	57
3.3. Typy danych	58
3.3.1. Typy całkowite	58
3.3.2. Typy zmiennoprzecinkowe	60
3.3.3. Typ char	61
3.3.4. Unicode i typ char	62
3.3.5. Typ boolean	64
3.4. Zmienne i stałe	65
3.4.1. Deklarowanie zmiennych	65
3.4.2. Inicjalizacja zmiennych	66
3.4.3. Stałe	67
3.4.4. Typ wyczerpieniowy	68
3.5. Operatory	68
3.5.1. Operatory arytmetyczne	69
3.5.2. Funkcje i stałe matematyczne	70
3.5.3. Konwersja typów numerycznych	71
3.5.4. Rzutowanie	72
3.5.5. Przypisanie	73
3.5.6. Operatory inkrementacji i dekrementacji	74
3.5.7. Operatory relacyjne i logiczne	74
3.5.8. Operator warunkowy	75
3.5.9. Wyrażenia switch	75
3.5.10. Operatory bitowe	77
3.5.11. Nawiasy i priorytety operatorów	78
3.6. Łańcuchy	79
3.6.1. Konkatenacja	79
3.6.2. Dzielenie łańcuchów	80
3.6.3. Indeksy i podłańcuchy	81
3.6.4. Łańcuchów nie można modyfikować	82
3.6.5. Porównywanie łańcuchów	83
3.6.6. Łańcuchy puste i łańcuchy null	84
3.6.7. API String	85
3.6.8. Dokumentacja API w internecie	87
3.6.9. Składanie łańcuchów	89
3.6.10. Bloki tekstowe	91
3.7. Wejście i wyjście	93
3.7.1. Odbieranie danych wejściowych	93
3.7.2. Formatowanie danych wyjściowych	96
3.7.3. Zapis i odczyt plików	99
3.8. Sterowanie wykonywaniem programu	101
3.8.1. Zasięg blokowy	102
3.8.2. Instrukcje warunkowe	102
3.8.3. Pętla	105
3.8.4. Pętla o określonej liczbie powtórzeń	109
3.8.5. Wybór wielokierunkowy — instrukcja switch	112
3.8.6. Instrukcje przerywające przepływ sterowania	116

3.9.	Wielkie liczby	119
3.10.	Tablice	122
3.10.1.	Deklarowanie tablic	122
3.10.2.	Dostęp do elementów tablicy	124
3.10.3.	Pętla typu for each	124
3.10.4.	Kopiowanie tablicy	125
3.10.5.	Argumenty wiersza poleceń	126
3.10.6.	Sortowanie tablicy	127
3.10.7.	Tablice wielowymiarowe	130
3.10.8.	Tablice postrzępione	133
Rozdział 4. Obiekty i klasy		136
4.1.	Wstęp do programowania obiektowego	136
4.1.1.	Klasy	137
4.1.2.	Obiekty	138
4.1.3.	Identyfikacja klas	139
4.1.4.	Relacje między klasami	140
4.2.	Używanie klas predefiniowanych	141
4.2.1.	Obiekty i zmienne obiektów	142
4.2.2.	Klasa LocalDate	144
4.2.3.	Metody udostępniające i zmieniające wartość elementu	146
4.3.	Definiowanie własnych klas	150
4.3.1.	Klasa Employee	150
4.3.2.	Używanie wielu plików źródłowych	153
4.3.3.	Analiza klasy Employee	153
4.3.4.	Pierwsze kroki w tworzeniu konstruktorów	154
4.3.5.	Deklarowanie zmiennych lokalnych za pomocą słowa kluczowego var	155
4.3.6.	Praca z referencjami null	156
4.3.7.	Parametry jawne i niejawne	157
4.3.8.	Korzyści z hermetyzacji	158
4.3.9.	Przywileje klasowe	161
4.3.10.	Metody prywatne	161
4.3.11.	Stałe jako pola klasy	162
4.4.	Pola i metody statyczne	162
4.4.1.	Pola statyczne	163
4.4.2.	Stałe statyczne	164
4.4.3.	Metody statyczne	164
4.4.4.	Metody fabryczne	166
4.4.5.	Metoda main	166
4.5.	Parametry metod	169
4.6.	Konstruowanie obiektów	175
4.6.1.	Przeciążanie	176
4.6.2.	Domyślna inicjalizacja pól	177
4.6.3.	Konstruktor bezargumentowy	177
4.6.4.	Jawna inicjalizacja pól	178
4.6.5.	Nazywanie parametrów	179
4.6.6.	Wywoływanie innego konstruktora	180
4.6.7.	Bloki inicjalizujące	180
4.6.8.	Niszczanie obiektów i metoda finalize	185

4.7.	Rekordy	185
4.7.1.	Koncepcja rekordu	186
4.7.2.	Konstruktory: kanoniczny, niestandardowy i kompaktowy	188
4.8.	Pakiety	190
4.8.1.	Nazwy pakietów	190
4.8.2.	Importowanie klas	191
4.8.3.	Importowanie statyczne	193
4.8.4.	Dodawanie klasy do pakietu	193
4.8.5.	Dostęp do pakietu	196
4.8.6.	Ścieżka klas	198
4.8.7.	Ustawianie ścieżki klas	200
4.9.	Pliki JAR	201
4.9.1.	Tworzenie plików JAR	201
4.9.2.	Manifest	202
4.9.3.	Wykonywalne pliki JAR	202
4.9.4.	Pliki JAR z wieloma wersjami klas	203
4.9.5.	Kilka uwag na temat opcji wiersza poleceń	205
4.10.	Komentarze dokumentacyjne	206
4.10.1.	Wstawianie komentarzy	206
4.10.2.	Komentarze do klas	207
4.10.3.	Komentarze do metod	207
4.10.4.	Komentarze do pól	208
4.10.5.	Komentarze do pakietów	209
4.10.6.	Kod HTML	209
4.10.7.	Łączy	210
4.10.8.	Komentarze ogólne	211
4.10.9.	Fragmenty kodu	212
4.10.10.	Pobieranie komentarzy	212
4.11.	Porady dotyczące projektowania klas	213
Rozdział 5. Dziedziczenie		217
5.1.	Klasy, nadklasy i podklasy	217
5.1.1.	Definiowanie podklas	218
5.1.2.	Przesłanianie metod	219
5.1.3.	Konstruktory podklas	221
5.1.4.	Hierarchia dziedziczenia	225
5.1.5.	Polimorfizm	225
5.1.6.	Zasady wywoływania metod	227
5.1.7.	Wyłączanie dziedziczenia — klasy i metody finalne	229
5.1.8.	Rzutowanie	232
5.1.9.	Operator instanceof i dopasowywanie wzorców	234
5.1.10.	Ograniczanie dostępu	236
5.2.	Kosmiczna klasa wszystkich klas — Object	238
5.2.1.	Zmienne typu Object	238
5.2.2.	Metoda equals	238
5.2.3.	Porównywanie a dziedziczenie	240
5.2.4.	Metoda hashCode	244
5.2.5.	Metoda toString	247

5.3.	Generyczne listy tablicowe	253
5.3.1.	Deklarowanie list tablicowych	253
5.3.2.	Dostęp do elementów listy tablicowej	256
5.3.3.	Zgodność pomiędzy typowanymi a surowymi listami tablicowymi	259
5.4.	Opakowania obiektów i automatyczne pakowanie	260
5.5.	Metody ze zmienną liczbą argumentów	264
5.6.	Klasy abstrakcyjne	265
5.7.	Klasy wyliczeniowe	270
5.8.	Klasy zapieczętowane	274
5.9.	Dopasowywanie do wzorca	280
5.9.1.	Obsługa wartości null	281
5.9.2.	Strażnicy	281
5.9.3.	Kompletność	282
5.9.4.	Dominacja	283
5.9.5.	Wzorce i stałe	284
5.9.6.	Zasięg zmiennych i przekazywanie sterowania w dół	284
5.10.	Refleksja	287
5.10.1.	Klasa Class	287
5.10.2.	Podstawy deklarowania wyjątków	290
5.10.3.	Zasoby	291
5.10.4.	Zastosowanie refleksji w analizie funkcjonalności klasy	294
5.10.5.	Refleksja w analizie obiektów w czasie działania programu	301
5.10.6.	Zastosowanie refleksji w generycznym kodzie tablicowym	306
5.10.7.	Wywoływanie dowolnych metod i konstruktorów	309
5.11.	Porady projektowe dotyczące dziedziczenia	313
Rozdział 6. Interfejsy, wyrażenia lambda i klasy wewnętrzne		316
6.1.	Interfejsy	317
6.1.1.	Koncepcja interfejsu	317
6.1.2.	Własności interfejsów	323
6.1.3.	Interfejsy a klasy abstrakcyjne	325
6.1.4.	Metody statyczne i prywatne	326
6.1.5.	Metody domyślne	326
6.1.6.	Wybieranie między metodami domyślnymi	328
6.1.7.	Interfejsy i wywołania zwrotne	330
6.1.8.	Interfejs Comparator	332
6.1.9.	Klonowanie obiektów	334
6.2.	Wyrażenia lambda	340
6.2.1.	Po co w ogóle są lambdy?	340
6.2.2.	Składnia wyrażeń lambda	342
6.2.3.	Interfejsy funkcyjne	344
6.2.4.	Typy funkcji	345
6.2.5.	Referencje do metod	347
6.2.6.	Referencje do konstruktorów	350
6.2.7.	Zakres dostępności zmiennych	351
6.2.8.	Przetwarzanie wyrażeń lambda	354
6.2.9.	Tworzenie komparatorów	357

6.3.	Klasy wewnętrzne	358
6.3.1.	Dostęp do stanu obiektu w klasie wewnętrznej	359
6.3.2.	Specjalne reguły składniowe dotyczące klas wewnętrznych	362
6.3.3.	Czy klasy wewnętrzne są potrzebne i bezpieczne?	363
6.3.4.	Lokalne klasy wewnętrzne	365
6.3.5.	Dostęp do zmiennych finalnych z metod zewnętrznych	365
6.3.6.	Anonimowe klasy wewnętrzne	367
6.3.7.	Klasy statyczne	370
6.4.	Moduły ładowania usług	374
6.5.	Klasy pośredniczące	376
6.5.1.	Kiedy używać klas pośredniczących?	377
6.5.2.	Tworzenie obiektów pośredniczących	377
6.5.3.	Właściwości klas pośredniczących	381
Rozdział 7. Wyjątki, asercje i dzienniki		384
7.1.	Obsługa błędów	385
7.1.1.	Klasyfikacja wyjątków	386
7.1.2.	Deklarowanie wyjątków kontrolowanych	388
7.1.3.	Zgłaszanie wyjątków	390
7.1.4.	Tworzenie klas wyjątków	391
7.2.	Przechwytywanie wyjątków	392
7.2.1.	Przechwytywanie wyjątku	393
7.2.2.	Przechwytywanie wielu typów wyjątków	395
7.2.3.	Powtórne generowanie wyjątków i budowanie łańcuchów wyjątków	397
7.2.4.	Klauzula finally	398
7.2.5.	Instrukcja try z zasobami	400
7.2.6.	Analiza danych ze stosu wywołań	402
7.3.	Wskazówki dotyczące stosowania wyjątków	406
7.4.	Asercje	410
7.4.1.	Koncepcja asercji	410
7.4.2.	Włączanie i wyłączanie asercji	411
7.4.3.	Zastosowanie asercji do sprawdzania parametrów	412
7.4.4.	Zastosowanie asercji do dokumentowania założeń	414
7.5.	Dzienniki	415
7.5.1.	Czy należy używać systemu dzienników Javy?	415
7.5.2.	Podstawy pracy z dziennikami	416
7.5.3.	API rejestrowania platformy	416
7.5.4.	Konfiguracja rejestrowania w dzienniku	418
7.5.5.	Handlerzy dzienników	419
7.5.6.	Filtry i formatery	421
7.5.7.	Przepis na dziennik	422
7.6.	Wskazówki dotyczące debugowania	429
Rozdział 8. Programowanie generyczne		435
8.1.	Dlaczego programowanie generyczne?	435
8.1.1.	Zalety parametrów typów	436
8.1.2.	Dla kogo programowanie generyczne?	437
8.2.	Definicja prostej klasy generycznej	438
8.3.	Metody generyczne	440

8.4.	Ograniczenia zmiennych typowych	441
8.5.	Kod generyczny a maszyna wirtualna	443
8.5.1.	Wymazywanie typów	444
8.5.2.	Translacja wyrażeń generycznych	445
8.5.3.	Translacja metod generycznych	445
8.5.4.	Używanie starego kodu	448
8.5.5.	Generyczne wzorce rekordów	449
8.6.	Zasady dziedziczenia dla typów generycznych	449
8.7.	Typy wieloznaczne	451
8.7.1.	Koncepcja typu wieloznacznego	451
8.7.2.	Ograniczenia nadtypów typów wieloznacznych	453
8.7.3.	Typy wieloznaczne bez ograniczeń	455
8.7.4.	Chwyatanie typu wieloznacznego	456
8.8.	Ograniczenia i braki	458
8.8.1.	Nie można podawać typów prostych jako parametrów typowych	459
8.8.2.	Sprawdzanie typów w czasie działania programu jest możliwe tylko dla typów surowych	459
8.8.3.	Nie można tworzyć tablic typów generycznych	459
8.8.4.	Ostrzeżenia dotyczące zmiennej liczby argumentów	460
8.8.5.	Generyczne parametry zmienne nie rozkładają tablic typów podstawowych	461
8.8.6.	Nie wolno tworzyć egzemplarzy zmiennych typowych	462
8.8.7.	Nie można utworzyć egzemplarza generycznej tablicy	463
8.8.8.	Zmiennych typowych nie można używać w statycznych kontekstach klas generycznych	465
8.8.9.	Obiektów klasy generycznej nie można zgłaszać ani przechwytywać	465
8.8.10.	Można wyłączyć sprawdzanie wyjątków kontrolowanych	466
8.8.11.	Uważaj na konflikty, które mogą powstać po wymazaniu typów	467
8.8.12.	Dedukcja typów w generycznych wzorcach rekordów jest ograniczona	468
8.9.	Refleksja a typy generyczne	470
8.9.1.	Generyczna klasa Class	470
8.9.2.	Zastosowanie parametrów Class<T> do dopasowywania typów	471
8.9.3.	Informacje o typach generycznych w maszynie wirtualnej	472
8.9.4.	Literały typowe	475
Rozdział 9. Kolekcje		481
9.1.	Architektura kolekcji Javy	481
9.1.1.	Oddzielenie warstwy interfejsów od warstwy klas konkretnych	482
9.1.2.	Interfejs Collection	484
9.1.3.	Iteratory	484
9.1.4.	Generyczne metody użytkowe	487
9.2.	Interfejsy w systemie kolekcji Javy	491
9.3.	Konkretne klasy kolekcyjne	493
9.3.1.	Listy powiązane	494
9.3.2.	Listy tablicowe	504
9.3.3.	Zbiór HashSet	504
9.3.4.	Zbiór TreeSet	508
9.3.5.	Kolejki Queue i Deque	512
9.3.6.	Kolejki priorytetowe	514

9.4.	Słowniki	515
9.4.1.	Podstawowe operacje słownikowe	515
9.4.2.	Modyfikowanie wpisów w słowniku	519
9.4.3.	Widoki słowników	521
9.4.4.	Klasa WeakHashMap	523
9.4.5.	Klasy LinkedHashSet i LinkedHashMap	524
9.4.6.	Klasy EnumSet i EnumMap	525
9.4.7.	Klasa IdentityHashMap	526
9.5.	Kopie i widoki	528
9.5.1.	Małe kolekcje	529
9.5.2.	Niemodyfikowalne kopie i widoki	531
9.5.3.	Przedziały	532
9.5.4.	Zbiory ze słowników wartości logicznych	533
9.5.5.	Widoki odwrócone	534
9.5.6.	Widoki kontrolowane	534
9.5.7.	Widoki synchronizowane	535
9.5.8.	Uwagi dotyczące operacji opcjonalnych	535
9.6.	Algorytmy	540
9.6.1.	Dlaczego algorytmy generyczne?	540
9.6.2.	Sortowanie i tasowanie	542
9.6.3.	Wyszukiwanie binarne	544
9.6.4.	Proste algorytmy	546
9.6.5.	Operacje zbiorowe	548
9.6.6.	Konwersja pomiędzy kolekcjami a tablicami	549
9.6.7.	Pisanie własnych algorytmów	550
9.7.	Stare kolekcje	551
9.7.1.	Klasa Hashtable	552
9.7.2.	Wyliczenia	552
9.7.3.	Słowniki własności	553
9.7.4.	Właściwości systemowe	555
9.7.5.	Stosy	557
9.7.6.	Zbiory bitów	558
Rozdział 10.	Współbieżność	562
10.1.	Uruchamianie wątków	563
10.2.	Stany wątków	567
10.2.1.	Wątki tworzone za pomocą operatora new	568
10.2.2.	Wątki wykonywalne	568
10.2.3.	Wątki BLOCKED i WAITING	569
10.2.4.	Zamykanie wątków	569
10.3.	Własności wątków	571
10.3.1.	Wątki wirtualne	571
10.3.2.	Przerywanie wątków	572
10.3.3.	Wątki demony	575
10.3.4.	Nazwy i identyfikatory wątków	576
10.3.5.	Procedury obsługi nieprzechwyconych wyjątków	576
10.3.6.	Priorytety wątków	578
10.3.7.	Fabryki i konstruktory wątków	579

10.4.	Koordinacja zadań	580
10.4.1.	Interfejsy Callable i Future	580
10.4.2.	Klasa Executors	583
10.4.3.	Wywoływanie grup zadań	587
10.4.4.	Zmienne lokalne wątków	593
10.4.5.	Metoda rozgałęzienie-złączenie	594
10.5.	Synchronizacja	597
10.5.1.	Przykład sytuacji powodującej wyścig	597
10.5.2.	Wyścigi	599
10.5.3.	Obiekty klasy Lock	601
10.5.4.	Warunki	604
10.5.5.	Zakleszczenia	609
10.5.6.	Słowo kluczowe synchronized	611
10.5.7.	Bloki synchronizowane	616
10.5.8.	Monitor	618
10.5.9.	Pola ulotne	619
10.5.10.	Pola finalne	620
10.5.11.	Zmienne atomowe	621
10.5.12.	Inicjalizacja na żądanie	623
10.5.13.	Bezpieczna publikacja	624
10.5.14.	Współdzielenie zmiennych lokalnych wątków	625
10.6.	Kolekcje bezpieczne wątkowo	626
10.6.1.	Kolejki blokujące	626
10.6.2.	Szybkie słowniki, zbiory i kolejki	632
10.6.3.	Atomowe modyfikowanie elementów słowników	634
10.6.4.	Operacje masowe na współbieżnych słownikach skrótów	637
10.6.5.	Współbieżne widoki zbiorów	639
10.6.6.	Tablice kopiowane przy zapisie	640
10.6.7.	Równoległe algorytmy tablicowe	640
10.6.8.	Starsze kolekcje bezpieczne wątkowo	641
10.7.	Obliczenia asynchroniczne	642
10.7.1.	Klasa CompletableFuture	643
10.7.2.	Tworzenie obiektów CompletableFuture	645
10.7.3.	Czasochłonne zadania w wywołaniach zwrotnych interfejsu użytkownika	651
10.8.	Procesy	657
10.8.1.	Budowanie procesu	658
10.8.2.	Uruchamianie procesu	660
10.8.3.	Uchwyty procesów	661

Rozdział 11. Adnotacje 666

11.1.	Stosowanie adnotacji	667
11.1.1.	Elementy adnotacji	667
11.1.2.	Powielanie i mnożenie adnotacji	668
11.1.3.	Adnotacje deklaracji	668
11.1.4.	Adnotacje zastosowań typów	669
11.1.5.	Określanie adresatów wprost	671
11.2.	Definiowanie adnotacji	672
11.3.	Adnotacje w API Javy	675
11.3.1.	Adnotacje kompilacji	675
11.3.2.	Metaadnotacje	677

11.4.	Przetwarzanie adnotacji podczas działania programu	678
11.5.	Przetwarzanie adnotacji w kodzie źródłowym	682
11.5.1.	Procesory adnotacji	682
11.5.2.	Interfejs programowy modelu języka	683
11.5.3.	Stosowanie adnotacji do generacji kodu źródłowego	684
11.6.	Inżynieria kodu bajtowego	689
11.6.1.	Modyfikowanie plików klasowych	689
11.6.2.	Modyfikacja kodu bajtowego podczas ładowania	695

Rozdział 12. System modułów platformy Javy 698

12.1.	Pojęcie modułu	698
12.2.	Nadawanie nazw modułom	700
12.3.	Modularny program „Witaj, świecie!”	701
12.4.	Żądanie użycia modułów	702
12.5.	Eksportowanie pakietów	704
12.6.	Modularne pliki JAR	708
12.7.	Moduły a technika refleksji	709
12.8.	Moduły automatyczne	712
12.9.	Moduł nienazwany	714
12.10.	Flagi wiersza poleceń stosowane podczas migracji	715
12.11.	Wymagania przechodnie i statyczne	717
12.12.	Eksport kwalifikowany i otwieranie	718
12.13.	Wczytywanie usług	719
12.14.	Narzędzia do pracy z modułami	721

Dodatek A 725

Skorowidz 729