

Spis treści

Wstęp	13
Do Czytelnika	13
O książce	15
Konwencje typograficzne	17
Przykłady kodu	17
Podziękowania	19
Rozdział 1. Wprowadzenie do Javy	21
1.1. Java jako platforma programistyczna	21
1.2. Słowa klucze białej księgi Javy	22
1.2.1. Prostota	23
1.2.2. Obiektowość	23
1.2.3. Sieciowość	24
1.2.4. Niezawodność	24
1.2.5. Bezpieczeństwo	24
1.2.6. Niezależność od architektury	25
1.2.7. Przenośność	26
1.2.8. Interpretacja	26
1.2.9. Wysoka wydajność	27
1.2.10. Wielowątkowość	27
1.2.11. Dynamiczność	27
1.3. Aplety Javy i internet	28
1.4. Krótka historia Javy	29
1.5. Główne nieporozumienia dotyczące Javy	32
Rozdział 2 Środowisko programistyczne Javy	35
2.1. Instalacja oprogramowania Java Development Kit	35
2.1.1. Pobieranie pakietu JDK	36
2.1.2. Instalacja pakietu JDK	37
2.1.3. Instalacja plików źródłowych i dokumentacji	39
2.2. Używanie narzędzi wiersza poleceń	40
2.3. Praca w zintegrowanym środowisku programistycznym	45
2.4. JShell	48
Rozdział 3. Podstawowe elementy języka Java	51
3.1. Prosty program w Javie	52
3.2. Komentarze	55
3.3. Typy danych	56
3.3.1. Typy całkowite	56
3.3.2. Typy zmiennoprzecinkowe	57

3.3.3. Typ char	58
3.3.4. Unicode i typ char	60
3.3.5. Typ boolean	61
3.4. Zmienne i stałe	61
3.4.1. Deklarowanie zmiennych	61
3.4.2. Inicjalizacja zmiennych	62
3.4.3. Stałe	63
3.4.4. Typ wyliczeniowy	64
3.5. Operatory	65
3.5.1. Operatory arytmetyczne	65
3.5.2. Funkcje i stałe matematyczne	66
3.5.3. Konwersja typów numerycznych	68
3.5.4. Rzutowanie	68
3.5.5. Łączenie przypisania z innymi operatorami	69
3.5.6. Operatory inkrementacji i dekrementacji	70
3.5.7. Operatory relacyjne i logiczne	70
3.5.8. Operatory bitowe	71
3.5.9. Nawiasy i priorytety operatorów	72
3.6. Łańcuchy	73
3.6.1. Podłańcuchy	73
3.6.2. Konkatenacja	74
3.6.3. Łańcuchów nie można modyfikować	74
3.6.4. Porównywanie łańcuchów	76
3.6.5. Łańcuchy puste i łańcuchy nuli	77
3.6.6. Współrzędne kodowe znaków i jednostki kodowe	77
3.6.7. API String	78
3.6.8. Dokumentacja API w internecie	81
3.6.9. Składanie łańcuchów	84
3.7. Wejście i wyjście	85
3.7.1. Odbieranie danych wejściowych	85
3.7.2. Formatowanie danych wyjściowych	88
3.7.3. Zapis i odczyt plików	92
3.8. Sterowanie wykonywaniem programu	94
3.8.1. Zasięg blokowy	94
3.8.2. Instrukcje warunkowe	95
3.8.3. Pętle	98
3.8.4. Pętle o określonej liczbie powtórzeń	101
3.8.5. Wybór wielokierunkowy — instrukcja switch	105
3.8.6. Instrukcje przerywające przepływ sterowania	107
3.9. Wielkie liczby	110
3.10. Tablice	112
3.10.1. Deklarowanie tablic	112
3.10.2. Dostęp do elementów tablicy	114
3.10.3. Pętla typu for each	114
3.10.4. Kopiowanie tablicy	115
3.10.5. Parametry wiersza poleceń	116
3.10.6. Sortowanie tablicy	117

3.10.7. Tablice wielowymiarowe	120
3.10.8. Tablice postrzępione	122
Rozdział 4. Obiekty i klasy	127
4.1. Wstęp do programowania obiektowego	128
4.1.1. Klasy	129
4.1.2. Obiekty	129
4.1.3. Identyfikacja klas	130
4.1.4. Relacje między klasami	131
4.2. Używanie klas predefiniowanych	132
4.2.1. Obiekty i zmienne obiektów	133
4.2.2. Klasa LocalDate	135
4.2.3. Metody udostępniające i zmieniające wartość elementu	137
4.3. Definiowanie własnych klas	141
4.3.1. Klasa Employee	141
4.3.2. Używanie wielu plików źródłowych	144
4.3.3. Analiza klasy Employee	144
4.3.4. Pierwsze kroki w tworzeniu konstruktorów	145
4.3.5. Deklarowanie zmiennych lokalnych za pomocą słowa kluczowego var	146
4.3.6. Praca z referencjami null	147
4.3.7. Parametry jawne i niejawne	148
4.3.8. Korzyści z hermetyzacji	149
4.3.9. Przywileje klasowe	151
4.3.10. Metody prywatne	152
4.3.11. Stałe jako pola klasy	152
4.4. Pola i metody statyczne	153
4.4.1. Pola statyczne	153
4.4.2. Stałe statyczne	154
4.4.3. Metody statyczne	155
4.4.4. Metody fabryczne	156
4.4.5. Metoda main	156
4.5. Parametry metod	159
4.6. Konstruowanie obiektów	165
4.6.1. Przeciążanie	165
4.6.2. Domyślna inicjalizacja pól	166
4.6.3. Konstruktor bezargumentowy	166
4.6.4. Jawna inicjalizacja pól	167
4.6.5. Nazywanie parametrów	168
4.6.6. Wywoływanie innego konstruktora	169
4.6.7. Bloki inicjalizujące	169
4.6.8. Niszczanie obiektów i metoda finalize	173
4.7. Pakiety	174
4.7.1. Nazwy pakietów	174
4.7.2. Importowanie klas	175
4.7.3. Importowanie statyczne	176
4.7.4. Dodawanie klasy do pakietu	177

4.7.5. Dostęp do pakietu	180
4.7.6. Ścieżka klas	181
4.7.7. Ustawianie ścieżki klas	183
4.8. Pliki JAR	184
4.8.1. Tworzenie plików JAR	184
4.8.2. Manifest	184
4.8.3. Wykonywalne pliki JAR	186
4.8.4. Pliki JAR z wieloma wersjami klas	187
4.8.5. Kilka uwag na temat opcji wiersza poleceń	188
4.9. Komentarze dokumentacyjne	189
4.9.1. Wstawianie komentarzy	190
4.9.2. Komentarze do klas	190
4.9.3. Komentarze do metod	191
4.9.4. Komentarze do pól	192
4.9.5. Komentarze ogólne	192
4.9.6. Komentarze do pakietów	193
4.9.7. Pobieranie komentarzy	194
4.10. Porady dotyczące projektowania klas	195
Rozdział 5. Dziedziczenie	199
5.1. Klasy, nadklasy i podklasy	200
5.1.1. Definiowanie podklas	200
5.1.2. Przesłanie metod	201
5.1.3. Konstruktory podklas	203
5.1.4. Hierarchia dziedziczenia	207
5.1.5. Polimorfizm	207
5.1.6. Zasady wywoływania metod	209
5.1.7. Wyłączanie dziedziczenia — klasy i metody finalne	211
5.1.8. Rzutowanie	212
5.1.9. Klasy abstrakcyjne	215
5.1.10. Ograniczanie dostępu	220
5.2. Kosmiczna klasa wszystkich klas — Object	221
5.2.1. Zmienne typu Object	221
5.2.2. Metoda equals	221
5.2.3. Porównywanie a dziedziczenie	223
5.2.4. Metoda hashCode	226
5.2.5. Metoda toString	228
5.3. Generyczne listy tablicowe	234
5.3.1. Deklarowanie list tablicowych	234
5.3.2. Dostęp do elementów listy tablicowej	237
5.3.3. Zgodność pomiędzy typowanymi a surowymi listami tablicowymi	240
5.4. Opakowania obiektów i automatyczne pakowanie	241
5.5. Metody ze zmienną liczbą parametrów	244
5.6. Klasy wyliczeniowe	246
5.7. Refleksja	248
5.7.1. Klasa Class	248
5.7.2. Podstawy deklarowania wyjątków	251

5.7.3. Zasoby	252
5.7.4. Zastosowanie refleksji w analizie funkcjonalności klasy	254
5.7.5. Refleksja w analizie obiektów w czasie działania programu	259
5.7.6. Zastosowanie refleksji w generycznym kodzie tablicowym	264
5.7.7. Wywoływanie dowolnych metod i konstruktorów	267
5.8. Porady projektowe dotyczące dziedziczenia	270

Rozdział 6. Interfejsy, wyrażenia lambda i klasy wewnętrzne **273**

6.1. Interfejsy	274
6.1.1. Koncepcja interfejsu	274
6.1.2. Własności interfejsów	280
6.1.3. Interfejsy a klasy abstrakcyjne	281
6.1.4. Metody statyczne i prywatne	282
6.1.5. Metody domyślne	283
6.1.6. Wybieranie między metodami domyślnymi	284
6.1.7. Interfejsy i wywołania zwrotne	286
6.1.8. Interfejs Comparator	289
6.1.9. Klonowanie obiektów	290
6.2. Wyrażenia lambda	296
6.2.1. Po co w ogóle są lambdy	296
6.2.2. Składnia wyrażeń lambda	297
6.2.3. Interfejsy funkcyjne	299
6.2.4. Referencje do metod	301
6.2.5. Referencje do konstruktorów	305
6.2.6. Zakres dostępności zmiennych	306
6.2.7. Przetwarzanie wyrażeń lambda	308
6.2.8. Poszerzenie wiadomości o komparatorach	311
6.3. Klasy wewnętrzne	312
6.3.1. Dostęp do stanu obiektu w klasie wewnętrznej	313
6.3.2. Specjalne reguły składniowe dotyczące klas wewnętrznych	316
6.3.3. Czy klasy wewnętrzne są potrzebne i bezpieczne?	317
6.3.4. Lokalne klasy wewnętrzne	319
6.3.5. Dostęp do zmiennych finalnych z metod zewnętrznych	320
6.3.6. Anonimowe klasy wewnętrzne	321
6.3.7. Statyczne klasy wewnętrzne	325
6.4. Moduły ładowania usług	328
6.5. Klasy pośredniczące	331
6.5.1. Kiedy używać klas pośredniczących	331
6.5.2. Tworzenie obiektów pośredniczących	331
6.5.3. Właściwości klas pośredniczących	335

Rozdział 7. Wyjątki, asercje i dzienniki **337**

7.1. Obsługa błędów	338
7.1.1. Klasyfikacja wyjątków	339
7.1.2. Deklarowanie wyjątków kontrolowanych	341
7.1.3. Zgłaszanie wyjątków	343
7.1.4. Tworzenie klas wyjątków	344

7.2. Przechwytywanie wyjątków	345
7.2.1. Przechwytywanie wyjątku	345
7.2.2. Przechwytywanie wielu typów wyjątków	347
7.2.3. Powtórne generowanie wyjątków i budowanie łańcuchów wyjątków	348
7.2.4. Klauzula finally	350
7.2.5. Instrukcja try z zasobami	352
7.2.6. Analiza danych ze stosu wywołań	354
7.3. Wskazówki dotyczące stosowania wyjątków	358
7.4. Asercje	360
7.4.1. Koncepcja asercji	361
7.4.2. Włączanie i wyłączanie asercji	362
7.4.3. Zastosowanie asercji do sprawdzania parametrów	362
7.4.4. Zastosowanie asercji do dokumentowania założeń	364
7.5. Dzienniki	365
7.5.1. Podstawy zapisu do dziennika	366
7.5.2. Zaawansowane techniki zapisu do dziennika	366
7.5.3. Zmiana konfiguracji menedżera dzienników	368
7.5.4. Lokalizacja	370
7.5.5. Obiekty typu Handler	371
7.5.6. Filtry	374
7.5.7. Formatery	374
7.5.8. Przepis na dziennik	375
7.6. Wskazówki dotyczące debugowania	383
Rozdział 8. Programowanie generyczne	389
8.1. Dlaczego programowanie generyczne	390
8.1.1. Zalety parametrów typów	390
8.1.2. Dla kogo programowanie generyczne	391
8.2. Definicja prostej klasy generycznej	392
8.3. Metody generyczne	394
8.4. Ograniczenia zmiennych typowych	396
8.5. Kod generyczny a maszyna wirtualna	398
8.5.1. Wymazywanie typów	398
8.5.2. Translacja wyrażeń generycznych	399
8.5.3. Translacja metod generycznych	400
8.5.4. Używanie starego kodu	402
8.6. Ograniczenia i braki	403
8.6.1. Nie można podawać typów prostych jako parametrów typowych	403
8.6.2. Sprawdzanie typów w czasie działania programu jest możliwe tylko dla typów surowych	403
8.6.3. Nie można tworzyć tablic typów generycznych	404
8.6.4. Ostrzeżenia dotyczące zmiennej liczby argumentów	405
8.6.5. Nie wolno tworzyć egzemplarzy zmiennych typowych	406
8.6.6. Nie można utworzyć egzemplarza generycznej tablicy	407
8.6.7. Zmiennych typowych nie można używać w statycznych kontekstach klas generycznych	408

8.6.8. Obiektów klasy generycznej nie można generować ani przechwytywać	409
8.6.9. Można wyłączyć sprawdzanie wyjątków kontrolowanych	409
8.6.10. Uważaj na konflikty, które mogą powstać po wymazaniu typów	411
8.7. Zasady dziedziczenia dla typów generycznych	412
8.8. Typy wieloznaczne	414
8.8.1. Koncepcja typu wieloznacznego	414
8.8.2. Ograniczenia nadtypów typów wieloznacznych	415
8.8.3. Typy wieloznaczne bez ograniczeń	418
8.8.4. Chwytnie typów wieloznacznych	418
8.9. Refleksja a typy generyczne	421
8.9.1. Generyczna klasa Class	421
8.9.2. Zastosowanie parametrów Class<T> do dopasowywania typów	422
8.9.3. Informacje o typach generycznych w maszynie wirtualnej	422
8.9.4. Literały typowe	426
Rozdział 9. Kolekcje	433
9.1. Architektura kolekcji Javy	434
9.1.1. Oddzielenie warstwy interfejsów od warstwy klas konkretnych	434
9.1.2. Interfejs Collection	436
9.1.3. Iteratory	437
9.1.4. Generyczne metody użytkowe	439
9.2. Interfejsy w systemie kolekcji Javy	442
9.3. Konkretnie klasy kolekcyjne	445
9.3.1. Listy powiązane	445
9.3.2. Listy tablicowe	454
9.3.3. Zbiór HashSet	454
9.3.4. Zbiór TreeSet	458
9.3.5. Kolejki Queue i Deque	462
9.3.6. Kolejki priorytetowe	464
9.4. Słowniki	465
9.4.1. Podstawowe operacje słownikowe	465
9.4.2. Modyfikowanie wpisów w słowniku	468
9.4.3. Widoki słowników	470
9.4.4. Klasa WeakHashMap	471
9.4.5. Klasy LinkedHashMap i LinkedHashMap	472
9.4.6. Klasy EnumSet i EnumMap	473
9.4.7. Klasa IdentityHashMap	474
9.5. Widoki i opakowania	476
9.5.1. Małe kolekcje	476
9.5.2. Przedziały	478
9.5.3. Widoki niemodyfikowalne	478
9.5.4. Widoki synchronizowane	480
9.5.5. Widoki kontrolowane	480
9.5.6. Uwagi dotyczące operacji opcjonalnych	481
9.6. Algorytmy	485
9.6.1. Dlaczego algorytmy generyczne	485

9.6.2. Sortowanie i tasowanie	486
9.6.3. Wyszukiwanie binarne	489
9.6.4. Proste algorytmy	490
9.6.5. Operacje zbiorowe	492
9.6.6. Konwersja pomiędzy kolekcjami a tablicami	493
9.6.7. Pisanie własnych algorytmów	493
9.7. Stare kolekcje	495
9.7.1. Klasa Hashtable	495
9.7.2. Wyliczenia	495
9.7.3. Słowniki własności	496
9.7.4. Stosy	500
9.7.5. Zbiory bitów	500
Rozdział 10. Graficzne interfejsy użytkownika	505
10.1. Historia zestawów narzędzi do tworzenia interfejsów użytkownika	505
10.2. Wyświetlanie ramki	507
10.2.1. Tworzenie ramki	507
10.2.2. Właściwości ramki	509
10.3. Wyświetlanie informacji w komponencie	512
10.3.1. Figury 2D	517
10.3.2. Kolory	523
10.3.3. Czcionki	524
10.3.4. Wyświetlanie obrazów	530
10.4. Obsługa zdarzeń	531
10.4.1. Podstawowe koncepcje obsługi zdarzeń	531
10.4.2. Przykład — obsługa kliknięcia przycisku	533
10.4.3. Zwiąże definiowanie procedur nasłuchowych	536
10.4.4. Klasy adaptacyjne	537
10.4.5. Akcje	539
10.4.6. Zdarzenia generowane przez mysz	545
10.4.7. Hierarchia zdarzeń w bibliotece AWT	550
10.5. API Preferences	552
Rozdział 11. Komponent» Swing interfejsu użytkownika	559
11.1. Swing i wzorzec model-widok-kontroler	560
11.2. Wprowadzenie do zarządzania rozkładem	563
11.2.1. Zarządcy układu	563
11.2.2. Rozkład brzegowy	565
11.2.3. Rozkład siatkowy	567
11.3. Wprowadzanie tekstu	568
11.3.1. Pola tekstowe	568
11.3.2. Etykiety komponentów	570
11.3.3. Pola haseł	571
11.3.4. Obszary tekstowe	572
11.3.5. Panele przewijane	573
11.4. Komponenty umożliwiające wybór opcji	575
11.4.1. Pola wyboru	575

11.4.2. Przełączniki	577
11.4.3. Obramowanie	581
11.4.4. Listy rozwijane	583
11.4.5. Suwaki	586
11.5. Menu	592
11.5.1. Tworzenie menu	592
11.5.2. Ikony w elementach menu	595
11.5.3. Pola wyboru i przełączniki jako elementy menu	596
11.5.4. Menu podręczne	597
11.5.5. Mnemoniki i akceleratory	598
11.5.6. Aktywowanie i dezaktywowanie elementów menu	600
11.5.7. Paski narzędzi	604
11.5.8. Dymki	606
11.6. Zaawansowane techniki zarządzania rozkładem	607
11.6.1. Rozkład GridBagLayout	607
11.6.2. Niestandardowi zarządcy rozkładu	616
11.7. Okna dialogowe	620
11.7.1. Okna dialogowe opcji	621
11.7.2. Tworzenie okien dialogowych	625
11.7.3. Wymiana danych	629
11.7.4. Okna dialogowe wyboru plików	634
Rozdział 12. Współbieżność	643
12.1. Czym są wątki	644
12.2. Stany wątków	648
12.2.1. Wątki tworzone za pomocą operatora new	649
12.2.2. Wątki RUNNABLE	649
12.2.3. Wątki BLOCKED i WAITING	650
12.2.4. Zamykanie wątków	650
12.3. Własności wątków	652
12.3.1. Przerywanie wątków	652
12.3.2. Wątki demony	655
12.3.3. Nazwy wątków	655
12.3.4. Procedury obsługi nieprzechwyconych wyjątków	655
12.3.5. Priorytety wątków	657
12.4. Synchronizacja	658
12.4.1. Przykład sytuacji powodującej wyścig	658
12.4.2. Wyścigi	660
12.4.3. Obiekty klasy Lock	662
12.4.4. Warunki	665
12.4.5. Słowo kluczowe synchronized	670
12.4.6. Bloki synchronizowane	674
12.4.7. Monitor	675
12.4.8. Pola ulotne	676
12.4.9. Zmienne finalne	677
12.4.10. Zmienne atomowe	677
12.4.11. Zakleszczenia	679

12.4.12. Zmienne lokalne wątków	682
12.4.13. Dlaczego metody stop i suspend są wycofywane	683
12.5. Kolekcje bezpieczne wątkowo	685
12.5.1. Kolejki blokujące	685
12.5.2. Szybkie słowniki, zbiory i kolejki	692
12.5.3. Atomowe modyfikowanie elementów słowników	693
12.5.4. Operacje masowe na współbieżnych słownikach skrótów	696
12.5.5. Współbieżne widoki zbiorów	698
12.5.6. Tablice kopiowane przy zapisie	699
12.5.7. Równoległe algorytmy tablicowe	699
12.5.8. Starsze kolekcje bezpieczne wątkowo	700
12.6. Zadania i pule wątków	701
12.6.1. Interfejsy Callable i Future	702
12.6.2. Klasa Executors	704
12.6.3. Kontrolowanie grup zadań	707
12.6.4. Metoda rozgałęzienie-złączenie	711
12.7. Obliczenia asynchroniczne	714
12.7.1. Klasa CompletableFuture	714
12.7.2. Tworzenie obiektów CompletableFuture	716
12.7.3. Czasochłonne zadania w wywołaniach zwrotnych interfejsu użytkownika	722
12.8. Procesy	728
12.8.1. Budowanie procesu	729
12.8.2. Uruchamianie procesu	730
12.8.3. Uchwyty procesów	731
Dodatek A	737
Skorowidz	741