

WYDANIE IX

JAVATM

Podstawy



CAY S. HORSTMANN • GARY CORNELL

Spis treści

Wstęp	13
Podziękowania	19
Rozdział 1. Wstęp do Javy	21
1.1. Java jako platforma programistyczna	21
1.2. Słowa kluczowe białej książki Javy	22
1.2.1. Prosty	23
1.2.2. Obiektowy	24
1.2.3. Sieciowy	24
1.2.4. Niezawodny	24
1.2.5. Bezpieczny	25
1.2.6. Niezależny od architektury	26
1.2.7. Przenośny	26
1.2.8. Interpretowany	27
1.2.9. Wysokowydajny	27
1.2.10. Wielowątkowy	28
1.2.11. Dynamiczny	28
1.3. Aplety Javy i internet	28
1.4. Krótka historia Javy	30
1.5. Główne nieporozumienia dotyczące Javy	32
Rozdział 2. Środowisko programistyczne Javy	37
2.1. Instalacja oprogramowania Java Development Kit	38
2.1.1. Pobieranie pakietu JDK	38
2.1.2. Ustawianie ścieżki dostępu	39
2.1.3. Instalacja bibliotek i dokumentacji	41
2.1.4. Instalacja przykładowych programów	42
2.1.5. Drzewo katalogów Javy	42
2.2. Wybór środowiska programistycznego	43
2.3. Używanie narzędzi wiersza poleceń	44
2.3.1. Rozwiązywanie problemów	45
2.4. Praca w zintegrowanym środowisku programistycznym	47
2.4.1. Znajdowanie błędów kompilacji	49

2.5.	Uruchamianie aplikacji graficznej	50
2.6.	Tworzenie i uruchamianie apletów	53

Rozdział 3. Podstawowe elementy języka Java 57

3.1.	Prosty program w Javie	58
3.2.	Komentarze	61
3.3.	Typy danych	62
3.3.1.	Typy całkowite	62
3.3.2.	Typy zmiennoprzecinkowe	63
3.3.3.	Typ char	65
3.3.4.	Typ boolean	66
3.4.	Zmienne	66
3.4.1.	Inicjacja zmiennych	68
3.4.2.	Stałe	68
3.5.	Operatory	69
3.5.1.	Operatory inkrementacji i dekrementacji	71
3.5.2.	Operatory relacyjne i logiczne	71
3.5.3.	Operatory bitowe	72
3.5.4.	Funkcje i stałe matematyczne	73
3.5.5.	Konwersja typów numerycznych	74
3.5.6.	Rzutowanie	75
3.5.7.	Nawiasy i priorytety operatorów	76
3.5.8.	Typ wyliczeniowy	77
3.6.	Łańcuchy	77
3.6.1.	Podłańcuchy	78
3.6.2.	Konkatenacja	78
3.6.3.	Łańcuchów nie można modyfikować	79
3.6.4.	Porównywanie łańcuchów	79
3.6.5.	Łańcuchy puste i łańcuchy null	81
3.6.6.	Współrzędne kodowe znaków i jednostki kodowe	81
3.6.7.	API String	83
3.6.8.	Dokumentacja API w internecie	85
3.6.9.	Składanie łańcuchów	86
3.7.	Wejście i wyjście	89
3.7.1.	Odbieranie danych wejściowych	89
3.7.2.	Formatowanie danych wyjściowych	91
3.7.3.	Zapis i odczyt plików	96
3.8.	Przepływ sterowania	98
3.8.1.	Zasięg blokowy	98
3.8.2.	Instrukcje warunkowe	99
3.8.3.	Pętle	101
3.8.4.	Pętle o określonej liczbie powtórzeń	106
3.8.5.	Wybór wielokierunkowy — instrukcja switch	109
3.8.6.	Instrukcje przerywające przepływ sterowania	111
3.9.	Wielkie liczby	114
3.10.	Tablice	116
3.10.1.	Pętla typu for each	117
3.10.2.	Inicjowanie tablic i tworzenie tablic anonimowych	118
3.10.3.	Kopiowanie tablicy	119
3.10.4.	Parametry wiersza poleceń	120
3.10.5.	Sortowanie tablicy	121

3.10.6. Tablice wielowymiarowe	124
3.10.7. Tablice postrzępione	127

Rozdział 4. Obiekty i klasy131

4.1. Wstęp do programowania obiektowego	132
4.1.1. Klasy	132
4.1.2. Obiekty	133
4.1.3. Identyfikacja klas	134
4.1.4. Relacje między klasami	135
4.2. Używanie klas predefiniowanych	137
4.2.1. Obiekty i zmienne obiektów	137
4.2.2. Klasa GregorianCalendar	139
4.2.3. Metody udostępniające i zmieniające wartość elementu	141
4.3. Definiowanie własnych klas	148
4.3.1. Klasa Employee	148
4.3.2. Używanie wielu plików źródłowych	151
4.3.3. Analiza klasy Employee	151
4.3.4. Pierwsze kroki w tworzeniu konstruktorów	152
4.3.5. Parametry jawne i niejawne	153
4.3.6. Korzyści z hermetyzacji	155
4.3.7. Przywileje klasowe	157
4.3.8. Metody prywatne	157
4.3.9. Stałe jako pola klasy	158
4.4. Pola i metody statyczne	158
4.4.1. Pola statyczne	159
4.4.2. Stałe statyczne	159
4.4.3. Metody statyczne	160
4.4.4. Metody fabryczne	161
4.4.5. Metoda main	162
4.5. Parametry metod	164
4.6. Konstruowanie obiektów	171
4.6.1. Przeciążanie	171
4.6.2. Inicjacja pól wartościami domyślnymi	171
4.6.3. Konstruktor bezargumentowy	172
4.6.4. Jawna inicjacja pól	172
4.6.5. Nazywanie parametrów	174
4.6.6. Wywoływanie innego konstruktora	174
4.6.7. Bloki inicjujące	175
4.6.8. Niszczenie obiektów i metoda finalize	179
4.7. Pakiety	180
4.7.1. Importowanie klas	180
4.7.2. Importy statyczne	182
4.7.3. Dodawanie klasy do pakietu	182
4.7.4. Zasięg pakietów	185
4.8. Ścieżka klas	187
4.8.1. Ustawianie ścieżki klas	189
4.9. Komentarze dokumentacyjne	190
4.9.1. Wstawianie komentarzy	190
4.9.2. Komentarze do klas	191
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 i ogólne	194
4.9.7.	Generowanie dokumentacji	194
4.10.	Porady dotyczące projektowania klas	195

Rozdział 5. Dziedziczenie 199

5.1.	Klasy, nadklasy i podklasy	200
5.1.1.	Hierarchia dziedziczenia	206
5.1.2.	Polimorfizm	207
5.1.3.	Wiązanie dynamiczne	209
5.1.4.	Wyłączanie dziedziczenia — klasy i metody finalne	211
5.1.5.	Rzutowanie	212
5.1.6.	Klasy abstrakcyjne	214
5.1.7.	Ochrona dostępu	219
5.2.	Klasa bazowa Object	220
5.2.1.	Metoda equals	221
5.2.2.	Porównywanie a dziedziczenie	222
5.2.3.	Metoda hashCode	225
5.2.4.	Metoda toString	228
5.3.	Generyczne listy tablicowe	233
5.3.1.	Dostęp do elementów listy tablicowej	236
5.3.2.	Zgodność pomiędzy typowanymi a surowymi listami tablicowymi	239
5.4.	Ośłony obiektów i autoboxing	241
5.5.	Metody ze zmienną liczbą parametrów	244
5.6.	Klasy wyliczeniowe	245
5.7.	Refleksja	247
5.7.1.	Klasa Class	248
5.7.2.	Podstawy przechwytywania wyjątków	250
5.7.3.	Zastosowanie refleksji w analizie funkcjonalności klasy	252
5.7.4.	Refleksja w analizie obiektów w czasie działania programu	257
5.7.5.	Zastosowanie refleksji w generycznym kodzie tablicowym	261
5.7.6.	Wywoływanie dowolnych metod	264
5.8.	Porady projektowe dotyczące dziedziczenia	268

Rozdział 6. Interfejsy i klasy wewnętrzne 271

6.1.	Interfejsy	272
6.1.1.	Własności interfejsów	276
6.1.2.	Interfejsy a klasy abstrakcyjne	279
6.2.	Klonowanie obiektów	280
6.3.	Interfejsy a sprzężenie zwrotne	286
6.4.	Klasy wewnętrzne	289
6.4.1.	Dostęp do stanu obiektu w klasie wewnętrznej	289
6.4.2.	Specjalne reguły składniowe dotyczące klas wewnętrznych	293
6.4.3.	Czy klasy wewnętrzne są potrzebne i bezpieczne?	294
6.4.4.	Lokalne klasy wewnętrzne	296
6.4.5.	Dostęp do zmiennych finalnych z metod zewnętrznych	297
6.4.6.	Anonimowe klasy wewnętrzne	300
6.4.7.	Statyczne klasy wewnętrzne	303
6.5.	Klasy proxy	306
6.5.1.	Własności klas proxy	311

Rozdział 7. Grafika	313
7.1. Wprowadzenie do pakietu Swing	314
7.2. Tworzenie ramki	318
7.3. Pozycjonowanie ramki	321
7.3.1. Własności ramek	322
7.3.2. Określanie rozmiaru ramki	323
7.4. Wyświetlanie informacji w komponencie	327
7.5. Figury 2D	332
7.6. Kolory	340
7.7. Czcionki	343
7.8. Wyświetlanie obrazów	351
Rozdział 8. Obsługa zdarzeń	355
8.1. Podstawy obsługi zdarzeń	355
8.1.1. Przykład — obsługa kliknięcia przycisku	357
8.1.2. Nabywanie biegłości w posługiwaniu się klasami wewnętrznymi	362
8.1.3. Tworzenie słuchaczy zawierających jedno wywołanie metody	364
8.1.4. Przykład — zmiana stylu	366
8.1.5. Klasy adaptacyjne	369
8.2. Akcje	373
8.3. Zdarzenia generowane przez mysz	380
8.4. Hierarchia zdarzeń w bibliotece AWT	387
8.4.1. Zdarzenia semantyczne i niskiego poziomu	388
Rozdział 9. Komponenty Swing interfejsu użytkownika	391
9.1. Swing a wzorzec projektowy Model-View-Controller	392
9.1.1. Wzorce projektowe	392
9.1.2. Wzorzec Model-View-Controller	393
9.1.3. Analiza MVC przycisków Swing	397
9.2. Wprowadzenie do zarządzania rozkładem	398
9.2.1. Rozkład brzegowy	400
9.2.2. Rozkład siatkowy	402
9.3. Wprowadzanie tekstu	406
9.3.1. Pola tekstowe	406
9.3.2. Etykiety komponentów	408
9.3.3. Pola haseł	410
9.3.4. Obszary tekstowe	410
9.3.5. Panele przewijane	411
9.4. Komponenty umożliwiające wybór opcji	413
9.4.1. Pola wyboru	413
9.4.2. Przełączniki	415
9.4.3. Obramowanie	419
9.4.4. Listy rozwijalne	423
9.4.5. Suwaki	426
9.5. Menu	432
9.5.1. Tworzenie menu	432
9.5.2. Ikony w elementach menu	435
9.5.3. Pola wyboru i przełączniki jako elementy menu	436
9.5.4. Menu podręczne	437
9.5.5. Mnemoniki i akceleratory	438

9.5.6.	Aktywowanie i dezaktywowanie elementów menu	440
9.5.7.	Paski narzędzi	444
9.5.8.	Dymki	446
9.6.	Zaawansowane techniki zarządzania rozkładem	448
9.6.1.	Rozkład GridBagLayout	449
9.6.2.	Rozkład grupowy	459
9.6.3.	Nieuzywanie żadnego zarządcy rozkładu	468
9.6.4.	Niestandardowi zarządcy rozkładu	469
9.6.5.	Kolejka dostępu	472
9.7.	Okna dialogowe	474
9.7.1.	Okna dialogowe opcji	474
9.7.2.	Tworzenie okien dialogowych	484
9.7.3.	Wymiana danych	489
9.7.4.	Okna dialogowe wyboru plików	495
9.7.5.	Okna dialogowe wyboru kolorów	505

Rozdział 10. Przygotowywanie apletów i aplikacji do użytku511

10.1.	Pliki JAR	512
10.1.1.	Manifest	512
10.1.2.	Wykonywalne pliki JAR	514
10.1.3.	Zasoby	515
10.1.4.	Pieczątowanie pakietów	518
10.2.	Java Web Start	519
10.2.1.	Piaskownica	522
10.2.2.	Podpisywanie kodu	523
10.2.3.	API JNLP	525
10.3.	Aplety	533
10.3.1.	Prosty aplet	533
10.3.2.	Znacznik applet i jego atrybuty	537
10.3.3.	Znacznik object	540
10.3.4.	Parametry przekazujące informacje do apletów	541
10.3.5.	Dostęp do obrazów i plików audio	546
10.3.6.	Środowisko działania apletu	547
10.4.	Zapisywanie preferencji użytkownika	549
10.4.1.	Mapy własności	550
10.4.2.	API Preferences	555

Rozdział 11. Wyjątki, dzienniki, asercje i debugowanie563

11.1.	Obsługa błędów	564
11.1.1.	Klasyfikacja wyjątków	565
11.1.2.	Deklarowanie wyjątków kontrolowanych	567
11.1.3.	Zgłaszanie wyjątków	569
11.1.4.	Tworzenie klas wyjątków	570
11.2.	Przechwytywanie wyjątków	571
11.2.1.	Przechwytywanie wielu typów wyjątków	574
11.2.2.	Powtórne generowanie wyjątków i budowanie łańcuchów wyjątków	575
11.2.3.	Klauzula finally	576
11.2.4.	Instrukcja try z zasobami	580
11.2.5.	Analiza danych ze śledzenia stosu	581
11.3.	Wskazówki dotyczące stosowania wyjątków	584

11.4.	Asercje	587
11.4.1.	Włączanie i wyłączanie asercji	588
11.4.2.	Zastosowanie asercji do sprawdzania parametrów	589
11.4.3.	Zastosowanie asercji do dokumentowania założeń	590
11.5.	Dzienniki	591
11.5.1.	Podstawy zapisu do dziennika	592
11.5.2.	Zaawansowane techniki zapisu do dziennika	592
11.5.3.	Zmiana konfiguracji menedżera dzienników	594
11.5.4.	Lokalizacja	596
11.5.5.	Obiekty typu Handler	596
11.5.6.	Filtry	600
11.5.7.	Formatery	600
11.5.8.	Przepis na dziennik	601
11.6.	Wskazówki dotyczące debugowania	609
11.7.	Wskazówki dotyczące debugowania aplikacji z GUI	614
11.7.1.	Zaprzęganie robota AWT do pracy	617
11.8.	Praca z debuggerem	621
Rozdział 12.	Programowanie ogólne	627
12.1.	Dlaczego programowanie ogólne	628
12.1.1.	Dla kogo programowanie ogólne	629
12.2.	Definicja prostej klasy ogólnej	630
12.3.	Metody ogólne	632
12.4.	Ograniczenia zmiennych typowych	633
12.5.	Kod ogólny a maszyna wirtualna	635
12.5.1.	Translacja wyrażeń generycznych	637
12.5.2.	Translacja metod ogólnych	637
12.5.3.	Używanie starego kodu	639
12.6.	Ograniczenia i braki	641
12.6.1.	Nie można podawać typów prostych jako parametrów typowych	641
12.6.2.	Sprawdzanie typów w czasie działania programu jest możliwe tylko dla typów surowych	641
12.6.3.	Nie można tworzyć tablic typów ogólnych	642
12.6.4.	Ostrzeżenia dotyczące zmiennej liczby argumentów	642
12.6.5.	Nie wolno tworzyć egzemplarzy zmiennych typowych	643
12.6.6.	Zmiennych typowych nie można używać w statycznych kontekstach klas ogólnych	645
12.6.7.	Obiektów klasy ogólnej nie można generować ani przechwytywać	646
12.6.8.	Uważaj na konflikty, które mogą powstać po wymazaniu typów	648
12.7.	Zasady dziedziczenia dla typów ogólnych	649
12.8.	Typy wieloznaczne	650
12.8.1.	Ograniczenia nadtypów typów wieloznacznych	652
12.8.2.	Typy wieloznaczne bez ograniczeń	655
12.8.3.	Chwyatanie typu wieloznacznego	655
12.9.	Refleksja a typy ogólne	658
12.9.1.	Zastosowanie parametrów Class<T> do dopasowywania typów	659
12.9.2.	Informacje o typach generycznych w maszynie wirtualnej	659
Rozdział 13.	Kolekcje	665
13.1.	Interfejsy kolekcyjne	665
13.1.1.	Oddzielenie warstwy interfejsów od warstwy klas konkretnych	666
13.1.2.	Interfejsy Collection i Iterator	668

13.2.	Konkretne klasy kolekcyjne	674
13.2.1.	Listy powiązane	674
13.2.2.	Listy tablicowe	684
13.2.3.	Zbiór HashSet	684
13.2.4.	Zbiór TreeSet	688
13.2.5.	Porównywanie obiektów	689
13.2.6.	Kolejki Queue i Deque	694
13.2.7.	Kolejki priorytetowe	696
13.2.8.	Mapy	697
13.2.9.	Specjalne klasy Set i Map	702
13.3.	Architektura kolekcji	706
13.3.1.	Widoki i obiekty opakowujące	709
13.3.2.	Operacje zbiorcze	717
13.3.3.	Konwersja pomiędzy kolekcjami a tablicami	718
13.4.	Algorytmy	718
13.4.1.	Sortowanie i tasowanie	720
13.4.2.	Wyszukiwanie binarne	722
13.4.3.	Proste algorytmy	723
13.4.4.	Pisanie własnych algorytmów	725
13.5.	Stare kolekcje	726
13.5.1.	Klasa Hashtable	726
13.5.2.	Wyliczenia	727
13.5.3.	Mapy własności	728
13.5.4.	Stosy	729
13.5.5.	Zbiory bitów	729
Rozdział 14.	Wielowątkowość	735
14.1.	Czym są wątki	736
14.1.1.	Wykonywanie zadań w osobnych wątkach	741
14.2.	Przerywanie wątków	746
14.3.	Stany wątków	749
14.3.1.	Wątki NEW	749
14.3.2.	Wątki RUNNABLE	750
14.3.3.	Wątki BLOCKED i WAITING	750
14.3.4.	Zamykanie wątków	752
14.4.	Własności wątków	752
14.4.1.	Priorytety wątków	752
14.4.2.	Wątki demony	753
14.4.3.	Procedury obsługi nieprzechwyconych wyjątków	754
14.5.	Synchronizacja	756
14.5.1.	Przykład sytuacji powodującej wyścig	756
14.5.2.	Wyścigi	760
14.5.3.	Obiekty klasy Lock	762
14.5.4.	Warunki	765
14.5.5.	Słowo kluczowe synchronized	769
14.5.6.	Bloki synchronizowane	774
14.5.7.	Monitor	775
14.5.8.	Pola ulotne	776
14.5.9.	Zmienne finalne	777
14.5.10.	Zmienne atomowe	777
14.5.11.	Zakleszczenia	778

14.5.12. Zmienne lokalne wątków	781
14.5.13. Testowanie blokad i odmierzanie czasu	782
14.5.14. Blokada odczytu-zapisu	783
14.5.15. Dlaczego metody stop i suspend są wycofywane	784
14.6. Kolejki blokujące	786
14.7. Kolekcje bezpieczne wątkowo	794
14.7.1. Szybkie mapy, zbiory i kolejki	794
14.7.2. Tablice kopiowane przy zapisie	796
14.7.3. Starsze kolekcje bezpieczne wątkowo	796
14.8. Interfejsy Callable i Future	797
14.9. Klasa Executors	802
14.9.1. Pule wątków	803
14.9.2. Planowanie wykonywania	807
14.9.3. Kontrolowanie grup zadań	808
14.9.4. Szkielet rozgałęzienie-złączenie	809
14.10. Synchronizatory	812
14.10.1. Semafor	812
14.10.2. Klasa CountdownLatch	813
14.10.3. Bariery	814
14.10.4. Klasa Exchanger	814
14.10.5. Kolejki synchroniczne	815
14.11. Wątki a biblioteka Swing	815
14.11.1. Uruchamianie czasochłonnych zadań	816
14.11.2. Klasa SwingWorker	820
14.11.3. Zasada jednego wątku	827

Dodatek A. Słowa kluczowe Javy	829
---	------------

Skorowidz	831
------------------------	------------