

Spis treści

Wstęp	15
Podziękowania	16
O autorze	17
1 Podstawowe struktury programistyczne	19
1.1. Nasz pierwszy program	20
1.1.1. Analiza programu „Witaj, świecie!”	20
1.1.2. Kompilacja i uruchamianie programu w języku Java	21
1.1.3. Wywołania metod	24
1.1.4. JShell	25
1.2. Typy proste	28
1.2.1. Typy całkowite ze znakiem	28
1.2.2. Typy zmiennoprzecinkowe	29
1.2.3. Typ char	30
1.2.4. Typ boolean	30
1.3. Zmienne	30
1.3.1. Deklaracje zmiennych	30
1.3.2. Identyfikatory	31
1.3.3. Inicjalizacja	31
1.3.4. Stałe	32
1.4. Działania arytmetyczne	33
1.4.1. Przypisanie	33
1.4.2. Podstawowa arytmetyka	34
1.4.3. Metody matematyczne	35
1.4.4. Konwersja typów liczbowych	35
1.4.5. Operatory relacji i operatory logiczne	37
1.4.6. Duże liczby	38
1.5. Ciągi znaków	38
1.5.1. Łączenie ciągów znaków	38
1.5.2. Wycinanie ciągów znaków	39
1.5.3. Porównywanie ciągów znaków	40
1.5.4. Konwersja liczb na znaki i znaków na liczby	41
1.5.5. API klasy String	41
1.5.6. Kodowanie znaków w języku Java	43
1.5.7. Bloki tekstu	45

1.6. Wejście i wyjście	46
1.6.1. Wczytywanie danych wejściowych	46
1.6.2. Formatowanie generowanych danych	47
1.7. Kontrola przepływu	49
1.7.1. Instrukcje warunkowe	49
1.7.2. Słowo kluczowe switch	50
1.7.3. Pętle	51
1.7.4. Przerywanie i kontynuacja	53
1.7.5. Zasięg zmiennych lokalnych	54
1.8. Tablice i listy tablic	55
1.8.1. Obsługa tablic	55
1.8.2. Tworzenie tablicy	56
1.8.3. Klasa ArrayList	57
1.8.4. Klasy opakujące typy proste	58
1.8.5. Rozszerzona pętla for	58
1.8.6. Kopiowanie tablic i obiektów ArrayList	59
1.8.7. Algorytmy tablic	60
1.8.8. Parametry wiersza poleceń	60
1.8.9. Tablice wielowymiarowe	61
1.9. Dekompozycja funkcjonalna	63
1.9.1. Deklarowanie i wywoływanie metod statycznych	63
1.9.2. Parametry tablicowe i zwracane wartości	63
1.9.3. Zmienna liczba parametrów	64
Ćwiczenia	65
2 Programowanie obiektowe	67
2.1. Praca z obiektami	68
2.1.1. Metody dostępne i modyfikujące	69
2.1.2. Referencje do obiektu	70
2.2. Implementowanie klas	71
2.2.1. Zmienne instancyjne	71
2.2.2. Nagłówki metod	72
2.2.3. Treści metod	72
2.2.4. Wywołania metod instancyjnych	73
2.2.5. Referencja this	73
2.2.6. Wywołanie przez wartość	74
2.3. Tworzenie obiektów	75
2.3.1. Implementacja konstruktorów	75
2.3.2. Przeciążanie	75
2.3.3. Wywoływanie jednego konstruktora z innego	76
2.3.4. Domyślna inicjalizacja	76
2.3.5. Inicjalizacja zmiennych instancyjnych	77
2.3.6. Zmienne instancyjne z modyfikatorem final	77
2.3.7. Konstruktor bez parametrów	78
2.4. Rekordy	78
2.4.1. Koncepcja rekordu	79
2.4.2. Konstruktory: kanoniczny, niestandardowy i kompaktowy	80
2.5. Statyczne zmienne i metody	81
2.5.1. Zmienne statyczne	81
2.5.2. Stałe statyczne	81
2.5.3. Statyczne bloki inicjalizacyjne	82
2.5.4. Metody statyczne	82
2.5.5. Metody wytwórcze	83

2.6. Pakiety	84
2.6.1. Deklarowanie pakietów	84
2.6.2. Polecenie jar	85
2.6.3. Ścieżka przeszukiwań dla klas	86
2.6.4. Dostęp do pakietu	87
2.6.5. Importowanie klas	87
2.6.6. Import metod statycznych	88
2.7. Klasy zagnieżdżone	89
2.7.1. Statyczne klasy zagnieżdżone	89
2.7.2. Klasy wewnętrzne	90
2.7.3. Specjalne reguły składni dla klas wewnętrznych	92
2.8. Komentarze do dokumentacji	93
2.8.1. Wstawianie komentarzy	93
2.8.2. Komentarze klasy	93
2.8.3. Komentarze metod	94
2.8.4. Komentarze zmiennych	94
2.8.5. Ogólne komentarze	94
2.8.6. Odnośniki	95
2.8.7. Opisy pakietów, modułów i ogólne	96
2.8.8. Wycinanie komentarzy	96
Ćwiczenia	97
3 Interfejsy i wyrażenia lambda	99
3.1. Interfejsy	100
3.1.1. Korzystanie z interfejsów	100
3.1.2. Deklarowanie interfejsu	101
3.1.3. Implementowanie interfejsu	101
3.1.4. Konwersja do typu interfejsu	103
3.1.5. Rzutowanie i operator instanceof	103
3.1.6. Forma operatora instanceof z „dopasowywaniem wzorca”	104
3.1.7. Rozszerzanie interfejsów	105
3.1.8. Implementacja wielu interfejsów	105
3.1.9. Stałe	105
3.2. Metody statyczne, domyślne i prywatne	105
3.2.1. Metody statyczne	106
3.2.2. Metody domyślne	106
3.2.3. Rozstrzyganie konfliktów metod domyślnych	107
3.2.4. Metody prywatne	108
3.3. Przykłady interfejsów	108
3.3.1. Interfejs Comparable	108
3.3.2. Interfejs Comparator	109
3.3.3. Interfejs Runnable	110
3.3.4. Wywołania zwrotne interfejsu użytkownika	111
3.4. Wyrażenia lambda	111
3.4.1. Składnia wyrażeń lambda	112
3.4.2. Interfejsy funkcyjne	113
3.5. Referencje do metod i konstruktora	113
3.5.1. Referencje do metod	114
3.5.2. Referencje konstruktora	115
3.6. Przetwarzanie wyrażeń lambda	115
3.6.1. Implementacja odroczonego wykonania	115
3.6.2. Wybór interfejsu funkcyjnego	116
3.6.3. Implementowanie własnych interfejsów funkcyjnych	118

3.7. Wyrażenia lambda i zasięg zmiennych	119
3.7.1. Zasięg zmiennej lambda	119
3.7.2. Dostęp do zmiennych zewnętrznych	119
3.8. Funkcje wyższych rzędów	121
3.8.1. Metody zwracające funkcje	121
3.8.2. Metody modyfikujące funkcje	122
3.8.3. Metody interfejsu Comparator	122
3.9. Klasy lokalne i anonimowe	123
3.9.1. Klasy lokalne	123
3.9.2. Klasy anonimowe	124
Ćwiczenia	125
4 Dziedziczenie i mechanizm refleksji	127
4.1. Rozszerzanie klas	128
4.1.1. Klasy nadrzędne i podrzędne	128
4.1.2. Definiowanie i dziedziczenie metod klas podrzędnych	129
4.1.3. Przesłanianie metod	129
4.1.4. Tworzenie klasy podrzędnej	130
4.1.5. Przypisania klas nadrzędnych	130
4.1.6. Rzutowanie	131
4.1.7. Anonimowe klasy podrzędne	132
4.1.8. Wywołania metod z super	132
4.2. Hierarchie dziedziczenia	133
4.2.1. Metody i klasy z modyfikatorem final	133
4.2.2. Abstrakcyjne metody i klasy	133
4.2.3. Ograniczony dostęp	134
4.2.4. Typy zapieczętowane	135
4.2.5. Dziedziczenie i metody domyślne	137
4.3. Object — najwyższa klasa nadrzędna	138
4.3.1. Metoda toString	138
4.3.2. Metoda equals	139
4.3.3. Metoda hashCode	141
4.3.4. Klonowanie obiektów	142
4.4. Wyliczenia	145
4.4.1. Sposoby wyliczania	145
4.4.2. Konstruktory, metody i pola	146
4.4.3. Zawartość elementów	146
4.4.4. Elementy statyczne	146
4.4.5. Wyrażenia switch ze stałymi wyliczeniowymi	147
4.5. Informacje o typie i zasobach w czasie działania programu	148
4.5.1. Klasa Class	148
4.5.2. Wczytywanie zasobów	150
4.5.3. Programy wczytujące klasy	150
4.5.4. Kontekstowy program wczytujący klasy	152
4.5.5. Programy do ładowania usług	153
4.6. Refleksje	154
4.6.1. Wyliczanie elementów klasy	154
4.6.2. Kontrolowanie obiektów	155
4.6.3. Wywoływanie metod	156
4.6.4. Tworzenie obiektów	156
4.6.5. JavaBeans	157
4.6.6. Praca z tablicami	158
4.6.7. Klasa Proxy	159
Ćwiczenia	161

5 Wyjątki, asercje i logi	163
5.1. Obsługa wyjątków	164
5.1.1. Wyrzucanie wyjątków	164
5.1.2. Hierarchia wyjątków	165
5.1.3. Deklarowanie wyjątków kontrolowanych	166
5.1.4. Przechwytywanie wyjątków	167
5.1.5. Wyrażenie try z określeniem zasobów	168
5.1.6. Klauzula finally	169
5.1.7. Ponowne wyrzucanie wyjątków i łączenie ich w łańcuchy	170
5.1.8. Nieprzechwycone wyjątki i ślad stosu wywołań	171
5.1.9. Metody API pozwalające wyrzucać wyjątki	172
5.2. Asercje	173
5.2.1. Użycie asercji	173
5.2.2. Włączanie i wyłączanie asercji	173
5.3. Rejestrowanie danych	174
5.3.1. Czy w Javie warto korzystać z frameworka rejestrowania danych?	174
5.3.2. ABC rejestrowania danych	175
5.3.3. Interfejs Platform Logging API	175
5.3.4. Konfiguracja mechanizmów rejestrowania danych	176
5.3.5. Programy obsługujące rejestrowanie danych	178
5.3.6. Filtry i formaty	180
Ćwiczenia	181
6 Programowanie uogólnione	183
6.1. Klasy uogólnione	184
6.2. Metody uogólnione	185
6.3. Ograniczenia typów	185
6.4. Zmienność typów i symbole wieloznaczne	186
6.4.1. Symbole wieloznaczne w typach podrzędnych	187
6.4.2. Symbole wieloznaczne typów nadrzędnych	188
6.4.3. Symbole wieloznaczne ze zmiennymi typami	189
6.4.4. Nieograniczone symbole wieloznaczne	190
6.4.5. Przechwytywanie symboli wieloznacznych	190
6.5. Uogólnienia w maszynie wirtualnej Javy	190
6.5.1. Wymazywanie typów	191
6.5.2. Wprowadzanie rzutowania	191
6.5.3. Metody pomostowe	192
6.6. Ograniczenia uogólnień	193
6.6.1. Brak typów prostych	193
6.6.2. W czasie działania kodu wszystkie typy są surowe	193
6.6.3. Nie możesz tworzyć instancji zmiennych opisujących typy	194
6.6.4. Nie możesz tworzyć tablic z parametryzowanym typem	196
6.6.5. Zmienne opisujące typ klasy nie są poprawne w kontekście statycznym	197
6.6.6. Metody nie mogą wywoływać konfliktów po wymazywaniu typów	197
6.6.7. Wyjątki i uogólnienia	198
6.7. Refleksje i uogólnienia	199
6.7.1. Klasa Class<T>	199
6.7.2. Informacje o uogólnionych typach w maszynie wirtualnej	199
Ćwiczenia	201

7	Kolekcje	203
7.1.	Mechanizmy do zarządzania kolekcjami	204
7.2.	Iteratory	207
7.3.	Zestawy	208
7.4.	Mapy	209
7.5.	Inne kolekcje	212
7.5.1.	Właściwości	212
7.5.2.	Zestawy bitów	213
7.5.3.	Zestawy wyliczeniowe i mapy	214
7.5.4.	Stosy, kolejki zwykłe i dwukierunkowe oraz kolejki z priorytetami	215
7.5.5.	Klasa WeakHashMap	216
7.6.	Widoki	216
7.6.1.	Małe kolekcje	216
7.6.2.	Zakresy	217
7.6.3.	Niemodyfikowalne widoki	218
	Ćwiczenia	219
8	Strumienie	221
8.1.	Od iteratorów do operacji strumieniowych	222
8.2.	Tworzenie strumienia	223
8.3.	Metody filter, map i flatMap	225
8.4.	Wycinanie podstrumieni i łączenie strumieni	227
8.5.	Inne przekształcenia strumieni	227
8.6.	Proste redukcje	228
8.7.	Typ Optional	229
8.7.1.	Tworzenie alternatywnej wartości	229
8.7.2.	Wykorzystywanie wartości tylko wtedy, gdy jest obecna	230
8.7.3.	Przetwarzanie potokowe wartości typu Optional	230
8.7.4.	Jak nie korzystać z wartości Optional	231
8.7.5.	Tworzenie wartości Optional	231
8.7.6.	Łączenie flatMap z funkcjami wartości Optional	232
8.7.7.	Zamiana Optional w Stream	232
8.8.	Kolekcje wyników	233
8.9.	Tworzenie map	234
8.10.	Grupowanie i partycjonowanie	235
8.11.	Kolektory strumieniowe	236
8.12.	Operacje redukcji	237
8.13.	Strumienie typów prostych	239
8.14.	Strumienie równoległe	240
	Ćwiczenia	243
9	Przetwarzanie danych wejściowych i wyjściowych	245
9.1.	Strumienie wejściowe i wyjściowe, mechanizmy wczytujące i zapisujące	246
9.1.1.	Pozyskiwanie strumieni	246
9.1.2.	Wczytywanie bajtów	247
9.1.3.	Zapisywanie bajtów	247
9.1.4.	Kodowanie znaków	248
9.1.5.	Wczytywanie danych tekstowych	250
9.1.6.	Generowanie danych tekstowych	251
9.1.7.	Wczytywanie i zapisywanie danych binarnych	252
9.1.8.	Pliki o swobodnym dostępie	252
9.1.9.	Pliki mapowane w pamięci	253
9.1.10.	Blokowanie plików	253

9.2. Ścieżki, pliki i katalogi	254
9.2.1. Ścieżki	254
9.2.2. Tworzenie plików i katalogów	255
9.2.3. Kopiowanie, przenoszenie i usuwanie plików	256
9.2.4. Odwiedzanie katalogów	257
9.2.5. System plików ZIP	259
9.3. Połączenia HTTP	260
9.3.1. Klasy URLConnection i HttpURLConnection	260
9.3.2. API klienta HTTP	261
9.4. Wyrażenia regularne	263
9.4.1. Składnia wyrażeń regularnych	263
9.4.2. Testowanie dopasowania	266
9.4.3. Odnajdywanie wszystkich dopasowań	267
9.4.4. Grupy	267
9.4.5. Dzielenie za pomocą znaczników	268
9.4.6. Zastępowanie dopasowań	269
9.4.7. Flagi	269
9.5. Serializacja	270
9.5.1. Interfejs Serializable	270
9.5.2. Chwilowe zmienne instancyjne	271
9.5.3. Metody readObject i writeObject	272
9.5.4. Metody readExternal i writeExternal	273
9.5.5. Metody readResolve i writeReplace	274
9.5.6. Wersjonowanie	275
9.5.7. Deserializacja i bezpieczeństwo	276
Ćwiczenia	278
10 Programowanie współbieżne	281
10.1. Zadania współbieżne	282
10.1.1. Uruchamianie zadań	282
10.1.2. Obiekty Future	284
10.2. Obliczenia asynchroniczne	286
10.2.1. Klasa CompletableFuture	286
10.2.2. Tworzenie obiektów typu CompletableFuture	287
10.2.3. Długie zadania obsługujące interfejs użytkownika	290
10.3. Bezpieczeństwo wątków	291
10.3.1. Widoczność	291
10.3.2. Wyścigi	293
10.3.3. Strategie bezpiecznego korzystania ze współbieżności	294
10.3.4. Klasy niemodyfikowalne	295
10.4. Algorytmy równoległe	296
10.4.1. Strumień równoległy	296
10.4.2. Równoległe operacje na tablicach	297
10.5. Struktury danych bezpieczne dla wątków	297
10.5.1. Klasa ConcurrentHashMap	298
10.5.2. Kolejki blokujące	299
10.5.3. Inne struktury danych bezpieczne dla wątków	300
10.6. Atomowe liczniki i akumulatory	301
10.7. Blokady i warunki	303
10.7.1. Blokady	303
10.7.2. Słowo kluczowe synchronized	304
10.7.3. Oczekiwanie warunkowe	305

10.8. Wątki	307
10.8.1. Uruchamianie wątku	307
10.8.2. Przerywanie wątków	308
10.8.3. Zmienne lokalne w wątku	309
10.8.4. Dodatkowe właściwości wątku	310
10.9. Procesy	311
10.9.1. Tworzenie procesu	311
10.9.2. Uruchamianie procesu	313
10.9.3. Uchwyty procesów	314
Ćwiczenia	315

11 Adnotacje	319
11.1. Używanie adnotacji	320
11.1.1. Elementy adnotacji	320
11.1.2. Wielokrotne i powtarzane adnotacje	321
11.1.3. Adnotacje deklaracji	321
11.1.4. Adnotacje wykorzystania typów	322
11.1.5. Jawne określanie odbiorców	323
11.2. Definiowanie adnotacji	324
11.3. Adnotacje standardowe	326
11.3.1. Adnotacje do kompilacji	327
11.3.2. Metaadnotacje	328
11.4. Przetwarzanie adnotacji w kodzie	329
11.5. Przetwarzanie adnotacji w kodzie źródłowym	331
11.5.1. Przetwarzanie adnotacji	331
11.5.2. API modelu języka	332
11.5.3. Wykorzystanie adnotacji do generowania kodu źródłowego	333
Ćwiczenia	335

12 API daty i czasu	337
12.1. Linia czasu	338
12.2. Daty lokalne	340
12.3. Modyfikatory daty	342
12.4. Czas lokalny	343
12.5. Czas strefowy	344
12.6. Formatowanie i przetwarzanie	346
12.7. Współpraca z przestarzałym kodem	349
Ćwiczenia	350

13 Internacjonalizacja	351
13.1. Lokalizacje	352
13.1.1. Określanie lokalizacji	352
13.1.2. Domyślna lokalizacja	355
13.1.3. Nazwy wyświetlane	355
13.2. Formaty liczb	356
13.3. Waluty	357
13.4. Formatowanie czasu i daty	357
13.5. Porównywanie i normalizacja	359
13.6. Formatowanie komunikatów	360
13.7. Pakiety z zasobami	361
13.7.1. Organizacja pakietów z zasobami	362
13.7.2. Klasy z pakietami	363
13.8. Kodowanie znaków	364
13.9. Preferencje	364
Ćwiczenia	366

14 Kompilacja i skryptowanie	367
14.1. API kompilatora	367
14.1.1. Wywołanie kompilatora	368
14.1.2. Uruchamianie zadania kompilacji	368
14.1.3. Przechwytywanie komunikatów diagnostycznych	369
14.1.4. Wczytywanie plików źródłowych z pamięci	369
14.1.5. Zapisywanie skompilowanego kodu w pamięci	369
14.2. API skryptów	371
14.2.1. Tworzenie silnika skryptowego	371
14.2.2. Uruchamianie skryptów za pomocą metody eval	371
14.2.3. Powiązania	372
14.2.4. Przekierowanie wejścia i wyjścia	372
14.2.5. Wywoływanie funkcji i metod skryptowych	373
14.2.6. Kompilowanie skryptu	374
Ćwiczenia	375
15 System modułów na platformie Java	377
15.1. Koncepcja modułu	378
15.2. Nazywanie modułów	379
15.3. Modularny program „Witaj, świecie!”	380
15.4. Dołączanie modułów	381
15.5. Eksportowanie pakietów	383
15.6. Moduły i dostęp przez refleksje	385
15.7. Modularne pliki JAR	387
15.8. Moduły automatyczne	388
15.9. Moduł nienazwany	389
15.10. Flagi wiersza poleceń dla migracji	390
15.11. Wymagania przechodnie i statyczne	391
15.12. Wybiórcze eksportowanie i otwieranie	392
15.13. Wczytywanie usług	393
15.14. Narzędzia do pracy z modułami	394
Ćwiczenia	396

Najważniejszym powodem korzystania z Javy jest konieczność programowania równoległego. Dzięki algorytmom i bezpiecznym dla współdzielonych danych, dostępnym w bibliotece języka Java, kompletnie zmienił się sposób, w jaki programiści aplikacji radzą sobie z programowaniem współbieżnym. Prezentuję tutaj świeże spojrzenie, pokazując, jak wykorzystywać potężne mechanizmy dostępne w bibliotekach zamiast podatnych na błędy, niskopoziomowych konstrukcji.

Tradycyjne książki na temat języka Java skupiają się na programowaniu interfejsu użytkownika — obecnie jednak niewielu deweloperów tworzy interfejsy użytkownika na komputery stacjonarne. Jeśli zamierzasz używać Javy do tworzenia oprogramowania działającego po stronie serwera lub w systemie Android, będziesz mógł korzystać efektywnie z tej książki bez rozpraszania się kodem GUI aplikacji desktopowych.

I wreszcie: ta książka jest napisana dla programistów aplikacji, a nie dla ekspertów systemowych ani na potrzeby kursu dla studentów. Książka omawia problemy, z którymi muszą zmagać się programiści aplikacji, takie jak logowanie i praca z plikami — ale nie będziesz się uczył, jak ręcznie implementować listę powiązań lub jak pisać serwer internetowy.

Mam nadzieję, że spodoba Ci się także dynamiczne wprowadzenie do nowoczesnej Javy i że uczyni to Twoją pracę z językiem Java produktywną i przyjemną.

Pod adresem <https://fp.helion.pl/przyklady/javap3.zip> znajdziesz archiwum zawierające wszystkie przykłady kodu wykorzystane w tej książce.