
Spis treści

Wstęp	13
1. Rozpoczynanie pracy: kompilacja i uruchamianie	27
1.0. Wprowadzenie	27
1.1. Kompilacja i uruchamianie programów napisanych w Javie — JDK	28
1.2. Kompilacja i uruchamianie: stosowanie GraalVM dla lepszej wydajności	30
1.3. Kompilacja, uruchamianie i testowanie programów przy użyciu IDE	32
1.4. Poznawanie Javy przy wykorzystaniu JShell	36
1.5. Efektywne wykorzystanie zmiennej środowiskowej CLASSPATH	39
1.6. Pobieranie przykładów dołączonych do tej książki i korzystanie z nich	41
1.7. Automatyzacja zależności, kompilacji, testowania i wdrażania przy użyciu programu Apache Maven	49
1.8. Automatyzacja zależności, kompilacji, testowania i wdrażania przy użyciu programu Gradle	53
1.9. Komunikaty o odrzuconych metodach	55
1.10. Zapewnianie poprawności kodu za pomocą testów jednostkowych: JUnit	57
1.11. Zarządzanie kodem z wykorzystaniem ciągłej integracji	60
1.12. Uzyskiwanie czytelnych komunikatów o wyjątkach	65
1.13. Poszukiwanie przykładowych kodów źródłowych	66
1.14. Poszukiwanie gotowych bibliotek Javy	67
2. Interakcja ze środowiskiem	71
2.0. Wprowadzenie	71
2.1. Pobieranie wartości zmiennych środowiskowych	71
2.2. Pobieranie informacji z właściwości systemowych	72
2.3. Stosowanie kodu zależnego od używanej wersji JDK lub systemu operacyjnego	74
2.4. Stosowanie rozszerzających interfejsów programistycznych lub innych API	79
2.5. Stosowanie systemu modułów Javy	80

3. Łańcuchy znaków i przetwarzanie tekstów	85
3.0. Wprowadzenie	85
3.1. Odczytywanie fragmentów łańcucha	88
3.2. Łączenie łańcuchów znaków przy użyciu klasy StringBuilder	93
3.3. Przetwarzanie łańcucha znaków po jednej literze	95
3.4. Wyrównywanie łańcuchów znaków	97
3.5. Konwersja pomiędzy znakami Unicode a łańcuchami znaków	100
3.6. Odwracanie kolejności słów lub znaków w łańcuchu	103
3.7. Rozwijanie i kompresja znaków tabulacji	104
3.8. Kontrola wielkości liter	108
3.9. Wprowadzanie znaków niedrukowalnych	109
3.10. Usuwanie odstępów z końca łańcucha	111
3.11. Tworzenie komunikatów z użyciem zasobów wielojęzycznych	113
3.12. Stosowanie konkretnych ustawień lokalnych	116
3.13. Tworzenie wiązki zasobów	117
3.14. Program — proste narzędzie do formatowania tekstów	118
3.15. Program — fonetyczne porównywanie nazwisk	121
4. Dopasowywanie wzorców przy użyciu wyrażeń regularnych	125
4.0. Wprowadzenie	125
4.1. Składnia wyrażeń regularnych	127
4.2. Wykorzystanie wyrażeń regularnych w języku Java — sprawdzanie występowania wzorca	134
4.3. Odnajdywanie tekstu pasującego do wzorca	138
4.4. Zastępowanie określonego tekstu	140
4.5. Wyświetlanie wszystkich wystąpień wzorca	143
4.6. Wyświetlanie wierszy zawierających fragment pasujący do wzorca	145
4.7. Kontrola wielkości znaków w metodach match() i subst()	147
4.8. Dopasowywanie znaków z akcentami lub znaków złożonych	148
4.9. Odnajdywanie znaków nowego wiersza	149
4.10. Program — analiza dziennika serwera Apache	151
4.11. Program — pełna wersja programu grep	153
5. Liczby	159
5.0. Wprowadzenie	159
5.1. Sprawdzanie, czy łańcuch znaków stanowi poprawną liczbę	162
5.2. Konwertowanie liczb na obiekty i na odwrót	163
5.3. Pobieranie ułamka z liczby całkowitej bez konwertowania go na postać zmiennoprzecinkową	164
5.4. Stosowanie liczb zmiennoprzecinkowych	166
5.5. Formatowanie liczb	170

5.6. Konwersje pomiędzy różnymi systemami liczbowymi — dwójkowym, ósemkowym, dziesiętnym i szesnastkowym	174
5.7. Operacje na grupie liczb całkowitych	175
5.8. Formatowanie z zachowaniem odpowiedniej postaci liczby mnogiej	177
5.9. Generowanie liczb losowych	179
5.10. Mnożenie macierzy	182
5.11. Operacje na liczbach zespolonych	184
5.12. Obsługa liczb o bardzo dużych wartościach	187
5.13. Program TempConverter	189
5.14. Program — generowanie liczbowych palindromów	191
6. Daty i godziny	195
6.0. Wprowadzenie	195
6.1. Określanie bieżącej daty	198
6.2. Wyświetlanie daty i czasu w zadanym formacie	200
6.3. Konwersja liczb określających datę i czas oraz liczbę sekund	202
6.4. Analiza łańcuchów znaków i ich zamiana na daty	203
6.5. Obliczanie różnicy pomiędzy dwiema datami	204
6.6. Dodawanie i odejmowanie dat	206
6.7. Obsługa zdarzeń cyklicznych	207
6.8. Obliczanie dat z uwzględnieniem stref czasowych	209
6.9. Stosowanie starych klas Date i Calendar	210
7. Strukturalizacja danych w języku Java	213
7.0. Wprowadzenie	213
7.1. Strukturalizacja danych przy użyciu tablic	214
7.2. Modyfikacja wielkości tablic	216
7.3. Framework kolekcji	217
7.4. Klasa podobna do tablicy, lecz bardziej dynamiczna	219
7.5. Stosowanie typów ogólnych we własnych klasach	222
7.6. Jak przeglądać zawartość kolekcji? Wyliczenie dostępnych sposobów	226
7.7. Unikanie powtórzeń dzięki zastosowaniu zbioru	229
7.8. Strukturalizacja danych z wykorzystaniem list połączonych	230
7.9. Odzworowywanie z wykorzystaniem klas Hashtable oraz HashMap	235
7.10. Zapisywanie łańcuchów znaków w obiektach Properties i Preferences	237
7.11. Sortowanie kolekcji	242
7.12. Unikanie konieczności sortowania danych	246
7.13. Odnajdywanie obiektu w kolekcji	248
7.14. Zamiana kolekcji na tablicę	251
7.15. Zapewnianie możliwości przeglądania danych przy użyciu iteratora	252
7.16. Używanie stosu obiektów	254

7.17. Struktury wielowymiarowe	257
7.18. Upraszczanie obiektów danych przy wykorzystaniu bibliotek Lombok lub Record	259
7.19. Program — porównanie szybkości działania	262
8. Techniki obiektowe	265
8.0. Wprowadzenie	265
8.1. Metody klasy Object — formatowanie obiektów przy użyciu metody toString() i porównywanie ich metodą equals()	268
8.2. Wykorzystanie klas wewnętrznych	275
8.3. Tworzenie metod zwrotnych z wykorzystaniem interfejsów	278
8.4. Polimorfizm i metody abstrakcyjne	282
8.5. Wartości wyliczeniowe bezpieczne dla typów	283
8.6. Unikanie wyjątków NPE z użyciem klasy Optional	287
8.7. Wymuszanie użycia wzorca Singleton	289
8.8. Zgłaszanie własnych wyjątków	291
8.9. Wstrzykiwanie zależności	293
8.10. Program Plotter	296
9. Techniki programowania funkcyjnego: interfejsy funkcyjne, strumienie i kolekcje równoległe	301
9.0. Wprowadzenie	301
9.1. Stosowanie wyrażeń lambda lub domknięć zamiast klas wewnętrznych	304
9.2. Stosowanie predefiniowanych interfejsów lambda zamiast własnych	307
9.3. Upraszczanie przetwarzania z wykorzystaniem interfejsu Stream	311
9.4. Upraszczanie strumieni przy użyciu kolektorów	312
9.5. Poprawianie przepustowości dzięki wykorzystaniu strumieni i kolekcji równoległych	315
9.6. Używanie istniejącego kodu w sposób funkcyjny dzięki wykorzystaniu odwołań do metod	317
9.7. Wstawianie istniejącego kodu metod	321
10. Wejście i wyjście: odczyt, zapis i sztuczki z katalogami	323
10.0. Wprowadzenie	323
10.1. Klasy InputStream/OutputStream oraz Reader/Writer	325
10.2. Odczytywanie pliku tekstowego	327
10.3. Odczytywanie informacji ze standardowego strumienia wejściowego bądź z konsoli/terminala	329
10.4. Wyświetlanie tekstów przy użyciu klasy Formatter i metody printf	334
10.5. Analiza zawartości pliku przy użyciu klasy StringTokenizer	338
10.6. Analiza danych wejściowych przy użyciu klasy Scanner	342
10.7. Analiza danych wejściowych o strukturze gramatycznej	345
10.8. Kopiowanie pliku	349

10.9. Zmiana skojarzeń standardowych strumieni	350
10.10. Powielanie strumienia podczas realizacji operacji zapisu; zmiana skojarzeń standardowych strumieni	351
10.11. Odczyt i zapis danych zakodowanych w innym zbiorze znaków	354
10.12. Te kłopotliwe znaki końca wiersza	356
10.13. Kod operujący na plikach w sposób zależny od systemu operacyjnego	356
10.14. Odczytywanie i zapisywanie danych binarnych	358
10.15. Odczytywanie i zapisywanie danych w archiwach JAR oraz ZIP	358
10.16. Odnajdywanie plików w sposób niezależny od systemu operacyjnego przy użyciu metod <code>getResource()</code> i <code>getResourceAsStream()</code>	362
10.17. Pobieranie informacji o pliku	364
10.18. Tworzenie nowego pliku lub katalogu	371
10.19. Zmiana nazwy pliku lub jego atrybutów	372
10.20. Usuwanie plików	375
10.21. Tworzenie plików tymczasowych	377
10.22. Tworzenie listy zawartości katalogu	379
10.23. Pobieranie katalogów głównych	380
10.24. Stosowanie usługi <code>WatchService</code> do uzyskiwania informacji o zmianach pliku	382
10.25. Zapisywanie danych użytkownika na dysku	384
10.26. Program <code>Find</code> — poruszanie się po drzewie katalogów	388
11. Nauka o danych i język R	393
11.1. Uczenie maszynowe z użyciem Javy	394
11.2. Stosowanie danych w Apache Spark	395
11.3. Interaktywne stosowanie R	398
11.4. Porównanie i wybór implementacji R	400
11.5. Używanie R w aplikacjach Javy: Renjin	401
11.6. Stosowanie kodu Javy w sesji R	403
11.7. Stosowanie FastR — implementacji R korzystającej z GraalVM	405
11.8. Stosowanie R w aplikacjach internetowych	406
12. Klienci sieciowe	409
12.0. Wprowadzenie	409
12.1. Internetowy klient HTTP/REST	412
12.2. Nawiązywanie połączenia z serwerem z użyciem gniazd	414
12.3. Odnajdywanie adresów sieciowych i zwracanie informacji o nich	415
12.4. Obsługa błędów sieciowych	417
12.5. Odczyt i zapis danych tekstowych	418
12.6. Odczyt i zapis danych binarnych	420
12.7. Datagramy UDP	423

12.8. URI, URL czy może URN?	426
12.9. Program — klient TFTP wykorzystujący protokół UDP	427
12.10. Program — klient pogawędek internetowych używający gniazd	432
12.11. Program — sprawdzanie odnośników HTTP	436
13. Programy Javy działające na serwerze	439
13.0. Wprowadzenie	439
13.1. Tworzenie serwera	440
13.2. Znajdowanie interfejsów sieciowych	443
13.3. Zwracanie odpowiedzi (łańcucha znaków bądź danych binarnych)	444
13.4. Zwracanie informacji o obiektach przez połączenie sieciowe	448
13.5. Obsługa wielu klientów	449
13.6. Serwer obsługujący protokół HTTP	454
13.7. Zabezpieczanie serwera WWW przy użyciu SSL i JSSE	456
13.8. Tworzenie usługi typu RESTful z użyciem JAX-RS	459
13.9. Rejestracja operacji sieciowych	462
13.10. Konfiguracja i stosowanie SLF4J	463
13.11. Rejestracja przez sieć przy użyciu Log4j	466
13.12. Rejestracja przez sieć przy użyciu pakietu java.util.logging	471
14. Przetwarzanie danych w formacie JSON	475
14.0. Wprowadzenie	475
14.1. Bezpośrednie generowanie danych w formacie JSON	477
14.2. Analiza i zapisywanie danych JSON przy użyciu pakietu Jackson	478
14.3. Analiza i zapis danych w formacie JSON przy użyciu pakietu org.json	480
14.4. Analiza i zapis danych w formacie JSON przy użyciu biblioteki JSON-B	481
14.5. Znajdowanie elementów danych w formacie JSON przy użyciu wskaźników	483
15. Pakiety i ich tworzenie	487
15.0. Wprowadzenie	487
15.1. Tworzenie pakietu	488
15.2. Tworzenie dokumentacji klas przy użyciu programu Javadoc	490
15.3. Więcej niż Javadoc — adnotacje i metadane	494
15.4. Tworzenie klasy w taki sposób, by była komponentem JavaBeans	496
15.5. Umieszczanie pakietów w plikach JAR	500
15.6. Wykonywanie programu spakowanego w pliku JAR	501
15.7. Pakowanie komponentów warstwy internetowej do pliku WAR	503
15.8. Tworzenie mniejszego pakietu dystrybucyjnego przy użyciu programu jlink	504
15.9. Stosowanie JPMS do tworzenia modułów	506

16. Stosowanie wątków w Javie	509
16.0. Wprowadzenie	509
16.1. Uruchamianie kodu w innym wątku	511
16.2. Animacja — wyświetlanie poruszających się obrazów	515
16.3. Zatrzymywanie działania wątku	519
16.4. Spotkania i ograniczenia czasowe	522
16.5. Synchronizacja wątków przy użyciu słowa kluczowego synchronized	523
16.6. Upraszczenie synchronizacji przy użyciu blokad	529
16.7. Upraszczenie programu producent-konsument przy użyciu interfejsu Queue	533
16.8. Optymalizacja działania równoległego przy użyciu Fork/Join	535
16.9. Planowanie zadań: operacje w przyszłości, zapis danych w tle w programach edycyjnych	539
17. Introspekcja lub „klasa o nazwie Class”	543
17.0. Wprowadzenie	543
17.1. Pobieranie deskryptora klasy	544
17.2. Określanie oraz stosowanie metod i pól	545
17.3. Uzyskiwanie dostępu do prywatnych pól i metod za pomocą introspekcji	549
17.4. Dynamiczne ładowanie i instalowanie klas	550
17.5. Tworzenie nowej klasy od podstaw przy użyciu obiektu ClassLoader	552
17.6. Tworzenie nowej klasy od podstaw z użyciem kompilatora Javy	554
17.7. Określanie efektywności działania	557
17.8. Wyświetlanie informacji o klasie	561
17.9. Wyświetlanie klas należących do pakietu	563
17.10. Stosowanie i definiowanie adnotacji	565
17.11. Zastosowanie adnotacji do odnajdywania klas pełniących rolę wtyczek	570
17.12. Program CrossRef	573
18. Wykorzystywanie Javy wraz z innymi językami programowania	577
18.0. Wprowadzenie	577
18.1. Uruchamianie zewnętrznego programu	578
18.2. Wykonywanie programu i przechwytywanie jego wyników	582
18.3. Wywoływanie kodu napisanego w innych językach przy użyciu javax.script	586
18.4. Łączenie różnych języków z użyciem GraalVM	588
18.5. Łączenie języków Java i Perl	590
18.6. Dołączanie kodu rodzimego	593
18.7. Wywoływanie kodu Javy z kodu rodzimego	599
Postowie	603
A Java kiedyś i obecnie	605