

Spis treści

Przedmowa	11
Podziękowania	15
Rozdział 1. Biblioteka strumieni Java SE 8	17
1.1. Od iteracji do operacji na strumieniach	18
1.2. Tworzenie strumieni	20
1.3. Metody filter, map oraz flatMap	24
1.4. Pobieranie podstrumieni i łączenie strumieni	25
1.5. Inne przekształcenia strumieni	26
1.6. Proste operacje redukcji	27
1.7. Typ Optional	29
1.7.1. Sposoby posługiwania się wartościami Optional	29
1.7.2. Jak nie należy używać wartości opcjonalnych	31
1.7.3. Tworzenie obiektów typu Optional	31
1.7.4. Łączenie funkcji zwracających wartości opcjonalne przy użyciu flatMap	32
1.8. Gromadzenie wyników	34
1.9. Gromadzenie wyników w mapach	39
1.10. Grupowanie i podział	43
1.11. Kolektory przetwarzające	44
1.12. Operacje redukcji	48
1.13. Strumień danych typów prostych	50
1.14. Strumień równoległy	55
Rozdział 2. Wejście i wyjście	61
2.1. Strumień wejścia-wyjścia	61
2.1.1. Odczyt i zapis bajtów	62
2.1.2. Zoo pełne strumieni	64
2.1.3. Łączenie filtrów strumieni wejścia-wyjścia	68
2.2. Strumień tekstowe	72
2.2.1. Zapisywanie tekstu	72
2.2.2. Wczytywanie tekstu	75
2.2.3. Zapis obiektów w formacie tekstowym	75
2.2.4. Zbiory znaków	78

2.3. Odczyt i zapis danych binarnych	81
2.3.1. Interfejsy DataInput oraz DataOutput	81
2.3.2. Strumienie plików o swobodnym dostępie	84
2.3.3. Archiwa ZIP	88
2.4. Strumienie obiektów i serializacja	91
2.4.1. Zapisywanie i wczytywanie obiektów serializowalnych	91
2.4.2. Format pliku serializacji obiektów	95
2.4.3. Modyfikowanie domyślnego mechanizmu serializacji	102
2.4.4. Serializacja singletonów i wyliczeń	104
2.4.5. Wersje	105
2.4.6. Serializacja w roli klonowania	107
2.5. Zarządzanie plikami	109
2.5.1. Ścieżki dostępu	110
2.5.2. Odczyt i zapis plików	112
2.5.3. Tworzenie plików i katalogów	114
2.5.4. Kopiowanie, przenoszenie i usuwanie plików	115
2.5.5. Informacje o plikach	117
2.5.6. Przeglądanie zawartości katalogu	118
2.5.7. Stosowanie strumieni katalogów	120
2.5.8. Systemy plików ZIP	123
2.6. Mapowanie plików w pamięci	124
2.6.1. Wydajność plików mapowanych w pamięci	124
2.6.2. Struktura bufora danych	131
2.6.3. Blokowanie plików	133
2.7. Wyrażenia regularne	135

Rozdział 3. Język XML

149

3.1. Wprowadzenie do języka XML	150
3.1.1. Struktura dokumentu XML	152
3.2. Parsowanie dokumentów XML	155
3.3. Kontrola poprawności dokumentów XML	166
3.3.1. Definicje typów dokumentów	167
3.3.2. XML Schema	174
3.3.3. Praktyczny przykład	176
3.4. Wyszukiwanie informacji i XPath	189
3.5. Przestrzenie nazw	195
3.6. Parsery strumieniowe	198
3.6.1. Wykorzystanie parsera SAX	198
3.6.2. Wykorzystanie parsera StAX	203
3.7. Tworzenie dokumentów XML	207
3.7.1. Dokumenty bez przestrzeni nazw	207
3.7.2. Dokumenty z przestrzenią nazw	208
3.7.3. Zapisywanie dokumentu	209
3.7.4. Przykład: tworzenie pliku SVG	209
3.7.5. Tworzenie dokumentu XML za pomocą parsera StAX	213
3.8. Przekształcenia XSL	220

Rozdział 4. Programowanie aplikacji sieciowych

231

4.1. Połączenia z serwerem	231
4.1.1. Stosowanie programu telnet	231
4.1.2. Nawiązywanie połączenia z serwerem z wykorzystaniem Javy	234
4.1.3. Limity czasu gniazd	235
4.1.4. Adresy internetowe	237

4.2. Implementacja serwerów	238
4.2.1. Gniazda serwera	239
4.2.2. Obsługa wielu klientów	241
4.2.3. Połączenia częściowo zamknięte	244
4.3. Przerwanie działania gniazd sieciowych	246
4.4. Połączenia wykorzystujące URL	252
4.4.1. URL i URI	252
4.4.2. Zastosowanie klasy URLConnection do pobierania informacji	254
4.4.3. Wysyłanie danych do formularzy	262
4.5. Wysyłanie poczty elektronicznej	270
Rozdział 5. Programowanie baz danych: JDBC	275
5.1. Architektura JDBC	276
5.1.1. Typy sterowników JDBC	276
5.1.2. Typowe zastosowania JDBC	278
5.2. Język SQL	278
5.3. Instalacja JDBC	284
5.3.1. Adresy URL baz danych	284
5.3.2. Pliki JAR zawierające sterownik	285
5.3.3. Uruchamianie bazy danych	285
5.3.4. Rejestracja klasy sterownika	286
5.3.5. Nawiązywanie połączenia z bazą danych	287
5.4. Stosowanie poleceń SQL	289
5.4.1. Wykonywanie poleceń SQL	290
5.4.2. Zarządzanie połączeniami, poleceniami i zbiorami wyników	293
5.4.3. Analiza wyjątków SQL	294
5.4.4. Wypełnianie bazy danych	296
5.5. Wykonywanie zapytań	300
5.5.1. Polecenia przygotowane	300
5.5.2. Odczyt i zapis dużych obiektów	306
5.5.3. Sekwencje sterujące	308
5.5.4. Zapytania o wielu zbiorach wyników	309
5.5.5. Pobieranie wartości kluczy wygenerowanych automatycznie	310
5.6. Przewijalne i aktualizowalne zbiory wyników zapytań	311
5.6.1. Przewijalne zbiory wyników	311
5.6.2. Aktualizowalne zbiory rekordów	313
5.7. Zbiory rekordów	318
5.7.1. Tworzenie zbiorów rekordów	318
5.7.2. Buforowane zbiory rekordów	319
5.8. Metadane	322
5.9. Transakcje	331
5.9.1. Programowanie transakcji w JDBC	332
5.9.2. Punkty kontrolne	332
5.9.3. Aktualizacje wsadowe	333
5.10. Zaawansowane typy języka SQL	335
5.11. Zaawansowane zarządzanie połączeniami	336
Rozdział 6. API dat i czasu	339
6.1. Oś czasu	340
6.2. Daty lokalne	343
6.3. Modyfikatory dat	346
6.4. Czas lokalny	347

6.5. Czas strefowy	348
6.6. Formatowanie i parsowanie	352
6.7. Współdziałanie ze starym kodem	356

Rozdział 7. Internacjonalizacja

359

7.1. Lokalizatory	360
7.2. Formaty liczb	365
7.3. Waluty	370
7.4. Data i czas	372
7.5. Porządek alfabetyczny i normalizacja	378
7.6. Formatowanie komunikatów	385
7.6.1. Formatowanie liczb i dat	385
7.6.2. Formatowanie z wariantami	387
7.7. Wczytywanie i wyświetlanie tekstów	389
7.7.1. Pliki tekstowe	389
7.7.2. Znaki końca wiersza	389
7.7.3. Konsola	390
7.7.4. Pliki dzienników	391
7.7.5. BOM — znacznik kolejności bajtów UTF-8	391
7.7.6. Kodowanie plików źródłowych	392
7.8. Komplet zasobów	392
7.8.1. Wyszukiwanie kompletów zasobów	393
7.8.2. Pliki właściwości	394
7.8.3. Klasy kompletów zasobów	395
7.9. Kompletny przykład	397

Rozdział 8. Skrypty, kompilacja i adnotacje

413

8.1. Skrypty na platformie Java	413
8.1.1. Wybór silnika skryptów	414
8.1.2. Wykonywanie skryptów i wiązania zmiennych	415
8.1.3. Przekierowanie wejścia i wyjścia	417
8.1.4. Wywoływanie funkcji i metod skryptów	418
8.1.5. Kompilacja skryptu	420
8.1.6. Przykład: skrypty i graficzny interfejs użytkownika	420
8.2. Interfejs kompilatora	425
8.2.1. Kompilacja w najprostszy sposób	426
8.2.2. Stosowanie zadań kompilacji	426
8.2.3. Przykład: dynamiczne tworzenie kodu w języku Java	432
8.3. Stosowanie adnotacji	436
8.3.1. Wprowadzenie do stosowania adnotacji	437
8.3.2. Przykład: adnotacje obsługi zdarzeń	438
8.4. Składnia adnotacji	443
8.4.1. Interfejsy adnotacji	443
8.4.2. Adnotacje	445
8.4.3. Adnotacje deklaracji	446
8.4.4. Adnotacje zastosowań typów	447
8.4.5. Adnotacje i this	449
8.5. Adnotacje standardowe	450
8.5.1. Adnotacje kompilacji	451
8.5.2. Adnotacje zarządzania zasobami	451
8.5.3. Metaadnotacje	452

8.6. Przetwarzanie adnotacji w kodzie źródłowym	455
8.6.1. Procesory adnotacji	455
8.6.2. Interfejs programowy modelu języka	455
8.6.3. Stosowanie adnotacji do generacji kodu źródłowego	456
8.7. Inżynieria kodu bajtowego	459
8.7.1. Modyfikowanie plików klasowych	459
8.7.2. Modyfikacja kodu bajtowego podczas ładowania	464

Rozdział 9. Bezpieczeństwo

9.1. Ładowanie klas	468
9.1.1. Proces wczytywania plików klas	468
9.1.2. Hierarchia klas ładowania	469
9.1.3. Zastosowanie procedur ładujących w roli przestrzeni nazw	471
9.1.4. Implementacja własnej procedury ładującej	473
9.1.5. Weryfikacja kodu maszyny wirtualnej	478
9.2. Menedżery bezpieczeństwa i pozwolenia	483
9.2.1. Sprawdzanie uprawnień	483
9.2.2. Bezpieczeństwo na platformie Java	484
9.2.3. Pliki polityki bezpieczeństwa	487
9.2.4. Tworzenie własnych klas pozwoleń	495
9.2.5. Implementacja klasy pozwoleń	496
9.3. Uwierzytelnianie użytkowników	502
9.3.1. Framework JAAS	502
9.3.2. Moduły JAAS	507
9.4. Podpis cyfrowy	516
9.4.1. Skróty wiadomości	517
9.4.2. Podpisywanie wiadomości	520
9.4.3. Weryfikacja podpisu	522
9.4.4. Problem uwierzytelniania	524
9.4.5. Podpisywanie certyfikatów	526
9.4.6. Żądania certyfikatu	527
9.4.7. Podpisywanie kodu	528
9.5. Szyfrowanie	534
9.5.1. Szyfrowanie symetryczne	534
9.5.2. Generowanie klucza	536
9.5.3. Strumienie szyfrujące	541
9.5.4. Szyfrowanie kluczem publicznym	542

Rozdział 10. Zaawansowane możliwości pakietu Swing

10.1. Listy	547
10.1.1. Komponent JList	548
10.1.2. Modele list	553
10.1.3. Wstawianie i usuwanie	558
10.1.4. Odrysowywanie zawartości listy	559
10.2. Tabele	563
10.2.1. Najprostsze tabele	563
10.2.2. Modele tabel	568
10.2.3. Wiersze i kolumny	571
10.2.4. Rysowanie i edycja komórek	586

10.3. Drzewa	598
10.3.1. Najprostsze drzewa	599
10.3.2. Modyfikacje drzew i ścieżek drzew	606
10.3.3. Przeglądanie węzłów	613
10.3.4. Rysowanie węzłów	615
10.3.5. Nasłuchiwanie zdarzeń w drzewach	618
10.3.6. Własne modele drzew	625
10.4. Komponenty tekstowe	633
10.4.1. Śledzenie zmian zawartości komponentów tekstowych	634
10.4.2. Sformatowane pola wejściowe	637
10.4.3. Komponent JSpinner	653
10.4.4. Prezentacja HTML za pomocą JEditorPane	661
10.5. Wskaźniki postępu	667
10.5.1. Paski postępu	667
10.5.2. Monitory postępu	670
10.5.3. Monitorowanie postępu strumieni wejścia	673
10.6. Organizatory komponentów i dekoratory	678
10.6.1. Panele dzielone	678
10.6.2. Panele z kartami	681
10.6.3. Panele pulpitu i ramki wewnętrzne	687
10.6.4. Warstwy	703
Rozdział 11. Zaawansowane możliwości biblioteki AWT	709
11.1. Potokowe tworzenie grafiki	710
11.2. Figury	712
11.2.1. Hierarchia klas Shape	713
11.2.2. Wykorzystanie klas obiektów graficznych	714
11.3. Pola	727
11.4. Ślad pędzla	728
11.5. Wypełnienia	735
11.6. Przekształcenia układu współrzędnych	737
11.7. Przycinanie	743
11.8. Przezroczystość i składanie obrazów	745
11.9. Wskazówki operacji graficznych	753
11.10. Czytanie i zapisywanie plików graficznych	758
11.10.1. Wykorzystanie obiektów zapisu i odczytu plików graficznych	759
11.10.2. Odczyt i zapis plików zawierających sekwencje obrazów	763
11.11. Operacje na obrazach	768
11.11.1. Dostęp do danych obrazu	769
11.11.2. Filtrowanie obrazów	775
11.12. Drukowanie	783
11.12.1. Drukowanie grafiki	784
11.12.2. Drukowanie wielu stron	792
11.12.3. Podgląd wydruku	794
11.12.4. Usługi drukowania	802
11.12.5. Usługi drukowania za pośrednictwem strumieni	806
11.12.6. Atrybuty drukowania	807
11.13. Schowek	813
11.13.1. Klasy i interfejsy umożliwiające przekazywanie danych	814
11.13.2. Przekazywanie tekstu	815
11.13.3. Interfejs Transferable i formaty danych	818

11.13.4. Przekazywanie obrazów za pomocą schowka	820
11.13.5. Wykorzystanie schowka systemowego do przekazywania obiektów Java	824
11.13.6. Zastosowanie lokalnego schowka do przekazywania referencji obiektów	827
11.14. Mechanizm „przeciągnij i upuść”	828
11.14.1. Przekazywanie danych pomiędzy komponentami Swing	829
11.14.2. Źródła przeciąganych danych	833
11.14.3. Cele upuszczanych danych	835
11.15. Integracja z macierzystą platformą	844
11.15.1. Ekran powitalny	844
11.15.2. Uruchamianie macierzystych aplikacji pulpitu	849
11.15.3. Zasobnik systemowy	853

Rozdział 12. Metody macierzyste 859

12.1. Wywołania funkcji języka C z programów w języku Java	860
12.2. Numeryczne parametry metod i wartości zwracane	866
12.3. Łańcuchy znaków jako parametry	868
12.4. Dostęp do składowych obiektu	873
12.4.1. Dostęp do pól instancji	874
12.4.2. Dostęp do pól statycznych	877
12.5. Sygnatury	878
12.6. Wywoływanie metod języka Java	880
12.6.1. Wywoływanie metod obiektów	880
12.6.2. Wywoływanie metod statycznych	883
12.6.3. Konstruktory	884
12.6.4. Alternatywne sposoby wywoływania metod	885
12.7. Tablice	886
12.8. Obsługa błędów	890
12.9. Interfejs programowy wywołań języka Java	895
12.10. Kompletny przykład: dostęp do rejestru systemu Windows	900
12.10.1. Rejestr systemu Windows	900
12.10.2. Interfejs dostępu do rejestru na platformie Java	902
12.10.3. Implementacja dostępu do rejestru za pomocą metod macierzystych	902

Skorowidz 917

Rozdziały niniejszej książki są w większości przypadków zupełnie niezależne od siebie i w praktyce można rozpocząć lekturę od najbardziej interesującego materiału, a pozostałe wadzyńy przeczytać w dowolnej kolejności.

Rozdział 1. opisuje szczegółowo bibliotekę strumieni języka Java 8, która zapewnia programi-
stom możliwość przetwarzania danych w nowoczesny sposób — poprzez strumienie typu
stream. Chcemy zrobić, bez podawania wszystkich szczegółów metody użycia, następującego
spójnie. Dzięki temu biblioteka strumieni może się koncentrować na dobrej i optymalnej stru-
mie przetwarzania, co jest szczególnie przydatne podczas optymalizacji obliczeń współbieżnych.