

Przedmowa	29
Wstęp	31
Zanim zaczniesz	49
1. Wprowadzenie do komputerów, internetu i Javy	55
1.1. Wprowadzenie	56
1.2. Sprzęt i oprogramowanie	58
1.2.1. Prawo Moore'a	59
1.2.2. Sposób organizacji komputera	59
1.3. Hierarchia danych	61
1.4. Języki maszynowe, języki assemblerowe i języki wysokiego poziomu	64
1.5. Wprowadzenie do technologii obiektowej	65
1.5.1. Samochód jako obiekt	65
1.5.2. Metody i klasy	66
1.5.3. Tworzenie egzemplarzy	66
1.5.4. Wielokrotne użycie	66
1.5.5. Komunikaty i wywoływanie metod	67
1.5.6. Atrybuty i zmienne instancji	67
1.5.7. Enkapsulacja i ukrywanie informacji	67
1.5.8. Dziedziczenie	67
1.5.9. Interfejsy	67
1.5.10. Obiektowa analiza i projektowanie	68
1.5.11. Język UML	68
1.6. Systemy operacyjne	68
1.6.1. Windows — własnościowy system operacyjny	69
1.6.2. Linux — system operacyjny o otwartym kodzie źródłowym	69
1.6.3. Systemy macOS i iOS firmy Apple dla urządzeń iPhone, iPad i iPod Touch	70
1.6.4. System Android firmy Google	70
1.7. Języki programowania	71
1.8. Java	74
1.9. Typowe środowisko programistyczne wykorzystujące Javę	74
1.10. Testowanie przykładowej aplikacji napisanej w Javie	78
1.11. Internet i sieć WWW	82
1.11.1. Internet, czyli sieć sieci	83
1.11.2. Sieć WWW — przyjazny internet	83
1.11.3. Usługi sieciowe i mashupy	83
1.11.4. Internet rzeczy	84
1.12. Technologie związane z oprogramowaniem	85
1.13. Otrzymywanie odpowiedzi na pytania	87
2. Wprowadzenie do aplikacji javy, wejścia - wyjścia i operatorów	91
2.1. Wprowadzenie	92
2.2. Twój pierwszy program — wyświetlenie wiersza tekstu	92

2.2.1. Kompilacja aplikacji	97
2.2.2. Wykonywanie aplikacji	98
2.3. Modyfikacja pierwszego programu	99
2.4. Wyświetlanie tekstu metodą printf	101
2.5. Inna aplikacja — dodawanie liczb całkowitych	102
2.5.1. Deklaracja import	102
2.5.2. Deklaracja i utworzenie obiektu Scanner w celu pobrania danych z klawiatury	103
2.5.3. Prośba o wprowadzenie danych	104
2.5.4. Deklaracja zmiennej dla liczby całkowitej i pobranie wartości z klawiatury	104
2.5.5. Pobranie drugiej liczby całkowitej	105
2.5.6. Użycie zmiennych w obliczeniach	105
2.5.7. Wyświetlenie wyniku obliczeń	105
2.5.8. Dokumentacja API Javy	106
2.5.9. Deklaracja i inicjalizacja zmiennej jako osobne instrukcje	106
2.6. Zagadnienia dotyczące pamięci	106
2.7. Operacje arytmetyczne	107
2.8. Podejmowanie decyzji — operatory równości i relacji	111
2.9. Podsumowanie	115
3. Wprowadzenie do klas, obiektów, metod i tekstów	127
3.1. Wprowadzenie	128
3.2. Zmienne instancji, metody ustawiające i metody pobierające	129
3.2.1. Klasa Account ze zmienną instancji oraz metodą ustawiającą i metodą pobierającą	129
3.2.2. Klasa AccountTest, która tworzy i stosuje obiekt klasy Account	133
3.2.3. Kompilacja i wykonanie aplikacji z wieloma klasami	136
3.2.4. Diagram klas UML dla klasy Account	136
3.2.5. Dodatkowe uwagi na temat klasy AccountTest	137
3.2.6. Inżynieria oprogramowania z prywatnymi zmiennymi instancji i publicznymi metodami dostępowymi	138
3.3. Klasa Account — inicjalizacja obiektów za pomocą konstruktorów	139
3.3.1. Deklaracja konstruktora klasy Account dla własnej inicjalizacji obiektów	140
3.3.2. Klasa AccountTest — inicjalizacja obiektów Account w momencie ich tworzenia	141
3.4. Klasa Account ze stanem konta — liczby zmiennoprzecinkowe	142
3.4.1. Klasa Account ze zmienną instancji balance typu double	143
3.4.2. Klasa AccountTest używająca klasy Account	145
3.5. Typy podstawowe i typy referencyjne	148
3.6. (Opcjonalnie) Studium przypadku GUI i grafiki — prosty interfejs graficzny	149
3.6.1. Czym jest interfejs graficzny?	150
3.6.2. FXML i narzędzie Scene Builder	150
3.6.3. Aplikacja powitalna — wyświetlenie tekstu i obrazka	150
3.6.4. Uruchomienie narzędzia Scene Builder i utworzenie pliku Welcome.fxml	151
3.6.5. Dodanie obrazu do folderu zawierającego plik Welcome.fxml	151
3.6.6. Tworzenie kontenera układu VBox	151
3.6.7. Konfiguracja kontenera układu VBox	151
3.6.8. Dodanie i konfiguracja etykiety	153
3.6.9. Dodanie i konfiguracja Image View	154

3.6.10. Podgląd wynikowego interfejsu aplikacji	155
3.7. Podsumowanie	156
4. Struktury sterujące, część 1.; przypisanie i operatory ++ i -	165
4.1. Wprowadzenie	166
4.2. Algorytmy	166
4.3. Pseudokod	167
4.4. Struktury sterujące	167
4.4.1. Struktury sekwencyjne w Javie	168
4.4.2. Instrukcje wyboru w Javie	169
4.4.3. Instrukcje iteracji w Javie	169
4.4.4. Podsumowanie instrukcji sterujących Javy	170
4.5. Instrukcja pojedynczego wyboru — if	170
4.6. Instrukcja podwójnego wyboru — if.. .else	171
4.6.1. Zagnieżdżone instrukcje if.. .else	172
4.6.2. Problem wiszącego else	173
4.6.3. Bloki	173
4.6.4. Operator warunku (?:)	174
4.7. Klasa Student — zagnieżdżone instrukcje if...else	175
4.8. Instrukcja iteracji while	177
4.9. Tworzenie algorytmów — iteracja sterowana licznikiem	179
4.10. Tworzenie algorytmów — iteracja sterowana znacznikiem	183
4.11. Tworzenie algorytmów — zagnieżdżone struktury sterujące	191
4.12. Złożone operatory przypisania	195
4.13. Operatory inkrementacji i dekrementacji	196
4.14. Typy podstawowe	199
4.15. Studium przypadku GUI i grafiki — obsługa zdarzeń i rysowanie linii	200
4.15.1. Test ukończonej aplikacji	200
4.15.2. Budowanie GUI aplikacji	201
4.15.3. Przygotowanie do interakcji z GUI w sposób programowy	205
4.15.4. Klasa DrawLinesController	207
4.15.5. Klasa DrawLines — główna klasa aplikacji	209
4.16. Podsumowanie	212
5. Struktury sterujące, część 2.; operatory logiczne	229
5.1. Wprowadzenie	230
5.2. Podstawy iteracji sterowanej licznikiem	230
5.3. Instrukcja iteracji for	231
5.4. Przykłady użycia instrukcji for	236
5.4.1. Aplikacja — suma liczb parzystych od 2 do 20	236
5.4.2. Aplikacja — kalkulator procentu składanego	237
5.5. Instrukcja iteracji do...while	240
5.6. Instrukcja switch dotycząca wielu wyborów	242
5.7. Studium przypadku klasy AutoPolicy — tekst jako wartość w instrukcji switch	247
5.8. Instrukcje break i continue	251
5.8.1. Instrukcja break	251
5.8.2. Instrukcja continue	251
5.9. Operatory logiczne	252

5.9.1. Operator warunkowy AND (&&)	253
5.9.2. Operator warunkowy OR ()	253
5.9.3. Skrócone obliczenie wartości złożonych warunków	254
5.9.4. Operatory logiczne AND (&) i OR ()	255
5.9.5. Operator logiczny XOR (^)	255
5.9.6. Operator negacji logicznej (!)	256
5.9.7. Przykład użycia operatorów logicznych	256
5.10. Podsumowanie programowania strukturalnego	258
5.11. Studium przypadku GUI i grafiki — rysowanie prostokątów i owali	263
5.12. Podsumowanie	266
6. Metody — dokładniejsze spojrzenie	279
6.1. Wprowadzenie	280
6.2. Jednostki programu w Javie	280
6.3. Metody statyczne, pola statyczne i klasa Math	282
6.4. Metody z wieloma parametrami	284
6.5. Uwagi na temat deklarowania i używania metod	288
6.6. Stos wywołań metod i rekordy aktywacyjne	289
6.6.1. Stos wywołań metod	289
6.6.2. Ramki stosu	290
6.6.3. Zmienne lokalne i ramki stosu	290
6.6.4. Przepełnienie stosu	290
6.7. Promocja argumentu i rzutowanie	291
6.8. Pakiety API Javy	292
6.9. Studium przypadku — bezpieczne generowanie liczb losowych	294
6.10. Studium przypadku — gra losowa i wprowadzenie do typów enum	299
6.11. Zasięg deklaracji	303
6.12. Przeciążanie metod	306
6.12.1. Deklarowanie przeciążonych metod	306
6.12.2. Rozróżnianie przeciążonych metod	307
6.12.3. Typy zwracane przez metody przeciążone	308
6.13. (Opcjonalnie) Studium przypadku GUI i grafiki — kolory i wypełnione kształty	308
6.14. Podsumowanie	312
7. Tablice i obiekty Array List	327
7.1. Wprowadzenie	328
7.2. Tablice	329
7.3. Deklaracja i tworzenie tablic	330
7.4. Przykłady użycia tablic	332
7.4.1. Tworzenie i inicjalizacja tablicy	332
7.4.2. Użycie inicjalizatora tablicy	333
7.4.3. Obliczenie wartości elementów tablicy	334
7.4.4. Sumowanie elementów tablicy	335
7.4.5. Graficzne przedstawienie danych z tablicy za pomocą wykresu słupkowego	336
7.4.6. Użycie elementów tablicy jako liczników	337
7.4.7. Użycie tablic do analizy wyników	338
7.5. Obsługa wyjątków — przetworzenie niepoprawnej odpowiedzi	340
7.5.1. Instrukcja try	341
7.5.2. Wykonywanie bloku catch	341

7.5.3. Metoda toString parametru wyjątku	341
7.6. Studium przypadku — tasowanie i rozdawanie kart	342
7.7. Rozszerzona instrukcja for	346
7.8. Przekazywanie tablic do metod	348
7.9. Przekazywanie przez wartość kontra przekazywanie przez referencję	350
7.10. Studium przypadku — klasa GradeBook wykorzystująca tablicę z ocenami	351
7.11. Tablice wielowymiarowe	356
7.11.1. Tablice tablic jednowymiarowych	357
7.11.2. Dwuwymiarowe tablice z wierszami o różnych długościach	357
7.11.3. Tworzenie tablic dwuwymiarowych za pomocą wyrażenia tworzenia tablic	357
7.11.4. Przykład tablicy dwuwymiarowej — wyświetlanie wartości elementów	358
7.11.5. Typowe operacje na tablicach wielowymiarowych wykonywane pętlami for	359
7.12. Studium przypadku — klasa GradeBook używająca tablicy dwuwymiarowej	360
7.13. Lista argumentów o zmiennej długości	365
7.14. Użycie argumentów wiersza poleceń	367
7.15. Klasa Arrays	369
7.16. Wprowadzenie do kolekcji i klasy ArrayList	372
7.17. (Opcjonalnie) Studium przypadku GUI i grafiki — rysowanie łuków	375
7.18. Podsumowanie	378
8. Obiekty i klasy — dokładniejsze spojrzenie	401
8.1. Wprowadzenie	402
8.2. Studium przypadku — klasa Time	402
8.3. Sterowanie dostępem do składowych	407
8.4. Odnoszenie się do składowych aktualnego obiektu referencją this	408
8.5. Studium przypadku klasy Time — przeciążanie konstruktorów	410
8.6. Konstruktory domyślne i bezargumentowe	416
8.7. Uwagi dotyczące metod dostępowych	416
8.8. Kompozycja	418
8.9. Typ enum	420
8.10. Mechanizm odśmiecania pamięci	423
8.11. Składowe statyczne klasy	424
8.12. Import statyczny	428
8.13. Zmienne instancji typu final	429
8.14. Dostęp na poziomie pakietu	430
8.15. Użycie klasy BigDecimal do precyzyjnych obliczeń pieniężnych	431
8.16. (Opcjonalnie) Studium przypadku GUI i grafiki — użycie obiektów z grafiką	434
8.17. Podsumowanie	438
9. Programowanie obiektowe — dziedziczenie	447
9.1. Wprowadzenie	448
9.2. Klasy nadrzędne i podklasy	449
9.3. Składowe chronione	451
9.4. Związek między klasami nadrzędnymi i podklasami	452
9.4.1. Tworzenie i wykorzystywanie klasy CommissionEmployee	453
9.4.2. Tworzenie i użycie klasy BasePlusCommissionEmployee	458

9.4.3. Tworzenie hierarchii dziedziczenia <code>CommissionEmployee</code> przez <code>BasePlusCommissionEmployee</code>	462
9.4.4. Tworzenie hierarchii dziedziczenia <code>CommissionEmployee</code> przez <code>BasePlusCommissionEmployee</code> używającej zmiennych chronionych	465
9.4.5. Tworzenie hierarchii dziedziczenia <code>CommissionEmployee</code> przez <code>BasePlusCommissionEmployee</code> używającej zmiennych prywatnych	468
9.5. Konstruktory w podklasach	473
9.6. Klasa <code>Object</code>	473
9.7. Projektowanie klas — kompozycja kontra dziedziczenie	474
9.8. Podsumowanie	476
10. Programowanie obiektowe — polimorfizm i interfejsy	483
10.1. Wprowadzenie	484
10.2. Przykłady polimorfizmu	486
10.3. Przykład zachowania polimorficznego	487
10.4. Metody i klasy abstrakcyjne	490
10.5. Studium przypadku — system płac wykorzystujący polimorfizm	492
10.5.1. Abstrakcyjna klasa nadrzędna <code>Employee</code>	495
10.5.2. Konkretna podklasa <code>SalariedEmployee</code>	496
10.5.3. Konkretna podklasa <code>HourlyEmployee</code>	498
10.5.4. Konkretna podklasa <code>CommissionEmployee</code>	500
10.5.5. Pośrednia konkretna podklasa <code>BasePlusCommissionEmployee</code>	501
10.5.6. Przetwarzanie polimorficzne, operator <code>instanceof</code> i rzutowanie w dół	503
10.6. Umożliwienie przypisywania między zmiennymi klas nadrzędnych i podklas	507
10.7. Metody i klasy finalne	508
10.8. Dokładniejszy opis problemów z wywoływaniem metod z konstruktorów	509
10.9. Tworzenie i stosowanie interfejsów	509
10.9.1. Tworzenie hierarchii <code>Payable</code>	512
10.9.2. Interfejs <code>Payable</code>	512
10.9.3. Klasa <code>Invoice</code>	513
10.9.4. Modyfikacja klasy <code>Employee</code> w celu implementacji interfejsu <code>Payable</code>	515
10.9.5. Użycie interfejsu <code>Payable</code> do polimorficznego przetwarzania klas <code>Invoice</code> i <code>Employee</code>	517
10.9.6. Wspólne interfejsy API Javy	518
10.10. Rozszerzenia interfejsów w Javie SE 8	519
10.10.1. Domyślne metody interfejsu	519
10.10.2. Statyczne metody interfejsu	520
10.10.3. Interfejsy funkcyjne	520
10.11. Prywatne metody interfejsów w Javie SE 9	521
10.12. Konstruktory prywatne	521
10.13. Programowanie do interfejsu, a nie do implementacji	522
10.13.1. Dziedziczenie implementacji działa najlepiej dla niewielkiej liczby ściśle powiązanych klas	522
10.13.2. Dziedziczenie interfejsów jest najlepsze dla elastyczności	522
10.13.3. Modyfikacja hierarchii <code>Employee</code>	523
10.14. Studium przypadku GUI i grafiki — rysowanie z użyciem polimorfizmu	524

10.15. Podsumowanie	526
11. Obsługa wyjątków — dokładniejsze spojrzenie	535
11.1. Wprowadzenie	536
11.2. Przykład — dzielenie przez 0 bez obsługi wyjątków	537
11.3. Przykład — obsługa wyjątków ArithmeticException i InputMismatchException	540
11.4. Kiedy używać obsługi wyjątków	545
11.5. Hierarchia wyjątków Javy	546
11.6. Blok finally	549
11.7. Rozwijanie stosu i uzyskiwanie informacji z wyjątku	554
11.8. Wyjątki łańcuchowe	557
11.9. Deklarowanie nowych rodzajów wyjątków	559
11.10. Warunki wstępne i warunki końcowe	560
11.11. Asercje	560
11.12. Instrukcja try z zasobami — automatyczne zwalnianie zasobów	562
11.13. Podsumowanie	563
12. Graficzny interfejs użytkownika JavaFX, część 1.	569
12.1. Wprowadzenie	570
12.2. Narzędzie Scene Builder dla JavaFX	571
12.3. Struktura okna aplikacji JavaFX	572
12.4. Aplikacja powitalna — wyświetlenie tekstu i obrazka	573
12.4.1. Uruchomienie narzędzia Scene Builder i utworzenie pliku Welcome.fxml	574
12.4.2. Dodanie obrazu do folderu zawierającego plik Welcome.fxml	574
12.4.3. Tworzenie kontenera układu VBox	575
12.4.4. Konfiguracja kontenera układu VBox	575
12.4.5. Dodanie i konfiguracja etykiety	575
12.4.6. Dodanie i konfiguracja Image View	576
12.4.7. Podgląd wynikowego interfejsu aplikacji	577
12.5. Aplikacja do wyliczania napiwków — wprowadzenie do obsługi zdarzeń	577
12.5.1. Testowanie kalkulatora napiwków	579
12.5.2. Przedstawienie technologii	580
12.5.3. Budowanie GUI aplikacji	583
12.5.4. Klasa TipCalculator	589
12.5.5. Klasa TipCalculatorController	591
12.6. Funkcjonalności opisywane w pozostałych rozdziałach dotyczących JavaFX	596
12.7. Podsumowanie	597
13. Graficzny interfejs użytkownika JavaFX, część 2.	605
13.1. Wprowadzenie	606
13.2. Układanie węzłów w grafie	606
13.3. Aplikacja Painter — przyciski opcji, zdarzenia myszy i kształty	608
13.3.1. Omówienie technologii	608
13.3.2. Utworzenie pliku Painter.fxml	611
13.3.3. Tworzenie GUI	611
13.3.4. Podklasa Painter klasy Application	614
13.3.5. Klasa PainterController	614

13.4. Aplikacja do wybierania kolorów	618
13.4.1. Omówienie technologii	618
13.4.2. Tworzenie GUI	620
13.4.3. Podklasa ColorChooser klasy Application	622
13.4.4. Klasa ColorChooserController	622
13.5. Aplikacja do przeglądania okładek	625
13.5.1. Omówienie technologii	625
13.5.2. Dodanie obrazków do folderu aplikacji	626
13.5.3. Budowanie interfejsu graficznego aplikacji	626
13.5.4. Podklasa CoverViewer klasy Application	627
13.5.5. Klasa CoverViewerController	628
13.6. Aplikacja do przeglądania okładek — dostosowanie komórek List View	630
13.6.1. Omówienie technologii	630
13.6.2. Kopiowanie przeglądarki okładek	631
13.6.3. Klasa własnej fabryki komórek — ImageTextCell	631
13.6.4. Klasa CoverViewerController	633
13.7. Dodatkowe możliwości JavaFX	634
13.8. JavaFX 9 — aktualizacja JavaFX w Javie SE 9	636
13.9. Podsumowanie	638
14. Tekst, znaki i wyrażenia regularne	649
14.1. Wprowadzenie	650
14.2. Podstawy znaków i tekstów	650
14.3. Klasa String	651
14.3.1. Konstruktory klasy String	651
14.3.2. Metody length, charAt i getChars	652
14.3.3. Porównywanie tekstów	653
14.3.4. Znajdowanie położenia znaków i fragmentów w tekstach	658
14.3.5. Wydobywanie fragmentu tekstu	660
14.3.6. Łączenie tekstów	661
14.3.7. Inne metody klasy String	661
14.3.8. Metoda valueOf klasy String	663
14.4. Klasa StringBuilder	664
14.4.1. Konstruktory StringBuilder	665
14.4.2. Metody length, capacity, setLength i ensureCapacity klasy StringBuilder	665
14.4.3. Metody charAt, setCharAt, getChars i reverse klasy StringBuilder	667
14.4.4. Metody append klasy StringBuilder	668
14.4.5. Metody wstawiania i usuwania klasy StringBuilder	670
14.5. Klasa Character	671
14.6. Tokenizacja tekstów	675
14.7. Wyrażenia regularne, klasy Pattern i Matcher	676
14.7.1. Zastępowanie fragmentów tekstu i podział tekstu	682
14.7.2. Klasy Pattern i Matcher	683
14.8. Podsumowanie	685
15. Pliki, strumienie wejścia - wyjścia, NIO i serializacja XML	697
15.1. Wprowadzenie	698
15.2. Pliki i strumienie	698
15.3. Użycie klas i interfejsów NIO do pobrania informacji o pliku	

lub folderze	700
15.4. Sekwencyjne pliki tekstowe	704
15.4.1. Tworzenie sekwencyjnego pliku tekstowego	704
15.4.2. Odczyt danych z sekwencyjnego pliku tekstowego	707
15.4.3. Studium przypadku — program sprawdzający salda płatności klientów	709
15.4.4. Aktualizacja plików sekwencyjnych	713
15.5. Serializacja XML	713
15.5.1. Tworzenie sekwencyjnego pliku używającego serializacji XML	714
15.5.2. Odczyt i deserializacja danych z pliku sekwencyjnego	719
15.6. Okna dialogowe FileChooser i DirectoryChooser	720
15.7. (Opcjonalnie) Dodatkowe klasy java.io	726
15.7.1. Interfejsy i klasy dotyczące wejścia - wyjścia danych bajtowych	727
15.7.2. Interfejsy i klasy dla znakowych operacji wejścia - wyjścia	729
15.8. Podsumowanie	729
16. Ogólne kolekcje	739
16.1. Wprowadzenie	740
16.2. Omówienie kolekcji	740
16.3. Klasy otoczkowe	742
16.4. Automatyczne pakowanie i rozpakowywanie	742
16.5. Interfejs Collection i klasa Collections	743
16.6. Listy	743
16.6.1. ArrayList i Iterator	744
16.6.2. Klasa LinkedList	747
16.7. Metody klasy Collections	751
16.7.1. Metoda sort	752
16.7.2. Metoda shuffle	755
16.7.3. Metody reverse, fill, copy, max i min	757
16.7.4. Metoda binarySearch	759
16.7.5. Metody addAll, frequency i disjoint	761
16.8. Klasa PriorityQueue i interfejs Queue	763
16.9. Zbiory	764
16.10. Odwzorowania	767
16.11. Synchronizowane kolekcje	771
16.12. Kolekcje niezmiennie	772
16.13. Implementacje abstrakcyjne	772
16.14. Java SE 9 — wygodne metody fabryczne dla niezmiennych kolekcji	773
16.15. Podsumowanie	776
17. Lambdy i strumienie	783
17.1. Wprowadzenie	784
17.2. Strumienie i redukcja	786
17.2.1. Sumowanie liczb od 1 do 10 pętlą for	786
17.2.2. Iteracja zewnętrzna za pomocą pętli for jest narażona na błędy	787
17.2.3. Sumowanie za pomocą strumienia i redukcji	787
17.2.4. Iteracja wewnętrzna	789
17.3. Odwzorowania i lambdy	789
17.3.1. Wyrażenia lambda	790
17.3.2. Składnia lambda	791
17.3.3. Operacje pośrednie i operacje kończące	792
17.4. Filtrowanie	793

17.5. Jak elementy poruszają się po potoku strumienia?	795
17.6. Referencje do metod	796
17.6.1. Tworzenie strumienia IntStream liczb losowych	797
17.6.2. Wykonywanie zadań dla każdego elementu strumienia za pomocą metody forEach i referencji do metody	797
17.6.3. Odzworowanie liczb całkowitych na obiekty String za pomocą metody mapToObj	798
17.6.4. Łączenie tekstów metodą collect	799
17.7. Operacje IntStream	799
17.7.1. Tworzenie obiektu IntStream i wyświetlanie jego wartości	800
17.7.2. Operacje kończące count, min, max, sum i average	801
17.7.3. Operacja kończąca reduce	802
17.7.4. Sortowanie wartości IntStream	804
17.8. Interfejsy funkcyjne	804
17.9. Lambdy — dokładniejsze spojrzenie	806
17.10. Obsługa Stream<Integer>	807
17.10.1. Tworzenie Stream<Integer>	808
17.10.2. Sortowanie strumienia i zbieranie wyników	808
17.10.3. Filtrowanie strumienia i przechowywanie wyników w celu ich późniejszego użycia	809
17.10.4. Filtrowanie i sortowanie strumienia i zbieranie wyników	809
17.10.5. Sortowanie zebranych wyników	810
17.11. Obsługa Stream<String>	810
17.11.1. Mapowanie tekstów na duże litery	811
17.11.2. Filtrowanie tekstów i sortowanie ich rosnąco bez uwzględniania wielkości liter	812
17.11.3. Filtrowanie tekstów i sortowanie ich malejąco bez uwzględniania wielkości liter	812
17.12. Obsługa Stream<Employee>	812
17.12.1. Tworzenie i wyświetlenie List<Employee>	814
17.12.2. Odfiltrowywanie pracowników z wynagrodzeniem mieszczącym się w określonym przedziale	815
17.12.3. Sortowanie pracowników na podstawie wielu pól	818
17.12.4. Odzworowanie pracowników na unikatowe nazwiska	819
17.12.5. Grupowanie pracowników według działów	821
17.12.6. Zliczenie pracowników poszczególnych działów	822
17.12.7. Suma i średnia zarobków pracowników	822
17.13. Utworzenie Stream<String> na podstawie pliku	823
17.14. Strumień wartości losowych	827
17.15. Strumień nieskończony	829
17.16. Obsługa zdarzeń za pomocą lambda	831
17.17. Dodatkowe uwagi na temat interfejsów Javy SE 8	831
17.18. Podsumowanie	832
18. Rekurencja	849
18.1. Wprowadzenie	850
18.2. Pojęcie rekurencji	851
18.3. Przykład użycia rekurencji — silnia	852
18.4. Implementacja FactorialCalculator z użyciem klasy BigInteger	855
18.5. Przykład użycia rekurencji — ciąg Fibonacciego	856

18.6. Rekurencja i stos wywołań metod	859
18.7. Rekurencja kontra iteracja	860
18.8. Wieże Hanoi	862
18.9. Fraktale	864
18.9.1. Krzywa Kocha	865
18.9.2. (Opcjonalnie) Studium przypadku — pióro Lo	866
18.9.3. (Opcjonalnie) Interfejs graficzny rysujący fraktal	868
18.9.4. (Opcjonalnie) Klasa FractalController	870
18.10. Rekurencyjne nawracanie	875
18.11. Podsumowanie	876
19. Wyszukiwanie, sortowanie i notacja dużego O	885
19.1. Wprowadzenie	886
19.2. Wyszukiwanie liniowe	887
19.3. Notacja dużego O	890
19.3.1. Algorytmy $O(1)$	890
19.3.2. Algorytmy $O(n)$	890
19.3.3. Algorytmy $O(n^2)$	891
19.3.4. Duże O dla wyszukiwania liniowego	891
19.4. Wyszukiwanie binarne	892
19.4.1. Implementacja wyszukiwania binarnego	893
19.4.2. Wydajność wyszukiwania binarnego	896
19.5. Algorytmy sortujące	896
19.6. Sortowanie przez wybieranie	897
19.6.1. Implementacja sortowania przez wybieranie	897
19.6.2. Wydajność sortowania przez wybieranie	900
19.7. Sortowanie przez wstawianie	900
19.7.1. Implementacja sortowania przez wstawianie	901
19.7.2. Wydajność sortowania przez wstawianie	903
19.8. Sortowanie przez scalanie	903
19.8.1. Implementacja algorytmu sortowania przez scalanie	904
19.8.2. Wydajność sortowania przez scalanie	908
19.9. Podsumowanie dużego O algorytmów wyszukiwania i sortowania z tego rozdziału	908
19.10. Duże zrównoleglenie i algorytmy równoległe	908
19.11. Podsumowanie	909
20. Uogólnione klasy i metody — dokładniejsze spojrzenie	917
20.1. Wprowadzenie	918
20.2. Po co nam metody uogólnione?	918
20.3. Metody uogólnione — implementacja i przekształcenia na etapie kompilacji	920
20.4. Dodatkowy problem przekształcenia w trakcie kompilacji	923
20.5. Przeciążanie metod uogólnionych	927
20.6. Klasy uogólnione	927
20.7. Wieloznaczność w metodach akceptujących parametry typu	934
20.8. Podsumowanie	938
21. Własne uogólnione struktury danych	943
21.1. Wprowadzenie	944
21.2. Klasy odnoszące się do samych siebie	945
21.3. Dynamiczna alokacja pamięci	946

21.4. Listy jednokierunkowe	946
21.4.1. Lista jednokierunkowa	947
21.4.2. Implementacja uogólnionej klasy List	947
21.4.3. Klasy uogólnione ListNode i List	950
21.4.4. Klasa ListTest	950
21.4.5. Metoda insertAtFront klasy List	952
21.4.6. Metoda insertAtBack klasy List	952
21.4.7. Metoda removeFromFront klasy List	954
21.4.8. Metoda removeFromBack klasy List	954
21.4.9. Metoda print klasy List	956
21.4.10. Tworzenie własnych pakietów	956
21.5. Stosy	960
21.6. Kolejki	963
21.7. Drzewa	966
21.8. Podsumowanie	972
Podrozdział specjalny — tworzenie własnego kompilatora	984
22. Grafika, animacje i wideo w JavaFX	997
22.1. Wprowadzenie	998
22.2. Sterowanie czcionkami poprzez kaskadowe arkusze stylów	999
22.2.1. CSS, który określa styl GUI	999
22.2.2. FXML, który definiuje GUI — wprowadzenie do XML	1002
22.2.3. Referencja do pliku CSS w dokumencie FXML	1005
22.2.4. Określenie klasy dla stylów w VBox	1006
22.2.5. Programowe wczytywanie CSS	1006
22.3. Wyświetlanie kształtów dwuwymiarowych	1006
22.3.1. Definiowanie dwuwymiarowych kształtów za pomocą FXML	1007
22.3.2. CSS, który zapewnia stylowanie dwuwymiarowych kształtów	1009
22.4. Klasy Polyline, Polygon i Path	1012
22.4.1. GUI i CSS	1012
22.4.2. Klasa PolyShapesController	1014
22.5. Przekształcenia	1017
22.6. Odtwarzanie materiału wideo	1019
22.6.1. Interfejs graficzny aplikacji	1020
22.6.2. Klasa VideoPlayerController	1022
22.7. Animacje Transition	1026
22.7.1. Plik TransitionAnimations.fxml	1026
22.7.2. Klasa TransitionAnimationsController	1028
22.8. Animacje Timeline	1031
22.9. Animacja klatka po klatce za pomocą klasy AnimationTimer	1034
22.10. Rysowanie na kanwie	1037
22.11. Kształty trójwymiarowe	1042
22.12. Podsumowanie	1045
23. Współbieżność	1063
23.1. Wprowadzenie	1064
23.2. Stany i cykl życia wątku	1066
23.2.1. Stan nowy i stan działający	1066
23.2.2. Stan oczekujący	1067
23.2.3. Stan oczekujący czasowo	1067

23.2.4. Stan zablokowany	1068
23.2.5. Stan zakończony	1068
23.2.6. Stan działający z punktu widzenia systemu operacyjnego	1068
23.2.7. Priorytety i harmonogramowanie wątków	1068
23.2.8. Odsuwanie wykonania w nieskończoność i blokady wzajemne	1069
23.3. Tworzenie i wykonywanie wątków za pomocą frameworku Executor	1070
23.4. Synchronizacja wątków	1074
23.4.1. Dane niezmiennie	1074
23.4.2. Monitory	1075
23.4.3. Współdzielenie modyfikowalnych danych bez synchronizacji	1076
23.4.4. Synchronizowany dostęp do współdzielonych i modyfikowanych danych — operacje niepodzielne	1080
23.5. Związek typu producent - konsument bez synchronizacji	1083
23.6. Związek typu producent - konsument — klasa ArrayBlockingQueue	1091
23.7. Związek typu producent - konsument	1094
23.8. Związek typu producent - konsument — bufor o określonym rozmiarze	1101
23.9. Związek typu producent - konsument — interfejsy Lock i Condition	1109
23.10. Kolekcje współbieżne	1116
23.11. Wielowątkowość w JavaFX	1117
23.11.1. Wykonywanie obliczeń w wątku roboczym — ciąg Fibonacciego	1119
23.11.2. Przetwarzanie wyników pośrednich — sito Eratostenesa	1125
23.12. Pomiar czasu działania metod sort i parallelSort	1130
23.13. Java SE 8 — strumień szeregowy i równoległy	1133
23.14. (Dla zaawansowanych) Interfejsy Callable i Future	1135
23.15. (Dla zaawansowanych) Framework Fork/Join	1140
23.16. Podsumowanie	1140
24. Dostęp do baz danych poprzez JDBC	1155
24.1. Wprowadzenie	1156
24.2. Relacyjne bazy danych	1157
24.3. Baza danych books	1158
24.4. Język SQL	1162
24.4.1. Podstawowe zapytanie SELECT	1162
24.4.2. Klauzula WHERE	1163
24.4.3. Klauzula ORDER BY	1165
24.4.4. Łączenie danych z wielu tabel — operator INNER JOIN	1166
24.4.5. Instrukcja INSERT	1168
24.4.6. Instrukcja UPDATE	1169
24.4.7. Instrukcja DELETE	1170
24.5. Konfiguracja bazy danych Java DB	1171
24.5.1. Tworzenie baz danych dla rozdziału w systemie Windows	1171
24.5.2. Tworzenie baz danych dla rozdziału w systemie macOS	1172
24.5.3. Tworzenie baz danych dla rozdziału w systemie Linux	1173
24.6. Połączenie się z bazą danych i wykonanie zapytania	1173
24.6.1. Automatyczne odnajdowanie sterowników	1175
24.6.2. Połączenie z bazą danych	1175
24.6.3. Utworzenie obiektu Statement do wykonywania poleceń SQL	1176
24.6.4. Wykonanie zapytania	1176
24.6.5. Przetwarzanie wyników zapytania	1176
24.7. Odpytywanie bazy danych books	1178

24.7.1. Klasa ResultSetTableModel	1178
24.7.2. Wyświetlenie interfejsu graficznego aplikacji	1185
24.7.3. Klasa DisplayQueryResultsController	1185
24.8. Interfejs RowSet	1190
24.9. Klasa PreparedStatement	1193
24.9.1. Aplikacja do przechowywania kontaktów używająca obiektów PreparedStatement	1194
24.9.2. Klasa Person	1194
24.9.3. Klasa PersonQueries	1196
24.9.4. Interfejs graficzny książki adresowej	1199
24.9.5. Klasa AddressBookController	1200
24.10. Procedury składowane	1204
24.11. Przetwarzanie transakcyjne	1205
24.12. Podsumowanie	1206
25. Wprowadzenie do JShell — interaktywnej konsoli Javy 9	1215
25.1. Wprowadzenie	1216
25.2. Instalacja JDK 9	1218
25.3. Wprowadzenie do JShell	1218
25.3.1. Uruchamianie sesji JShell	1219
25.3.2. Wykonywanie poleceń	1219
25.3.3. Jawne deklarowanie zmiennych	1221
25.3.4. Wylistowanie i wykonanie wcześniejszych fragmentów	1223
25.3.5. Wyliczanie wyrażeń i niejawne deklarowanie zmiennych	1224
25.3.6. Wykorzystywanie niejawnie zadeklarowanych zmiennych	1225
25.3.7. Sprawdzanie zawartości zmiennej	1225
25.3.8. Czyszczenie sesji JShell	1225
25.3.9. Instrukcje wielowierszowe	1226
25.3.10. Edycja fragmentów kodu	1226
25.3.11. Wyjście z JShell	1229
25.4. Wejście danych z konsoli w JShell	1229
25.5. Deklarowanie i stosowanie klas	1231
25.5.1. Tworzenie klasy w JShell	1231
25.5.2. Jawne deklarowanie zmiennych będących referencjami	1232
25.5.3. Tworzenie obiektów	1232
25.5.4. Modyfikowanie obiektów	1233
25.5.5. Tworzenie własnej nazwy zmiennej dla wyrażenia	1233
25.5.6. Zapisywanie i otwieranie plików z fragmentami kodu	1234
25.6. Automatyczne uzupełnianie	1234
25.6.1. Automatyczne uzupełnianie identyfikatorów	1235
25.6.2. Automatyczne uzupełnianie poleceń JShell	1236
25.7. Przeglądanie składowych klas i dokumentacji	1236
25.7.1. Wyświetlanie składowych statycznych klasy Math	1237
25.7.2. Przeglądanie parametrów metod	1238
25.7.3. Przeglądanie dokumentacji metody	1238
25.7.4. Przeglądanie dokumentacji pól publicznych	1238
25.7.5. Przeglądanie dokumentacji klasy	1239
25.7.6. Przeglądanie przeciążeń metod	1240
25.7.7. Odkrywanie składowych konkretnego obiektu	1240
25.8. Deklarowanie metod	1242

25.8.1. Użycie w przód niezadeklarowanej metody — deklaracja metody displayCubes	1242
25.8.2. Deklarowanie wcześniej niezadeklarowanej metody	1243
25.8.3. Testowanie metody cube i zmiana jej deklaracji	1243
25.8.4. Testowanie uaktualnionej metody cube i metody displayCubes	1244
25.9. Wyjątki	1244
25.10. Import klas i dodawanie pakietów do CLASSPATH	1245
25.11. Korzystanie z zewnętrznego edytora	1247
25.12. Podsumowanie poleceń JShell	1250
25.12.1. Uzyskiwanie pomocy	1251
25.12.2. Polecenie /edit — dodatkowe funkcjonalności pomocy	1251
25.12.3. Polecenie /reload	1251
25.12.4. Polecenie /drop	1252
25.12.5. Tryby informacji zwrotnej	1253
25.12.6. Inne opcje JShell konfigurowalne za pomocą /set	1254
25.13. Skróty-klawiaturowe w edycji fragmentów	1255
25.14. W jaki sposób JShell modyfikuje Jave w celu jej interaktywnego użycia?	1256
25.15. Obsługa JShell w IDE	1256
25.16. Podsumowanie	1256
26. Sieć	1273
(online)	
26.1. Wprowadzenie	1274
26.2. Odczyt pliku z serwera WWW	1275
26.3. Wykonanie prostego serwera przy użyciu gniazd strumieniowych	1278
26.4. Wykonanie prostego klienta przy użyciu gniazd strumieniowych	1280
26.5. Interakcja klienta i serwera wykorzystująca gniazda strumieniowe	1281
26.6. Datagramy — bezpołączeniowa interakcja między klientem i serwerem	1293
26.7. Kółko i krzyżyk w wersji klient - serwer z serwerem wielowątkowym	1301
26.8. Opcjonalne studium przypadku — aplikacja DeitelMessenger	1315
26.9. Podsumowanie	1315
27. System modułów platformy Java	1321
(online)	
27.1. Wprowadzenie	1323
27.2. Deklaracja modułu	1328
27.2.1. Dyrektywa requires	1328
27.2.2. Dyrektywa requires transitive — niejawne czytanie	1328
27.2.3. Dyrektywy exports i exports...to	1329
27.2.4. Dyrektywa uses	1329
27.2.5. Dyrektywa provides...with	1329
27.2.6. Modyfikator open oraz dyrektywy opens i opens...to	1329
27.2.7. Ograniczone słowa kluczowe	1331
27.3. Modułowa wersja aplikacji powitalnej	1331
27.3.1. Struktura aplikacji	1331
27.3.2. Klasa Welcome	1335
27.3.3. Plik module-info.java	1335
27.3.4. Graf zależności modułu	1335

27.3.5. Kompilacja modułu	1337
27.3.6. Uruchamianie aplikacji z poziomu rozbitego folderu aplikacji	1338
27.3.7. Spakowanie modułu do pliku JAR	1338
27.3.8. Uruchamianie aplikacji z modułowego pliku JAR	1339
27.3.9. Dodatkowa uwaga — ścieżka klas a ścieżka modułów	1339
27.4. Tworzenie i użycie własnych modułów	1340
27.4.1. Eksport pakietu w celu użycia w innych modułach	1341
27.4.2. Wykorzystanie klasy pakietu w innym module	1342
27.4.3. Kompilacja i uruchomienie przykładu	1344
27.4.4. Zapakowanie aplikacji do modułowych plików JAR	1345
27.4.5. Silna enkapsulacja i dostępność	1345
27.5. Graf zależności modułu — dokładniejsze spojrzenie	1346
27.5.1. Moduł <code>java.sql</code>	1346
27.5.2. Moduł <code>java.se</code>	1347
27.5.3. Wyświetlenie grafu zależności modułów JDK	1347
27.5.4. Błąd — graf modułu z cyklem	1347
27.6. Migracja kodu do Javy 9	1350
27.6.1. Moduł nienazwany	1351
27.6.2. Moduły automatyczne	1351
27.6.3. Narzędzie <code>jdeps</code> — analiza zależności	1352
27.7. Zasoby w modułach — wykorzystanie modułu automatycznego	1354
27.7.1. Moduły automatyczne	1355
27.7.2. Wymaganie kilku modułów	1356
27.7.3. Otwarcie modułu na potrzeby mechanizmu refleksji	1356
27.7.4. Graf zależności modułu	1356
27.7.5. Kompilacja modułu	1357
27.7.6. Uruchomienie aplikacji po modularyzacji	1357
27.8. Tworzenie własnych systemów wykonawczych narzędziem <code>jlink</code>	1358
27.8.1. Wyświetlenie listy modułów JRE	1358
27.8.2. Własny system wykonawczy zawierający tylko moduł <code>java.base</code>	1359
27.8.3. Tworzenie własnego systemu wykonawczego dla aplikacji powitalnej	1361
27.8.4. Wykonywanie aplikacji powitalnej we własnym systemie wykonawczym	1362
27.8.5. Użycie mechanizmu rozwiązywania modułów z własnym systemem wykonawczym	1362
27.9. Usługi i klasa <code>ServiceLoader</code>	1363
27.9.1. Interfejs dostawcy usług	1365
27.9.2. Wczytywanie i użycie dostawców usług	1366
27.9.3. Dyrektywa <code>uses</code> modułu i konsumpcja usług	1368
27.9.4. Uruchomienie aplikacji bez dostawców usług	1369
27.9.5. Implementacja dostawcy usług	1369
27.9.6. Dyrektywa <code>provides...with</code> modułu i deklaracja dostawcy usług	1370
27.9.7. Uruchomienie aplikacji z jednym dostawcą usług	1370
27.9.8. Implementacja drugiego dostawcy usług	1371
27.9.9. Uruchomienie aplikacji z dwoma dostawcami usług	1372
27.10. Podsumowanie	1372
28. Pozostałe tematy związane z Javą 9	1375
(online)	
28.1. Wprowadzenie	1376
28.2. Przypomnienie — funkcjonalności Javy 9 omówione w poprzednich rozdziałach	1377

28.3. Nowa wersja formatu tekstowego	1378
28.4. Wyrażenia regularne — nowe metody klasy Matcher	1378
28.4.1. Metody appendReplacement i appendTail	1380
28.4.2. Metody replaceFirst i replaceAll	1380
28.4.3. Metoda results	1380
28.5. Nowe metody interfejsu Stream	1381
28.5.1. Metody take While i drop While strumienia	1382
28.5.2. Metoda iterate interfejsu Stream	1383
28.5.3. Metoda ofNullable interfejsu Stream	1383
28.6. Moduły w JShell	1384
28.7. API skórek dostępne w JavaFX 9	1385
28.8. Inne usprawnienia związane z interfejsem graficznym i grafiką	1386
28.8.1. Obrazy o wielu rozdzielczościach	1386
28.8.2. Obsługa obrazów TIFF	1386
28.8.3. Funkcjonalności pulpitu zależne od platformy	1386
28.9. Tematy związane z bezpieczeństwem Javy 9	1387
28.9.1. Filtrowanie nadchodzących danych serializowanych	1387
28.9.2. Domyślne tworzenie magazynów kluczy PKCS12	1387
28.9.3. Obsługa protokołu DTLS (Datagram Transport Layer Security)	1387
28.9.4. Obsługa walidacji OCSP dla TLS	1388
28.9.5. Rozszerzenie umożliwiające negocjację protokołu warstwy aplikacyjnej w TLS	1388
28.10. Inne tematy związane z Javą 9	1388
28.10.1. Usprawnienie łączenia tekstów	1388
28.10.2. Obsługa usług i API logów na poziomie platformy	1388
28.10.3. Aktualizacja API procesów	1389
28.10.4. Podpowiedzi dotyczące oczekiwania	1389
28.10.5. Obsługa paczek zasobów z kodowaniem UTF-8	1389
28.10.6. Domyślne korzystanie z danych CLDR	1389
28.10.7. Usunięcie ostrzeżeń o wycofaniu z instrukcji importu	1390
28.10.8. Wielowydaniowe pliki JAR	1390
28.10.9. Unicode 8	1390
28.10.10. Rozbudowa obsługi współbieżności	1390
28.11. Elementy usunięte z JDK i Javy 9	1391
28.12. Elementy zaproponowane do usunięcia w przyszłych wersjach Javy	1392
28.12.1. Ulepszone wycofywanie	1392
28.12.2. Elementy, które prawdopodobnie zostaną usunięte w przyszłych wydaniach Javy	1392
28.12.3. Znajdowanie wycofywanych funkcjonalności	1393
28.12.4. Aplety Javy	1393
28.13. Podsumowanie	1391
A Tabela kolejności wykonywania operatorów	1395
B Zbiór znaków ASCII	1397
C Słowa kluczowe i słowa zarezerwowane	1399
D Typy podstawowe	1401

E Korzystanie z debuggera	1401
E.1. Wprowadzenie	1402
E.2. Punkty wstrzymania oraz polecenia run, stop, cont i print	1402
E.3. Polecenia print i set	1406
E.4. Sterowanie wykonywaniem za pomocą poleceń step, step up i next	1408
E.5. Polecenie watch	1410
E.6. Polecenie elear	1412
E.7. Podsumowanie	1415
 Skorowidz	 1417