

Spis treści

O autorze	23
O redaktorze merytorycznym	24
Przedmowa	25

CZĘŚĆ I Język Java

1 Historia i ewolucja języka Java	31
Rodowód Javy	31
Narodziny nowoczesnego języka — C	31
Język C++ — następny krok	33
Podwaliny języka Java	33
Powstanie języka Java	33
Powiązanie z językiem C#	35
Jak Java wywarła wpływ na internet	35
Aplety Javy	35
Bezpieczeństwo	36
Przenośność	36
Magia języka Java — kod bajtowy	36
Wychodząc poza aplety	37
Szybszy harmonogram udostępniania	38
Serwlety — Java po stronie serwera	39
Hasła języka Java	39
Prostota	40
Obiektowość	40
Niezawodność	40
Wielowątkowość	41
Neutralność architektury	41
Interpretowalność i wysoka wydajność	41
Rozproszenie	41
Dynamika	41
Ewolucja Javy	41
Kultura innowacji	46

2	Podstawy języka Java	47
	Programowanie obiektowe	47
	Dwa paradygmaty	47
	Abstrakcja	48
	Trzy zasady programowania obiektowego	48
	Pierwszy przykładowy program	52
	Wpisanie kodu programu	52
	Kompilacja programów	53
	Bliższe spojrzenie na pierwszy przykładowy program	53
	Drugi prosty program	55
	Dwie instrukcje sterujące	56
	Instrukcja if	57
	Pętla for	58
	Bloki kodu	59
	Kwestie składniowe	60
	Znaki białe	60
	Identyfikatory	60
	Stałe	60
	Komentarze	61
	Separatory	61
	Słowa kluczowe języka Java	61
	Biblioteki klas Javy	62
3	Typy danych, zmienne i tablice	63
	Java to język ze ścisłą kontrolą typów	63
	Typy proste	63
	Typy całkowitoliczbowe	64
	Typ byte	64
	Typ short	65
	Typ int	65
	Typ long	65
	Typy zmiennoprzecinkowe	65
	Typ float	66
	Typ double	66
	Typ znakowy	66
	Typ logiczny	68
	Bliższe spojrzenie na stałe	68
	Stałe całkowitoliczbowe	68
	Stałe zmiennoprzecinkowe	69
	Stałe logiczne	70
	Stałe znakowe	70
	Stałe łańcuchowe	71
	Zmienne	71
	Deklaracja zmiennej	71
	Inicjalizacja dynamiczna	72
	Zasięg i czas życia zmiennych	72
	Konwersja typów i rzutowanie	74
	Automatyczna konwersja typów	74
	Rzutowanie niezgodnych typów	75
	Automatyczne rozszerzanie typów w wyrażeniach	76
	Zasady rozszerzania typu	76
	Tablice	77
	Tablice jednowymiarowe	77
	Tablice wielowymiarowe	79
	Alternatywna składnia deklaracji tablicy	82

Wnioskowanie typów zmiennych lokalnych	83
Ograniczenia var	84
Kilka słów o łańcuchach	85
4 Operatory	86
Operatory arytmetyczne	86
Podstawowe operatory arytmetyczne	87
Operator reszty z dzielenia	87
Operatory arytmetyczne z przypisaniem	88
Inkrementacja i dekrementacja	89
Operatory bitowe	90
Logiczne operatory bitowe	91
Przesunięcie w lewo	93
Przesunięcie w prawo	94
Przesunięcie w prawo bez znaku	95
Operatory bitowe z przypisaniem	96
Operatory relacji	97
Operatory logiczne	98
Operatory logiczne ze skracaniem	99
Operator przypisania	100
Operator ?	100
Kolejność wykonywania operatorów	101
Stosowanie nawiasów okrągłych	101
5 Instrukcje sterujące	103
Instrukcje wyboru	103
Instrukcja if	103
Tradycyjna instrukcja switch	106
Instrukcje iteracyjne	110
Pętla while	110
Pętla do-while	111
Pętla for	113
Wersja for-each pętli for	116
Wnioskowanie typów zmiennych lokalnych w pętlach for	120
Pętłe zagnieżdżone	121
Instrukcje skoku	122
Instrukcja break	122
Instrukcja continue	125
Instrukcja return	126
6 Wprowadzenie do klas	128
Klasy	128
Ogólna postać klasy	128
Prosta klasa	129
Deklarowanie obiektów	131
Bliższe spojrzenie na operator new	132
Przypisywanie zmiennych referencyjnych do obiektów	132
Wprowadzenie do metod	133
Dodanie metody do klasy Box	133
Zwracanie wartości	135
Dodanie metody przyjmującej parametry	136
Konstruktor	138
Konstruktor sparymetryzowany	139
Słowo kluczowe this	140
Ukrywanie zmiennych składowych	140

Mechanizm odzyskiwania pamięci	141
Klasa stosu	141
7 Dokładniejsze omówienie metod i klas	144
Przeciążanie metod	144
Przeciążanie konstruktorów	146
Obiekty jako parametry	148
Dokładniejsze omówienie przekazywania argumentów	150
Zwracanie obiektów	151
Rekurencja	152
Wprowadzenie do kontroli dostępu	154
Składowe statyczne	157
Słowo kluczowe final	158
Powtórka z tablic	159
Klasy zagnieżdżone i klasy wewnętrzne	160
Omówienie klasy String	162
Wykorzystanie argumentów wiersza poleceń	164
Zmienna liczba argumentów	165
Przeciążanie metod o zmiennej liczbie argumentów	167
Zmienna liczba argumentów i niejednoznaczności	168
Stosowanie wnioskowania typów zmiennych lokalnych z typami referencyjnymi	170
8 Dziedziczenie	172
Podstawy dziedziczenia	172
Dostęp do składowych a dziedziczenie	173
Bardziej praktyczny przykład	174
Zmienna klasy bazowej może zawierać referencję do obiektu klasy pochodnej	176
Słowo kluczowe super	177
Wykorzystanie słowa kluczowego super do wywołania konstruktora klasy bazowej	177
Drugie zastosowanie słowa kluczowego super	180
Tworzenie hierarchii wielopoziomowej	180
Kiedy są wykonywane konstruktory?	183
Przesłanianie metod	184
Dynamiczne przydzielanie metod	186
Dlaczego warto przesłaniać metody?	187
Zastosowanie przesłaniania metod	187
Klasy abstrakcyjne	189
Słowo kluczowe final i dziedziczenie	191
Słowo kluczowe final zapobiega przesłanianiu	191
Słowo kluczowe final zapobiega dziedziczeniu	192
Wnioskowanie typów zmiennych lokalnych a dziedziczenie	192
Klasa Object	193
9 Pakiety i interfejsy	195
Pakiety	195
Definiowanie pakietu	195
Znajdowanie pakietów i ścieżka CLASSPATH	196
Prosty przykład pakietu	197
Dostęp do pakietów i składowych	197
Przykład dostępu	198
Import pakietów	201
Interfejsy	202
Definiowanie interfejsu	203
Implementacja interfejsu	204
Interfejsy zagnieżdżone	206
Stosowanie interfejsów	207

Zmienne w interfejsach	209
Interfejsy można rozszerzać	211
Metody domyślne	211
Podstawy metod domyślnych	212
Bardziej praktyczny przykład	214
Problemy wielokrotnego dziedziczenia	214
Metody statyczne w interfejsach	215
Stosowanie metod prywatnych w interfejsach	216
Ostatnie uwagi dotyczące pakietów i interfejsów	217
10 Obsługa wyjątków	218
Podstawy obsługi wyjątków	218
Typy wyjątków	219
Nieprzechwycone wyjątki	219
Stosowanie instrukcji try i catch	220
Wyświetlenie opisu wyjątku	221
Wiele klauzul catch	222
Zagnieżdżone instrukcje try	223
Instrukcja throw	225
Klauzula throws	226
Słowo kluczowe finally	227
Wyjątki wbudowane w język Java	228
Tworzenie własnej klasy pochodnej wyjątków	229
Łańcuch wyjątków	231
Trzy dodatkowe cechy wyjątków	232
Wykorzystanie wyjątków	233
11 Programowanie wielowątkowe	234
Model wątków języka Java	235
Priorytety wątków	236
Synchronizacja	236
Przekazywanie komunikatów	236
Klasa Thread i interfejs Runnable	237
Wątek główny	237
Tworzenie wątku	238
Implementacja interfejsu Runnable	239
Rozszerzanie klasy Thread	240
Wybór odpowiedniego podejścia	241
Tworzenie wielu wątków	241
Stosowanie metod isAlive() i join()	243
Priorytety wątków	245
Synchronizacja	245
Synchronizacja metod	246
Instrukcja synchronized	247
Komunikacja międzywątkowa	249
Zakleszczenie	252
Zawieszanie, wznawianie i zatrzymywanie wątków	254
Uzyskiwanie stanu wątku	256
Stosowanie metody wytwórczej do tworzenia i uruchamiania wątku	257
Korzystanie z wielowątkowości	258
12 Wyliczenia, automatyczne opakowywanie typów prostych i adnotacje	259
Typy wyliczeniowe	259
Podstawy wyliczeń	259
Metody values() i valueOf()	261

Wyliczenia Javy jako typy klasowe	262
Wyliczenia dziedziczą po klasie Enum	264
Inny przykład wyliczenia	265
Opakowania typów	267
Klasa Character	267
Klasa Boolean	267
Opakowania typów numerycznych	268
Automatyczne opakowywanie typów prostych	269
Automatyczne opakowywanie i metody	270
Automatyczne opakowywanie i rozpakowywanie w wyrażeniach	270
Automatyczne opakowywanie typów znakowych i logicznych	272
Automatyczne opakowywanie pomaga zapobiegać błędom	272
Słowo ostrzeżenia	273
Adnotacje	273
Podstawy tworzenia adnotacji	273
Określanie strategii zachowywania adnotacji	274
Odczytywanie adnotacji w trakcie działania programu za pomocą refleksji	275
Interfejs AnnotatedElement	279
Wartości domyślne	279
Adnotacje znacznikowe	280
Adnotacje jednoelementowe	281
Wbudowane adnotacje	282
Adnotacje typów	284
Adnotacje powtarzalne	288
Ograniczenia	289
13 Wejście-wyjście, instrukcja try z zasobami i inne tematy	290
Podstawowa obsługa wejścia i wyjścia	290
Strumienie	291
Strumienie znakowe i bajtowe	291
Predefiniowane strumienie	293
Odczyt danych z konsoli	293
Odczyt znaków	294
Odczyt łańcuchów	295
Wyświetlanie informacji na konsoli	296
Klasa PrintWriter	297
Odczyt i zapis plików	297
Automatyczne zamykanie pliku	303
Modyfikatory transient i volatile	306
Prezentacja operatora instanceof	306
Modyfikator strictfp	308
Metody napisane w kodzie rdzennym	308
Stosowanie asercji	309
Opcje włączania i wyłączania asercji	311
Import statyczny	311
Wywoływanie przeciążonych konstruktorów za pomocą this()	313
Kilka słów o klasach wartościowych	315
14 Typy sparametryzowane	316
Czym są typy sparametryzowane?	316
Prosty przykład zastosowania typów sparametryzowanych	317
Typy sparametryzowane działają tylko dla typów referencyjnych	320
Typy sparametryzowane różnią się, jeśli mają inny argument typu	320
W jaki sposób typy sparametryzowane zwiększają bezpieczeństwo?	320
Klasa sparametryzowana z dwoma parametrami typu	322

Ogólna postać klasy sparametryzowanej	323
Typy ograniczone	323
Zastosowanie argumentów wieloznacznych	325
Ograniczony argument wieloznaczny	328
Tworzenie metody sparametryzowanej	332
Konstruktory sparametryzowane	333
Interfejsy sparametryzowane	334
Typy surowe i starszy kod	336
Hierarchia klas sparametryzowanych	338
Zastosowanie sparametryzowanej klasy bazowej	338
Podklasa sparametryzowana	340
Porównywanie typów w hierarchii klas sparametryzowanych w czasie wykonywania	341
Rzutowanie	342
Przesłanie metod w klasach sparametryzowanych	343
Wnioskowanie typów a typy sparametryzowane	344
Wnioskowanie typów zmiennych lokalnych a typy sparametryzowane	345
Znoszenie	345
Metody mostu	345
Błędy niejednoznaczności	347
Pewne ograniczenia typów sparametryzowanych	348
Nie można tworzyć obiektu parametru typu	348
Ograniczenia dla składowych statycznych	348
Ograniczenia tablic typów sparametryzowanych	348
Ograniczenia wyjątków typów sparametryzowanych	349
15 Wyrażenia lambda	350
Wprowadzenie do wyrażeń lambda	350
Podstawowe informacje o wyrażeniach lambda	351
Interfejsy funkcyjne	351
Kilka przykładów wyrażeń lambda	352
Blokowe wyrażenia lambda	355
Sparametryzowane interfejsy funkcyjne	357
Przekazywanie wyrażeń lambda jako argumentów	358
Wyrażenia lambda i wyjątki	361
Wyrażenia lambda i przechwytywanie zmiennych	362
Referencje do metod	363
Referencje do metod statycznych	363
Referencje do metod instancyjnych	364
Referencje do metod a typy sparametryzowane	367
Referencje do konstruktorów	369
Predefiniowane interfejsy funkcyjne	373
16 Moduły	375
Podstawowe informacje o modułach	375
Przykład prostego modułu	376
Kompilowanie i uruchamianie przykładowej aplikacji	379
Dokładniejsze informacje o instrukcjach <code>requires</code> i <code>exports</code>	380
<code>java.base</code> i moduły platformy	381
Stary kod i moduł nienazwany	382
Eksportowanie do konkretnego modułu	383
Wymagania przechodnie	384
Stosowanie usług	388
Podstawowe informacje o usługach i dostawcach usług	388
Słowa kluczowe związane z usługami	388
Przykład stosowania usług i modułów	389

Grafy modułów	395
Trzy wyspecjalizowane cechy modułów	395
Moduły otwarte	395
Instrukcja opens	396
Instrukcja requires static	396
Wprowadzenie do jlink i plików JAR modułów	396
Dołączanie plików dostarczonych jako struktura katalogów	396
Konsolidacja modularnych plików JAR	397
Pliki JMOD	398
Kilka słów o warstwach i modułach automatycznych	398
Końcowe uwagi dotyczące modułów	398
17 Wyrażenia switch, rekordy oraz inne najnowsze możliwości języka	399
Rozszerzenia switch	400
Listy stałych	401
Wprowadzanie wyrażenia switch i instrukcji yield	402
Wprowadzenie strzałek do klauzul case	403
Dokładniejsza prezentacja zapisu ze strzałką	405
Inny przykład wyrażenia switch	408
Blok tekstowy	408
Podstawowe informacje o blokach tekstu	408
Traktowanie początkowych białych znaków	409
Użycie cudzysłowów w blokach tekstu	411
Sekwencje specjalne w blokach tekstu	411
Rekordy	412
Podstawowe informacje o rekordach	412
Tworzenie konstruktorów rekordów	414
Inny przykład konstruktora rekordu	418
Tworzenie metod pobierających w rekordach	419
Dopasowywanie wzorców z użyciem operatora instanceof	420
Zmienne wzorców i logiczne wyrażenia AND	421
Dopasowywanie wzorców w innych wyrażeniach	422
Klasy i interfejsy „zapieczętowane”	423
Klasy zapieczętowane	423
Interfejsy zapieczętowane	425
Kierunki rozwoju	426

CZĘŚĆ II

Biblioteka języka Java

18 Obsługa łańcuchów	429
Konstruktory klasy String	429
Długość łańcucha	431
Specjalne operacje na łańcuchach	431
Literały tekstowe	431
Konkatenacja łańcuchów	432
Konkatenacja łańcuchów z innymi typami danych	432
Konwersja łańcuchów i metoda toString()	433
Wyodrębnianie znaków	434
Metoda charAt()	434
Metoda getChars()	434
Metoda getBytes()	434
Metoda toCharArray()	435

Porównywanie łańcuchów	435
Metody equals() i equalsIgnoreCase()	435
Metoda regionMatches()	436
Metody startsWith() i endsWith()	436
Metoda equals() kontra operator ==	436
Metoda compareTo()	437
Przeszukiwanie łańcuchów	438
Modyfikowanie łańcucha	439
Metoda substring()	439
Metoda concat()	440
Metoda replace()	440
Metody trim() i strip()	441
Konwersja danych za pomocą metody valueOf()	442
Zmiana wielkości liter w łańcuchu	442
Łączenie łańcuchów	443
Dodatkowe metody klasy String	443
Klasa StringBuffer	445
Konstruktory klasy StringBuffer	445
Metody length() i capacity()	445
Metoda ensureCapacity()	446
Metoda setLength()	446
Metody charAt() i setCharAt()	446
Metoda getChars()	447
Metoda append()	447
Metoda insert()	447
Metoda reverse()	448
Metody delete() i deleteCharAt()	448
Metoda replace()	449
Metoda substring()	449
Dodatkowe metody klasy StringBuffer	449
Klasa StringBuilder	450
19 Pakiet java.lang	451
Opakowania typów prostych	451
Klasa Number	452
Klasy Double i Float	452
Metody isInfinite() i isNaN()	455
Klasy Byte, Short, Integer i Long	456
Klasa Character	464
Dodatki wprowadzone w celu obsługi punktów kodowych Unicode	466
Klasa Boolean	467
Klasa Void	468
Klasa Process	468
Klasa Runtime	469
Wykonywanie innych programów	470
Runtime.Version	471
Klasa ProcessBuilder	473
Klasa System	475
Wykorzystanie metody currentTimeMillis()	
do obliczania czasu wykonywania programu	476
Użycie metody arraycopy()	477
Właściwości środowiska	477
Interfejs System.Logger i klasa System.LoggerFinder	478
Klasa Object	478

Wykorzystanie metody clone() i interfejsu Cloneable	478
Klasa Class	480
Klasa ClassLoader	483
Klasa Math	483
Funkcje trygonometryczne	483
Funkcje wykładnicze	484
Funkcje zaokrągleń	484
Inne metody klasy Math	485
Klasa StrictMath	487
Klasa Compiler	487
Klasa Thread i ThreadGroup oraz interfejs Runnable	487
Interfejs Runnable	488
Klasa Thread	488
Klasa ThreadGroup	490
Klasa ThreadLocal i InheritableThreadLocal	493
Klasa Package	493
Klasa Module	494
Klasa ModuleLayer	495
Klasa RuntimePermission	495
Klasa Throwable	495
Klasa SecurityManager	495
Klasa StackTraceElement	495
Klasa StackWalker i interfejs StackWalker.StackFrame	496
Klasa Enum	496
Klasa Record	497
Klasa ClassValue	497
Interfejs CharSequence	497
Interfejs Comparable	498
Interfejs Appendable	498
Interfejs Iterable	499
Interfejs Readable	499
Interfejs AutoCloseable	499
Interfejs Thread.UncaughtExceptionHandler	500
Podpakiety pakietu java.lang	500
Podpakiet java.lang.annotation	500
Podpakiet java.lang.constant	500
Podpakiet java.lang.instrument	500
Podpakiet java.lang.invoke	500
Podpakiet java.lang.management	500
Podpakiet java.lang.module	501
Podpakiet java.lang.ref	501
Podpakiet java.lang.reflect	501
20 Pakiet java.util, część 1. — kolekcje	502
Wprowadzenie do kolekcji	503
Interfejsy kolekcji	504
Interfejs Collection	505
Interfejs List	507
Interfejs Set	508
Interfejs SortedSet	509
Interfejs NavigableSet	509
Interfejs Queue	510
Interfejs Deque	511

Klasy kolekcji	512
Klasa ArrayList	513
Klasa LinkedList	516
Klasa HashSet	517
Klasa LinkedHashSet	518
Klasa TreeSet	518
Klasa PriorityQueue	519
Klasa ArrayDeque	520
Klasa EnumSet	521
Dostęp do kolekcji za pomocą iteratora	521
Korzystanie z iteratora Iterator	523
Pętla typu for-each jako alternatywa dla iteratora	524
Spliterator	525
Przechowywanie w kolekcjach własnych klas	527
Interfejs RandomAccess	529
Korzystanie z map	529
Interfejsy map	529
Klasy map	534
Komparatory	538
Stosowanie komparatora	540
Algorytmy kolekcji	545
Klasy Arrays	550
Starsze klasy i interfejsy	554
Interfejs Enumeration	554
Klasa Vector	555
Klasa Stack	558
Klasa Dictionary	559
Klasa Hashtable	560
Klasa Properties	563
Wykorzystanie metod store() i load()	566
Ostatnie uwagi na temat kolekcji	567
21 Pakiet java.util, część 2. — pozostałe klasy użytkowe	568
Klasa StringTokenizer	568
Klasa BitSet	570
Klasy Optional, OptionalDouble, OptionalInt oraz OptionalLong	572
Klasa Date	575
Klasa Calendar	576
Klasa GregorianCalendar	579
Klasa TimeZone	580
Klasa SimpleTimeZone	581
Klasa Locale	582
Klasa Random	583
Klasy Timer i TimerTask	585
Klasa Currency	587
Klasa Formatter	588
Konstruktory klasy Formatter	589
Metody klasy Formatter	589
Podstawy formatowania	590
Formatowanie łańcuchów i znaków	592
Formatowanie liczb	592
Formatowanie daty i godziny	593
Specyfikatory %n i %%	595
Określanie minimalnej szerokości pola	595

Określanie precyzji	596
Używanie znaczników (flag) formatów	597
Wyrównywanie danych wyjściowych	597
Znaczniki spacji, plusa, zera i nawiasów	598
Znacznik przecinka	599
Znacznik #	599
Opcja wielkich liter	599
Stosowanie indeksu argumentu	600
Zamykanie obiektu klasy Formatter	601
Metoda printf() w Javie	602
Klasa Scanner	602
Konstruktory klasy Scanner	602
Podstawy skanowania	603
Kilka przykładów użycia klasy Scanner	606
Ustawianie separatorów	609
Pozostałe elementy klasy Scanner	610
Klasy ResourceBundle, ListResourceBundle i PropertyResourceBundle	611
Dodatkowe klasy i interfejsy użytkowe	615
Podpakiety pakietu java.util	616
java.util.concurrent, java.util.concurrent.atomic oraz java.util.concurrent.locks	616
java.util.function	617
java.util.jar	619
java.util.logging	619
java.util.prefs	619
java.util.random	619
java.util.regex	619
java.util.spi	620
java.util.stream	620
java.util.zip	620
22 Operacje wejścia-wyjścia: analiza pakietu java.io	621
Klasy i interfejsy obsługujące operacje wejścia-wyjścia	622
Klasa File	622
Katalogi	625
Stosowanie interfejsu FilenameFilter	626
Alternatywna metoda listFiles()	627
Tworzenie katalogów	627
Interfejsy AutoCloseable, Closeable i Flushable	627
Wyjątki operacji wejścia-wyjścia	628
Dwa sposoby zamykania strumieni	628
Klasy strumieni	629
Strumienie bajtów	630
Klasa InputStream	630
Klasa OutputStream	631
Klasa FileInputStream	632
Klasa FileOutputStream	633
Klasa ByteArrayInputStream	635
Klasa ByteArrayOutputStream	637
Filtrowane strumienie bajtów	638
Buforowane strumienie bajtów	638
Klasa SequenceInputStream	641
Klasa PrintStream	643
Klasy DataOutputStream i DataInputStream	645
Klasa RandomAccessFile	646

Strumienie znaków	647
Klasa Reader	647
Klasa Writer	647
Klasa FileReader	649
Klasa FileWriter	649
Klasa CharArrayReader	650
Klasa CharArrayWriter	651
Klasa BufferedReader	652
Klasa BufferedWriter	654
Klasa PushbackReader	654
Klasa PrintWriter	655
Klasa Console	656
Serializacja	658
Interfejs Serializable	658
Interfejs Externalizable	658
Interfejs ObjectOutput	659
Klasa ObjectOutputStream	659
Interfejs ObjectInput	660
Klasa ObjectInputStream	661
Przykład serializacji	662
Korzyści wynikające ze stosowania strumieni	665
23 System NIO	666
Klasy systemu NIO	666
Podstawy systemu NIO	667
Bufory	667
Kanały	669
Zestawy znaków i selektory	670
Udoskonalenia dodane w systemie NIO.2	671
Interfejs Path	671
Klasa Files	672
Klasa Paths	674
Interfejsy atrybutów plików	675
Klasy FileSystem, FileSystems i FileStore	676
Stosowanie systemu NIO	677
Stosowanie systemu NIO dla operacji wejścia-wyjścia na kanałach	677
Stosowanie systemu NIO dla operacji wejścia-wyjścia na strumieniach	685
Stosowanie systemu NIO dla operacji na ścieżkach i systemie plików	687
24 Obsługa sieci	695
Podstawy działania sieci	695
Klasy i interfejsy pakietu java.net obsługujące komunikację sieciową	696
Klasa InetAddress	697
Metody wytwórcze	697
Metody klasy	698
Klasy InetAddress oraz Inet6Address	699
Gniazda klientów TCP/IP	699
URL	702
Klasa URLConnection	703
Klasa HttpURLConnection	705
Klasa URI	707
Pliki cookie	707
Gniazda serwerów TCP/IP	707
Datagramy	708

Klasa DatagramSocket	708
Klasa DatagramPacket	709
Przykład użycia datagramów	710
Prezentacja pakietu java.net.http	711
Trzy kluczowe elementy	712
Prosty przykład użycia API klienta HTTP	714
Czego jeszcze warto dowiedzieć się o pakiecie java.net.http?	716
25 Obsługa zdarzeń	717
Dwa mechanizmy obsługi zdarzeń	717
Model obsługi zdarzeń oparty na ich delegowaniu	718
Zdarzenia	718
Źródła zdarzeń	718
Obiekty nasłuchujące zdarzeń	719
Klasy zdarzeń	719
Klasa ActionEvent	721
Klasa AdjustmentEvent	721
Klasa ComponentEvent	722
Klasa ContainerEvent	722
Klasa FocusEvent	723
Klasa InputEvent	724
Klasa ItemEvent	724
Klasa KeyEvent	725
Klasa MouseEvent	726
Klasa MouseWheelEvent	727
Klasa TextEvent	728
Klasa WindowEvent	728
Źródła zdarzeń	729
Interfejsy nasłuchujące zdarzeń	730
Interfejs ActionListener	731
Interfejs AdjustmentListener	731
Interfejs ComponentListener	731
Interfejs ContainerListener	731
Interfejs FocusListener	731
Interfejs ItemListener	731
Interfejs KeyListener	731
Interfejs MouseListener	732
Interfejs MouseMotionListener	732
Interfejs MouseWheelListener	732
Interfejs TextListener	732
Interfejs WindowFocusListener	732
Interfejs WindowListener	732
Stosowanie modelu delegowania zdarzeń	733
Kluczowe zagadnienia tworzenia aplikacji graficznych z użyciem AWT	733
Obsługa zdarzeń generowanych przez mysz	734
Obsługa zdarzeń generowanych przez klawiaturę	737
Klasy adapterów	740
Klasy wewnętrzne	742
Anonimowa klasa wewnętrzna	744
26 Wprowadzenie do AWT: praca z oknami, grafiką i tekstem	746
Klasy AWT	747
Podstawy okien	749
Klasa Component	749
Klasa Container	749

Klasa Panel	749
Klasa Window	750
Klasa Frame	750
Klasa Canvas	750
Praca z oknami typu Frame	750
Ustawianie wymiarów okna	750
Ukrywanie i wyświetlanie okna	751
Ustawianie tytułu okna	751
Zamykanie okna typu Frame	751
Metoda paint()	751
Wyświetlanie łańcuchów znaków	751
Określanie koloru tekstu i tła	752
Żądanie ponownego wyświetlenia zawartości okna	752
Tworzenie aplikacji korzystających z klasy Frame	753
Wprowadzenie do stosowania grafiki	754
Rysowanie odcinków	754
Rysowanie prostokątów	754
Rysowanie elips, kół i okręgów	755
Rysowanie łuków	755
Rysowanie wielokątów	755
Prezentacja metod rysujących	756
Dostosowywanie rozmiarów obiektów graficznych	757
Praca z klasą Color	758
Metody klasy Color	759
Ustawianie bieżącego koloru kontekstu graficznego	760
Program demonstrujący zastosowanie klasy Color	760
Ustawianie trybu rysowania	761
Praca z czcionkami	762
Określanie dostępnych czcionek	763
Tworzenie i wybieranie czcionek	765
Uzyskiwanie informacji o czcionkach	767
Zarządzanie tekstowymi danymi wyjściowymi z wykorzystaniem klasy FontMetrics	768
27 Stosowanie kontroltek AWT, menedżerów układu graficznego oraz menu	771
Podstawy kontroltek AWT	772
Dodawanie i usuwanie kontroltek	772
Odpowiadanie na zdarzenia kontroltek	772
Wyjątek HeadlessException	773
Etykiety	773
Stosowanie przycisków	774
Obsługa zdarzeń przycisków	775
Stosowanie pól wyboru	778
Obsługa zdarzeń pól wyboru	779
Klasa CheckboxGroup	780
Kontrolki list rozwijanych	782
Obsługa zdarzeń list rozwijanych	783
Stosowanie list	784
Obsługa zdarzeń generowanych przez listy	785
Zarządzanie paskami przewijania	787
Obsługa zdarzeń generowanych przez paski przewijania	788
Stosowanie kontroltek typu TextField	790
Obsługa zdarzeń generowanych przez kontrolkę TextField	791
Stosowanie kontroltek typu TextArea	792

Wprowadzenie do menedżerów układu graficznego komponentów	794
FlowLayout	795
BorderLayout	796
Stosowanie obramowań	797
GridLayout	799
Klasa CardLayout	800
Klasa GridBagLayout	803
Menu i paski menu	807
Okna dialogowe	812
Przesłanie metody paint()	815
28 Obrazy	817
Formaty plików	817
Podstawy przetwarzania obrazów: tworzenie, wczytywanie i wyświetlanie	818
Tworzenie obiektu obrazu	818
Ładowanie obrazu	818
Wyświetlanie obrazu	819
Podwójne buforowanie	820
Interfejs ImageProducer	822
Klasa MemoryImageSource	822
Interfejs ImageConsumer	825
Klasa PixelGrabber	825
Klasa ImageFilter	827
Klasa CropImageFilter	828
Klasa RGBImageFilter	830
Dodatkowe klasy obsługujące obrazy	840
29 Narzędzia współbieżności	841
Pakiety interfejsu Concurrent API	842
Pakiet java.util.concurrent	842
Pakiet java.util.concurrent.atomic	843
Pakiet java.util.concurrent.locks	843
Korzystanie z obiektów służących do synchronizacji	843
Klasa Semaphore	843
Klasa CountDownLatch	848
CyclicBarrier	850
Klasa Exchanger	852
Klasa Phaser	853
Korzystanie z egzekutorów	860
Przykład prostego egzekutora	860
Korzystanie z interfejsów Callable i Future	862
Typ wyliczeniowy TimeUnit	864
Kolekcje współbieżne	865
Blokady	865
Operacje atomowe	868
Programowanie równoległe przy użyciu frameworku Fork/Join	869
Najważniejsze klasy frameworku Fork/Join	870
Strategia dziel i zwyciężaj	873
Prosty przykład użycia frameworku Fork/Join	873
Znaczenie poziomu równoległości	875
Przykład użycia klasy RecursiveTask<V>	878
Asynchroniczne wykonywanie zadań	880
Anulowanie zadania	880
Określanie statusu wykonania zadania	881

Ponowne uruchamianie zadania	881
Pozostałe zagadnienia	881
Wskazówki dotyczące stosowania frameworku Fork/Join	883
Pakiet Concurrency Utilities a tradycyjne metody języka Java	883
30 API strumieni	884
Podstawowe informacje o strumieniach	884
Interfejsy strumieni	885
Jak można uzyskać strumień?	887
Prosty przykład stosowania strumieni	888
Operacje redukcji	891
Stosowanie strumieni równoległych	893
Odwzorowywanie	895
Tworzenie kolekcji	899
Iteratory i strumień	902
Stosowanie typu Iterator i strumieni	902
Stosowanie spliteratorów	903
Inne możliwości API strumieni	906
31 Wyrażenia regularne i inne pakiety	907
Przetwarzanie wyrażeń regularnych	907
Klasa Pattern	908
Klasa Matcher	908
Składnia wyrażeń regularnych	909
Przykład dopasowywania do wzorca	909
Dwie opcje dopasowywania do wzorca	914
Przegląd wyrażeń regularnych	914
Refleksja	914
Zdalne wywoływanie metod (RMI)	918
Prosta aplikacja typu klient-serwer wykorzystująca RMI	918
Formatowanie dat i czasu przy użyciu pakietu java.text	921
Klasa DateFormat	921
Klasa SimpleDateFormat	922
Interfejs API dat i czasu — java.time	924
Podstawowe klasy do obsługi dat i czasu	924
Formatowanie dat i godzin	926
Analiza łańcuchów zawierających daty i godziny	928
Inne możliwości pakietu java.time	928

CZĘŚĆ III

Wprowadzenie do programowania GUI przy użyciu pakietu Swing

32 Wprowadzenie do pakietu Swing	933
Geneza powstania biblioteki Swing	933
Bibliotekę Swing zbudowano na bazie zestawu narzędzi AWT	934
Podstawowe cechy biblioteki Swing	934
Komponenty biblioteki Swing są lekkie	934
Biblioteka Swing obsługuje dołączany wygląd i sposób obsługi	934
Podobieństwo do architektury MVC	935
Komponenty i kontenery	936
Komponenty	936
Kontenery	937
Panele kontenerów najwyższego poziomu	937
Pakiety biblioteki Swing	938

Prosta aplikacja na bazie biblioteki Swing	938
Obsługa zdarzeń	941
Rysowanie w bibliotece Swing	944
Podstawy rysowania	944
Wyznaczanie obszaru rysowania	945
Przykład rysowania	946
33 Przewodnik po pakiecie Swing	949
Klasy JLabel i ImageIcon	949
Klasa JTextField	951
Przyciski biblioteki Swing	953
Klasa JButton	953
Klasa JToggleButton	955
Poła wyboru	957
Przyciski opcji	958
Klasa JTabbedPane	960
Klasa JScrollPane	963
Klasa JList	964
Klasa JComboBox	967
Drzewa	969
Klasa JTable	972
34 Wprowadzenie do systemu menu pakietu Swing	975
Podstawy systemu menu	975
Przegląd klas JMenuBar, JMenu oraz JMenuItem	977
Klasa JMenuBar	977
Klasa JMenu	977
Klasa JMenuItem	978
Tworzenie menu głównego	979
Dodawanie mnemoników i kombinacji klawiszy do opcji menu	983
Dodawanie obrazów i etykiet ekranowych do menu	984
Stosowanie klas JRadioButtonMenuItem i JCheckBoxMenuItem	986
Tworzenie menu podręcznych	988
Tworzenie paska narzędzi	991
Stosowanie akcji	993
Finalna postać programu MenuDemo	998
Dalsze poznawanie pakietu Swing	1003

CZĘŚĆ IV

Stosowanie Javy w praktyce

35 Java Beans	1007
Czym jest komponent typu Java Bean?	1007
Zalety komponentów Java Beans	1008
Introspekcja	1008
Wzorce właściwości	1008
Wzorce projektowe dla zdarzeń	1009
Metody i wzorce projektowe	1010
Korzystanie z interfejsu BeanInfo	1010
Właściwości ograniczone	1010
Trwałość	1011
Interfejs Customizer	1011
Interfejs Java Beans API	1011
Klasa Introspector	1013
KlasaPropertyDescriptor	1013

Klasa EventSetDescriptor	1013
Klasa MethodDescriptor	1013
Przykład komponentu Java Bean	1013
36 Serwlety	1017
Podstawy	1017
Cykl życia serwletu	1018
Sposoby tworzenia serwletów	1018
Korzystanie z serwera Tomcat	1019
Przykład prostego serwletu	1020
Tworzenie i kompilacja kodu źródłowego serwletu	1020
Uruchamianie serwera Tomcat	1021
Uruchamianie przeglądarki i generowanie żądania	1021
Interfejs Servlet API	1021
Pakiet javax.servlet	1021
Interfejs Servlet	1022
Interfejs ServletConfig	1022
Interfejs ServletContext	1023
Interfejs ServletRequest	1023
Interfejs ServletResponse	1024
Klasa GenericServlet	1024
Klasa ServletInputStream	1024
Klasa ServletOutputStream	1025
Klasy wyjątków związanych z serwletami	1025
Odczytywanie parametrów serwletu	1025
Pakiet jakarta.servlet.http	1026
Interfejs HttpServletRequest	1027
Interfejs HttpServletResponse	1027
Interfejs HttpSession	1028
Klasa Cookie	1029
Klasa HttpServlet	1030
Obsługa żądań i odpowiedzi HTTP	1031
Obsługa żądań GET protokołu HTTP	1031
Obsługa żądań POST protokołu HTTP	1032
Korzystanie ze znaczników kontekstu użytkownika	1033
Śledzenie sesji	1035

DODATKI

A Komentarze dokumentujące	1039
Znaczniki narzędzia javadoc	1039
Znacznik @author	1040
Znacznik {@code}	1040
Znacznik @deprecated	1041
Znacznik {@docRoot}	1041
Znacznik @exception	1041
Znacznik @hidden	1041
Znacznik {@index}	1041
Znacznik {@inheritDoc}	1041
Znacznik {@link}	1041
Znacznik {@linkplain}	1042
Znacznik {@literal}	1042
Znacznik @param	1042
Znacznik @provides	1042

Znacznik @return	1042
Znacznik @see	1042
Znacznik @serial	1043
Znacznik @serialData	1043
Znacznik @serialField	1043
Znacznik @since	1043
Znacznik {@summary}	1043
Znacznik {@systemProperty}	1043
Znacznik @throws	1044
Znacznik @uses	1044
Znacznik {@value}	1044
Znacznik @version	1044
Ogólna postać komentarzy dokumentacyjnych	1044
Wynik działania narzędzia javadoc	1044
Przykład korzystający z komentarzy dokumentacyjnych	1045
B Wprowadzenie do JShell	1046
Podstawy JShell	1046
Wyświetlanie, edytowanie i ponowne wykonywanie kodu	1048
Dodanie metody	1049
Utworzenie klasy	1050
Stosowanie interfejsu	1050
Przetwarzanie wyrażeń i wbudowane zmienne	1051
Importowanie pakietów	1052
Wyjątki	1052
Inne polecenia JShell	1053
Dalsze poznawanie możliwości JShell	1054
C Kompilowanie i uruchamianie prostych programów w jednym kroku	1055
Skorowidz	1057