

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 37

Wychodząc poza aplety 38

Szybszy harmonogram udostępniania 38

Serwlety - Java po stronie serwera 39

Hasła języka Java 39

Prostota 40

Obiektowość 40

Niezawodność 40

Wielowątkowość 40

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 85

Kilka słów o łańcuchach 85

4. Operatory 87

Operatory arytmetyczne 87

Podstawowe operatory arytmetyczne 88

Operator reszty z dzielenia 88

Operatory arytmetyczne z przypisaniem 89

Inkrementacja i dekrementacja 90

Operatory bitowe 91

Logiczne operatory bitowe 92

Przesunięcie w lewo 94

Przesunięcie w prawo 95

Przesunięcie w prawo bez znaku 96

Operatory bitowe z przypisaniem 97

Operatory relacji 98

Operatory logiczne 99

Operatory logiczne ze skracaniem 100

Operator przypisania 101

Operator ? 101

Kolejność wykonywania operatorów 102

Stosowanie nawiasów okrągłych 102

5. Instrukcje sterujące 105

Instrukcje wyboru 105

Instrukcja if 105

Instrukcja switch 108

Instrukcje iteracyjne 111

Pętla while 112

Pętla do-while 113

Pętla for 115

Wersja for-each pętli for 118

Wnioskowanie typów zmiennych lokalnych w pętlach 122

Pętle zagnieżdżone 123

Instrukcje skoku 123

Instrukcja break 124

Instrukcja continue 127

Instrukcja return 128

6. Wprowadzenie do klas 129

Klasy 129

Ogólna postać klasy 129

Prosta klasa 130

Deklarowanie obiektów 132

Bliższe spojrzenie na operator new 133

Przypisywanie zmiennych referencyjnych do obiektów 133

Wprowadzenie do metod 134

Dodanie metody do klasy Box 134

Zwracanie wartości 136

Dodanie metody przyjmującej parametry 137

Konstruktor 139

Konstruktor sparametryzowany 140

Słowo kluczowe this 141

Ukrywanie zmiennych składowych 141

Mechanizm odzyskiwania pamięci 142

Klasa stosu 142

7. Dokładniejsze omówienie metod i klas 145

Przeciążanie metod 145

Przeciążanie konstruktorów 147

Obiekty jako parametry 149

Dokładniejsze omówienie przekazywania argumentów 151

Zwracanie obiektów 152

Rekurencja 153

Wprowadzenie do kontroli dostępu 155

Składowe statyczne 158

Słowo kluczowe final 159

Powtórka z tablic 160

Klasy zagnieżdżone i klasy wewnętrzne 161

Omówienie klasy String 164

Wykorzystanie argumentów wiersza poleceń 165

Zmienna liczba argumentów 166

Przeciążanie metod o zmiennej liczbie argumentów 169

Zmienna liczba argumentów i niejednoznaczności 170

Stosowanie wnioskowania typów zmiennych lokalnych z typami referencyjnymi 171

8. Dziedziczenie 173

Podstawy dziedziczenia 173

Dostęp do składowych a dziedziczenie 174

Bardziej praktyczny przykład 175

Zmienna klasy bazowej może zawierać referencję do obiektu klasy pochodnej 177

Słowo kluczowe super 178

Wykorzystanie słowa kluczowego super do wywołania konstruktora klasy bazowej 178

Drugie zastosowanie słowa kluczowego super 181

Tworzenie hierarchii wielopoziomowej 182

Kiedy są wykonywane konstruktory? 184

Przesłanie metod 185

Dynamiczne przydzielanie metod 187

Dlaczego warto przesłaniać metody? 188

Zastosowanie przesłaniania metod 188

Klasy abstrakcyjne 190

Słowo kluczowe final i dziedziczenie 192

Słowo kluczowe final zapobiega przesłanianiu 192

Słowo kluczowe final zapobiega dziedziczeniu 193

Wnioskowanie typów zmiennych lokalnych a dziedziczenie 193

Klasa Object 195

9. Pakiety i interfejsy 197

Pakiety 197

Definiowanie pakietu 197

Znajdowanie pakietów i ścieżka CLASSPATH 198

Prosty przykład pakietu 199

Dostęp do pakietów i składowych 199

Przykład dostępu 200

Import pakietów 203

Interfejsy 204

Definiowanie interfejsu 205

Implementacja interfejsu 206

Interfejsy zagnieżdżone 208

Stosowanie interfejsów 209

Zmienne w interfejsach 211

Interfejsy można rozszerzać 213

Metody domyślne 214

Podstawy metod domyślnych 215

Bardziej praktyczny przykład 216

Problemy wielokrotnego dziedziczenia 217

Metody statyczne w interfejsach 217

Stosowanie metod prywatnych w interfejsach 218

Ostatnie uwagi dotyczące pakietów i interfejsów 219

10. Obsługa wyjątków 221

Podstawy obsługi wyjątków 221

Typy wyjątków 222

Nieprzechwycone wyjątki 222

Stosowanie instrukcji try i catch 223

Wyświetlenie opisu wyjątku 224

Wiele klauzul catch 225

Zagnieżdżone instrukcje try 226

Instrukcja throw 228

Klauzula throws 229

Słowo kluczowe finally 229

Wyjątki wbudowane w język Java 231

Tworzenie własnej klasy pochodnej wyjątków 231

Łącuch wyjątków 234

Trzy dodatkowe cechy wyjątków 235

Wykorzystanie wyjątków 236

11. Programowanie wielowątkowe 237

Model wątków języka Java 238

Priorytety wątków 239

Synchronizacja 239

Przekazywanie komunikatów 240

Klasa Thread i interfejs Runnable 240

Wątek główny 240

Tworzenie wątku 242

Implementacja interfejsu Runnable 242

Rozszerzanie klasy Thread 244

Wybór odpowiedniego podejścia 244

Tworzenie wielu wątków 245

Stosowanie metod isAlive() i join() 246

Priorytety wątków 248

Synchronizacja 249

Synchronizacja metod 249

Instrukcja synchronized 251

Komunikacja międzywątkowa 252

Zakleszczenie 256

Zawieszanie, wznawianie i zatrzymywanie wątków 258

Uzyskiwanie stanu wątku 260

Stosowanie metody wytwórczej do tworzenia i uruchamiania wątku 261

Korzystanie z wielowątkowości 262

12. Wyliczenia, automatyczne opakowywanie typów prostych i adnotacje 263

Typy wyliczeniowe 263

Podstawy wyliczeń 263

Metody values() i valueOf() 265

Wyliczenia Javy jako typy klasowe 266

Wyliczenia dziedziczą po klasie Enum 268

Inny przykład wyliczenia 269

Opakowania typów 271

Klasa Character 271

Klasa Boolean 271

Opakowania typów numerycznych 272

Automatyczne opakowywanie typów prostych 273

Automatyczne opakowywanie i metody 274

Automatyczne opakowywanie i rozpakowywanie w wyrażeniach 274

Automatyczne opakowywanie typów znakowych i logicznych 276

Automatyczne opakowywanie pomaga zapobiegać błędom 276

Słowo ostrzeżenia 277

Adnotacje 277

Podstawy tworzenia adnotacji 278

Określanie strategii zachowywania adnotacji 278

Odczytywanie adnotacji w trakcie działania programu za pomocą refleksji 279

Interfejs AnnotatedElement 283

Wartości domyślne 283

Adnotacje znacznikowe 285

Adnotacje jednoelementowe 285

Wbudowane adnotacje 287

Adnotacje typów 288

Adnotacje powtarzalne 292

Ograniczenia 294

13. Wejście-wyjście, instrukcja try z zasobami i inne tematy 295

Podstawowa obsługa wejścia i wyjścia 295

Strumienie 296

Strumienie znakowe i bajtowe 296

Predefiniowane strumienie 298

Odczyt danych z konsoli 298

Odczyt znaków 298

Odczyt łańcuchów 299

Wyświetlanie informacji na konsoli 301

Klasa PrintWriter 301

Odczyt i zapis plików 302

Automatyczne zamykanie pliku 307

Modyfikatory transient i volatile 310

Operator instanceof 311

Modyfikator strictfp 313

Metody napisane w kodzie rdzennym 313

Stosowanie asercji 313

Opcje włączania i wyłączania asercji 315

Import statyczny 316

Wywoływanie przeciążonych konstruktorów za pomocą this() 318

Kilka słów o kompaktowych profilach API 320

14. Typy sparametryzowane 321

Czym są typy sparametryzowane? 321

Prosty przykład zastosowania typów sparametryzowanych 322

Typy sparametryzowane działają tylko dla typów referencyjnych 325

Typy sparametryzowane różnią się, jeśli mają inny argument typu 325

W jaki sposób typy sparametryzowane zwiększają bezpieczeństwo? 325

Klasa sparametryzowana z dwoma parametrami typu 327

Ogólna postać klasy sparametryzowanej 328

Typy ograniczone 328

Zastosowanie argumentów wieloznacznych 330

Ograniczony argument wieloznaczny 333

Tworzenie metody sparametryzowanej 337

Konstruktory sparametryzowane 338

Interfejsy sparametryzowane 339

Typy surowe i starszy kod 341

Hierarchia klas sparametryzowanych 343

Zastosowanie sparametryzowanej klasy bazowej 343

Podklasa sparametryzowana 345

Porównywanie typów w hierarchii klas sparametryzowanych w czasie wykonywania 346

Rzutowanie 348

Przesłanianie metod w klasach sparametryzowanych 348

Wnioskowanie typów a typy sparametryzowane 349

Wnioskowanie typów zmiennych lokalnych a typy sparametryzowane 350

Znoszenie 350

Metody mostu 351

Błędy niejednoznaczności 352

Pewne ograniczenia typów sparametryzowanych 353

Nie można tworzyć egzemplarza parametru typu 353

Ograniczenia dla składowych statycznych 354

Ograniczenia tablic typów sparametryzowanych 354

Ograniczenia wyjątków typów sparametryzowanych 355

15. Wyrażenia lambda 357

Wprowadzenie do wyrażeń lambda 357

Podstawowe informacje o wyrażeniach lambda 358

Interfejsy funkcyjne 358

Kilka przykładów wyrażeń lambda 359

Blokowe wyrażenia lambda 362

Sparametryzowane interfejsy funkcyjne 364

Przekazywanie wyrażeń lambda jako argumentów 365

Wyrażenia lambda i wyjątki 368

Wyrażenia lambda i przechwytywanie zmiennych 369

Referencje do metod 370

Referencje do metod statycznych 370

Referencje do metod instancyjnych 371

Referencje do metod a typy sparametryzowane 374

Referencje do konstruktorów 376

Predefiniowane interfejsy funkcyjne 380

16. Moduły 383

Podstawowe informacje o modułach 383

Przykład prostego modułu 384

Kompilowanie i uruchamianie przykładowej aplikacji 388

Dokładniejsze informacje o instrukcjach requires i exports 389

java.base i moduły platformy 390

Stary kod i moduł nienazwany 390

Eksportowanie do konkretnego modułu 391

Wymagania przechodnie 392

Stosowanie usług 396

Podstawowe informacje o usługach i dostawcach usług 396

Słowa kluczowe związane z usługami 397

Przykład stosowania usług i modułów 397

Grafy modułów 403

Trzy wyspecjalizowane cechy modułów 404

Moduły otwarte 404

Instrukcja opens 404

Instrukcja requires static 404

Wprowadzenie do jlink i plików JAR modułów 405

Dołączanie plików dostarczonych jako struktura katalogów 405

Konsolidacja modularnych plików JAR 405

Pliki JMOD 406

Kilka słów o warstwach i modułach automatycznych 406

Końcowe uwagi dotyczące modułów 407

CZĘŚĆ II. BIBLIOTEKA JĘZYKA JAVA

17. Obsługa łańcuchów 411

Konstruktory klasy String 411

Długość łańcucha 413

Specjalne operacje na łańcuchach 413

Literały tekstowe 413

Konkatenacja łańcuchów 414

Konkatenacja łańcuchów z innymi typami danych 414

Konwersja łańcuchów i metoda toString() 415

Wyodrębnianie znaków 416

Metoda charAt() 416

Metoda getChars() 416

Metoda getBytes() 416

Metoda toCharArray() 417

Porównywanie łańcuchów 417

Metody equals() i equalsIgnoreCase() 417

Metoda regionMatches() 418

Metody `startsWith()` i `endsWith()` 418

Metoda `equals()` kontra operator `==` 418

Metoda `compareTo()` 419

Przeszukiwanie łańcuchów 420

Modyfikowanie łańcucha 421

Metoda `substring()` 421

Metoda `concat()` 422

Metoda `replace()` 422

Metody `trim()` i `strip()` 423

Konwersja danych za pomocą metody `valueOf()` 424

Zmiana wielkości liter w łańcuchu 424

Łączenie łańcuchów 425

Dodatkowe metody klasy `String` 425

Klasa `StringBuffer` 426

Konstruktory klasy `StringBuffer` 427

Metody `length()` i `capacity()` 427

Metoda `ensureCapacity()` 427

Metoda `setLength()` 428

Metody `charAt()` i `setCharAt()` 428

Metoda `getChars()` 428

Metoda `append()` 429

Metoda `insert()` 429

Metoda `reverse()` 430

Metody `delete()` i `deleteCharAt()` 430

Metoda `replace()` 431

Metoda `substring()` 431

Dodatkowe metody klasy `StringBuffer` 431

Klasa `StringBuilder` 432

18. Pakiet `java.lang` 433

Opakowania typów prostych 433

Klasa `Number` 434

Klasy `Double` i `Float` 434

Metody `isInfinite()` i `isNaN()` 437

Klasy `Byte`, `Short`, `Integer` i `Long` 437

Klasa `Character` 445

Dodatki wprowadzone w celu obsługi punktów kodowych Unicode 446

Klasa `Boolean` 449

Klasa `Void` 449

Klasa `Process` 450

Klasa `Runtime` 451

Zarządzanie pamięcią 452

Wykonywanie innych programów 453

`Runtime.Version` 453

Klasa `ProcessBuilder` 455

Klasa `System` 457

Wykorzystanie metody `currentTimeMillis()` do obliczania czasu wykonywania programu 458

Użycie metody `arraycopy()` 459

Właściwości środowiska 459

Interfejs `System.Logger` i klasa `System.LoggerFinder` 460

Klasa `Object` 460

Wykorzystanie metody `clone()` i interfejsu `Cloneable` 460

Klasa `Class` 462

Klasa `ClassLoader` 465

Klasa `Math` 465

Funkcje trygonometryczne 465

Funkcje wykładnicze 466

Funkcje zaokrąglenia 466

Inne metody klasy Math 467

Klasa StrictMath 469

Klasa Compiler 469

Klasy Thread i ThreadGroup oraz interfejs Runnable 469

Interfejs Runnable 469

Klasa Thread 470

Klasa ThreadGroup 472

Klasy ThreadLocal i InheritableThreadLocal 475

Klasa Package 475

Klasa Module 476

Klasa ModuleLayer 477

Klasa RuntimePermission 477

Klasa Throwable 477

Klasa SecurityManager 477

Klasa StackTraceElement 477

Klasa StackWalker i interfejs StackWalker.StackFrame 478

Klasa Enum 478

Klasa ClassValue 479

Interfejs CharSequence 479

Interfejs Comparable 480

Interfejs Appendable 480

Interfejs Iterable 480

Interfejs Readable 481

Interfejs AutoCloseable 481

Interfejs Thread.UncaughtExceptionHandler 481

Podpakiety pakietu java.lang 481

Podpakiet java.lang.annotation 482

Podpakiet java.lang.instrument 482

Podpakiet java.lang.invoke 482

Podpakiet java.lang.management 482

Podpakiet java.lang.module 482

Podpakiet java.lang.ref 482

Podpakiet java.lang.reflect 482

19. Pakiet java.util, część 1. - kolekcje 483

Wprowadzenie do kolekcji 484

Interfejsy kolekcji 485

Interfejs Collection 486

Interfejs List 488

Interfejs Set 489

Interfejs SortedSet 490

Interfejs NavigableSet 490

Interfejs Queue 491

Interfejs Deque 492

Klasy kolekcji 493

Klasa ArrayList 494

Klasa LinkedList 497

Klasa HashSet 498

Klasa LinkedHashSet 499

Klasa TreeSet 499

Klasa PriorityQueue 501

Klasa ArrayDeque 501

Klasa EnumSet 502

Dostęp do kolekcji za pomocą iteratora 502

Korzystanie z iteratora Iterator 504

Pętla typu for-each jako alternatywa dla iteratora 505

Splitatory 506

Przechowywanie w kolekcjach własnych klas 509

Interfejs RandomAccess 510

Korzystanie z map 510

Interfejsy map 510

Klasy map 516

Komparatory 520

Wykorzystanie komparatora 522

Algorytmy kolekcji 526

Klasa Arrays 531

Starsze klasy i interfejsy 535

Interfejs Enumeration 535

Klasa Vector 536

Klasa Stack 539

Klasa Dictionary 540

Klasa Hashtable 541

Klasa Properties 544

Wykorzystanie metod store() i load() 547

Ostatnie uwagi na temat kolekcji 548

20. Pakiet java.util, część 2. - pozostałe klasy użytkowe 549

Klasa StringTokenizer 549

Klasa BitSet 551

Klasy Optional, OptionalDouble, OptionalInt oraz OptionalLong 553

Klasa Date 556

Klasa Calendar 557

Klasa GregorianCalendar 560

Klasa TimeZone 561

Klasa SimpleTimeZone 562

Klasa Locale 563

Klasa Random 564

Klasy Timer i TimerTask 566

Klasa Currency 568

Klasa Formatter 569

Konstruktory klasy Formatter 570

Metody klasy Formatter 570

Podstawy formatowania 570

Formatowanie łańcuchów i znaków 573

Formatowanie liczb 573

Formatowanie daty i godziny 574

Specyfikatory %n i %% 575

Określanie minimalnej szerokości pola 576

Określanie precyzji 577

Używanie znaczników (flag) formatów 578

Wyrównywanie danych wyjściowych 578

Znaczniki spacji, plusa, zera i nawiasów 579

Znacznik przecinka 580

Znacznik # 580

Opcja wielkich liter 580

Stosowanie indeksu argumentu 581

Zamykanie obiektu klasy Formatter 582

Metoda printf() w Javie 582

Klasa Scanner 582

Konstruktory klasy Scanner 583

Podstawy skanowania 584

Kilka przykładów użycia klasy Scanner 587

Ustawianie separatorów 590

Pozostałe elementy klasy Scanner 591

Klasy ResourceBundle, ListResourceBundle i PropertyResourceBundle 592

Dodatkowe klasy i interfejsy użytkowe 596

Podpakiety pakietu java.util 597

java.util.concurrent, java.util.concurrent.atomic oraz java.util.concurrent.locks 598

java.util.function 598

java.util.jar 600

java.util.logging 600

java.util.prefs 600

java.util.regex 600

java.util.spi 601

java.util.stream 601

java.util.zip 601

21. Operacje wejścia-wyjścia: analiza pakietu java.io 603

Klasy i interfejsy obsługujące operacje wejścia-wyjścia 604

Klasa File 604

Katalogi 607

Stosowanie interfejsu FilenameFilter 608

Alternatywna metoda listFiles() 609

Tworzenie katalogów 609

Interfejsy AutoCloseable, Closeable i Flushable 609

Wyjątki operacji wejścia-wyjścia 610

Dwa sposoby zamykania strumieni 610

Klasy strumieni 611

Strumienie bajtów 612

Klasa InputStream 612

Klasa OutputStream 613

Klasa FileInputStream 613

Klasa FileOutputStream 615

Klasa ByteArrayInputStream 617

Klasa ByteArrayOutputStream 618

Filtrowane strumienie bajtów 620

Buforowane strumienie bajtów 620

Klasa SequenceInputStream 623

Klasa PrintStream 625

Klasy DataOutputStream i DataInputStream 627

Klasa RandomAccessFile 628

Strumienie znaków 629

Klasa Reader 629

Klasa Writer 629

Klasa FileReader 629

Klasa FileWriter 631

Klasa CharArrayReader 632

Klasa CharArrayWriter 633

Klasa BufferedReader 634

Klasa BufferedWriter 636

Klasa PushbackReader 636

Klasa PrintWriter 637

Klasa Console 638

Serializacja 639

Interfejs Serializable 640

Interfejs Externalizable 640

Interfejs ObjectOutputStream 640

Klasa ObjectOutputStream 641

Interfejs ObjectInput 642

Klasa ObjectInputStream 642

Przykład serializacji 643

Korzyści wynikające ze stosowania strumieni 645

22. System NIO 647

Klasy systemu NIO 647

Podstawy systemu NIO 648

Bufory 648

Kanały 648

Zestawy znaków i selektory 651

Udoskonalenia dodane w systemie NIO.2 651

Interfejs Path 651

Klasa Files 651

Klasa Paths 654

Interfejsy atrybutów plików 655

Klasy FileSystem, FileSystems i FileStore 657

Stosowanie systemu NIO 657

Stosowanie systemu NIO dla operacji wejścia-wyjścia na kanałach 658

Stosowanie systemu NIO dla operacji wejścia-wyjścia na strumieniach 666

Stosowanie systemu NIO dla operacji na ścieżkach i systemie plików 668

23. Obsługa sieci 675

Podstawy działania sieci 675

Klasy i interfejsy pakietu java.net obsługujące komunikację siecią 676

Klasa InetAddress 677

Metody wytwórcze 677

Metody klasy 678

Klasy InetAddress oraz Inet6Address 679

Gniazda klientów TCP/IP 679

URL 682

Klasa URLConnection 683

Klasa HttpURLConnection 685

Klasa URI 687

Pliki cookie 687

Gniazda serwerów TCP/IP 688

Datagramy 688

Klasa DatagramSocket 689

Klasa DatagramPacket 689

Przykład użycia datagramów 690

Prezentacja pakietu java.net.http 692

Trzy kluczowe elementy 692

Prosty przykład użycia API klienta HTTP 695

Czego jeszcze warto dowiedzieć się o pakiecie java.net.http? 696

24. Obsługa zdarzeń 697

Dwa mechanizmy obsługi zdarzeń 697

Model obsługi zdarzeń oparty na ich delegowaniu 698

Zdarzenia 698

Źródła zdarzeń 698

Obiekty nasłuchujące zdarzeń 699

Klasy zdarzeń 699

Klasa ActionEvent 701

Klasa AdjustmentEvent 701

Klasa ComponentEvent 702

Klasa ContainerEvent 702

Klasa FocusEvent 703

Klasa InputEvent 704

Klasa ItemEvent 704

Klasa KeyEvent 705

Klasa MouseEvent 706

Klasa MouseWheelEvent 707

Klasa TextEvent 708

Klasa WindowEvent 708

Źródła zdarzeń 709

Interfejsy nasłuchujące zdarzeń 710

Interfejs ActionListener 711

Interfejs AdjustmentListener 711

Interfejs ComponentListener 711

Interfejs ContainerListener 711

Interfejs FocusListener 711

Interfejs ItemListener 711

Interfejs KeyListener 711

Interfejs MouseListener 712

Interfejs MouseMotionListener 712

Interfejs MouseWheelListener 712

Interfejs TextListener 712

Interfejs WindowFocusListener 712

Interfejs WindowListener 712

Stosowanie modelu delegowania zdarzeń 713

Kluczowe zagadnienia tworzenia aplikacji graficznych z użyciem AWT 713

Obsługa zdarzeń generowanych przez mysz 714

Obsługa zdarzeń generowanych przez klawiaturę 717

Klasy adapterów 720

Klasy wewnętrzne 722

Anonimowa klasa wewnętrzna 724

25. Wprowadzenie do AWT: praca z oknami, grafiką i tekstem 727

Klasy AWT 728

Podstawy okien 730

Klasa Component 730

Klasa Container 730

Klasa Panel 730

Klasa Window 731

Klasa Frame 731

Klasa Canvas 731

Praca z oknami typu Frame 731

Ustawianie wymiarów okna 731

Ukrywanie i wyświetlanie okna 732

Ustawianie tytułu okna 732

Zamykanie okna typu Frame 732

Metoda paint() 732

Wyświetlanie łańcuchów znaków 732

Określanie koloru tekstu i tła 733

Żądanie ponownego wyświetlenia zawartości okna 733

Tworzenie aplikacji korzystających z klasy Frame 734

Wprowadzenie do stosowania grafiki 735

Rysowanie odcinków 735

Rysowanie prostokątów 735

Rysowanie elips, kół i okręgów 736

Rysowanie łuków 736

Rysowanie wielokątów 736

Prezentacja metod rysujących 737

Dostosowywanie rozmiarów obiektów graficznych 738

Praca z klasą Color 739

Metody klasy Color 740

Ustawianie bieżącego koloru kontekstu graficznego 741

Program demonstrujący zastosowanie klasy Color 741

Ustawianie trybu rysowania 742

Praca z czcionkami 743

Określanie dostępnych czcionek 745

Tworzenie i wybieranie czcionek 746

Uzyskiwanie informacji o czcionkach 748

Zarządzanie tekstowymi danymi wyjściowymi z wykorzystaniem klasy FontMetrics 749

26. Stosowanie kontrolek AWT, menedżerów układu graficznego oraz menu 753

Podstawy kontrolek AWT 754

Dodawanie i usuwanie kontrolek 754

Odpowiadanie na zdarzenia kontrolek 754

Wyjątek HeadlessException 755

Etykiety 755

Stosowanie przycisków 756

Obsługa zdarzeń przycisków 757

Stosowanie pól wyboru 760

Obsługa zdarzeń pól wyboru 761

Klasa CheckboxGroup 762

Kontrolki list rozwijanych 764

Obsługa zdarzeń list rozwijanych 765

Stosowanie list 766

Obsługa zdarzeń generowanych przez listy 767

Zarządzanie paskami przewijania 769

Obsługa zdarzeń generowanych przez paski przewijania 770

Stosowanie kontrolek typu TextField 772

Obsługa zdarzeń generowanych przez kontrolkę TextField 773

Stosowanie kontrolek typu TextArea 774

Wprowadzenie do menedżerów układu graficznego komponentów 776

FlowLayout 777

BorderLayout 778

Stosowanie obramowań 779

GridLayout 781

Klasa CardLayout 782

Klasa GridBagLayout 785

Menu i paski menu 789

Okna dialogowe 794

Przesłanianie metody paint() 797

27. Obrazy 799

Formaty plików 799

Podstawy przetwarzania obrazów: tworzenie, wczytywanie i wyświetlanie 800

Tworzenie obiektu obrazu 800

Ładowanie obrazu 800

Wyświetlanie obrazu 801

Podwójne buforowanie 802

Interfejs ImageProducer 804

Klasa MemoryImageSource 805

Interfejs ImageConsumer 806

Klasa PixelGrabber 806

Klasa ImageFilter 809

Klasa CropImageFilter 809

Klasa RGBImageFilter 810

Dodatkowe klasy obsługujące obrazy 821

28. Narzędzia współbieżności 823

Pakiety interfejsu Concurrent API 824

Pakiet java.util.concurrent 824

Pakiet java.util.concurrent.atomic 825

Pakiet java.util.concurrent.locks 825

Korzystanie z obiektów służących do synchronizacji 825

Klasa Semaphore 825

Klasa CountdownLatch 830

CyclicBarrier 832

Klasa Exchanger 834

Klasa Phaser 835

Korzystanie z egzekutorów 842

Przykład prostego egzekutora 842

Korzystanie z interfejsów Callable i Future 844

Typ wyliczeniowy TimeUnit 846

Kolekcje współbieżne 847

Blokady 847

Operacje atomowe 850

Programowanie równoległe przy użyciu frameworku Fork/Join 851

Najważniejsze klasy frameworku Fork/Join 852

Strategia dziel i zwyciężaj 855

Prosty przykład użycia frameworku Fork/Join 855

Znaczenie poziomu równoległości 858

Przykład użycia klasy RecursiveTask 860

Asynchroniczne wykonywanie zadań 862

Anulowanie zadania 863

Określanie statusu wykonania zadania 863

Ponowne uruchamianie zadania 863

Pozostałe zagadnienia 863

Wskazówki dotyczące stosowania frameworku Fork/Join 865

Pakiet Concurrency Utilities a tradycyjne metody języka Java 866

29. API strumieni 867

Podstawowe informacje o strumieniach 867

Interfejsy strumieni 868

Jak można uzyskać strumień? 870

Prosty przykład stosowania strumieni 871

Operacje redukcji 874

Stosowanie strumieni równoległych 876

Odwzorowywanie 878

Tworzenie kolekcji 882

Iteratory i strumienie 885

Stosowanie typu Iterator i strumieni 885

Stosowanie spliteratorów 886

Inne możliwości API strumieni 889

30. Wyrażenia regularne i inne pakiety 891

Przetwarzanie wyrażeń regularnych 891

Klasa Pattern 892

Klasa Matcher 892

Składnia wyrażeń regularnych 893

Przykład dopasowywania do wzorca 893

Dwie opcje dopasowywania do wzorca 898

Przegląd wyrażeń regularnych 898

Refleksje 898

Zdalne wywoływanie metod (RMI) 901

Prosta aplikacja typu klient-serwer wykorzystująca RMI 902

Formatowanie dat i czasu przy użyciu pakietu java.text 905

Klasa DateFormat 905

Klasa SimpleDateFormat 906

Interfejs API dat i czasu - java.time 908

Podstawowe klasy do obsługi dat i czasu 908

Formatowanie dat i godzin 910

Analiza łańcuchów zawierających daty i godziny 912

Inne możliwości pakietu java.time 913

CZĘŚĆ III. WPROWADZENIE DO PROGRAMOWANIA GUI PRZY UŻYCIU PAKIETU SWING

31. Wprowadzenie do pakietu Swing 917

Geneza powstania biblioteki Swing 917

Bibliotekę Swing zbudowano na bazie zestawu narzędzi AWT 918

Podstawowe cechy biblioteki Swing 918

Komponenty biblioteki Swing są lekkie 918

Biblioteka Swing obsługuje dołączany wygląd i sposób obsługi 919

Podobieństwo do architektury MVC 919

Komponenty i kontenery 920

Komponenty 920

Kontenery 921

Panele kontenerów najwyższego poziomu 921

Pakiety biblioteki Swing 922

Prosta aplikacja na bazie biblioteki Swing 922

Obsługa zdarzeń 926

Rysowanie w bibliotece Swing 929

Podstawy rysowania 929

Wyznaczanie obszaru rysowania 930

Przykład rysowania 930

32. Przewodnik po pakiecie Swing 933

Klasy JLabel i ImageIcon 933

Klasa JTextField 935

Przyciski biblioteki Swing 936

Klasa JButton 937

Klasa JToggleButton 939

Pola wyboru 940

Przyciski opcji 942

Klasa JTabbedPane 944

Klasa JScrollPane 946

Klasa JList 948

Klasa JComboBox 951

Drzewa 953

Klasa JTable 955

33. Wprowadzenie do systemu menu pakietu Swing 959

Podstawy systemu menu 959

Przegląd klas JMenuBar, JMenu oraz JMenuItem 961

Klasa JMenuBar 961

Klasa JMenu 962

Klasa JMenuItem 963

Tworzenie menu głównego 963

Dodawanie mnemonik i kombinacji klawiszy do opcji menu 967

Dodawanie obrazów i etykiet ekranowych do menu 969

Stosowanie klas JRadioButtonMenuItem i JCheckBoxMenuItem 970

Tworzenie menu podręcznych 972

Tworzenie paska narzędzi 974

Stosowanie akcji 977

Finalna postać programu MenuDemo 981

Dalsze poznawanie pakietu Swing 987

CZĘŚĆ IV. STOSOWANIE JAVY W PRAKTYCE

34. Java Beans 991

Czym jest komponent typu Java Bean? 991

Zalety komponentów Java Beans 992

Introspekcja 992

Wzorce właściwości 992

Wzorce projektowe dla zdarzeń 993

Metody i wzorce projektowe 994

Korzystanie z interfejsu BeanInfo 994

Właściwości ograniczone 994

Trwałość 995

Interfejs Customizer 995

Interfejs Java Beans API 995

Klasa Introspector 997

Klasa PropertyDescriptor 997

Klasa EventSetDescriptor 997

Klasa MethodDescriptor 997

Przykład komponentu Java Bean 997

35. Serwlety 1001

Podstawy 1001

Cykl życia serwletu 1002

Sposoby tworzenia serwletów 1002

Korzystanie z serwera Tomcat 1003

Przykład prostego serwletu 1004

Tworzenie i kompilacja kodu źródłowego serwletu 1004

Uruchamianie serwera Tomcat 1005

Uruchamianie przeglądarki i generowanie żądania 1005

Interfejs Servlet API 1005

Pakiet javax.servlet 1005

Interfejs Servlet 1006

Interfejs ServletConfig 1006

Interfejs ServletContext 1007

Interfejs ServletRequest 1007

Interfejs ServletResponse 1007

Klasa GenericServlet 1007

Klasa ServletInputStream 1007

Klasa ServletOutputStream 1009

Klasy wyjątków związanych z serwletami 1009

Odczytywanie parametrów serwletu 1009

Pakiet javax.servlet.http 1010

Interfejs HttpServletRequest 1011

Interfejs HttpServletResponse 1011

Interfejs HttpSession 1011

Klasa Cookie 1013

Klasa HttpServlet 1014

Obsługa żądań i odpowiedzi HTTP 1014

Obsługa żądań GET protokołu HTTP 1014

Obsługa żądań POST protokołu HTTP 1016

Korzystanie ze znaczników kontekstu użytkownika 1017

Śledzenie sesji 1019

DODATKI

A. Komentarze dokumentujące 1023

Znaczniki narzędzia javadoc 1023

Znacznik @author 1024

Znacznik {@code} 1024

Znacznik @deprecated 1025

Znacznik {@docRoot} 1025

Znacznik @exception 1025

Znacznik @hidden 1025

Znacznik {@index} 1025

Znacznik {@inheritDoc} 1025

Znacznik {@link} 1025

Znacznik {@linkplain} 1026

Znacznik {@literal} 1026

Znacznik @param 1026

Znacznik @provides 1026

Znacznik @return 1026

Znacznik @see 1026

Znacznik @serial 1026

Znacznik @serialData 1027

Znacznik @serialField 1027

Znacznik @since 1027

Znacznik {@summary} 1027

Znacznik @throws 1027

Znacznik @uses 1027

Znacznik {@value} 1027

Znacznik @version 1028

Ogólna postać komentarzy dokumentacyjnych 1028

Wynik działania narzędzia javadoc 1028

Przykład korzystający z komentarzy dokumentacyjnych 1028

B. Wprowadzenie do JShell 1031

Podstawy JShell 1031

Wyświetlanie, edytowanie i ponowne wykonywanie kodu 1033

Dodanie metody 1034

Utworzenie klasy 1035

Stosowanie interfejsu 1036

Przetwarzanie wyrażeń i wbudowane zmienne 1037

Importowanie pakietów 1037

Wyjątki 1038

Inne polecenia JShell 1038

Dalsze poznawanie możliwości JShell 1039

C. Kompilowanie i uruchamianie prostych programów w jednym kroku 1041