

Index

0–9

2D shapes, 281

A

abstract

- class, 250, 254, 255, 261, 359, 587
- method, 253, 254, 268, 359, 360, 361, 369, 370, 371, 374, 378, 383, 385, 412, 413, 436, 447, 458, 460, 482, 588, 649, 652, 656, 692, 693, 694

Abstract Windows Toolkit (AWT), 266

actors - use case model, 604

actual parameters, 95, 101, 110, 116

aggregation, 211, 310

airport case study, *see* case study, airport

Alert class, 518, 520

algorithms

- scheduling, 582

aliasing, 694, 695, 697

animations, 591, 593, 596, 597, 600

anonymous class, 357, 364–368, 378, 391, 587, 591, 597, 598

API, *see* Application Programming Interface

append mode - files, 535

Application Programming Interface (API), 369, 453, 641

applications

- deploying, 563

JavaFX, 267–269, 302, 310, 336, 337, 413–415, 496, 541, 553, 554, 577, 587, 599, 677, 680, 682

running from command line, 268, 415, 553, 560, 561, 575

application software, 3, 4, 16

arithmetic operators, 25, 26

array(s)

- accessing elements, 119, 124, 126, 187, 394
- creating, 120, 148, 149, 186

elements, 120–122, 124–126, 137, 139, 142, 185, 210, 232

index, 125–127, 137, 139, 143, 150, 154, 186, 395

length attribute, 151, 156, 198, 200–202

maximum, 122, 139, 140

of objects, 163, 185, 186, 191

returning from a method, 134

membership, 141

passing as parameters, 101, 129, 131

ragged, 119, 155, 156, 160, 162

search, 142, 143, 451

sort, 457, 458, 463

summation, 141

two-dimensional, 119, 148–151, 153, 155, 156, 160, 232

varargs, 131–134, 138, 159, 161

ArrayIndexOutOfBoundsException, 394, 395, 405, 422

ArrayList class, 188, 193, 211, 315, 317, 553, 376, 427–430, 450, 465

Arrays class, 456, 465, 647

assignment, 23–28, 34, 46, 103, 124–126, 200

asynchronous threads, 577, 599

attributes

class, 109, 165, 224, 227, 236, 237, 328, 376, 477, 570, 608, 701

length, 151, 156, 198, 199, 201, 202, 241

private, 195, 262, 697, 698

protected, 238, 262

public, 195, 238, 262, 315

static, 206, 676

autoboxing, 259

AWT, *see* Abstract Windows Toolkit

B

base class, 237, 394, 692

behaviour specifications, 603–605, 624

- BiConsumer interface, 447
- BiFunction interface, 378
- binary encoding, 530
- binary files – reading and writing, 539, 540
- BinaryOperator interface, 378
- BiPredicate interface, 378
- boolean** type, 20, 122
- BorderPane class, 292, 293, 303
- BorderStroke class, 289, 294, 344, 346, 629, 631
- BorderStrokeStyle class, 289, 294, 344, 346, 629, 631
- BorderWidths class, 289, 294, 344, 346, 629, 631
- bounded type parameters, 379
- break** statement, 56–59, 61, 65, 84–86
- BufferedReader class, 529, 537, 538, 542
- bugs, *see* errors
- busy waiting, 577, 585, 599
- byte** type, 20

- C**
- C ++, 4, 164, 208, 690, 691, 703, 705, 706
- C# (C Sharp), 705, 706
- calling method, 100, 105, 191, 312, 401, 413
- card menu, 499, 513
- Cascading Style Sheets (CSS), 469, 491, 496, 639
- case** statement, 55–58, 113, 115, 615
- case study
 - airport, 605
 - student hostel, 308, 334–336, 350, 354
- catch**, 393, 401–406, 408, 421, 423, 540, 573, 667, 673
- catching an exception, 401, 422
- char** type, 22, 23
- Character class, 362
- check boxes, 267, 282, 499, 509, 511, 512
- checked exceptions, 395, 422
- ChoiceDialog class, 518, 520, 523, 639
- claiming an exception, 397, 398, 422
- class(es)
 - anonymous, 357, 364–368, 378, 391, 587, 591, 597, 598
 - attributes, 109, 165, 200, 203, 236–238, 312, 339, 469, 529, 550, 570, 593, 701
 - abstract, 235, 250, 251, 254, 255, 261, 359
 - collection, *see* collection classes
 - final, 257
 - inheritance extending with, 236, 246, 251, 260, 357, 359, 691
 - inner, 364, 391
 - methods, 116, 117, 170, 205, 208, 229, 238, 374, 396, 399, 402, 411, 412, 456, 457
 - class method, 205, 208, 229, 238, 396, 411, 412, 456, 457
 - classpath, 562, 564, 569, 574
 - client-server model, 661, 676
 - clone method, 689, 697–704, 707, 708
 - Cloneable interface, 702
 - code layout, 319
 - collection classes
 - generic, 188, 378, 383
 - Collection interface, 428, 435, 436, 645
 - combo boxes, 499, 507
 - context menu, 499, 503–506, 524
 - Color class, 272, 296
 - colours
 - creating, 646
 - command line running applications from, 268, 415, 553, 560–563, 574, 575
 - comments
 - Javadoc, 14, 307, 310, 313, 317, 318, 330, 332, 334, 421, 424, 609, 621, 635
 - Comparable interface, 458–460, 463, 465, 466
 - Comparator interface, 460, 461, 463, 465
 - comparison operators, 46, 446
 - compiling programs, 4
 - composition, 211, 603, 607, 638
 - compound containers, 301
 - concatenation operator, 15, 30
 - concatenating streams, 656
 - concurrent processes, 578
 - constants, creating, 25
 - constructor
 - copy, 689, 703, 704, 707, 708
 - default, 200, 201
 - user defined, 200, 237
 - containment, 606, 607, 638
 - context menus, 499, 503, 505, 524
 - continue (key word), 86, 401
 - convenience method, 279, 384, 470, 472, 475, 476, 482, 497, 503
 - Consumer interface, 413, 435, 436, 442, 652
 - CornerRadii class, 344, 515, 516, 635
 - creating streams, 646
 - critical sections, 584
 - CSS, *see* Cascading Style Sheets

- D**
- data types, 19–21, 163, 223, 565
- database, 353, 553, 564, 565, 567–575, 578, 642, 663

- DataInputStream class, 540, 664–666, 669, 676, 682
- DataOutputStream class, 539, 545, 664, 677, 685
- DecimalFormat class, 297, 301, 344, 555
- declaring variables, 21, 35
- decrement operator, 29
- default methods, 360, 361, 693, 694
- deploying applications, 563
- derived class, 237
- design
 - case study airport, 607, 625, 639
 - case study student hostel, 309, 310, 335, 336
 - program, 35
- Dialog class, 348, 499, 518, 520, 523, 671, 680
- diamond problem, 692, 693
- Driver class, 564
- DriverManager class, 567
- double colon operator, 374
- documentation, 317, 318, 330, 425, 453, 482, 562, 564, 567, 575, 604
- dot operator, 169, 172, 186
- double** type, 21, 23, 24, 179
- Double class, 287
- do...while** loop, 79–84, 90
- driver(s)
 - JDBC, 564
- drop-down menus, 499, 500

- E**
- embedding
 - images, 483
 - videos, 486
 - webpages, 469
- embedded software, 4, 5, 16
- encapsulation, 169, 195–197, 224, 238
- encoding files, 527, 530
- enum**, 609, 610
- enumerated types
 - implementing in Java, 603, 609, 622
 - using with switch statements, 610, 615
- enhanced **for** loop, 119, 138, 139, 141, 143, 158, 187, 190, 316, 331, 427, 434–436, 438–440, 446, 447, 464, 623, 639
- environment variable, 562, 574
- equals method
 - defining, 450
 - String class, 433, 445
 - Object class, 450
- errors
 - compile-time errors, 564
 - runtime errors, 564, 569
- event-handling
 - key events, 477
 - mouse events, 470
- exception(s)
 - catching, 401, 403, 422
 - checked, 393, 395, 398, 422, 587, 622, 702
 - claiming, 397, 398, 422
 - documenting, 421, 422
 - exception classes creating, 419
 - exception class hierarchy, 395
 - in JavaFX applications, 415
 - handling, 393, 395, 414, 544, 694, 705
 - IOException, 394
 - NullPointerException, 408, 445
 - RuntimeException, 398, 400, 417–419
 - throwing, 394, 401, 417, 422, 424
 - unchecked, 393, 395, 422–424
- executable JAR files, 563
- expressions, 26, 27, 29, 30, 105, 106, 279, 280, 341, 357, 358, 368–371, 374, 387, 652, 655
- extending classes with inheritance, 236, 357

- F**
- fields data, 570
- files
 - access, 527, 545, 550
 - append mode, 535
 - binary, 530, 539, 540, 550
 - closing, 403, 534
 - encoding, 530, 549
 - executable JAR, 563, 575, 639
 - file-handling, 527, 530
 - file pointers, 531, 545, 548
 - object serialization, 542
 - random access, 527, 531, 544, 550
 - serial access, 527, 531, 550
 - streams, 530
 - text, 7, 530–532, 535, 536, 539, 541, 549, 551
- final**
 - class, 257
 - method, 257, 654
 - variable, 25
- final operations-streams, 656, 657
- finally**, 403–408, 423, 534, 573
- File class, 488
- file pointers, 531, 545, 548
- File Transfer Protocol (FTP), 662, 663
- FileInputStream class, 540
- FileReader class, 537
- FileWriter class, 534
- float** type, 21
- Float class, 380, 389

- FlowPane class, 292, 303, 485
- fonts
 creating, 274, 295
- for** loop
 enhanced, 119, 137–139, 141, 143, 158, 187, 190, 316, 331, 427, 434–436, 438, 439, 446, 447, 464, 623, 639
- forEach loop/method, 427, 435, 436, 438–440, 446, 447, 455, 464, 465, 623, 639, 645, 646, 649, 652
- formatting
 numbers, 297
- Function interface, 648
- functional interfaces, 383, 387
- FXML, 707
- G**
- garbage collection, 704, 705, 708
- generics
 out of the box interfaces, 378
 upper bound, 357, 379, 389
 wild card, 357, 382, 389, 482
- generic collection classes, 428
- Graphical User Interfaces (GUIs)
 design, 304, 334, 336
 exceptions, 525
- GridPane class, 290, 291, 303
- GUIs, *see* Graphical User Interfaces
- H**
- hashCode method, 450–453, 464, 606, 614, 697
- HashMap class, 427, 443, 446, 451, 452
- HashSet class, 427, 436, 437
- Hbox class, 267, 278, 279, 288, 289, 292, 303
- heavyweight components - user interface, 266
- host - network programming, 353
- hostel case study, *see* case study student hostel
- HTML, *see* Hypertext Markup Language
- Hypertext Markup Language (HTML), 317, 318, 491
- Hypertext Transfer Protocol (HTTP), 662
- I**
- IDE, *see* integrated development environment
- if** statement, 41, 43–45, 47–49, 51, 55, 59, 60, 68, 72, 73, 87, 126, 140, 176, 216, 248, 393, 410, 441
- if...else** statement
 nested, 41, 53, 55
- Image class, 484
- ImageView class, 484
- immutable objects, 689, 700, 701, 704
- import** statement, 32, 411, 429, 430, 555, 556, 574
- increment operator, 29
- indentation, 319
- infinite streams, 656
- InetAddress class, 672
- information hiding, 196
- inheritance
 defining, 236
 extending classes with, 236, 357
 implementing, 237, 361, 363
- init method, 587
- initializing
 attributes, 208
 variables, 24, 103, 208
- inner class, 357, 364, 391
- input
 devices, 527, 528
 from keyboard, 32, 33
 stream, 527, 529, 544, 664–667, 669, 676, 677, 682, 683
 validation, 65, 77–79, 92, 154, 287, 350, 531
- input events
 key events, 469, 473, 477
 mouse events, 469, 470
- InputStream class, 396, 666
- InputStreamReader class, 529
- instantiation, 167
- int** type, 20, 21, 23, 24
- Integer class, 259, 288, 344, 397
- integrated development environment (IDE), 6, 16
- integration testing, 307, 308, 320, 354, 624
- io package, 399, 553
- interface
 abstract methods, 359–361, 692
 default methods, 360, 361, 693, 694
 functional, 369, 371, 374, 378, 383, 384, 387, 436, 458, 460, 582, 649, 652, 656
 out of the box, 378, 648
 static methods, 360, 378, 461
- intermediate operations - streams, 642, 648, 658
- Internet Protocol (IP) address, 567, 663
- is-a-kind-of relationship, 237, 257
- iteration
 do...while loop, 65, 66, 79–84, 90, 92
 enhanced **for** loop, 119, 138, 139, 141, 143, 158, 187, 190, 316, 331, 427, 434–436, 438–440, 446, 447, 464, 623, 639

- for** loop, 65–73, 76, 77, 80, 84, 85, 90–92, 119, 120, 126–128, 137, 138, 141, 143, 150, 155, 158, 435, 537
 - forEach** loop, 427, 435, 436, 438, 440, 446, 447, 455, 464, 465, 623, 639, 646, 649, 652
 - vs streams, 643
 - while** loop, 65, 66, 77–80, 83, 84, 90, 92, 115, 154, 192, 441, 540, 666, 667
 - Iterator interface, 440
 - Iterator objects, 427, 440, 441
- J**
- Java Archive (JAR) files
 - executable, 563, 575, 639
 - Java API, *see* Application Programming Interface
 - Java byte code, 5, 8, 16, 555
 - Java Collections Framework (JCF), 428, 463
 - Java Database Connectivity (JDBC), 553, 564
 - Java Development Kit (JDK), 6, 165, 317
 - Javadoc
 - @author tag, 317, 318
 - @param tag, 318, 330
 - @return tag, 318
 - @throws tag, 421, 422, 611, 614, 620, 635, 639
 - @version tag, 317, 318, 344, 609–611, 614, 620, 635
 - JavaFX
 - containers, 265
 - init method, 587
 - Scene class, 274
 - Stage class, 271, 507
 - start method, 268, 271, 274, 301, 336–340, 517, 567, 582, 585, 587, 671, 680
 - stop method, 268, 587, 682
 - Java Runtime Environment (JRE), 6, 578
 - Java Virtual Machine (JVM), 5, 16
 - JVM, *see* Java Virtual Machine
- L**
- lambda expressions, 265, 279, 280, 284, 287, 341, 358, 368–371, 373–375, 378, 379, 384, 387, 389, 391, 412, 413, 435, 436, 442, 443, 461, 473, 475, 477, 482, 582, 646, 649, 652–655, 657
 - layout policies
 - BorderLayout, 292
 - FlowLayout, 292
 - lazy evaluation, 641, 642, 659
 - length attribute—arrays, 151, 156, 198, 200–202
 - lightweight components, 266
 - lightweight process, 578
 - List interface, 383, 428–431, 456, 460
 - logical operators, 51, 52
 - loops
 - do...while** loop, 65, 66, 79–84, 90, 92
 - enhanced **for** loop, 119, 138, 139, 141, 143, 158, 187, 190, 316, 331, 427, 434–436, 438–440, 446, 447, 464, 623, 639
 - for** loop, 65–73, 76, 77, 80, 84, 85, 90–92, 119, 120, 126–128, 137, 138, 141, 143, 150, 155, 158, 435, 439, 446, 535
 - picking the right loop, 83
 - while** loop, 65, 66, 77–80, 83, 84, 90, 92, 115, 154, 192, 441, 540, 666, 667
- M**
- main method, 12, 29, 35, 36, 41, 95, 97–99, 103, 107, 112, 116, 117, 129, 133, 136, 161, 162, 165, 172, 185, 197, 207, 208, 215, 253, 268–270, 274, 311, 337, 374, 399, 402, 561, 562
 - Map interface, 383, 428, 443
 - math package, 380, 575
 - MediaPlayer class, 488
 - menu-driven programs, 112, 117, 219, 258, 264, 531
 - menus
 - popup, 505
 - pull-down, 499, 500, 503
 - method
 - abstract, 253, 254, 268, 359–361, 369–371, 374, 378, 383–385, 412, 413, 482, 588, 649, 692, 693
 - actual parameters, 95, 101, 110, 116
 - calling, 95, 98, 100, 105, 132, 165, 191, 227, 278, 312, 401, 413, 416, 621, 680
 - class, 205, 208, 229, 238, 396, 411, 412, 456, 457
 - constructor, 167
 - declaring and defining, 96
 - formal parameters, 95, 100, 101, 103, 107, 108, 202
 - header, 13, 97, 134, 224, 371, 397, 398, 400, 402, 403, 430, 584
 - helper, 212, 268, 348, 382, 383, 517, 518, 520, 532, 542, 595, 598, 621, 671, 680
 - overloading, 109, 110, 116, 235, 245, 249, 261, 385, 390
 - overriding, 235, 245, 246, 248, 249, 261, 385, 390, 692, 694, 697
 - return value, 102, 103, 129, 186, 318
 - method references, 374, 461, 462

- modal dialogues, 499, 505
- modifier, 205, 235, 257, 261, 263, 560, 584, 697
- modulus operator, 26, 27, 72, 105
- multiple inheritance
 - diamond problem, 692, 693
- multitasking, 577, 578, 641, 658
- mutual exclusion threads, 584
- MySQL™ database, 564, 569
- N**
- Nested statements, 53, 55
- Netbeans™, 7, 8, 564, 569
- network programming
 - client-server architecture, 686
 - remote databases, 642
- non-modal dialogues, 524
- null** value, 171, 213, 216, 232, 316, 379, 408–410, 412, 413, 416, 445, 446, 660
- number formatting, 297
- O**
- object encoding, 527, 549, 551
- Object class, 235, 257, 258, 262, 449–451, 585, 697, 698, 701, 702
- ObjectInputStream class, 542
- object-orientation
 - benefits of, 223, 235
- object-oriented programming languages, 236, 358
- ObjectOutputStream class, 542–544
- object serialization, 542
- operators
 - arithmetic, 25
 - concatenation, 3, 15, 30
 - increment/decrement, 29
 - logical, 51, 52
 - overloading, 385, 390
- Optional class, 393, 410–413, 423
- Oracle™, 707
- out of the box interfaces, *see* generics
- output
 - devices, 527, 528, 549
 - stream, 529, 530, 549, 642, 665–667, 670, 672, 682
 - to file, 337, 534, 563
 - to screen, 9, 14, 15, 528, 535
- overloading method, 109, 110, 116, 235, 245, 249, 261, 385, 390
- overriding method, 697
- P**
- package(s)
 - accessing, 555
- Java API, 369
- deploying, 563
- developing, 558, 559, 609
- hierarchy, 554, 555
- scope, 559, 560, 574
- parallelism - streams, 658
- parameters
 - actual, 95, 101, 110, 116
 - arrays, 129, 134, 189
 - formal, 95, 100–103, 107, 108, 110, 116, 202
 - objects, 195, 209, 482, 562
 - varargs, 131–134, 138
- parameterized types, 376
- platform independence, 5
- pointers, 293, 531, 545, 548, 689–691, 694
- polymorphic types, 357, 385
- polymorphism, 95, 110, 116, 245, 246, 357, 385, 386, 390
- popup menus, 505
- Predicate interface, 378, 648
- PrintWriter class, 535
- primitive types, 19–21, 30, 176, 185, 190, 191, 259, 376, 452, 545
- private**, 195, 197, 200, 204, 205, 212, 215, 238, 239, 246, 247, 262, 337, 338, 341, 606, 620, 637, 696–698, 700
- processes concurrent, 578, 582
- producer–consumer relationship, 584
- program design, 35, 311
- programming languages, 3–5, 20, 110, 163, 223, 236, 358, 705, 706
- protected**, 238, 239, 246, 262, 697
- pseudocode, 19, 35, 37, 38, 43, 78, 95, 104, 140, 141, 143, 308, 316, 331, 335, 341–344, 354, 538
- public**, 11–13, 195, 197, 200, 201, 205, 238, 262, 315, 328, 559, 560, 606, 612, 614, 621, 693, 694, 697
- pull-down menus, 499, 500
- Q**
- quantum, 582
- R**
- radio buttons, 499, 509, 511–513, 525
- random access files, 527, 544, 545, 550
- record data, 567
- reference, 121, 122, 130, 138, 167, 168, 171, 185–187, 205, 210, 211, 255, 256, 336, 374, 375, 379, 408, 430, 488, 556, 558, 574, 588, 593, 595, 674, 677, 681, 683, 687, 690, 695–701, 707
- repetition, *see* iteration

ResultSet class, 567, 568
 Runnable interface, 383, 580, 582
 run-time error, 564

S

scalar types, *see* primitive types
 Scanner class, 19, 31, 32, 165, 168, 169, 175, 179, 209, 395, 396, 424, 529, 556–558
 scenario testing, 625
 scheduling algorithms, 582
 scheduling thread, 582
 scope, *see* variable scope
 ScrollPane class, 485
 selection, 41–43, 49, 54, 59, 65, 67, 106, 338, 642
 sequence, 12, 41, 42, 50, 59, 65, 173, 269, 428, 529, 600, 658
 serial access files, 527, 531, 550
 Serializable interface, 543, 544, 550, 551
 serialization object, 542
 ServerSocket class, 661, 663, 665
 Service class, 577, 590
 Set interface, 428, 436–438
short type, 20
 sleep method, 585–587
 sliders, 479, 481, 482, 497
 sockets, 661–663, 673
 Socket class, 663, 665, 666, 672, 686
 software

- application software, 3, 4, 16
- embedded software, 4, 16
- systems software, 3, 4, 16, 528

 sort methods

- in the Arrays class, 456
- in the Collections class, 456

 source code, 3–5, 7, 16, 230, 550, 561
 specification, 227, 307, 349, 603, 604, 606, 624
 SQL, *see* Structured Query Language
 sql package, 567
 StackPane, 291, 303, 513
 standard error stream, 529
 standard input stream, 529
 standard output stream, 529
 stateful operations - streams, 641, 657, 659
 stateless operations - streams, 658
 state transition diagram threads, 577, 585
static attribute, 206, 207, 676
static, 11, 13, 97, 100, 195, 205–209, 269, 328, 336, 360, 361, 378, 461, 463, 488, 575, 586, 656, 676
static method, 360, 461

streams

- collecting results, 654
- concatenating, 654, 656
- creating, 645, 646
- infinite, 656
- intermediate operations, 642, 645, 648, 658
- parallelism, 658
- terminating operations, 652
- vs iteration, 643

 Stream API, 641
 stream - files, 530
 string(s)

- comparing, 176, 445
- methods, 163, 174, 176
- using with switch statements, 610

 String class, 163, 172–174, 176, 177, 385, 433, 445, 458, 544
 Structured Query Language (SQL), 565, 568, 572, 573, 642, 706
 student hostel case study, *see* case study
 subclass, 237–240, 246, 249, 254, 256, 257, 359, 361, 385, 386, 395, 398, 400
 Sun™, 6
super, 239, 244, 694
 supplier interface, 378
 superclass, 237–239, 244, 248, 249, 251, 254, 257, 359, 380, 386, 420, 449, 691
 system software, 3, 4, 16, 528
 Swing, 266, 292
switch statement

- with enumerated types, 610, 615, 622
- with strings, 174

 synchronizing threads, 584
 syntax, 5, 17, 32, 35, 77, 79, 131, 132, 161, 274, 357, 369, 372, 401, 403, 412, 417, 491, 690

T

TabPane class, 603, 626
 Task class, 577, 587
 termination operations - streams, 645
 testing, 217, 223, 253, 258, 308, 310–312, 317, 320, 321, 328, 330, 335, 350, 354, 357, 421, 449, 603, 604, 624, 625, 639
 test log, 307, 321, 323, 324, 328, 334, 350, 354
 text encoding, 530, 551
 text files, 531
 text formatting, 539
 TextInputDialog class, 348, 669, 671, 680
this, 135, 136, 170
 Thread class, 580, 582
 threads

- asynchronous, 577, 599
 - execution, 577, 582, 584, 585, 599
 - lightweight process, 578
 - mutual exclusion, 577, 584
 - scheduling, 582
 - states, 585
 - state transition diagram, 577, 585
 - synchronizing, 584
 - throwing exceptions, 401, 417, 424
 - throw** command, 393, 417
 - throws** clause, 393, 398
 - time-slicing, 577, 578, 582, 599
 - tool tips GUI, 635
 - `toString` method, 258, 311–313, 316, 317, 321, 324, 328, 334, 403, 431, 446, 447, 449, 453, 457, 464, 465, 610, 622, 643, 703, 704
 - try...catch** block, 401, 567
 - try-with-resources* construct, 393, 423, 549, 623, 624
 - two-dimensional arrays
 - creating, 148
 - initializing, 149
 - ragged, 155
 - `txt` package, 297, 344, 555
 - type casting, 242, 244, 583
 - type inference, 189, 370, 377, 429, 482
- U**
- UML, *see* Unified Modeling Language
 - `UnaryOperator` interface, 656
 - unboxing, 259, 463
 - unchecked exceptions, 395, 422
 - Unicode, 20, 122, 178, 208, 396, 452, 477, 530, 539, 542, 583
 - Unified Modeling Language (UML), 195–197, 199–202, 206, 211, 212, 223, 224, 228, 230, 231, 235, 237, 239, 246, 247, 260, 261, 307, 308, 310, 328, 330, 331, 334, 336, 357, 360, 386, 448, 454, 455, 464, 466, 467, 603, 604, 606–608, 638, 700
 - unit testing, 307, 308, 311, 320, 624
 - upper bound, *see* generics
 - use-case model, 603, 604
 - user-defined constructor, 237
 - user interface, 9, 285, 304, 335, 353, 525, 587, 603, 626
 - `util` package, 31, 47, 429, 430, 456, 553, 556, 557, 575, 606
- V**
- validation
 - input, 65, 77, 78, 92, 153, 154, 287, 350, 531
 - varargs, 131–134, 138, 159, 161
 - variable(s)
 - declaring, 21, 23, 33, 35, 100, 119, 167, 205
 - environment, 562, 574
 - local, 107, 108, 114, 200, 208, 244, 364, 371
 - scope, 107
 - `Vbox` class, 267, 278, 279, 284, 288, 289, 292, 295, 299, 340, 503, 513, 568, 627, 628
- W**
- `WebView` class, 482, 489–491
 - wild card, *see* generics
 - while** loop, 79–83, 90
 - wrapper classes, 190, 259, 287, 376, 389, 463
- X**
- XML, 570, 572, 707