

# Index

## Symbols

- 64  
-- 70  
^ 79, 84  
^= 80  
\_ 36, 1128  
, 1151  
; 101  
: 159, 348, 1125, 1317  
:: 741  
! 79, 1126  
!= 76, 436  
? 594, 596, 602  
? extends T 595, 603  
? super T 596, 603  
?: 92  
. 11, 127, 128, 334, 346, 1151, 1318  
.. 1319  
... 137, 142  
' 37, 38  
" 39  
"" 468  
[] 93, 118, 121  
{ } 101, 120, 363  
@ 1596, 1601  
@Deprecated 1618  
@Documented 1612  
@FunctionalInterface 1617  
@Inherited 1612  
@Override 202, 228, 632, 1616  
@Override annotation 245  
@param 100, 112  
@Repeatable 1613  
@Retention 1605  
@return 187

@SafeVarargs 646, 664, 673, 881, 1623  
@SuppressWarnings 1620  
@SuppressWarnings("unchecked") 641, 646  
@Target 1607  
@throws 399  
\* 59, 62, 338  
\*= 67  
/ 62, 1317  
/\* and \*/ 40  
/\*\* and \*/ 41  
// 40  
/= 67  
\ 38, 1125, 1317  
& 79, 84  
&& 81  
&= 80  
# 1126, 1151, 1172  
% 62, 63  
%= 67  
+ 64, 68  
+ concatenation 458  
++ 70  
+= 67  
< 75, 1172  
<= 75  
<> 581, 584, 660  
-= 67  
= 55, 1125  
== 76, 436, 443  
-> 93, 162, 695  
> 75  
>= 75  
| 79, 84, 408, 1172  
|= 80  
|| 81  
~ 84

## Numerics

0 1151

## A

ability interfaces

*see* marker interfaces

abrupt method completion 376

absolute adjusters 1057

absolute path 1318

absolute values 489

abstract

interfaces 243

methods 244, 691

abstract 243, 244, 691

abstract classes 223

abstract methods 228

considered incomplete 226

declaring 223

extending 224

incomplete 224

polymorphic references 225

UML class diagram 224

abstract data type 579

abstract keyword 223, 228

abstract method declarations 691

in enum types 300

in interfaces 244

abstract methods 228

declaration 228

implementing 228

method header 228

overloading 229

overriding 228

abstraction 7, 15

access modifiers 99, 100, 109, 353

access order 858, 859

accessibility 11, 21, 361, 1199

access modifiers 353

anonymous classes 538

local classes 525, 527

members 355, 361

modifiers 505

non-static member classes 517

static member classes 511

top-level types 353

UML notation 356

accessibility for class members

package 356, 359

private 360

protected 359

public 358

accessibility for top-level types

package 353

public 353

accessibility rules 1200

accessor methods *see* get methods

accounting currency formatting 1143

acquisition order 1498

activation frame 548

*see* method execution 373

actual parameter 128

actual parameter list 128, 270

actual type parameters 583

adapter 1013, 1014, 1015

adding to class 195

additive operators 64

aggregate module 1194

aggregate of values 306

aggregate operations 899

aggregation 15, 17, 199

hierarchy 199

algorithms 799

aliases 55, 77, 78, 131

*see also* symbolic link

all-or-none model 1510

ambiguous call 206, 271

analyzing program code 1658

*and* operator 84

AND-ing 87

AnnotatedElement 1626

Annotation 1595

annotation 1595

@ 1601

applying annotation type 1601

declaration annotation 1607

default values 1603

marker annotation 1601

metadata 1595

normal annotation 1601

processing 1625

single-element annotation 1601

type annotations 1608

annotation type 1595, 1596

@ 1596

access modifier 1597

annotation type element 1598

containing annotation type 1614

declaring 1596

marker annotation type 1598

member declarations 1597

multi-element annotation type 1599

repeatable annotation type 1614

- simple name 1597
- single-element annotation type 1599
- syntax 1596
- annotation type declaration
  - see* annotation type
- annotation type element 1598
  - access modifier 1599
  - attributes 1598
  - default values 1600
  - element type 1599
  - method declaration 1598
  - optional annotation element 1600
  - required annotation element 1600
  - value() element 1599
- annotations 243
  - @Deprecated 1618
  - @FunctionalInterface 691, 1617
  - @Override 202, 632
  - @SafeVarargs 646
  - @SuppressWarnings("unchecked") 641
- anonymous arrays 123, 126
  - [] 123
- anonymous classes 507, 534, 585, 706
  - accessing members in enclosing context 538
  - compared with lambda expressions 708
  - declaration 534
  - event listeners 535
  - extending an existing class 535
  - hiding members in enclosing context 539
  - implementing an interface 537
  - in non-static context 534
  - in static context 534
  - instantiation 538
  - objects on the fly 534
  - referencing instances 538
  - shadowing local variables 539, 709
  - using qualified this 539, 709
- anti-symmetric 778
- API (application programming interface) 3
- apostrophe 38
- application 21
- architecture neutral 5
- argument
  - see* actual parameter
- arguments to main method 141
- arithmetic compound assignment
  - operators 67
- ArithmeticException 380
- arity 137, 711
- array constructor references 742, 748
- array creation expression 119, 123, 144
- array element initializer 1602
- array initializer 120, 123, 126, 147
- array list 818
- array mismatch 889
- array operations 815
- array store check 265, 641
- array store check report
  - false positive 645
- array types 593
  - see* arrays
- ArrayBlockingQueue 1535
- ArrayDeque 839
- ArrayIndexOutOfBoundsException 121, 380, 406
- ArrayList 460, 818
  - add collection 665
  - add element 661, 665
  - autoboxing 669
  - capacity 660
  - clear list 666
  - comparison with arrays 676
  - constructing 660
  - constructors 662
  - converting to array 672
  - element search 670
  - filtering 704
  - import 660
  - inheritance hierarchy 659
  - initial capacity 661
  - insertion order 658
  - list of lists 662
  - membership test 670
  - modifying 665
  - nested lists 662
  - object value equality 670
  - open range-view operations 659
  - ordered 658
  - position-based access 659
  - positional index 670
  - positional order 658
  - positional retrieve 670
  - querying 669
  - references 660
  - remove collection 666
  - remove element 666
  - replace element 666
  - size 660, 669
  - subtype relationship 662
  - text representation 661
  - trim to size 667
  - type-safety 660, 662
  - unchecked conversion warning 660

- zero-based index 658
- Arrays 886
  - creating streams 916
- arrays 118, 435
  - [] 118, 119, 121
  - { } 120
  - anonymous 123, 126
  - array creation expression 119
  - array initialize list 120, 123
  - array initializer 120, 126
  - array name 118
  - array size 119
  - array store check 265, 641
  - ArrayIndexOutOfBoundsException 121
  - binary search 884
  - bounds 121
  - collection operations 815
  - common prefix 887, 889
  - comparators 885
  - comparing 886
  - construction 119
  - converting to list 673
  - declarations 118
  - default initialization 119
  - dynamic 659, 818
  - element access expression 121
  - element default value 265
  - element type 118
  - elements 118, 121
  - equality 886
  - filling 891
  - index 118
  - index expression 121
  - initialization 119, 120, 125
  - initializing 891
  - iterating over 179
  - iterating over an array 121, 122
  - length 118
  - lexicographic comparison 887
  - mismatch 889
  - multidimensional 123, 124
  - objects 435
  - proper prefix 888, 890
  - ragged 125
  - reference 119, 121, 265
  - searching 884
  - sequential streams 916
  - sorting 882
  - subtype covariance 264
  - text representation 890
  - using 121
- arrays of arrays 118, 124
  - multidimensional 124
- ArrayStoreException 265, 381, 662, 672, 815, 816
- array-valued single-element annotation 1604
- arrow notation (->) 162
- arrow operator (->) 93
- ascending-order key set 864
- ASCII 37, 42
- AssertionError 383
- assessor methods
  - see* get methods
- assignable 48, 269
- assignment
  - widening reference 598
- assignment compatible 48, 269, 660
- assignment conversions 48
- assignment operator 9
- assignments
  - arithmetic compound operators 67
  - bitwise 86
  - cascading 56
  - compound operators 80, 86
  - expression statement 56
  - extended operator 91
  - implicit narrowing 57
  - multiple 56
  - numeric conversions 57
  - operator 51, 55
  - primitive values 55
  - references 55
  - widening reference conversions 200
- association 17
  - aggregation 199
  - composition 199
  - realization 250
- associativity 52
- asynchronous 1404
- atomic actions 1491, 1493, 1506
  - CAS: Compare and Set 1494
- atomic classes 1493
- atomic move 1339
- atomic operation 1335, 1371
- atomic variables 1493
  - atomic actions 1493
- AtomicInteger 1493, 1495
- atomicity 1491
- AtomicMoveNotSupportedException 1343
- at-sign 1596
- attributes 1550
- attributes *see* properties
- AttributeView interface 1366

- autoboxing 440
  - for(·) statement 179
  - see* boxing conversion
- AutoCloseable 418, 421, 550, 1262, 1273, 1557
  - streams 908
- automatic garbage collection 547
- automatic memory management 547
- automatic module 1233, 1234
  - naming 1234
- Automatic Resource Management (ARM) 420
- automatic variables *see* local variables
- auxiliary methods 257
- available time zones 1096

## B

- backing collection 874
- backslash 38
- backspace 38
- backtrace 414
- backward compatibility 579
- bags 821
- balanced tree 805, 829
- barrier
  - see* cyclic barrier
- barrier action 1508
- barrier point 1508
- base 33, 441, 445
- base case 1484
- base class 195
- base name 1126
- basic collection operations 807
- basic for statement 176
- basic map operations 848
- basic statement 1561
- BasicFileAttributes interface 1362
- BasicFileAttributeView interface 1366
- Before Current Era (BCE) 1051
- behavior 691
- behavior parameterization 691
- behavioral parameters 928
- BiConsumer functional interface 725
  - primitive type specializations 728
- BiFunction interface 734
  - composing two-arity functions 735
  - primitive type specializations 736
- big numbers 495, 496
- BigDecimal
  - customizing formatting 1150
  - formatting 1142
  - parsing 1144
  - rounding 1148
- BigDecimal class 495
- BigInteger 495
- binary
  - numeric promotion 50
  - operators 51
  - shift operators 87
- binary files 1266, 1267, 1268
- binary operator 53
- binary representation
  - see* binary values
- binary search
  - arrays 884
  - lists 878
- binary values 1266, 1267, 1268
- BinaryOperator interface 738
  - composing binary operators 738
  - primitive type specializations 739
- bind parameter 1564
- bind variable 1564
- binding of method call 210, 231
- BiPredicate functional interface 720
- bit mask 85
- bit masking 87
- bit patterns 55
- bit shifting 87
- bitwise
  - and* operator 84
  - assignment 86
  - complement 84
  - compound assignment 86
  - operators 84
  - or* operator 84
  - shifting 87
  - xor* 84
- bitwise AND
  - & 84
- bitwise complement
  - ~ 84
- bitwise exclusive OR
  - ^ 84
- bitwise OR
  - | 84
- bitwise XOR
  - ^ 84
- blank final local variables 236
- blank final variables 136, 234, 235
- block 163, 168, 171
- block scope 171, 363, 701
- BLOCKED state 1417

- blocking collection 1520
  - blocking operations 1262, 1440
  - blocking queues 1533
  - BlockingDeque 1535, 1537
  - BlockingQueue 1535, 1536
  - blocks 101, 363
    - scope 363
    - try 385
  - boilerplate code 418, 707
  - Boolean
    - condition 154
  - Boolean wrapper class 447
  - booleans 41, 43
    - casting 49
    - expressions 75
    - literals 37
  - BooleanSupplier functional interface 716
  - bounded collection 1520
  - bounded instance method references 744
  - bounded type parameters
    - constraint 605
  - bounded wildcard 595
  - bounds
    - constraint 605
  - boxing conversions 46
  - break statement 159, 182
  - bridge method 629, 780
  - BS *see* backspace
  - buckets 769
  - buffered readers 1280
    - creating streams 1280
    - end of stream 1280
    - reading text lines 1280
  - buffered writers 1279
  - BufferedReader class 1270, 1346
    - streams 920
  - BufferedWriter class 1272, 1347
  - buffering streams 1266
  - building abstractions 15
  - built-in functional interfaces
    - java.util.function package 711
    - BiConsumer 725
    - BiFunction 734
    - BinaryOperator 738
    - BiPredicate 720
    - BooleanSupplier 716
    - Consumer 725
    - DoubleBinaryOperator 738
    - DoubleConsumer 725
    - DoubleFunction 729
    - DoublePredicate 720
    - DoubleSupplier 716
    - DoubleToIntFunction 729
    - DoubleToLongFunction 729
    - DoubleUnaryOperator 737
    - Function 729
    - IntBinaryOperator 738
    - IntConsumer 725
    - IntFunction 729
    - IntPredicate 720
    - IntSupplier 716
    - IntToDoubleFunction 729
    - IntToLongFunction 729
    - IntUnaryOperator 737
    - LongBinaryOperator 738
    - LongConsumer 725
    - LongFunction 729
    - LongPredicate 720
    - LongSupplier 716
    - LongToDoubleFunction 729
    - LongToIntFunction 729
    - LongUnaryOperator 737
    - ObjDoubleConsumer 726
    - ObjIntConsumer 725
    - ObjLongConsumer 726
    - Predicate 719
    - Supplier 715
    - ToDoubleBiFunction 734
    - ToDoubleFunction 729
    - ToIntBiFunction 734
    - ToIntFunction 729
    - ToLongBiFunction 734
    - ToLongFunction 729
    - UnaryOperator 736, 737
  - built-in synchronization
    - see* intrinsic locking
  - bulk map operations 856
  - bulk operations on collections 807
  - bulk task execution 1476
  - bundling resources 1126
  - byte 32, 42
  - byte streams 1261
    - reading bytes 1262
    - writing bytes 1262
  - bytecode 5, 22
  - bytecode verification 1649
- ## C
- C++ 5
  - cached thread pool 1463
  - caching 367
  - call by reference 133
  - call by value 133

- call chaining 128
- call signature 271
- call stack
  - see* JVM stack 373
- callable statement 1568
  - stored functions 1568
  - stored procedures 1568
- callee 127
- caller 127, 186
- candidate bundle name 1132
- capacity 660
- capture conversion 621
- capture of 619
- carriage return 38, 40
- CAS: Compare and Set 1494
- cascaded grouping 1008
- cascaded partitioning 1012
- cascading assignments 56
- cascading if-else statements 157
- case constants 158, 159, 162
- case labels 158, 159, 162
- case sensitivity 30
- case-insensitive ordering 787
- cast 645
  - references 287
- cast operator 45, 49, 51, 58, 67, 68, 77, 144, 274
- casting 47, 49
  - see also* conversions
- catch clause 385
  - multi- 406
  - rethrowing exception 410
  - uni- 384
- catching exceptions 373
- catch-or-declare rule 397
- catch-or-specify requirement 397
- cause 413
- ceiling entry 864
- ceiling key 864
- chaining 569
  - constructors 217, 569
  - new operator 520
- chaining exceptions 413
- changing list contents 881
- char 42
- character case 458
- character encoding 1270
  - see also* character sets
- character sequences
  - see* strings *and* string builders
- character set
  - ASCII 37, 42, 1270
  - ISO Latin-1 37, 42
  - Unicode 37, 42, 1270
- character streams 1261, 1270
  - character encoding 1270
  - end of stream 1271
  - reading characters 1271
  - writing characters 1272
- Character wrapper class 447
- character-based device
  - console 1284
- characters 42
  - literals 37
  - searching for 461, 476
- CharSequence interface 454, 459, 919
- charsets
  - see also* character sets
- checked exceptions 383
- child class 195
- child threads 1407, 1412, 1490
- choice format 1172, 1174
  - limits 1172
  - subformat pattern 1172
  - syntax 1172
- ChoiceFormat 1140, 1174
- choosing between String and StringBuilder class 475
- ChronoField enum type 1058, 1068, 1102
  - list of enum constants 1068
- ChronoUnit enum type 1062, 1063, 1066, 1104, 1105
  - list of enum constants 1067
- Class 1626
- Class class 436
- class design 1649
- class file 22
- class file name 509
- class hierarchy
  - see* inheritance hierarchy
- class initialization 234
- class literals 650
- class lock 1415, 1425, 1427
- class method 14
- class path 345, 1233
  - absolute pathnames 348
  - entries order 348
  - entry-separator character 348
  - fully qualified package name 348
  - path-separator character 348
  - relative pathnames 348
  - search in a named package 347
  - searching for classes 345
  - whitespace 348

- class search path
  - see* class path
- class variable 14
- class versioning 1302
- ClassCastException 276, 381, 1129
- classes
  - abstract 223
  - accessibility 353
  - adding to 195
  - anonymous 504, 507, 534
  - base 195
  - body 99
  - child 195
  - class file name 509
  - concrete 223, 225, 230
  - constructors 109
  - contract 223, 247
  - declaration context 505
  - declarations 99, 333
  - definitions 7, 9
  - derived 195
  - diagram 12, 14
  - encapsulation 332
  - extending 195, 230
  - final 230, 366
  - final vs. abstract 230
  - fully qualified name 345
  - fully qualified package name 336
  - generalized 198
  - grouping 334
  - header 99
  - immutable 364
  - implementation 223
  - implementing interfaces 245
  - incomplete 224
  - initialization 234, 571
  - inner *see* inner classes
  - instances 9
  - local 504, 506, 525
  - members 10
  - methods 115
  - name 334
  - nested *see* nested classes
  - non-static member 504, 506, 514
  - Object 435
  - on-the-fly 504
  - parent 195
  - partial implementation 223
  - records 305
  - runtime 436
  - scope 361
  - sealed 318
  - searching for 345
  - specialized 198
  - static fields 113
  - static member 507
  - static members 113, 506, 507
  - static methods 115
  - static variables 113
  - subclass 15, 195
  - superclass 15, 195
  - top-level 503, 506
  - utility 115
  - wrappers 435, 439
- class-level lock
  - see* class lock
- ClassLoader class 435
- ClassNotFoundException 378
- CLASSPATH environment variable
  - see* class path
- classpath option
  - see* class path
- clauses
  - catch 385
  - extends 195
  - finally 386, 391
  - implements 245
  - permits 319
  - throws 396
- cleaning up 550
- clean-up code 391, 1412
- client 11
- Cloneable interface 437
- CloneNotSupportedException 437
- cloning objects 437
- Closeable 421
- closing an I/O stream 1262, 1267, 1268
- closing delimiter 468
- code corruption 1644
- code injection 1643
- code migration
  - bottom-up strategy 1236
  - top-down strategy 1237
- code optimizations 233
- Code Ranch ([www.coderanch.com](http://www.coderanch.com)) 1654
- code reuse 4, 195, 199
- codebase 1646
- Collection 658
- Collection interface 800
- collections 600, 658, 799, 800
  - adding elements 881
  - array operations 815
  - ArrayList 818
  - as single entity 658



- basic operations 807, 848
- bulk operations 807, 856
- `Cloneable` 802
- compare with streams 906
- data structures 804
- duplicate elements 817, 821
- elements 658, 799
- filtering 813
- first element 827, 828, 1524
- `HashSet` 823
- implementations 799, 802, 804
- interfaces 799
- iteration 808, 809
- iteration order 811
- iterators 810
- last element 827, 828, 1524
- `LinkedHashSet` 823
- `LinkedList` 818
- `List` 818
- lists 817
- map views 856
- maximum 880
- minimum 880
- `NavigableSet` 827
- optional operations 806
- ordered 658, 806, 817
- parallel streams 915
- range-views 817, 827, 828, 1525
- retaining elements 807
- sequential streams 915
- `Serializable` 802
- `Set` 821
- shuffle elements 878
- sorted 806
- `SortedSet` 827, 828
- sorting 658
- subset 827, 828, 1525
- text representation 812
- thread-safety 818, 857
- `TreeSet` 829, 864
- unmodifiable view 874
- unordered 806
- unsupported operations 848
- utility methods 799
- `Vector` 818
- views 670, 827, 828, 856, 1525
- collections framework 799
- `Collections` utility class 876, 878
- `Collector` interface 908, 999
- `Collectors` 908, 999
  - importing 999
- collectors 967, 984, 998
  - averaging 1022
  - classification mapping 1000
  - classification maps 1000
  - classifier function 1000
  - collecting to a `Collection` 1000
  - collecting to a `ConcurrentMap` 1004
  - collecting to a `List` 1001
  - collecting to a `Map` 1001
    - key mapper 1002
    - map supplier 1002
    - merge function 1002
    - value mapper 1002
  - collecting to a `Set` 1001
  - comparison with terminal operations 1030
  - composing 999
  - concurrent 1004, 1010
  - counting 1019
  - downstream 999, 1006, 1013, 1014, 1015, 1018
  - filtering 1013
    - downstream collector 1013
    - predicate 1013
  - finding min/max 1019
  - finishing 1018
    - downstream collector 1018
    - finisher function 1018
  - flat mapping 1015
    - downstream collector 1015
    - mapper function 1015
  - functional reduction 1018, 1024
  - grouping 999, 1005
    - classification map 1006
    - classifier function 1006
    - default downstream collector 1006
    - downstream collector 1006
    - map supplier 1006
    - multilevel 1008
    - to a `ConcurrentMap` 1010
  - joining to a string 1004
  - mapping 1014
    - downstream collector 1014
    - mapper function 1014
  - partitioning 999, 1010
    - classification map 1010, 1011
    - default downstream collector 1011
    - downstream collector 1010, 1011
    - multilevel 1012
    - partitioning predicate 1010

- reducing 1024
  - binary operator 1024
  - identity value 1024
  - mapper function 1024
- summarizing 1023
- summary of 1026
- summing 1021
- colon notation (:) 158
- columns 1550
- command line 22, 142
- command prompt 22
- commands
  - java 23
  - javac 22
- comments 40
- common memory space 1401
- common prefix 887, 889
- communication 127
- compact canonical record constructor 310, 312
- compact form styles 1146
- compact number formatting 1145
- compact number parsing 1147
- compact strings 450
- CompactNumberFormat 1140
- Comparable interface 442, 457, 777
  - compareTo 777
- comparable objects 777
- Comparator interface
  - compare 785
  - functional interface 785
- comparators 777
  - conditional 782, 788, 790
  - reverse order 876
- comparing objects 436, 760, 785
  - see also* equality
- comparing strings 457
- comparison 75
- compilation unit 336
  - deriving permitted direct types 323
- compiler
  - Xlint:unchecked option 591
- compiler optimizations 1490
- compiling and running a service 1231
- compiling Java source code 22
- complement
  - ~ 84
- completes abruptly
  - see* exception propagation 376
- component fields 308
- component list 306
- composite object 15
- composition 17, 199
  - see* aggregation
- compound declaration 147
- compound messages
  - conditional formatting 1172
  - formatting 1169
  - handling plurals 1173
- compound mutually exclusive operations 1517
- compound state 1415
- compound statement 101
- concatenation of strings 458
- concatenation operator 68
- concise try-with-resources 420
- concrete class 223
- concrete method 231
- concrete parameterized types 594
- concrete type 587, 594, 600
- concurrency 1402, 1578
- concurrent activities 1401
- concurrent collections 1519, 1520
  - blocking 1520
  - bounded 1520
  - duplicate elements 1520
  - happens-before relationship 1519
  - non-blocking 1520
  - ordering 1520
  - unbounded 1520
- concurrent collector 1004, 1010
- concurrent maps 1004, 1010, 1528
  - segments 1530
- ConcurrentHashMap 1530
- ConcurrentLinkedDeque 1524, 1525
- ConcurrentLinkedQueue 1524, 1525
- ConcurrentMap 1004, 1010, 1530
- ConcurrentModificationException 672, 811, 812
- ConcurrentNavigableMap 1531
- ConcurrentSkipListMap 1530
- ConcurrentSkipListSet 1524
- condition
  - Boolean 154
  - expressions 154
- conditional 75
  - operators 92
  - statements 154
- conditional *and* operator 81
  - pattern variable 282
- conditional comparator 782, 788, 790
- conditional expressions 92
  - associativity 93
  - nested 93

- precedence 92
- short-circuit evaluation 92
- side effects 92
- conditional formatting 1140, 1172
- conditional operators 81
- conditional *or* operator 81
  - pattern variable 283
- conditions 75
- configuration 6
- configuration data 1124
- congruent 778
- connecting punctuation character 30
- connection 1557
- Console 1179
- console 1284
  - formatted prompting 1286
  - printing formatted values 1285
  - reading password 1285
- Console class 1284
- const 32
- constant 560
- constant expression 47, 57, 70, 159, 234, 560, 1600
  - labels 159
- constant field values
  - case labels 164
- constant string expressions 164
- constant values 32, 233
- constant variable 57, 234, 560
- constants 259
- constant-specific class bodies
  - using anonymous classes 299
- constituent objects 15
- constituents 17
- constraints 1550
- constructing array 119
- constructor chaining 214, 217, 218, 569
- constructor references 742, 747
- constructors 8, 109
  - access modifier 109
  - accessibility 360
  - body 109
  - canonical 308
  - chaining 214, 217, 218
  - class name 109
  - declaration 99
  - default 110
  - header 109
  - implicit default 110
  - local declarations 109
  - no-argument 110, 214, 218
  - non-zero argument 111, 218, 219
  - overloading 112
  - superclass constructor 110
- constructs 30
  - high-level 30
  - loops *see* iteration statements
  - synchronized statements 1426, 1427
- Consumer functional interface 725
  - composing consumers 727
  - primitive type specializations 727
- contained annotation type 1600
- containers 600
  - see also* collections
- containing annotation type 1600, 1614
- contains characters 462
- context switching 1402
- contextual keywords 31
  - non-sealed 318
  - permits 318
  - sealed 318, 321
  - yield 166
- continue statement 185
- contract 7, 199, 223, 244, 332
- control flow
  - break 159, 182
  - continue 185
  - do-while 176
  - for(;;) 176
  - for(:) 179
  - if 154
  - if-else 155
  - iteration *see* iteration statements
  - loops *see* iteration statements
  - return 186
  - statements 101
  - switch expression 166
  - switch statement 157
  - throw 395
  - transfer statements 181
  - while 175
  - yield 166
- control transfer 181
- conversion categories 47
- conversion contexts 47
- conversions 266
  - assignment 48
  - contexts 47
  - explicit 44
  - identity 68
  - implicit 44
  - implicit narrowing 68
  - method invocation 48
  - narrowing reference 275, 599, 619

- numeric promotions 50
- parameters 129
- reference casting 274
- string concatenation 69
- to strings 464
- truncation 58
- type-safe 270
- unsafe casts 276
- widening reference 200, 275, 598
- converting values 441, 442, 444, 445, 447, 464
- copy-on-write collections 1540
- CopyOnWriteArrayList 1541
- CopyOnWriteArraySet 1541, 1542
- CopyOption interface 1332
- Core Collections Interfaces 803, 1523, 1534
- core interfaces 799
- Core Map Interfaces 804
- core stream interfaces 908
- count-down latch 1510
- CountDownLatch 1510
- counter-controlled loops 176
- country codes 1121
- covariant return 201, 205, 633
- cp option
  - see* class path
- CPU core 1031
- CR *see* carriage return
- crab 179
- CREATE statement 1552
- creating
  - objects 93
  - threads 1404
- creating a runtime image 1231
- creating modular JAR 1216
- creating records 309
- critical section 1422
- CRUD operations 1552
- currency 1120, 1141
  - currency symbol 1151
  - formatting 1141, 1144
- currency formatter 1142, 1145
- currency symbol 30, 1151
- current directory 1319
  - . 346, 1318
  - system property 1322
- Current Era (CE) 1051
- current object 106
- currying functions 739
  - partial application 740
- cursor 818, 1572
- cyclic barrier 1507

- all-or-none model 1510
- barrier action 1508
- barrier point 1508
- CyclicBarrier 1507

## D

- d option 343
- daemon 1412, 1437
- dangling references 547
- Data Definition Language (DDL) 1552
- data encryption 1642
- data integrity 1644
- Data Manipulation Language (DML) 1552
- Data Query Language (DQL) 1552
- data source 902
- data streams 1264, 1266, 1267
- data structures 658, 804
- data types *see* types
- database capabilities 1581
- database connection 1557
- database cursor 1572, 1579
- Database Management System (DBMS) 1550
- database protocol 1558
- database transactions 1583
- DataFormat 1140
- DataInputStream class 1264
- DataOutputStream class 1264
- Date 1046, 1101, 1167
- date 777, 1120
  - see* temporal objects
- date fields 1046
- date units 1062
- DateFormat 1167
- date-time
  - see* temporal objects
- date-time formatters
  - format styles 1158
  - immutability 1154
  - ISO-based default 1153, 1155
  - ISO-based predefined 1153, 1155
  - letter pattern 1161
  - parsing 1154
  - pattern letters 1161
  - pattern-based 1160
  - style-based 1153, 1157
  - thread-safety 1154
- DateTimeException 382, 1047, 1055, 1098
- DateTimeFormatter 1140, 1153
- DateTimeParseException 382, 1073, 1080, 1088, 1157, 1160

- daylight saving time (DST) 1096, 1105
  - time gap 1105
  - time overlap 1105
  - time zone rules 1105, 1107
  - vs. standard time 1105
- DayOfWeek enum type 1055
- DBMS 1550
- DDL 1552
- DDL operations 1552
- deadlocks 1443
- decimal separator 1151
- DecimalFormat 1140, 1150
- declaration 99
- declaration annotations 1607
- declaration context 505, 1607
- declaration statement 9, 66, 71, 82, 102, 103, 178
- declaration-before-reading rule 561
- declarations
  - arrays 93, 118
  - classes 99, 333
  - interfaces 333
  - local 101
  - main method 142
  - methods 100
  - multidimensional arrays 123
  - packages 333, 336
  - statements 101
  - variable arity method 137
- declare-before-use rule 586
- declared type 207, 270
  - see* static type
- declared-type parameters 696
- declaring *see* declarations
- decoupling 288
- decrement operator 70
- deep copying 436
- deep reflection 1218
- default
  - constructor 110
  - exception handler 376
  - values 103, 555, 569
- default 251
  - label 158, 163, 169
  - method 251
- default accessibility for class members
  - see* package accessibility for class members
- default accessibility for top-level types
  - see* package accessibility for top-level types
- default constructor 110
- default downstream collectors 1006
- default file system 1319
- default locale 1122, 1123, 1129, 1132, 1133, 1144
- default method 251, 691, 692
  - conflict resolution 253
  - multiple inheritance of implementation 253
  - overriding 252
- default ordering
  - see* natural order
- default package 336
- default resource bundle 1127, 1128, 1134
- default values 1600, 1603
- defensive copying 368
- deferred execution 696, 700
- definite assignment analysis 105, 237
- definitions
  - interfaces 242
  - polymorphism 284
- DelayQueue 1535
- DELETE statement 1554
- delimiters 1145
- Denial of Service (DoS) 1640
- dependency cycles 1198
- deprecated 439
- deprecated code
  - @Deprecated 1618
  - deprecation warning 1619
  - ordinarily deprecated 1618
  - terminally deprecated 1618
- deprecation warning 1619, 1621
- Deque 838
- deque 801, 804
  - ArrayDeque 839
  - Deque 838
  - FIFO queue 838
  - head 838
  - LinkedList 839
  - tail 838
- Derby database 1558, 1559
  - connection 1558
- derived class 195
- deserialization 1649
- deserialization of objects 1289
- destination directory 343, 1214, 1247
- destination object 1179
- destination of data 1261
- destination stream 26
- destructive collection operations 807
- diagrams
  - class 7

- object 10
  - see also* UML
- diamond operator (<>) 584
- dictionary 801
- dictionary order 457
- difference 821
- direct sealed superclass 319
- direct superclass 216
- directory
  - copy entire directory 1384
  - see also* directory entry
- directory entry 1318
  - accessing individual file attributes 1352
  - atomic move 1344
  - copy entire directory 1384
  - copy options 1339
  - copying 1339
  - copying using I/O streams 1342
  - creating 1371
  - creating regular directories 1375
  - creating regular files 1371, 1372
  - creating streams 1378
  - creating symbolic links 1374
  - creating temporary directories 1375, 1377
  - creating temporary files 1371, 1373
  - creation time 1355
  - deleting 1338
  - denotes same as another entry 1337
  - determining existence 1336
  - determining file accessibility 1355
  - determining kind of entry 1355
  - determining last modification time 1355
  - determining owner 1356
  - determining size 1354
  - handling POSIX file permissions 1357
  - hidden 1355
  - last access time 1355
  - last modified time 1355
  - managing file attributes 1352
  - moving and renaming 1343
  - reading and writing bytes 1350
  - reading and writing character data 1346
  - stream to read text lines 1378
- directory hierarchy
  - breadth-first traversal 1380
  - cyclic path dependency 1384
  - depth 1379
  - depth-first traversal 1380
  - finding/searching 1386
  - limit traversal by depth 1384
  - root 1379
  - siblings 1379
  - traversal strategies 1380
  - walking/traversing 1382
- directory-level separator 1215
- walking/traversing 1382
- directory-level separator 1215
- DirectoryNotEmptyException 1338
- disjunct collections 825
- dispensing canned advice application 1206
- distance of shift 87
- distributed 5
- divide-and-conquer
  - base case 1484
  - forking 1484
  - joining 1484
  - parallel processing 1484
- divide-and-conquer algorithm 885
- divide-and-conquer paradigm 1484
- dividend 63
- division
  - floating-point 63
  - integer 62
- division operator
  - / 62
- divisor 63
- DML 1552
- documentation 40
- documentation comment 40, 41
  - tags 41
- documenting *see* documentation
- Domain Name System (DNS) 335, 1197
- DosFileAttributes interface 1362, 1365
- DosFileAttributeView interface 1366, 1370
- dot 334
- double 36, 43
- double quote 38
- DoubleBinaryOperator interface 738
- DoubleConsumer functional interface 725
- double-ended queues
  - deque 838
- DoubleFunction interface 729
- DoublePredicate functional interface 720
- DoubleStream interface 907
- DoubleSummaryStatistics 908
- DoubleSupplier functional interface 716
- DoubleToIntFunction interface 729
- DoubleToLongFunction interface 729
- DoubleUnaryOperator interface 737
  - composing unary operators 737
- doubly linked list 805
- do-while statement 176
- downcasting 45
- downstream collectors 999, 1006, 1025
  - default 1006

- filtering adapter 1013
- finishing adapter 1018
- flat mapping adapter 1015
- mapping adapter 1014
- DQL 1552
- dropping stream elements 929, 931
- duplicate elements 817, 821
- duplicating objects 437
- duration
  - accessing units 1090
  - comparable 1091
  - compared with period 1095
  - creating 1087
  - equality 1091
  - get methods 1090
  - immutable 1080, 1087
  - ISO format 1089
  - parsing 1088
  - plus/minus methods 1092
  - supported ChronoUnits 1089, 1090, 1093
  - temporal arithmetic 1092
  - thread-safe 1080, 1087
  - time-based 1079, 1087
  - unit conversion 1091
  - with methods 1092
- Duration class 1087
  - see* duration
- dynamic 5
- dynamic arrays 659
- dynamic binding
  - see* dynamic method lookup
- dynamic method lookup 211, 284, 288
- dynamic type 200, 207
- dynamically resizable arrays 818

## E

- eager execution 905
- effectively final 527, 703
- element type 118
- element type values for @Target 1610
- elements 118, 658, 799
- elements of collections 799
- eligible for garbage collection 548
- ellipsis 137
- else clause matching 157
- embedded applications 3
- empty set 821
- empty statement 101
- empty string 451
- encapsulation 4, 332, 334, 514, 1189, 1644
  - nested types 506

- encapsulation of implementation 199
- enclosing class 504, 513, 515
- enclosing context 505, 513, 518
- enclosing instance 505, 506, 511, 538
- enclosing object reference 516
- encoding of characters 1270
- encounter order 909
- end of file
  - see* end of I/O stream
- end of file (EOF) 378
- end of I/O stream 1262, 1268, 1271, 1280, 1282, 1290
- ends with characters 462
- enhanced for loop 175
- enterprise applications 3
- entries 801, 847
- entry point 21, 23, 141, 1215, 1217, 1218, 1240
- entry set 847, 1414, 1423
- enum class
  - see* enum types
- enum constants 293
  - symbolic names 293
  - values 293
- enum types 165, 260, 293, 341, 649
  - abstract method declarations 300
  - accessibility 353
  - canonical form 294
  - constant-specific class bodies 299
  - declaring 294
  - enum constants 293
  - enum constructors 296
  - equality 299
  - final methods 299
  - implementing interfaces 304
  - implicit static methods 298
  - in switch expression 303
  - inherited methods 298
  - initializer blocks 297
  - local 507, 533
  - members 296
  - named constants 293
  - natural order 299
  - nested 504
  - ordinal value 165, 299
  - permitted direct subtypes 323
  - selector expression 158
  - static member 506, 507
  - top-level 503, 506
  - using 295
- enumerated types
  - see* enum types

- EOFException 378
- epoch 1046, 1071
- epoch-second 1071
- equality 76, 436, 760
  - equals method 78, 436, 760
  - object value 78
  - objects 78
  - primitive values 76
  - reference values 77
- equals method 78, 436, 760
- equivalence relation 762
- equivalent collections 825
- erasure 627, 630, 633
  - of the supertype method 635
  - same 634
- Error 382
- escape character 1125
- escape sequences 38, 469
- essential whitespace 472
- evaluation order 52, 82
  - arithmetic expressions 60
- evaluation short-circuits 82
- event listeners
  - anonymous classes 535
- exact integer arithmetic functions 491
- exam 1653
  - multiple-choice 1659
  - Oracle exam attempt 1655
  - program 1656
  - questions 1658
  - registration 1655
  - result 1657
- exam objectives
  - OCP Java SE 11 1667
  - OCP Java SE 17 1661
- exam question assumptions
  - OCP Java SE 17 1658
- Exception 378, 380
- exception handler 373
  - see also* exceptions
- exception handling
  - advantages 425
- exception type 396
- exceptions 373, 384
  - ArrayStoreException 815
  - backtrace 414
  - cause 413
  - chaining 413
  - checked 378
  - ClassNotFoundException 1291
  - ConcurrentModificationException 811
  - customized 383
  - default handler 376
  - detail message 377
  - EOFException 1266
  - FileNotFoundException 1265
  - handler 373
  - IllegalThreadStateException 1412
  - IOException 1262, 1272
  - propagation 373
  - rethrowing 410
  - situations 378
  - suppressed 424
  - throw 395
  - throwing *see* throwing exceptions
  - thrown by JVM 378
  - thrown by method 101
  - thrown programmatically 378
  - throws 396
  - types 377
  - uncaught 376
  - unchecked 383
  - UnsupportedOperationException 806
- exchanging information 127
- execution 1403
- execution paths 1401
- execution stack
  - see* JVM stack
- Executor 1460
- executor 1459
- Executor Framework 1459
- executor service 1463
  - bulk task execution 1476
  - creating 1465, 1467
  - executor lifecycle 1464
  - shutdown 1465, 1468, 1469
  - submitting tasks 1465, 1467
  - task cancellation 1481
  - task execution policy 1462
  - tasks 1459
  - termination 1465, 1468, 1469
  - thread pool 1461
- Executors 1460, 1461
- ExecutorService 1463
- explicit
  - garbage collection 551
  - referencing 529
- explicit module 1233, 1235
- exploded module 1207
- exploded module directory 1207
- exponent 36
- exponential functions 491
- exports directive 1195, 1197, 1204
  - accessibility 1199



- exports-to directive 1204
- expression statements 56, 71, 101, 162, 178, 698
- expressions 168
  - actual parameters 128
  - boolean 75
  - conditional 92
  - deterministic evaluation 51
  - return 186
  - statements 101
  - switch expression 166
  - throw 395
  - yield 166
- extending 535
  - inner class 521
  - interfaces 248
  - Thread 1409
- extends 595, 603, 605
  - bounded type parameters 605
  - constraint 605
- extends clause 16, 195
- extensibility 1645
- extensions
  - .class 22
  - .java 22, 24
- external iteration 907
- extracting substrings 463, 476

## F

- fairness 1443
- fairness policy 1498, 1506
  - acquisition order 1498
- fallback 1134
- fall-through 158, 159, 167
- false literal 32, 37
- fetch size 1574
- FF *see* form feed
- field declarations 99
- field hiding 207, 235
- field initialization 569
- fields 7
  - final 234
  - hiding from superclass 207
  - shadowing by local variables 115
- FIFO queue 838
- file 777
  - see also* directory entry
- file attribute views 1360, 1365, 1366
  - view name 1359
- file attributes 1352
  - accessing by attribute name 1359
  - accessing by bulk operations 1360
  - accessing individually 1352
  - file attribute name format 1359
  - file attribute views 1360
  - file attributes interfaces 1362
  - file attributes interfaces 1360, 1362
  - File class 1320, 1322
  - file extension 1318
    - .properties 1127
  - file name 333
  - file path 343
    - separator character 343
  - file streams 1264
  - file system 1261, 1317
    - accessing 1648
    - default 1319
  - file system resources 1378
  - file visit option 1381
  - FileAlreadyExistsException 1340, 1371
  - FileAttribute interface 1371
  - FileAttributeView interface 1366
  - FileInputStream class 1263
  - FileNotFoundException 378, 380
  - FileOutputStream class 1264
  - FileOwnerAttributeView interface 1366
  - FileReader class 1270
  - files
    - binary 1266
    - low-level I/O streams 1264
    - not found 1265
  - Files class 1317
    - method characteristics 1334
    - streams 921
  - FileSystem class 1319
  - FileSystems class 1319
  - FileTime class 1355
  - FileVisitOption enum type 1382
  - FileWriter class 1272
  - filtering 704
  - filtering adapter 1013
  - filtering streams 928
    - using a Predicate 931
  - FilterInputStream class 1263
  - FilterOutputStream class 1264
  - filters 928
    - see also* I/O filter streams
  - final
    - effectively 539, 708
    - variables 539, 708
  - final fields 234
    - hiding 235
  - final instance fields

- initialization 235
- final keyword 230
- final local variables 234, 235
- final members 230
- final methods 231
  - binding of method call 231
- final parameters 136, 236
- final static fields
  - initialization 235
  - manifest constant 235
- final variables 233
  - blank 234
  - constant variables 234
  - definite assignment analysis 237
  - definitely assigned 237
  - definitely unassigned 238
  - fields 234
  - local variables 234
  - object state 233
- finalize method 437
- finally clause 386, 391
- finishing adapter 1018
- finite streams 910
  - extracting lines from a String 920
  - from a BufferedReader 920
  - from a CharSequence 919
  - from arrays 916
  - from collections 915
  - from factory methods in the Files class 921
  - from specified values 911
  - from the Random class 918
- first element 827, 828, 1524
- first map entry 863
- fixed arity method 137
- fixed arity method call 140
- Fixed Thread Pool 1462
- fixed-size list view 673
- flat mapping adapter 1015
- float 36, 43
- floating-point 41
  - double 43
  - float 43
  - literals 36
- floating-point arithmetic 61
- floating-point data types 36
- floating-point division 63
- floating-point remainder 64
- floor entry 864
- floor key 864
- flow control *see* control flow
- flushing I/O streams 1262
- for(;;) statement 176
  - backward 177
  - collections 811
  - forward 177
  - iterating over array 121
  - iterator 811
- for(:) statement 179, 812
  - iterating over array 122
- for-each loop 175
- Fork/Join Framework 1031, 1484
- forking 1484
- fork-join pool 1485
- ForkJoinPool 1485
- ForkJoinTask 1485
- form feed 38, 40
- formal parameters 100, 109, 128, 270, 363
  - modifier 100
  - name 100
  - type 100
- formal type parameter 581
- Format class 1139
- format elements 1166
  - format style 1166
  - format type 1166
- format specifications 26, 467
- format specifiers 26, 1275
- format style 1153, 1158, 1166
- format type 1166
- FormatStyle enum type 1153, 1157
- formatted output 26
  - format specifiers 26
- formatted prompting 1286
- formatted string 467
- formatted values
  - Console 1285
  - PrintStream 1275
  - PrintWriter 1275
- Formatter 1179
- formatter
  - destination object 1179
- formatting 40, 1140
- formatting date and time values 1153
- formatting messages 1165
  - conditional formatting 1172
- formatting numbers/currency values 1141
- forward reference 555, 556, 559, 565, 569
- fractional signed numbers 41
- frame stack
  - see* JVM stack
- framework for collections 799
- full permissions 1375
- fully 1218

- fully qualified class name 345, 1215, 1217, 1218
- fully qualified name 320, 353, 509
- fully qualified package name 334, 336, 338
- fully qualified type name 334, 335, 339
- function 700
- function arrow -> 695
- Function interface 729
  - composing functions 732
- function type 699
- functional interface
  - @FunctionalInterface 691, 1617
  - abstract method 691
  - built-in functional interfaces 711
  - function type 699
  - functional method 691
  - primitive values 712
  - target type 699
  - see also* lambda expressions
- functional method 691
- functional programming 6
- functional reduction 974, 975, 976, 993, 1004, 1018
- functional requirements (FRs) 1638
- FunctionalInterface annotation 691
- functionality 691
- functional-style programming 691
- functional-style programming paradigm 4
- functions
  - currying functions 739

## G

- garbage collection 550, 551
  - automatic 547
  - facilitate 551
- general abstractions 198
- general contract
  - compareTo() 778
  - equals() 762
  - hashCode() 772
- general loops 176
- generalization 15
- generalized classes 198
- generic class 581, 582
- generic instance method 608
- generic interface 586
- generic method 607, 624, 671
  - calling 608
- generic method call
  - inference 609
- generic static method 608

- generic subtype method 635
- generic type 581, 659
  - extending 587
  - implementing 612
- generics 147
  - arrays 641
  - casting 639
  - class literals 650
  - collections 592
  - constructors 607
  - enums 649
  - erasure 627, 634
  - exception handling 646
  - inherited methods 634
  - instanceof operator 638
  - JVM 579
  - method signature 630
  - methods 607
  - nested classes 647
  - non-reifiable types 638
  - overloading 629, 630
  - overriding 629, 631
  - reifiable types 637, 638
  - subsignature 630
  - type erasure 590, 627
  - variable arity parameter 644
  - wildcards 593
- GESS 604
- get methods 306, 308, 367, 604, 996, 1054, 1123, 1569, 1572, 1575
- get operation 600
- getter methods *see* get methods
- GMT (Greenwich Mean Time) 1096
- goto 32
- grammar rules 30
- grant 1646
- graph
  - nodes 1193
  - unidirectional edges 1193
- grouping 334
- grouping separator 1151
- groups of objects 799
- GUI applications 1412

## H

- half-open interval 883, 885, 1172
- handle-or-declare rule 397
- handles *see* references
- happens-before relationship 1449, 1490
  - transitivity 1449
- hard link

- see* symbolic link
- has-a* relationship 199
- hash code 108, 436, 769, 770
- hash function 769
- hash sign 1126
- hash table 770, 804, 805
  - see* map
- hash value 436
  - see* hash code
- hashing 769
  - collision 769
  - hash code 769
- HashSet 823
- Haskell Curry 740
- head map view 862, 863
- head set view 827, 828, 1525
- heap 547, 1403
- heap pollution 1623
- heap pollution warning 644, 664, 673, 822, 850, 881
- helper methods 257
- heuristics 772
- hidden members
  - accessing 518, 519
- hiding internals 332
- hierarchical file system 1261
- higher entry 864
- higher key 864
- high-level I/O streams 1261
- high-performance 6
- holdability 1578
- horizontal tab 38
- hotspots 6
- HT *see* horizontal tab
- human time 1071

## I

- I/O buffering 1266
- I/O filter streams 1263, 1264, 1266
- I/O package 1261
- I/O streams 1261
  - buffering 1266
  - chaining 1266
  - closing 1262, 1267, 1268
  - end of stream 1268
  - files 1264
  - filter streams 1266
  - flushing 1262
  - high-level 1261
  - input filter streams 1263
  - low-level 1261

- of binary values 1264, 1266
- of characters 1270
- of objects 1264
- output filter streams 1264
- skipping bytes 1267
- IANA Time Zone Database 1096
- IDE (integrated development environment) 1654
- idempotent method 417, 422
- identifiers 30
  - variable 102
- identity container 985
- identity conversion 47, 68
- identity of object 9
- identity() function 1002
- IEEE 754-1985 43
- if block 154
- if-else statement 155
  - scope of pattern variable 282
- illegal thread state 1412
- IllegalArgumentException 381, 827, 1161
- IllegalStateException 381, 810
- immediately enclosing instance 504
- immutability 1489, 1645
- immutable 308, 812, 1046, 1098
- immutable classes
  - in Java SE platform documentation 364
- immutable object pattern
  - see* immutable objects
- immutable objects 439, 450
  - advantages 368
  - caching 367
  - consistency of state 367
  - defensive copying 368
  - guidelines 366
  - state 364
  - static factory method 368
  - thread-safety 364
- immutable primitive values 364
- immutable strings 450
- implementation inheritance 250
- implementations 7, 197, 332
  - collections 799
  - inheritance hierarchy 230
- implementations of collections 802
- implementing
  - Comparable 777
  - Comparator 785
  - equals 760
  - hashCode 769
  - interfaces 245
- implements clause 245

- implicit
  - inheritance 195
  - narrowing conversions 68
- implicit canonical record constructor 308
- implicit default constructor 110
- implicit dependency 1193, 1198, 1244
- implied dependency 1201
- import 353
  - declaration 337
  - see also* static import
  - single-type-import declaration 337
  - statement 333
  - type-import-on-demand declaration 338
- importing
  - enum constants 341
  - nested types 510
  - reference types 337
  - static members 339
- in lockstep 814
- IN parameters 1568
- incidental whitespace 472
- increment operator 70
- indenting lines 465
- index 118
- index expression 121
- `IndexOutOfBoundsException` 409, 454, 455, 463, 475, 477, 478, 479, 665, 666, 670
- individual array elements 121
- individual module compilation 1213
- inequality 76
  - see also* equality
- inferred-type lambda parameters 147
- inferred-type parameters 696
- infinite loop 179
- infinite streams 910
  - building with generator functions 913
  - from the `Random` class 918
- infinity 61, 442
  - negative 61
  - positive 61
- information hiding 332
- inheritance 15, 199, 242
  - hierarchy 198, 520
  - serialization 1299
  - subtype-supertype relationship 199
- inheritance of implementation 242
- inheritance of type 242
- initial capacity 661
- initial state of object 569
- initialization
  - arrays 120, 125
  - code 120
  - default values 103
  - for statement 176
  - objects 10
  - references 103
  - variables 102
- initialization expression 102, 143
- initializer 554
  - declaration-before-reading rule 556
  - static 555, 556, 567, 568, 571
- initializer blocks
  - instance 565
  - static 559
- initializer expression 103, 234, 554
  - static 557
- initializers
  - declaration order 559
  - exception handling 563
  - forward references 559
  - non-static block 100
  - non-static field 100
  - static block 99
  - static field 99
  - using class name 561
  - using simple name 559
- initializing *see* initialization
- inlining 231
- inner classes 503, 504
  - anonymous 504, 507, 534
  - local 504, 506, 525
  - non-static member 504, 506
  - qualified superclass constructor
    - invocation 523
  - synchronization 1427
- `INOUT` parameters 1570
- input 1261
- input stream 1261
- input validation 1643
- `InputStream` abstract class 1262
- `InputStreamReader` class 1270
- `INSERT` statement 1553
- insertion order 658, 806, 824, 859
- insertion point 879, 885
- instance
  - members 14
  - methods 14, 101, 106
  - variable initialization 103
  - variables 14, 106
  - see also* object
- instance initializer blocks 565
  - local variable type inference 566
- instance methods 10
- instance variables 10, 569

- instanceof operators 93, 274, 280, 638
- instanceof pattern match operator 280
  - pattern variable 280
  - type pattern 280
- instanceof type comparison operator 275, 287
- instant
  - accessing fields 1074
  - comparing 1075
  - convert to temporal objects 1079
  - creating 1072
  - creating from date-time 1073
  - current instant 1072
  - epoch 1071
  - epoch-second 1071
  - immutable 1071
  - ISO format 1072
  - max value 1072
  - min value 1072
  - nano-of-second 1071
  - parsing 1073
  - plus/minus methods 1077
  - representation 1071
  - supported ChronoFields 1076
  - supported ChronoUnits 1078
  - temporal arithmetic 1077
  - text representation 1073
  - thread-safe 1071
  - timestamps 1071
  - with methods 1076
  - zero UTC offset 1071
- Instant class
  - see* instant
- instant timeline 1097
- instantiation 9
  - anonymous classes 538
  - local classes 530
  - non-static member classes 515
  - parameterized type 583
  - static member classes 507
- int 32, 42
- IntBinaryOperator interface 738
- IntConsumer functional interface 725
- integer arithmetic 60
- integer bitwise operators 84
- integer constant expressions 48
- integer data types 32
- integer division 62
- integer remainder operation 63
- integers 41
  - and* operator 84
  - byte 42
  - complement 84
  - data types 42
  - int 42
  - literals 32
  - long 42
  - or* operator 84
  - representation 55
  - short 42
  - types 42
  - xor* 84
- integral types 41, 42, 44
- interface constant antipattern 340
- interface-based 802
- interfaces
  - abstract 243
  - abstract methods 244
  - accessibility 353
  - body 242
  - collection core 799
  - constants 259
  - contract 244
  - declarations 333
  - default methods 251
  - evolution 255
  - extending 248
  - functional interfaces 691
  - header 242
  - implementing 245, 304, 537
  - inheriting from multiple interfaces 635
  - initialization 571
  - local 507, 533
  - markers 243
  - member declarations 243, 244
  - multiple inheritance of implementation 253
  - nested 503, 504
  - overriding default methods 252
  - partial implementation 247
  - private methods 257
  - public methods from Object class 250
  - realization 250
  - references 250
  - static member 506, 507
  - static methods 255
  - subinterfaces 249
  - superinterfaces 248
  - top-level 503, 506
  - UML 250
  - UML stereotype 250
  - variables 259
  - with single abstract method 694
- intermediate stream operations 903, 923

- examining elements 939
- filtering streams 931
- flattening streams 942
- input stream 903
- lazy execution 905
- mapping streams 940
- non-interfering behavioral parameters 928
- order of 926
- output stream 903
- replacing each element with multiple elements 946
- selecting distinct elements 933
- setting execution mode 952
- setting stream as unordered 951
- short-circuit execution 926, 935, 937
- skipping elements 934
- sorting streams 949
- stateful operations 926
- stateless behavioral parameters 928
- stateless operations 926
- truncating a stream 935
- internal iteration 907
- internal packages 1197
- internationalization 1120
- interned strings 450, 452
- interned values 443
- interpackage accessibility 332
- interrupt handling 1428
  - interrupt status 1428
  - InterruptedException 1428
  - spurious wakeup 1434
- InterruptedException 1428
- interruptible locking 1499, 1504
- intersection 821
- IntFunction interface 729
- IntPredicate functional interface 720
- intrinsic lock
  - see* object lock
- intrinsic locking 1422, 1496
- IntStream interface 907
- IntSummaryStatistics 908
- IntSupplier functional interface 716
- IntToDoubleFunction interface 729
- IntToLongFunction interface 729
- IntUnaryOperator interface 737
  - composing unary operators 737
- invocation
  - parameterized type 583
- invocation stack
  - see* JVM stack
- invoker 186

- invoking garbage collection 551
- IOException 378
- is-a* relationship 198, 199
- ISO 8859-1 encoding scheme 449
- ISO Latin-1 37, 42
- ISO standard 1153, 1155
- Iterable interface 460, 672, 808, 812
- iteration 177, 809
- iteration order 811
- iteration statements 174
  - next iteration 185
  - termination 175, 184
- iterators 659, 672, 810, 813
  - fail-fast 811, 1521
  - ListIterator 818
  - ordered collection 813
  - snapshot-style 1521
  - sorted collection 813
  - unordered collection 813
  - weakly consistent 1521

## J

- Jakarta EE (Enterprise Edition) 3
- JAR 1189
- JAR file 1215
- jar tool 1215
- java 1320, 1555
- java 23
- Java Application Launcher 23
  - see* JDK tool java
- Java Archive
  - see* JAR
- Java Archive Tool
  - see* JDK tool jar
- Java bytecode 22
- Java Card 3
- Java Class Dependency Analyzer Tool
  - see* JDK tool jdeps
- Java Collections Framework 658
- Java compilation unit 1194
- Java Database Connectivity
  - see* JDBC
- Java Development Kit (JDK) 3
- Java ecosystem 3
- Java EE (Enterprise Edition) 3
- Java Language Compiler 22
  - see* JDK tool javac
- Java Linker
  - see* JDK tool jlink
- Java ME (Micro Edition) 3
- Java Platform Module System 1189

- Java Platforms 3
- Java Runtime Environment (JRE) 4
- Java SE (Standard Edition) 3
- Java Shell Tool (jshe11) 24
- Java Virtual Machine (JVM) 3
- java.base module 1193, 1241
- java.io package 1261, 1320
- java.lang.annotation package 1595
- java.lang.Record class 309
- java.lang.reflect package 1625
- java.math package 495
- java.net package 1320
- java.nio package 1317
- java.nio.file package 1317
- java.nio.file.attribute package 1355
- java.se module 1194
- java.sql package 1555
- java.text package 1139
- java.time package 1046, 1140
- java.time.format package 1140
- java.util package 658
- java.util.concurrent package 1460
- java.util.function package 711
- javac 22
- Javadoc comment 40
  - @param tag 100, 112
  - @return tag 187
  - @throws tag 399
- javadoc tool 41
- JDBC
- JDBC API 1555
- JDBC Drivers
  - legacy drivers 1559
- JDBC drivers 1555
- JDBC interfaces 1557
- JDBC resources 1556
- JDBC URL 1558
- JDK 1654
  - licencing 4
- JDK tools
  - jar 1216, 1239, 1241, 1248
  - java 23, 1215, 1240, 1241, 1247
  - javac 22, 1213, 1246
  - javadoc 41
  - jdeps 1193, 1242, 1249
  - jlink 1231, 1250
  - jmod 1192
  - jshe11 24
  - summary of operations 1245
- JMOD archive 1192
- JMOD files 1240
- join

- starvation 1447
- joining 1484
- joining threads 1418
- JPMS 1189
- just-in-time (JIT) 5
- JVM 548, 551
  - heap 1403
  - JVM stack 1403
  - method area 1404
  - program counter (PC) 1403
  - runtime constant pool 1404
  - runtime data areas 1403
- JVM stack 373, 548, 1403

## K

- key 884
- key insertion order 858
- key objects 801
- key sets 847, 864
- key sorting 862
- key-based lookup 1129
- keys 847
- keystore 1646
- key-value entries 770
- key-value pairs 1126
- keywords 31
  - abstract 223, 244, 691
  - boolean 43
  - break statement 182
  - byte 42
  - case 158, 162, 166
  - catch 385
  - char 42
  - class 99, 243
  - contextual words 31
  - continue 185
  - default 158, 163, 169, 251
  - do 176
  - double 43
  - else 155
  - extends 195, 595, 603, 605
  - final 136, 230, 236, 318, 366, 409, 419, 1492
  - finally 391
  - float 43
  - for 176, 179
  - if 154
  - implements 245
  - import 337, 353
  - instanceof 93, 275, 280, 287, 638
  - int 42



- interface 242
- long 42
- new *see* new operator
- null 49, 78, 274
- package 336, 353
- private 360
- protected 359
- public 358
- reserved words 31
- return 186
- short 42
- static 21, 113, 256, 339, 507, 559
- super 110, 205, 210, 216, 253, 520, 596
- switch 157, 166
- synchronized 1423, 1492
- this 106, 518
- throw 395
- throws 396
- try 385
- unused words 31
- void 21, 439
- volatile 1490
- while 175, 176
- keywords (contextual)
  - exports 31, 1195, 1203
  - module 31, 1203
  - non-sealed 31
  - open 31, 1203
  - opens 31, 1203
  - permits 31
  - provides 31, 1203
  - record 31
  - requires 31, 1196, 1201, 1203
  - sealed 31
  - to 31, 1203
  - transitive 31, 1203
  - uses 31, 1203
  - var 31
  - with 31, 1203
  - yield 31
- keywords (reserved)
  - const 32
  - goto 32

## L

- labeled break statement 184
- labels 182, 184
  - break 184
  - case constants 158, 162
  - constant expressions 159
  - default 158, 163, 169
  - labeled statement 182
  - switch statement 158, 162
- lambda body 695, 697
- lambda expressions 695, 707
  - access class members 701
  - block scope 701, 703
  - compared with anonymous classes 708
  - contexts providing target types 750
  - declared-type parameters 696
  - deferred execution 696, 700
  - expression 698
  - expression statement 698
  - function 700
  - function arrow -> 695
  - implicitly typed 696
  - inferred-type parameters 696
  - lambda body 695, 697
  - lambda parameters 696
  - non-void return 697
  - parameter list 695
  - shadowing members 701
  - single expression 697
  - statement block 698
  - target type 699
  - target typing 700
  - type checking 699
  - using this and super 701
  - variable capture 703
  - var-type inferred parameters 696
  - void return 697
- lambda parameters 696
- language codes 1121
- language tag 1143
- last element 827, 828, 1524
- last map entry 863
- latch
  - see* count-down latch
- late binding
  - see* dynamic method lookup
- LATIN-1 encoding scheme 449, 450
- lazy execution 905, 925
- least significant bit 34
- left
  - shift 87
- left associativity 52
- legacy class 580
- legacy code 579
- legacy supertypes 633
- legal assignments 269
- length method 455, 920
- letter pattern 1161
- lexical nesting 509

- lexical scope
  - see* block scope
- lexical tokens 30
- lexicographic array comparison 887
- lexicographic comparison 887
- lexicographical ordering 457
- LF *see* linefeed
- lifetime 549
  - see* scope 106
- limits 1172
- line continuation 38
- line separator 26, 1275
- line terminator 40, 455
- linear implementation inheritance 197
- linear inheritance 242
- line-continuation character 1213
- linefeed 38
- lines of text 1281
- LinkageError 382
- linked data structures 580
- linked list 804, 818
- LinkedBlockingDeque 1535
- LinkedBlockingQueue 1535
- LinkedHashSet 823
- LinkedList 661, 818, 832, 839
- LinkedTransferQueue 1535
- LinkOption enum type 1332
- List 659, 817
- list views 670, 673, 675
- listing observable modules 1240
- ListIterator 818
- lists 800, 817, 818
  - ArrayList 818
  - binary search 878
  - changing contents 881
  - copying 881
  - LinkedList 818
  - List 817
  - ListIterator 818
  - positional 801
  - replacing elements 880, 881
  - reverse order 876
  - sorted 878
  - sorting 876, 877
  - sublist view 670
  - sublists 879
  - unmodifiable 663, 818
  - unmodifiable views 874
  - Vector 818
  - views 673
  - see also* ArrayList
- literals 32
- boolean 37
- character 37
- default type 33, 36
- double 36
- escape sequences 38
- false 37
- float 36
- floating-point 36
- integer 32
- null 32
- predefined 31
- prefix 33
- quoting 37
- scientific notation 36
- string 39
- suffix 33, 36
- true 37
- live threads 1417
- livelock 1445
- liveness 1443
- local 102, 104, 533
  - block 504
  - chaining of constructors 214, 569
  - classes 506, 525
  - enum types 504, 507
  - interfaces 504, 507
  - record classes 504, 507
  - variables 106, 363
- local classes 504, 506, 525
  - accessing local declarations 527
  - accessing members in enclosing context 528
  - declaration 525
  - hiding members from enclosing context 529
  - in non-static context 526
  - in static context 526
  - instantiation 530
  - scope 525
  - shadowing local variables 527
  - using qualified this 529
- local date
  - conversion to Date 1111, 1167
- local date-time
  - conversion to Date 1111, 1167
  - converting to instant 1074
- local declarations 101
- local enum types 533
- local interfaces 533
- local record class 533
- local scope 171
- local time 1096

- conversion to Date 1111, 1167
  - local timeline 1097
  - local variable type inference 143, 419
    - arrays 144
    - instance initializer blocks 566
    - parameterized types 585
    - static initializer blocks 559, 561
  - local variables 109, 143
    - block scope 701
  - LocalDate
    - see* temporal objects
  - LocalDateTime class
    - see* temporal objects
  - Locale 1121
  - locale 458, 1120
    - country codes 1121
    - country name 1122
    - default 1122
    - language codes 1121
    - language name 1122
    - predefined 1121
  - locale qualifiers 1122
  - locale-sensitive 1140
  - locale-sensitive operations 1120
  - localization 1120
  - localizing an application 1129
  - localizing information 332
  - LocalTime class
    - see* temporal objects
  - locations
    - see* class path
  - Lock 1497
  - lock 1415
    - acquire 1414, 1423
    - class 1425
    - entry set 1414, 1423
    - release 1414
  - lock acquisition
    - starvation 1448
  - logarithm 491
  - logical 283
  - logical AND
    - $\&$  79
  - logical complement
    - $!$  79
  - logical complement (!) operator
    - pattern variable 283
  - logical exclusive OR
    - $\wedge$  79
  - logical inclusive OR
    - $|$  79
  - logical XOR
    - $\wedge$  79
  - long 32, 42
    - suffix 33
  - LongBinaryOperator interface 738
  - LongConsumer functional interface 725
  - LongFunction interface 729
  - LongPredicate functional interface 720
  - LongStream interface 907
  - LongSummaryStatistics 908
  - LongSupplier functional interface 716
  - LongToDoubleFunction interface 729
  - LongToIntFunction interface 729
  - LongUnaryOperator interface 737
    - composing unary operators 737
  - loop body 174, 176
  - loop condition 174, 175, 176
  - loop variable 176
  - loops *see* iteration statements
  - loss of precision 44
  - lower bound 595
  - lower bounded wildcard
    - ? super T 596
  - lower bounded wildcard references 603
  - lower entry 864
  - lower key 864
  - low-level I/O streams 1261
  - LTS (long-term support) release 4
- ## M
- magnitude 44
  - main memory 1490
  - main method 23, 141
    - arguments 142
    - modifiers 142
  - main thread 1407, 1412
  - Main-Class attribute 1239, 1240
  - MANIFEST.MF file 1239, 1240
  - many-to-one relation 847
  - Map 847
  - map
    - key-value entries 770
  - map views 856
  - Map.Entry 850, 863
  - mapping adapter 1014
  - mappings 801
    - see* entries 801
  - map-reduce operation 1024
  - map-reduce transformation 977
  - maps 801, 847
    - access order 858
    - entries 801

- iteration 856
- key insertion order 858
- key-based operations 848, 852
- Map 847, 857
- map views 856
- mappings 801
- multimap 611
- NavigableMap 863
- ordering mode 858
- replacing values 856
- SortedMap 862
- submap view 863
- text representation 858
- TreeMap 864
- unmodifiable 849
- unmodifiable views 875
- views 856
- marker annotation 1601, 1603
- marker annotation type 1598
- marker interfaces 243, 1291
- marker parameter 1564, 1568
- mask
  - see* bit mask
- Math class 108, 488
- math constants 489
- mathematical set 821
- MAX\_VALUE constant 444
- maximum value 489
- member declarations 99, 313
- member type declarations 99
  - non-static member class 504
  - static member class 504
  - static member enum type 504
  - static member interface 504
  - static record class 504
- members 8, 361
  - access 106
  - accessibility 355
  - default values 103
  - inheritance 195
  - modified 195
  - of objects 10
  - scope 361
  - shorthand 108
  - static 11
  - terminology 14
  - variables *see* fields
- memory consistency errors 1449
- memory management 547
- memory organization 547
- MessageFormat 1140
- meta-annotation 1597, 1605
  - @Documented 1612
  - @Inherited 1612
  - @Repeatable 1613
  - @Retention 1605
  - @Target 1607
- metadata 243, 1595
- META-INF directory 1239
- method area 1404
- method call 11, 100, 127
  - chaining 477, 479
  - fixed arity 140
  - variable arity 140
- method chaining 924, 1058, 1064, 1083, 1092
- method declaration 99
  - variable arity parameter 644
- method header 228
- method invocation conversions 48, 270
- method modifiers 100
- method naming conventions 1047
- method overloading 108, 206, 229
- method overriding 206, 228, 231, 570
  - covariant return 205
- method references 740, 742
  - bounded instance method references 744
  - static method references 742
  - unbounded instance method references 745
- method signature 101, 201
- method type 699
- methods 8
  - @Override 202
  - abstract 228, 244, 691
  - abstract method declarations 244
  - advisory 1420, 1421
  - ambiguous call 206, 271
  - automatic variables *see* local variables
  - behavior 691
  - binding of method call 210
  - blocks 101
  - body 100
  - bridge 629
  - call chaining 128
  - call *see* method call
  - calling variable arity method 138
  - chained 459
  - clone 436
  - concrete 230, 231
  - declaration 100, 128
  - default 251
  - dynamic lookup 288
  - equals 78, 436, 760

- exceptions 101
- final 231
- finalize 437
- fixed arity 137
- functional 691
- generic 607
- getClass 436
- header 100
- hiding 231
- hiding static methods 208
- implementation 228
- invocation *see* method call
- local declarations 101
- local variables
- locale-sensitive 1123
- main *see* main method
- method invocation conversions 270
- method type 699
- modifiers 100
- most specific 271, 669
- mutually exclusive 1426
- name 128
- objects 106
- overloaded resolution 271
- overloading *see* method overloading
- overriding *see* method overriding
- overriding vs. overloading 206
- parameters 100
- return 186
- return value 100
- signature 101, 108, 206, 630
- static 115, 256
- subsignature 631
- synchronized 1423
- termination 186
- throws clause 396
- toString 436
- variable arity 137
- migration to modules 1236
- MIN\_VALUE constant 444
- minimizing overhead 550
- minimum value 489
- MissingResourceException 382, 1129
- mixing raw types and generic types 589
- mobile applications 3
- modifiers
  - abstract 244
  - access 353
  - default 251
  - final 230
  - static 256
- modular JAR 1216, 1233
  - creating 1216
  - describe a JAR 1241
  - extract contents 1239
  - list contents 1239
  - listing observable modules 1240
  - Main-Class attribute 1239, 1240
  - MANIFEST.MF file 1239, 1240
  - META-INF directory 1239
  - running from 1218
- modular JDK 4, 1190
  - goals 1191
  - java.base module 1193, 1241
  - java.se module 1194
- modular JDK installation 1191
- module application directory structure 1206
- module declaration 1194
  - module as contextual keyword 1195
- module dependency 1196
- module descriptor 1196, 1214
- module directives 1195
  - exports directive 1195, 1197, 1204
  - exports-to directive 1204
  - opens directive 1204, 1218, 1219
  - opens-to directive 1204, 1222
  - overview of directives 1203
  - provides-with directive 1204, 1227
  - qualified export 1204
  - requires directive 1195, 1196, 1203
  - requires static directive 1203
  - requires transitive directive 1201, 1203
  - unqualified export 1195, 1204
  - uses directive 1204, 1229
- module graph 1193, 1198, 1206, 1244
  - acyclic 1198
  - dependency cycles 1198
  - implicit dependency 1193, 1198
  - implied dependency 1201
  - module dependency 1196
  - sink nodes 1199
- module name 1197
- module path 1233
- module resolution 1198, 1213
- module root directory 1207
- module-info.java file 1207
- modules 1194
  - accessibility 1199
  - accessibility rules 1197, 1200
  - aggregate module 1194
  - automatic 1233, 1234
  - benefits 1189
  - comparison 1235

- creating modular JAR 1216
  - dependencies 1193
  - explicit 1233, 1235
  - exploded module 1207
  - exploded module directory 1207
  - exploring modules 1238
  - implicit dependency 1244
  - individual module compilation 1213
  - internal packages 1197
  - migration to 1236
  - modular JAR 1216
  - module application directory structure 1206
  - module declaration 1194
  - module descriptor 1196
  - module graph 1193
  - module name 1197
  - module resolution 1213
  - module root directory 1207
  - module-info.java file 1207
  - multi-module compilation 1214
  - named 1235
  - non-standard modules 1192
  - open modifier 1204
  - platform modules 1192
  - principle exported package 1197
  - public API 1189, 1197, 1200
  - readability 1199
  - readable 1197
  - reliable configuration 1189
  - running from a modular JAR 1218
  - services 1223
  - standard modules 1192
  - stronger encapsulation 1189
  - system modules 1240
  - unnamed 1233
  - viewing dependencies 1242
  - monitor 1415
    - entry set 1414
    - lock 1414
  - monitor lock
    - see* object lock
  - Month enum type 1050
  - most specific method 271, 669
  - multi-catch clause 406
    - final parameter 409
  - multicore architectures 6
  - multidimensional arrays 123, 124
  - multi-element annotation type 1599
  - multilevel classification 1008
  - multilevel grouping 1008
  - multilevel partitioning 1012
  - multimap 611
  - multi-module compilation 1214
  - multi-paradigm programming 4
  - multiple assignments 56
  - multiple bounds 606
  - multiple catch clauses 384, 406
  - multiple inheritance of implementation 197, 242, 253
  - multiple inheritance of type 242
  - multiple-line comment 40
  - multiplication operator
    - \* 62
  - multiplicative operators 62
  - multisets 821
  - multitasking 1401
  - multithreaded 6
  - multithreaded programming 1402
  - mutability 675
  - mutable character sequences 474
  - mutable containers 967, 984
  - mutable objects 367
  - mutable reduction 991, 992
    - mutable containers 984
    - see also* collectors
  - mutator method *see* set methods
  - mutex
    - see* lock
  - mutual exclusion 1492
  - mutually comparable 876, 878, 885
  - mutually exclusive
    - actions 156
    - locks 1414
  - MVC design pattern 1205
- ## N
- name 30
  - name clash 634, 635, 636
  - name elements 1318
  - name separator in path 1318, 1321
  - named constants 235
  - named modules 320, 1235
  - named package 320
  - namespaces 110, 1197
  - naming conventions of paths 1318
  - NaN 442
  - NaN (Noit-a-Number) 1644
  - NaN (Not-a-Number) 62
  - nano-of-second 1071
  - narrower range 45
  - narrowing conversions
    - primitive 44

- reference 45
- narrowing reference conversions 200, 275, 599, 619
- natural logarithms 489, 491
- natural order 777, 876, 883
- natural ordering 883
- navigable map 801
  - first entry 863
  - last entry 863
  - range-views 863
  - views 863
- navigable set 801, 827
  - ceiling 828, 1525
  - floor 828, 1525
  - higher 828, 1525
  - iterator 828, 1525
  - lower 828, 1525
- `NavigableMap` 863
- `NavigableSet` 827
- navigation methods 863
  - closest match 828
- `nbsp`
  - see* non-breaking space
- negative zero 61
- nested classes 504
  - anonymous 504, 507, 534
  - class file name 509
  - fully qualified name 509
  - generic 647
  - local 504, 506, 525
  - non-static member 504, 506, 514
  - qualified name 509
  - static member 503, 506, 507
  - synchronization 1427
- nested enum types
  - local 507
  - static member 503, 506
- nested interfaces 503
  - static member 506
- nested lists 662
- nested loops 127
- nested parameterization 589
- nested parameterized types 622
- nested record classes 504
  - local 507
  - static member 506
- nested type declaration 503
- nested types 503
  - accessibility 505
  - enclosing context 505
  - enclosing instance 505
  - importing 510
- nested wildcards 598
- new operator 9, 93, 109, 119, 520, 534, 569
  - qualified class instance creation
    - expression 515
- NEW state 1416, 1421
- newline *see* linefeed
- NIO.2 1317
- NL *see* newline
- no-argument constructor 110, 214, 218
- `NoClassDefFoundError` 383
- nodes 580, 1193
- non-access method modifier 99, 100
- non-associativity 51
- non-atomic operation 1491
- non-blocking collection 1520
- non-blocking lock-free synchronization 1493
- non-breaking space (`nbsp`) 1142, 1143, 1145
- non-functional requirements (NFRs) 1638
- non-interfering behavioral parameters 928
- non-modular JARS
  - see* plain JARS
- non-observable substate 1415
- non-reifiable types 637, 645
- non-runnable states 1415
- non-standard modules 1192
- non-static code 100
  - see* non-static context 99
- non-static context 99, 100
- non-static field 14
- non-static field initializers 100
- non-static initializer block 100
- non-static member classes 506
  - access hidden members 518
  - access in enclosing context 517
  - declaration 514
  - fully qualified name 509
  - hiding in enclosing context 520
  - inheritance 520
  - instantiation 515
  - qualified class instance creation
    - expression 515, 516
  - qualified name 509
  - qualified superclass constructor
    - invocation 523
  - qualified `this` 518
- non-static nested classes
  - see* inner classes
- non-static type 582
- non-varargs call
  - see* fixed arity call
- non-void return 697

- non-zero argument constructor 111, 218, 219
- normal annotation 1601
  - array element initializer 1602
  - element-value pair 1601
  - single-element array-valued element-value pair 1602
- normal canonical record constructor 310, 311
- normal execution 374
- normal non-canonical record constructor 310, 313
- normal threads 1412
- NoSuchElementException 810, 828, 832
- NoSuchFileException 1337, 1338
- Not-a-Number
  - see* NAN
- notifying threads 437, 1431
- NotSerializableException 378, 380
- null literal 32
  - casting 49
  - equality comparison 78
- null reference 32
  - casting 274
- nulling references 550
- NullPointerException 381
- null-safe 759
- Number class 444
- number format pattern symbols 1151
- number formatter 1142, 1144
  - accounting currency 1143
  - compact form 1145
  - currency 1141, 1142
  - numbers 1141
  - percentage 1141, 1142
- number formatting
  - accounting currency 1143
  - compact form 1145
  - compact form styles 1146
  - pattern symbols 1150
  - static factory methods 1141
  - string pattern 1150
- number systems
  - base 33
  - decimal 33
  - hexadecimal 33
  - octal 33
  - radix 33
- NumberFormat 1140, 1141
- NumberFormatException 381, 406, 441
- numbers 1120, 1141
  - formatting 1141

- numeric optional classes 964
- numeric promotions 50
  - assignment 57
  - binary 50
  - unary 50
- numeric streams 902, 910
  - averaging 994
  - from a range 917
  - from arrays 916
  - from the Random class 918
  - functional reduction 993
  - summary statistics 993, 995
  - summation 994
- numeric types 41
- numeric wrapper classes 444
- numerical literals
  - using underscore 36

## O

- ObjDoubleConsumer functional interface 726
- Object class 198, 435
- object creation 1299
- object graph 1295
- object hierarchy 199
- object lock 1415, 1423, 1426, 1432, 1433, 1434, 1443
- Object lock rule 1449
- object reference equality 761
- object references 9, 102
- object serialization
  - see* serialization
- object state 10, 109, 132, 569
  - immutable 364, 1490
- object streams 902, 910
- object type *see* dynamic type
- object value equality 761
- ObjectInputStream class 1264
- object-level lock
  - see* object lock
- object-oriented programming 7
- object-oriented programming paradigm 4
- ObjectOutputStream class 1264, 1290
- objects 9
  - aggregate 17
  - alive 548
  - arrays 118
  - behaviors 7
  - callee 127
  - caller 127
  - Class class 436
  - cleaning up 550



- cloning 437
- communication 127
- comparing 436, 760, 785
- compound 548
- constituent 17, 548
- constructing 569
- contract 332
- creation 1299
- decoupling 288
- deserialization 1289
- eligible 550
- equality 78, 436, 760
- exchanging information 127
- garbage collection 547
- I/O streams 1264
- identity 9
- immutable 439
- implementation 332
- initial state 569
- initialization 10, 109
- initializer block 565
- internals 332
- lifetime 549
- members 10
- methods 106
- Object class 435
- persistence 1289
- properties 7
- reachable 547, 548
- reading 1289
- serialization 1289
- services 332
- state 233
- state *see* object state
- value equality 78
- wait set for threads 1431
- writing 1289
- Objects class
  - comparing objects 759
  - computing hash code 759
  - null-safe methods 759
- ObjIntConsumer functional interface 725
- ObjLongConsumer functional interface 726
- observable modules 1240
- on/off latch 1512
- one's complement 35
- one-arity 711
- one-dimensional arrays 118
- one-shot tasks 1476
- one-to-many mapping 942, 946
- one-to-one mapping 940
- one-to-one transformation 946
- online proctored exam 1656
- on-the-fly classes 504
- open modifier 1204, 1223
- open module 1223
- open package 1219
- open range-view operations 817
- opening delimiter 468
- OpenOption interface 1332
- opens directive 1204, 1218, 1219
- opens-to directive 1204, 1222
- operands 49
  - evaluation order 52
- operations 7
  - put 600
  - read 600
  - write 600
- operators 51
  - 59, 64
  - 70
  - ^ 79, 84
  - ^= 80, 86
  - ! 79
  - != 76, 77
  - ? : 92
  - . 11, 334
  - [] 93, 121
  - \* 59, 62
  - \*= 67
  - / 59, 62
  - /= 67
  - & 79, 84
  - && 81
  - &= 80, 86
  - % 59, 62, 63
  - %= 67
  - + 59, 64, 68
  - ++ 70
  - += 67
  - < 75
  - << 87, 88
  - <=<= 91
  - <= 75
  - = 67
  - = 55
  - == 76, 77
  - > 93
  - > 75
  - >= 75
  - >> 87, 89
  - >>= 91
  - >>> 87
  - >>> 90

- >>= 91
- | 79, 84
- |= 80, 86
- || 81
- ~ 84
- arithmetic compound assignment 67
- assignment 51, 55
- associativity 51
- binary 51
- bitwise 84, 87
- boolean 75, 76, 79
- cast 51
- comparisons 75
- compound assignment 80, 86
- conditional 81, 92
- decrement 70
- dot 11
- equality 76
- execution order 52
- extended assignments 91
- floating-point 61
- floating-point division 63
- floating-point remainder 64
- increment 70
- instanceof 93
- instanceof pattern match operator 280
- instanceof type comparison operator 275
- integer 84
- integer arithmetic 60
- integer division 62
- integer remainder 63
- logical 79
- multiplicative 62
- new *see* new operator
- overflow 61
- overloaded 60, 62
- postfix 51
- precedence 51
- relational 75
- shift 87
- short-circuited 81
- string concatenation 68
- ternary 51
- unary 51, 62
- unary - 62
- unary + 62
- optimizations 6
- optional annotation element 1600, 1604
- Optional class 908, 959
  - absence of a value 959
  - creating 960
  - declaring and returning 960
  - querying 962
    - see also* numeric optional classes 964
- optional dependency 1203
- optional operations 806
- OptionalDouble 908, 964
- OptionalInt 908, 964
- OptionalLong 908, 964
- optionally bounded collections 1520
- or operator 84
- Oracle CertView 1655
- Oracle exam attempt 1655
- Oracle JDK 4
- Oracle MyLearn 1655
- ordered collections 800, 817
- ordered streams 909
  - extracting lines from a String 920
  - from a BufferedReader 920
  - from a CharSequence 919
  - from factory methods in the Files class 921
- ordering mode 858
- ordinal value 165, 299
- OUT parameters 1569
- outer instance 521, 523
- OutOfMemoryError 382, 552
- output 25, 1261
- output stream 1261
- OutputStream abstract class 1262
- OutputStreamWriter class 1272
- overflow 34, 61
- overflow errors 491
- overloaded 60
- overloaded method resolution 271
- overloaded methods
  - ambiguous call 206
- overloading 630
  - constructors 112
  - method resolution 271
  - methods 108, 206
- overloading vs. overriding 206
- override criteria 631
- override-equivalent 630, 635
- overriding 399, 635
  - @Override 632, 1616
  - covariant return
  - equals 78, 760
  - hashCode 769
  - methods 206
  - override-equivalent 630
  - toString 69
- overriding vs. overloading 206

**P**

- package accessibility for class members 356, 359
- package accessibility for top-level types 353
- package directory 343
- package statement 333, 336, 353
- package-private accessibility for class members
  - see* package accessibility for class members
- package-private accessibility for top-level types
  - see* package accessibility for top-level types
- packages 334
  - classpath 347
  - declaration 333
  - definition 336
  - destination directory 343
  - hierarchy 334
  - java.io 1261
  - java.lang 435
  - java.util 799
  - location 347
  - members 334
  - package directory 343
  - root 347
  - running code from 345
  - shorthand 337
  - statement *see* package statement
  - unnamed 336
  - using 337
- palindromes 704
- parallel processing 1484
- parallel streams 909, 1030
  - autoboxing 1037
  - behavioral parameters 928
  - benchmarking 1031
  - building parallel streams 953, 1030
  - collecting to a `ConcurrentMap` 1004
  - computation-intensive operations 1032
  - concatenating 914
  - concurrent collectors 1004, 1010
  - data size 1032
  - determine if parallel 953
  - encounter order 972
  - execution 982, 986, 1031
  - factors affecting performance 1031
  - from collections 915
  - grouping to a `ConcurrentMap` 1010
  - ordered 910
  - ordering 1036
  - parallel functional reduction 982
  - parallel mutable reduction 986
  - setting parallel execution mode 952, 953
  - shared state 1036
  - side effects 928, 968
    - non-interfering behaviors 1036
    - stateless behaviors 1036
  - splittable 1032
  - stateful operations 930, 1036
  - unordered 910
- parallelism 1402
- parameter list 695
- parameter list *see* formal parameters
- parameter passing
  - by value 128
  - variable arity 137
- parameterized types 583, 621
  - as formal parameters 623
- parameters 100
  - actual 128
  - array elements 134
  - effectively final 409
  - final 136, 236, 409
  - fixed arity 137
  - formal *see* formal parameters
  - implicit 106
  - main method 142
  - passing 127
  - primitives 129
  - program 142
  - references 131
  - this 106
  - variable arity 137
- parent class 195
- parent directory (`..`) 1319
- parent resource bundle 1133
- parentheses 51
- `ParseException` 380
- parsing 1140
  - compact number form 1147
  - currency values 1145
  - date and time 1153
  - numbers 1144
  - percentage values 1145
  - using patterns 1177
- parsing numeric values 444
- partial application 740
- partial implementation 223, 247
- partly synchronizing a method 1426
- pass by value 128
- passing
  - parameters 127

- references 131
  - variable arity parameter 137
- passwords
  - reading from console 1285
- path 1318
  - comparing with another path 1333
  - constructing relative path to another path 1331
  - converting 1327
  - converting to a real path 1332
  - converting to an absolute path 1328
  - creating 1320, 1321
  - interoperability with `File` class 1322
  - interoperability with `URI` class 1323
  - name elements 1318
  - name separator 1318, 1321
  - naming conventions 1318
  - normalizing 1329
  - querying 1324
  - resolving with another path 1330
- Path interface 1317
  - see also* path
- paths
  - see* class path
- paths of execution 1401
- path-separator character 348
- pattern letters 1153, 1161
- pattern match operator 280
- pattern matching 280
- pattern symbols 1150
- pattern variable 280
  - cannot shadow local variable 281
  - declare final 281
  - scope 280, 282
  - shadow a field 281
- patterns
  - format elements 1166
- percentage 1141, 1142
  - formatting 1141, 1144
- percentage formatter 1142, 1145
- performance 6
- period 1079
  - compared with duration 1095
  - creating 1080
  - date-based 1079, 1087
  - equality 1083
  - get methods 1081
  - immutable 1080
  - normalization 1083
  - parsing 1080
  - period-based loop 1085
  - plus/minus methods 1083
  - supported `ChronoUnits` 1082
  - temporal arithmetic 1083
  - text representation 1080, 1081, 1089
  - thread-safe 1080
    - with methods 1083
- Period class 1079
  - see* period
- periodic tasks 1478
- permits clause 319
- permitted direct subclasses 318
  - accessibility 320
  - contract 319
  - extending direct sealed superclass 319
  - final 319
  - locality 320
  - non-sealed 320
  - sealed 320
- permitted direct subtypes
  - contract 322
  - enum types 323
  - record classes 323
- permute elements 878
- persistence
  - see* serialization
- plain data 305
- plain data class 305
- plain data objects 305
- plain JAR 1216, 1233
- Plain Old Java Objects (POJOs) 305
- platform modules 1192
- plurals 1173
- poison value 1538
- POJOs 305
- polled locking 1499, 1504
- polymorphic method 607
- polymorphism 265, 284, 288, 532
- polynomial 775
- portability 5
- Portable Operating System Interface (POSIX) 1357
- positional order 658
- position-based access 817
- positive zero 61
- POSIX file permissions
  - enum type 1357
  - full permissions 1375
  - group 1357
  - others 1357
  - owner 1357
  - read (r), write (w), execute (x) 1357
  - string format 1357
- PosixFileAttributes interface 1362, 1364

- PosixFileAttributeView interface 1366, 1368
- PosixFilePermission enum type 1357
- PosixFilePermissions class 1358
- postfix operators 51
- power 491
- precedence rules 52
- precision 57
- predefined literals 31
- predicate 705
- Predicate functional interface 719
  - composing predicates 721
  - primitive type specializations 722
- preempted threads 1420
- preemptive scheduling 1420
- prefix 33
  - 0 33
  - 0x 33
- prepared statement 1564
  - substitutional parameterization 1564
- primitive data types 41
  - see* primitive types
- primitive types 44
  - autoboxing 440
  - unboxing 442
  - see also* primitive values
- primitive values
  - assignment 55
  - binary representation 1266
  - equality 76
  - passing 129
- principle exported package 1197
- principle of least privilege (PoLP) 1649
- print writers 1274
  - formatted values 1275
  - text representation of values 1275
- printing values 25
- PrintWriter class 1272
- priorities 1419
- priority
  - priority queue 833
- priority heap 805, 832
- priority of thread 1420
  - starvation 1447
- priority queue 805, 831
- PriorityBlockingQueue 1535
- PriorityQueue 832
- private members 360
- privileged code 1647
- privileges 1647
- PRNG
  - see* pseudorandom number generators 493
- procedural programming paradigm 4
- process of elimination 1657
- process-based multitasking 1401
- processes 1401
- program
  - application 21
  - arguments 142
  - command line 142
  - compiling 22
  - correctness 1420
  - formatting 40
  - performance 1420
  - running 23
  - single-file source-code 23
- program arguments 142
- program counter (PC) 1403
- program output 25
- programmable locking 1497
  - interruptible locking 1499, 1504
  - lock acquisition 1497
  - lock disciplines 1497
  - polled locking 1499, 1504
  - read-write locking 1502
  - reentrant 1501
  - reentrant locking 1497
  - timed locking 1499, 1504
  - unconditional locking 1503
- programming to interfaces 661, 1223
- proleptic year 1051
- promotion 50
- proper 888
- proper prefix 888, 890
- properties 7
  - see also* class members
- properties file
  - comment 1126
  - escaping whitespace 1125
  - key-value pair syntax 1125
  - key-value pairs 1124
  - properties 1124
  - property list 1125
  - text file 1126
  - see also* Properties table
  - see also* property resource file
- Properties table 1126
- property
  - key-value pair 1124
  - see also* properties file
  - see also* Properties table
  - see also* property resource file
- property list 1124
  - see also* properties file

- see also* property resource file
- property resource file 1127
  - .properties file extension 1127
  - naming scheme 1127
  - properties 1127
  - see also* properties file
  - see also* resource bundle
- PropertyResourceBundle 1129, 1133
- protected members 359
- provides-with directive 1204, 1227
- pseudorandom generator 1458
- pseudorandom number generators 918
  - see* Math class
  - see* Random class
- pseudorandom numbers 493
- public 21
- public API 1189, 1197, 1200
- public members 358
- punctuators 32

## Q

- qualified class instance creation expression
  - 515, 516
- qualified export 1202
  - see* exports-to directive
- qualified name 256, 504, 509, 510, 513, 514
- qualified opens directive
  - see* opens-to directive
- qualified superclass constructor invocation
  - 523
- qualified this 518, 529
- query results 1572
- Queue 831
- queue 801, 804
  - capacity-restricted 831
  - FIFO 838
  - head 831
  - LinkedList 832
  - PriorityQueue 832
  - Queue 831
  - unbounded 831
- quotation mark 38, 39
- quotient 63

## R

- race conditions 1036, 1424
- radix
  - prefix 33
  - see* base 441
- ragged arrays 125

- raised to power 491
- random access by index 818
- Random class 493, 1458
- random numbers 493
- random order 878
- range
  - character values 42
  - floating-point values 43
  - integer values 42
- range of values for date fields 1051
- range of values for time fields 1051
- range-view 670, 817, 827, 828, 863, 1525
- ranking criteria 658
- raw types 585, 589, 877
  - unchecked warnings 589
- read lock 1503
- read operation 555
- readability 1199
- readable 1197
- readers 1270
- Read-Evaluate-Print Loop (REPL) 24
- reading binary values 1266
- reading bytes 1262
- reading characters 1278, 1279
- reading data 1261
- ReadLock 1503
- read-write lock
  - fairness policy 1506
  - read lock 1503
  - reentrant 1507
  - write lock 1503
- read-write locking 1502
- realization 250
- reclaiming memory 547
- record classes 170, 305, 306, 900
  - augmenting 310
  - basic syntax 306
  - compact canonical record constructor
    - 310, 312
  - component fields 308
  - component list 306
  - generating methods of the Object class
    - 309
  - generic 317
  - get methods 306, 308
  - immutable 308
  - implementing interfaces 316
  - implicit canonical record constructor 308
  - local 507, 533
  - member declarations 313
  - nested 317, 504
  - normal canonical record constructor 310,

- 311
- normal non-canonical record constructor
  - 310, 313
- permitted direct subtypes 323
- permitted in sealed interfaces 317
- record construction 309
- record elements 306
- record header 306
- record name 306
- record serialization 316
- restrictions 309
- serialization 1292
- shallowly immutable 308
- static member 506
- top-level 506
- record components 306
- record header 306
- record name 306
- recursive type bound 605, 627
- RecursiveAction 1484
- RecursiveTask 1484
- reduced graph 1193
- reducing complexity 332
- reduction operations 966
- reentrant 1501
- reentrant locking 1497
  - fairness policy 1498
- reentrant synchronization 1425
- ReentrantLock 1497
- reference types 99, 102, 199, 503
  - classes 99
  - enum types 293, 294
  - interface types 242, 250
- reference values 9
- reference variables 102
- references 9, 12, 102, 113, 128
  - abstract types 225
  - aliases 78, 131
  - array 119, 121, 265
  - assignment 55
  - casting 49, 274
  - dangling 547
  - downcasting 45
  - dynamic type 200
  - equality 77
  - field 548
  - local 548
  - narrowing conversions 45
  - null *see* null reference
  - parameterized types 583
  - passing 131
  - raw types 602
  - reachable 547, 548
  - static type 200
  - super 210, 529
  - this 106, 529
  - upcasting 45
  - widening conversions 45
- reflection 643, 1191, 1204
- Reflection API 1625
- reflective access 1218
- reifiable types 637
- relational databases 1550
- relational model 1550
- relational operators 75
- relational tables
  - columns 1550
  - constraints 1550
  - rows 1551
- relative adjusters 1062
- relative path 1318
- reliability 6
- reliable configuration 1189
- remainder 63
- remainder operator
  - % 63
- remove whitespace 463
- repeatable annotation type 1614
- replacing characters 462
- required annotation element 1600
- requires directive 1195, 1196, 1203
  - readability 1199
- requires static directive 1203
- requires transitive directive 1201, 1203
- reserved keywords 31
- reserved literals
  - false 32, 37
  - null *see* null reference
  - true 32, 37
- resizable arrays 804, 805, 818
- resource
  - see* resource bundle
- resource bundle 1126
  - base name 1126
  - candidate bundle name 1132
  - default resource bundle 1127, 1128, 1134
  - fallback 1134
  - key-based lookup 1129
  - key-value pairs 1126
  - locating resources 1132
  - naming convention 1126
  - parent resource bundle 1133
  - property resource file 1127
  - resource 1126

- resource bundle class 1127
  - result bundle 1133
  - resource bundle class 1127
  - resource bundle family 1126
    - see also* resource bundle
  - resource declaration statement 419
  - resource leakage 416
  - resource variable
    - `final` 419
  - `ResourceBundle` 1126, 1127
  - resources 415, 550
  - restricted type name var 100
  - result bundle 1133
  - result set 1572
    - concurrency 1577, 1578, 1579
    - concurrency mode 1579
    - cursor 1572
    - customizing 1577
    - extracting column values 1575
    - fetch size 1574
    - holdability 1577, 1578, 1579
    - holdable 1578
    - metadata 1581, 1582
    - navigational direction 1577, 1578
    - scrollable 1578
    - type 1577, 1578
    - updatability 1577, 1578
  - retaining elements 807
  - retention policy values for `@Retention` 1607
  - rethrowing exceptions 410
  - return statement 186
    - `@return` tag 187
  - return type
    - covariant
  - return value 11
  - reuse of code 195, 199
  - reverse DNS 335, 1197
  - reverse-order key set 864
  - reverse-order view 864
  - rhyming order 787
  - right associativity 52
  - rightmost bit 34
  - ripple effect 199
  - robustness 6, 425, 1645
  - role relationship 199
  - root
    - see* inheritance hierarchy
  - root component 1317
  - root node 1317
  - round method 490
  - rounding functions 489
  - rounding mode 1148
  - rounding numbers 490
  - round-robin scheduling 1420
  - rows 1551
  - `Runnable` functional interface 1405, 1410, 1460
  - `RUNNABLE` state 1417
  - Running 1231
  - running a Java application 23
  - running from a modular JAR 1218
  - running from a runtime image 1232
  - runtime
    - bounds checking 121
  - runtime checks 49, 662
  - `Runtime` class 435, 551
  - runtime class 436
  - runtime constant pool 1404
  - runtime data areas 1403
  - runtime environment 547, 1124
  - runtime image 4, 1231
  - runtime stack
    - see* JVM stack
  - runtime type
    - see* dynamic type
  - `RuntimeException` 380, 406
- ## S
- scheduled executor service
    - scheduling one-shot tasks 1476
    - scheduling periodic tasks 1478
  - scheduled thread pool 1463
  - `ScheduledExecutorService` 1476
  - schedulers 1420
  - schema 1323
  - scientific notation 36
  - scope 361
    - block 363
    - catch clause 389, 419
    - class 361
    - disjoint 363
  - scope of local variables 332
  - sealed classes 318
    - direct sealed superclass 319
    - in compilation unit 323
    - permits clause 319
    - permitted direct subclasses 318, 319
  - sealed interfaces
    - direct sealed superinterface 322
    - in compilation unit 323
    - permits clause 321
    - permitted direct subtypes 321
  - searching 799, 874



- arrays 884
- collections 878
- searching in string 461
- searching in string builder 476
- secure 6
- secure coding 1638
- security guidelines 1648
- security policies 1646
- security threats 1640
- SecurityManager class 435
- seed 495
- SELECT statement 1553
- selection statements 154
- selector expression 162
- self-reference 147
- self-referential data structure 580
- semantic definition 30
- semicolon 101
- sensitive information leakage 1641
- separators 32, 51
- sequence 800, 801
- sequential I/O 1261
- sequential streams 909
  - execution 977
  - extracting lines from a String 920
  - from a BufferedReader 920
  - from a CharSequence 919
  - from arrays 916
  - from collections 915
  - from factory methods in the Files class 921
  - ordered 910
  - unordered 910
- serial access 1515
- serialization 1289
  - customizing 1297
  - inheritance 1299
  - serialVersionUID 1303
  - transient fields 1296
  - versioning 1302
- serialVersionUID 1303
- service 1204, 1223
- service consumer 1204, 1224, 1230
- service interface 1223, 1226
- service loader 1224, 1227
- service locator 1204, 1224, 1227
- service provider 1204, 1224, 1226
- service provider interface
  - see* service interface
- ServiceLoader class 1224
- ServiceProvider.Provider interface 1228
- services 332
- Set 821
- set 770, 800
  - capacity 824
  - HashSet 823
  - LinkedHashSet 823
  - load factor 824
  - NavigableSet 827
  - Set 821
  - SortedSet 827
  - TreeSet 829
  - unmodifiable 821
  - unmodifiable views 874
- set methods 367
- set operation 600
- setter methods *see* set methods
- SHA-256 digest algorithm 1641
- shadowing fields by local variables 115
- shadowing local variables 527
- shadowing members 701
- shallow copying 436
- shallowly immutable 308
- shared field 1490
- shared memory 1403
- shift
  - compound operators 91
  - distance 87
  - left 87
  - operators 87
- shifting 87
- short 32, 42
- short cut
  - see* symbolic link
- short-circuit 81
  - evaluation 82
- short-circuit execution 926, 966, 972
- shuffle elements 878
- sign bit 88
- sign fill 88
- signature 108, 206, 630
- signed shift 88
- simple
  - assignment operator 55
  - if 154
  - statement 101
- simple name 353
- simple type name 334
- SimpleFormat 1140
- simplicity 5
- single implementation inheritance 197
- single inheritance of implementation 197
- single quote (') 37, 38
- single source file 23

- single static import 339
- single thread executor 1462
- single thread scheduled executor 1463
- single-element annotation 1601, 1603
  - array-valued single-element annotation 1604
  - optional annotation element 1604
  - single-element array-valued single-element annotation 1605
  - value() element 1603
- single-element annotation type 1599
- single-file source-code program 23
- single-line comment 8, 40
- singleton 1284
- single-valued maps 847
- sink nodes 1199
- skeletal source file 333
- skip list 1531
- sleeping 1430
- software security 1638
- sorted map 801
  - comparator 862
- sorted set 800, 801, 827, 828
  - comparator 827
  - half-open interval 827
- SortedMap 862
- SortedSet 827
- sorting 799, 874, 949
- sorting arrays 882
- sorting lists 876
- source
  - file 22, 336
  - file name 333
  - file structure 333
- source directory 1247
- source of data 1261
- SP *see* space
- space 38
- spaces 40
- spawning threads 1412
- special character values 38
- specialization 15
- specialized classes 198
- spurious wakeup 1434
- SQL 1552
- SQL injection 1563, 1643
- SQL statement API
  - basic statement 1561
  - callable statements 1568
  - prepared statement 1564
- SQL statements 1560
- SQLException 380, 1556
- square root 491
- stability guarantee 930, 949
- stack
  - generic 612
  - LIFO deque 838
- stack frame
  - see* method execution
- stack trace 376, 377, 425
  - see* method execution
- StackOverflowError 382
- stacks 831
- standard error stream 377, 1283
- standard input stream 1283
- standard modules 1192
  - JMOD archive 1192
- standard out 25
- standard output stream 1283
- standard time 1105
- StandardCopyOption enum type 1339
- starting threads 1405
- starvation 1447
- state *see* object state
- stateful operations 926
- stateless behavioral parameters 928
- stateless operations 926
- statement block 698
- statement terminator 1552
- statements 101
  - break 182
  - compound 101
  - conditional 154
  - continue 185
  - control flow 101
  - control transfer 181
  - declaration 66, 71, 82, 101
  - do-while 176
  - empty 101
  - expression 71, 101
  - for(;;) 176
  - for(:) 179
  - if 154
  - if-else 155
  - iteration 174
  - labeled 182
  - resource declaration 419
  - return 186
  - selection 154
  - simple 101
  - simple if 154
  - switch statement 157, 162
  - synchronized 1426, 1427
  - throw 395

- transfer 181
- try 385
- try-with-resources 415
- while 175
- yield 166
- static
  - members *see* static members
  - methods 12, 14, 101
  - variable initialization 103
  - variables *see* static variables
- static 256, 339
- static 12
- static code
  - see* static context 99, 100
- static context 99, 100
- static factory method 368, 1141
- static field 14
- static field initializers 99
- static fields 113
- static import 339
  - conflicts 342
  - on demand 339
  - shadow static members 341
  - single static import 339
- static initializer block 99
- static initializer blocks 559
  - declaration order 559
  - exception handling 563
  - local variable type inference 559, 561
- static keyword 113, 559
  - static fields 113
  - static methods 115
- static local enum types 504, 507
- static local interfaces 504, 507
- static local nested types
  - see* static local types 503
- static local record classes 504, 507
- static local types 503
  - enum types 504, 507
  - interfaces 504, 507
  - record classes 504, 507
  - see* local enum types
  - see* local interfaces
  - see* local record class
- static member classes 506, 507
  - class file name 509
  - declaration 507
  - direct access 511
  - fully qualified name 509
  - instantiation 510
  - qualified name 509
  - UML class diagram 511
  - static member enum types 506
  - static member interfaces 506
  - static member record classes 506
  - static member types 503
    - static member class 503
    - static member enum type 504
    - static member interface 503
    - static member record class 504
- static members 11, 12, 14
- static method references 742
- static methods 115
  - hiding from superclass 208
- static nested classes
  - see* static member types
- static nested enum types
  - see* static member enum types
- static nested interfaces
  - see* static member interfaces
- static nested record class
  - see* static member record class
- static nested types
  - see* static member types
- static type 200
- static variables 12, 14, 106
- stored functions 1568
  - IN parameter 1568
  - INOUT parameter 1570
  - OUT parameter 1569
- stored procedures 1568
  - IN parameter 1568
  - INOUT parameter 1570
  - OUT parameter 1569
- stream 814
  - transformations 923
- Stream API 907
  - BaseStream interface 907
  - building streams 909
  - Collector interface 908
  - Collectors 908
  - DoubleStream interface 907
  - IntStream interface 907
  - LongStream interface 907
  - Stream interface 907
- Stream interface 907
- stream mapping 924
- stream operations 899
  - non-interfering behavioral parameters 928
  - stateless behavioral parameters 928
- stream pipelines
  - composing 904
  - executing 905

- streams 899
  - see also* I/O streams
  - aggregate operations 899
  - building from a `BufferedReader` 920
  - building from a `CharSequence` 919
  - building from a range 917
  - building from arrays 916
  - building from collections 915
  - building from directory entries 1378
  - building from specified values 911
  - building from the `Random` class 918
  - building using factory methods in the `Files` class 921
  - collecting into a list 992
  - collecting into an array 991
  - compare with collections 906
  - concatenating 914
  - conversion 954
  - counting elements 973
  - data source 902
  - dropping elements 929, 931
  - empty 911
  - encounter order 909
  - examining elements 939
  - execution mode 952
  - extracting lines from a `String` 920
  - filtering 928
  - finding first/any 971
  - finding min/max 973
  - finite 910
  - flattening 942
  - functional reduction 973, 974, 984, 993
  - infinite 910
  - intermediate operations 903, 923, 926
  - lazy execution 925
  - mapping 923, 940
  - map-reduce transformations 923
  - matching operations 969
  - method chaining 924
  - mutable reduction 967, 984
    - see also* collectors
  - numeric 902, 910
  - numeric summary statistics 993
  - object 902, 910
  - operations with side effects 966, 967
  - ordered 909
  - parallel streams 909, 982, 986, 1030
  - reduction operations 966, 984
  - replacing each element with multiple elements 946
  - searching operations 966
  - selecting distinct elements 933
  - sequential 909
  - sequential execution 977
  - setting as unordered 951
  - short-circuit execution 926, 966, 972
  - skipping elements 934
  - sorting 949
  - stability guarantee 930, 949
  - stateful operations 926
  - stateless operations 926
  - `Stream` API 907
  - stream mapping 924
  - stream operations 899
  - stream pipelines 904
  - taking elements 929, 931
  - terminal stream operations 903, 966
  - truncating 935
  - unordered 909
- `strictfp` 59
- string builders 70
  - appending 477
  - capacity 474, 480
  - `compareTo` 477
  - construction 475
  - copying characters 475
  - deleting 479
  - extracting substrings 476
  - individual characters 475
  - inserting 478
  - joining 460
  - length 476
  - mutable 474
  - reading characters 475
  - replacing 478
  - reversing 479
  - searching 476
  - setting length 480
  - substrings 476
  - trimming 480
- `String` class
  - see* strings
- string conversion 47, 68, 465
- string literal pool 450
  - interned 450
- string literals 450
  - case labels 164
  - hash value 164
  - interned 450
- string pattern 1150
- `StringBuffer` legacy class
  - thread-safe 474
- `StringBuilder` class
  - see* string builders

- strings
  - capacity 480
  - changing case 458
  - compact 450
  - compareTo 457
  - comparing 454, 457, 777
  - concatenation 68, 458
  - concatenation operator + 70
  - contains 462
  - conversions 465
  - convert to character array 455
  - copying characters 455
  - creating 450
  - empty 451, 454
  - ends with 462
  - equals 457
  - extracting lines 920
  - extracting substrings 463
  - finding index 461
  - formatted 467
  - ignoring case in comparison 457
  - immutable 450
  - indenting lines 465
  - individual characters 455, 475
  - initializing 450
  - integer stream 454
  - internal representation 449
  - interned 450, 452
  - joining 459, 460, 861, 1004
  - length 454, 455
  - lexicographical ordering 457
  - literals 39, 450
  - natural order 777
  - read character at index 454, 455
  - repeating 459
  - replacing 462
  - searching 461
  - selector expression 158
  - starts with 462
  - streams 919, 920
  - string literal pool 450
  - stripping 463
  - substrings 463
  - text blocks 468
  - thread-safe 450
  - transforming 465
  - trimming 463
- strongly typed language 49
- Structured Query Language 1552
- style-based formatter for dates and times 1157
- subclass 15, 17, 195
- subdirectories 1317
- subformat pattern 1172
- subinterface 249
- sublist 670
- submap view 862, 863
- sub-maps 1530
- subsequence 454
- subset 821, 825, 827, 828, 1525
- subsignature 631
- substitutional parameterization 1564
- substring searching 461, 476
- substrings 461, 463, 476
- subtags 1122
- subtype bivariance 595
- subtype contravariance 595
- subtype covariance 264, 265, 594, 595
- subtype invariance 595
- subtype notation <: 598
- subtype relationship 662
- subtypes 247, 284, 595, 606
- subtype-supertype relationship 45, 199
- suffix
  - D 36
  - F 36
  - L 33
- super 253
  - construct 216
  - keyword 205, 210
  - reference 210
- superclass 15, 195
- superclass constructor 110
- superinterfaces 248
- superset 825
- supertypes 247, 284, 596
- Supplier functional interface 715
  - primitive type specializations 717
- suppressed exceptions 377, 424
- suppressed warnings
  - deprecation warnings 1621
  - unchecked warnings 1621
- switch block
  - local variable scope 171
  - switch rules 162
- switch expression 166
  - arrow notation 168
  - case labels 166
  - colon notation 166
  - comparison with switch statement 171
  - default clause 169
  - exhaustive 167, 169
  - fall-through 167
  - local variable scope 171

- return multiple values 170
- yield statement 166, 168
- switch rules 162
  - mutually exclusive 163
- switch statement 157
  - arrow notation 162
  - break 159
  - case labels 158
  - colon notation 157, 159
  - comparison with switch expression 171
  - default clause 158, 163
  - enum types 165
  - execution 158
  - exhaustive 165
  - fall-through 158
  - local variable scope 171
  - selector expression 162
  - using strings 164
- switching between threads 1402
- symbolic link 1319, 1335
- synchronization 1422
- synchronized
  - keyword 1423
  - methods 1423
  - starvation 1447
  - statements 1426, 1427
  - vs. volatile 1492
- synchronized block 1426
- synchronized collections 1513
  - compound mutually exclusive operations 1517
  - serial access 1515
  - underlying collection 1514
- synchronized method 1423
- synchronized views of collections
  - see* synchronized collections
- SynchronousQueue 1535
- syntactically legal 30
- System 435
  - out 25
  - reassigning standard streams 1283
  - standard streams 1283
- system clock 1052
- system default zone ID 1097
- system modules 1240

## T

- TAB *see* horizontal tab
- table definition 1551
- tabs 40
- tabulators 40

- tagging interface
  - see* marker interfaces
- tags 41
- tail map view 862, 863
- tail set view 827, 828, 1525
- taking stream elements 929, 931
- target directory 1214
- target parallelism level 1462
- target reference 741
- target type 584, 699
- target typing 700
- task 1401, 1459
- task cancellation 1481
- task execution policy 1462
- TCL 1552
- telephone directory order 457
- temporal arithmetic 1062, 1077, 1083, 1092
- temporal fields 1049
- temporal objects
  - access specific temporal values 1054, 1100
  - before/after methods 1056
  - combining date and time 1052
  - common methods 1048
  - comparing 1056
  - conversion to Date 1111
  - creating 1049, 1098
  - date 1046
  - date fields 1046
  - date units 1062
  - date-time 1046
  - fields 1046, 1047
  - formatting 1153
  - get methods 1054, 1100
  - immutable 1046, 1098
  - method naming convention 1048
  - parsing 1153
  - plus/minus methods 1062, 1104
  - range of values for date fields 1051
  - range of values for time fields 1051
  - temporal arithmetic 1062, 1083, 1084, 1093, 1103
  - temporal fields 1049, 1058
  - thread-safe 1046, 1098
  - time 1046
  - time fields 1046
  - time units 1062
  - with methods 1057, 1101
  - zoned date-time 1046, 1097
- TemporalAmount interface 1084
  - see also* Duration
  - see also* Period

- TemporalField interface
  - see also* ChronoField
- TemporalUnit interface
  - see also* ChronoUnit
- terminal stream operations 903, 966
  - averaging 994
  - collecting into a list 992
  - collecting into an array 991
  - counting elements 973
  - eager execution 905
  - finding first/any 966, 971
  - finding min/max 973
  - functional reduction 967, 973, 974, 993
    - accumulator 976, 977, 981
    - combiner 976, 982
    - identity value 975, 977, 981
    - parallel execution 982
    - sequential execution 977
  - matching operations 966, 969
  - mutable reduction 967, 984, 991, 992
    - accumulator 985
    - combiner 985
    - parallel stream execution 986
    - sequential stream execution 985
    - supplier 985
    - see also* collectors
  - operations with side effects 966, 967
  - reduction operations 966, 984
    - see also* collectors
  - searching operations 966
  - short-circuit operations 966, 972
  - summary statistics 993, 995
  - summation 994
- TERMINATED state 1417, 1440
- terminating loops 183
- ternary conditional expressions
  - see also* conditional expressions 92
- ternary conditional operator 51, 92
- text block 468
  - closing delimiter 468
  - content 468
  - escape sequences 469
  - essential whitespace 472
  - incidental whitespace 472
  - indentation 472
  - opening delimiter 468
- text files
  - reading from 1278
  - stream to read lines 1378
  - writing to 1275
- text representation 436
- this
  - reference 106
- this keyword
  - qualified form 518
  - with new 516
- this() constructor call 213, 569
- thousands separator 1151
- Thread class 1405
- thread coordination 1431
- thread execution
  - runtime organization 1403
- thread interference 1408, 1424
- thread interruption rule 1450
- thread lifecycle 1414
- thread of execution 1401
- thread pool 1459, 1461
- thread safety 1402
- Thread start rule 1449
- Thread termination rule 1450
- thread-based multitasking 1401
- thread-local variable 1490
- ThreadLocalRandom 1458
- threads 6, 435, 548, 1437
  - acquire lock 1423
  - alive 1412, 1417
  - BLOCKED state 1417
  - blocking operation 1440
  - child 1407, 1412
  - class 1405
  - code executed 1405
  - controlling termination 1440
  - creation 1404
  - current thread 1406
  - daemon 1412
  - deadlocks 1443
  - death 376
  - deprecated methods 1443
  - determining state 1415
  - entry set for 1414
  - exception propagation 376
  - extending Thread 1409
  - fairness 1443
  - happens-before relationship 1449
  - IllegalThreadStateException 1412
  - interrupt status 1428
  - interrupted 1434
  - InterruptedException 1428
  - joining 1418, 1437
  - JVM stack 548
  - live 548
  - livelock 1445
  - liveness 1443
  - main thread 1407, 1412

- names 1406, 1407
- NEW state 1416, 1421
- non-observable substate 1415
- non-runnable state 1415
- normal 1412
- notification 1431, 1433
- notifying 437, 1431
- Object lock rule 1449
- priorities 1419
- priority 1418
- run() method 1406
- Runnable 1405, 1410
- RUNNABLE state 1417
- running 1409
- running thread 1415
- scheduler 1419
- shared field 1490
- shared memory 1403
- sleeping 1418, 1430
- spawning 1412
- spurious wakeup 1434
- start() method 1406
- starting 1405, 1409, 1421
- starvation 1447
- states 1415
- switching 1402
- synchronization 1422
- synchronized 1423
- synchronized method 1423
- tasks 1459
- TERMINATED state 1417, 1440
- termination 1440
- Thread class 1405
- thread coordination 1431
- Thread interruption rule 1450
- thread lifecycle 1414
- Thread start rule 1449
- thread status 1412
- Thread termination rule 1450
- thread-local 1490
- TIMED\_WAITING state 1417
- transitions 1415
- Volatile field rule 1449, 1490
- wait set for 1432
- waiting 437, 1431
- WAITING state 1417
- yielding 1418, 1421
- thread-safe 474, 659, 806, 1046, 1098
  - immutable classes 364
- thread-safe code 1422, 1487
- thread-safe collection 1514
- thread-safety 364
  - atomic variables 1493
  - immutability 1489
  - intrinsic locking 1496
  - programmatic locking 1497
  - synchronized collections 1513
  - unmodifiable collections 1514
  - volatile fields 1490
- throw statement 395
- Throwable 377, 414, 435
- throw-and-catch paradigm 373
- throwing exceptions 373
- throws clause 396
  - and rethrowing 410
- time fields 1046
- time gap 1105
- time overlap 1105
- time standard 1096
- time units 1062, 1457
- time zone 1046
  - local time 1096
  - see* zone ID
- time zone offset 1096
- time zone rules 1097, 1105, 1107
- timed locking 1499, 1504
- TIMED\_WAITING state 1417
- timeline 1046
  - instant 1097
  - local 1097
- time-sliced scheduling 1420
- timestamps 1071
- TimeUnit 1457
- ToDoubleBiFunction interface 734
- ToDoubleFunction interface 729
- ToIntBiFunction interface 734
- ToIntFunction interface 729
- tokens 30
- ToLongBiFunction interface 734
- ToLongFunction interface 729
- top-level classes 503, 506
- top-level enum types 503, 506
- top-level interfaces 503, 506
- top-level record classes 503, 506
- top-level types 503
- toString method 436, 441
- total order 777
- total ordering 883
- Transaction Control Language (TCL) 1552
- transactions 1583
  - auto-commit mode 1584
  - commit 1583
  - rollback 1583, 1584
  - savepoint 1584



- transfer statements 181
- TransferQueue 1535, 1537
- transforming strings 465
- transient fields 1296
- transitive relation 198
- transitive relationship 1449
- tree 1317
  - traversal 1379
- tree map 864
- tree set 829, 864
- TreeMap 864
- trees 805
- TreeSet 829
- trim method 463
- true literal 32, 37
- truth-values 37, 43
- try block 385
- try-catch-finally construct 384
- try-with-resources 147, 415, 1557
  - basic 419
  - concise syntax 420
  - extended 419
  - header 419
  - var in header 419
- tuples 1551
- two's complement 34, 35, 87
- two-arity 711
- type annotation 1608
- type cast 49
- type cast expression 274
- type checking 699
- type comparison operator 275
- type contexts 1608
- type declarations 333
  - nested types 503
  - top-level types 503, 506
- type erasure 590, 627, 636
- type hierarchy 45, 199
- type import
  - see* import
- type inference 143
- type information 579
- type instance 583
- type parameter constraints 605
- type parameters 647
  - actual 583
  - bounds 605
  - formal 581
  - recursive bounds 625
  - restrictions 582
- type pattern 280
- type variables 581

- type witness 609, 947
- types
  - arrays 593
  - boolean 41, 43
  - byte 32, 42
  - casting 49
  - char 42
  - classes *see* classes
  - comparison 275
  - compatibility 49
  - double 43
  - exceptions 377
  - float 43
  - floating-point 41, 43
  - int 32, 42
  - integers 41, 42
  - integral types 41
  - long 32, 42
  - numeric types 41
  - parsing 444
  - short 32, 42
  - unrelated 584
  - wrappers 439
  - see also* classes
- type-safe 270
- type-safety 660, 662
- typeValue method 442, 444

## U

- UML 7
  - accessibility 356
  - aggregation 17
  - classes 7
  - inheritance 15
  - see also* diagrams
- UML class diagram
  - non-static member classes 522
  - static member classes 511
- UML notation for members 356
- UN M.49 three-digit area code 1121
- unary arithmetic operators 62
- unary numeric promotion 50
- unary operators 51
- UnaryOperator interface 736, 737
  - composing unary operators 737
  - primitive type specializations 737
- unbounded collections 1520
- unbounded instance method references 745
- unbounded type parameter 595, 597
- unbounded type references 604

- unbounded wildcard
  - ? 595, 596
- unbounded wildcard references 602
- unboxing 442
  - do-while statement 176
  - for(;;) statement 177
  - for(:) statement 179
  - if statement 154
  - if-else statement 155
  - switch statement 158
  - while statement 175
- unboxing conversions 46
- uncaught exceptions 376
- unchecked 641, 644
- unchecked array creation warning 644
- unchecked call warning 590, 602
- unchecked cast warning 639, 641
- unchecked conversion warning 585, 590, 599, 641, 660
- unchecked conversions 47
- unchecked exceptions 383
- unchecked generic array creation warning 644, 664, 673, 822, 850, 881
- unchecked warnings 46, 589, 590, 1621
  - @SuppressWarnings("unchecked") 641
  - call warning 590, 602
  - cast warning 639
  - conversion warning 590
  - generic array creation warning 644
  - heap pollution warning 644, 1623
  - unchecked generic array warning 1623
- unconditional locking 1503
- underflow 34, 61
- underlying collection 1514
- uni-catch clause 384, 406
  - effectively final parameter 409
- Unicode 37, 42, 447, 457, 1270
- Unicode character encoding 1126
- Unicode locale/language extension 1143
- unidirectional edges 1193
- Unified Modeling Language *see* UML
- Uniform Resource Identifier (URI) 1320
- union 821
- union of alternatives 408
- unmodifiable 675
- unmodifiable collections 663, 821, 875, 1514
- unmodifiable lists 663, 675, 818
- unmodifiable maps 849
- unmodifiable sets 821
- unmodifiable view of a collection 874, 875
- unmodifiable view of a list 874
- unmodifiable view of a map 875
- unmodifiable view of a navigable map 875
- unmodifiable view of a navigable set 874
- unmodifiable view of a set 874
- unmodifiable view of a sorted map 875
- unmodifiable view of a sorted set 874
- unmodifiable views 874, 875
- unmodifiable views of collections 1514
- unnamed module 320, 1233
- unnamed package 320
- unordered streams 909
  - from the Random class 918
- unqualified export 1204
  - see* exports directive
- unreachable code 390, 410
- unsafe casts 276
- unsigned integer 445
- unsupported operations 806, 848
- UnsupportedOperationException 382, 810, 814
- UnsupportedTemporalTypeException 1055, 1084, 1090, 1093, 1098, 1157
- unused keywords 31
- upcasting 45
- updatability 1577
- update expression 176
- UPDATE statement 1554
- upper bound 595, 605
- upper bounded wildcard
  - ? extends T 595
- upper bounded wildcard references 603
- URI class 1320, 1323
  - schema 1323
- user threads
  - see* normal thread
- uses directive 1204, 1229
- using arrays 121
- using packages 337
- using variables 102
- UTC (Coordinated Universal Time) 1096
- UTC/Greenwich 1096
  - Z 1096
- UTF-16 encoding scheme 449, 450
- utility class 115, 255, 368

## V

- value collection 847
- value obfuscation 1641
- value objects 801
- value() element 1599, 1603
- valueOf method 440, 441, 464
- values 34, 847

- constants 32
- overflow 34
- underflow 34
- wrap-around 34
- see also* variables
- var 696
- var 585
- var declaration
  - see* local variable type inference
- varargs
  - variable arity 137
- varargs method
  - see* variable arity method
- varargs parameter
  - see* variable arity parameter
- variable arity argument 673, 881
- variable arity call 140
- variable arity method 137
- variable arity parameter 137, 644, 881, 1335
  - generic array creation 644
  - heap pollution 644
- variable capture 703
- variable declarations 102, 363
- variable initialization 12, 102, 104
- variables 9, 102
  - blank final 136
  - blank final local 235, 236
  - constant values 233
  - default values 103
  - effectively final 703
  - final 233
  - identifiers 102
  - in interfaces 259
  - initialization *see* variable initialization
  - lifetime 106
  - local 143, 363
  - parameters 100, 128
  - reference variable 102
  - references 102
  - static 12
- var-type inferred parameters 696
- Vector 818
- vector 818
- Venn diagrams 807
- versioning 1302
- view of a list 670, 675
- viewing dependencies 1242
  - class-level dependencies 1244
  - module-level dependencies 1243
  - package-level dependencies 1242
- views 827, 856

- virtual method invocation
  - see* dynamic method lookup
- Virtual Private Network (VPN) 1638
- VirtualMachineError 382
- visibility 1490
- visibility modifiers
  - see* access modifiers
- void 21, 186, 439
- void return 697
- Void wrapper class 439
- volatile
  - vs. synchronized 1492
- Volatile field rule 1449, 1490
- volatile fields 1490
  - visibility 1490
- volume name 1317

## W

- wait set 1431, 1432
  - starvation 1447
- waiting 1431
  - timeout 1434
- WAITING state 1417
- waiting threads 437
- WHERE clause 1553, 1554, 1555
- while statement 175
- whitespace 40, 463, 464
- whole-part relationship 199
- widening conversions
  - primitive 44
  - references 45
- widening reference conversions 200, 275, 598
- wider range 44
- wildcard parameterized types 598
  - type hierarchy 595, 596, 597
- wildcard types 594
  - see* wildcards
- wildcards 593, 594
  - ? 602
  - ? extends T 603
  - ? super T 603
  - assignment 598
  - bounded 595
  - capture 619
  - capture conversion 621
  - nested 621
  - parameterized types 598
  - restrictions 598
  - subtype relationships 593
  - type hierarchy 595, 596, 597

- withers 1057
- work stealing pool 1462
  - target parallelism level 1462
- worker threads 1459
- wrapper 1261
- wrapper classes 41, 435, 436, 439, 760, 770, 777
  - interned values 443
- wrapper type 46, 60, 71
- write lock 1503
- write operation 556
- WriteLock 1503
- writers 1270
- writing binary values 1266
- writing bytes 1262
- writing characters 1266, 1277, 1282
- writing data 1261

## X

- xor* 84

## Y

- `yield` statement 166, 168
- yielding 1421

## Z

- Z 1096
- zero
  - negative 61
  - positive 61
- zero fill shift 88
- zero-arity 711
- zero-based index 658, 817
- zone ID 1096
  - available zone IDs 1096
  - system default 1097
- zoned date-time
  - accessing time zone 1100
  - accessing time zone offset 1100
  - conversion to `Date` 1111
  - converting to different time zone 1102
  - converting to instant 1101
  - see also* daylight saving time (DST)
  - see* temporal objects
- `ZonedDateTime` class 1095, 1097
  - see* temporal objects
  - see* zoned date-time
- `ZoneId` class 1095
- `ZoneOffset` class 1095