

# Index

## Symbols

! 194  
 - 180, 556, 597  
 -- 186  
 != 191  
 " 25  
 # 597  
 \$ 557, 595  
 % 178, 595  
 %= 182  
 & 194  
 && 196  
 &= 195  
 ( 597  
 () 558  
 \* 108, 121, 174, 178, 558  
 \*= 182  
 + 180, 185, 558, 597  
 + concatenation 446  
 ++ 186  
 += 182  
 , 597  
 . 7, 105, 117, 470, 557, 596  
 .. 470  
 ... 90, 95, 269, 270  
 / 178, 468  
 /\* and \*/ 26  
 /\*\* and \*/ 27  
 // 26  
 /= 182  
 : 119, 468  
 ; 45, 468  
 < 190, 595  
 <= 190  
 <> 664  
 -= 182

= 123, 169  
 == 191  
 > 190  
 >= 190  
 ? 558, 675, 677, 682  
 ? extends T 675, 682  
 ? super T 676, 683  
 ?: 201  
 @Override 289, 718  
 @SuppressWarnings("unchecked") 726  
 [] 70, 72, 555  
 \ 24, 468, 555, 561  
 ^ 194, 555, 557  
 ^= 195  
 {} 45, 71, 131  
 | 194, 558  
 |= 195  
 || 196  
 ' 23

## Numerics

o 597  
 2's complement 1010

## A

absolute pathnames 469, 471  
 abstract  
   classes 135  
   interfaces 135  
   methods 150  
 abstract data type 662  
 abstract method declarations  
   in interfaces 310  
 abstract pathnames 469

- abstraction 2, 10
- access order 823, 824
- accessibility 7, 16, 129
  - anonymous classes 380
  - default 132, 142
  - local classes 372
  - members 129, 135, 138
  - modifiers 132
  - non-static member classes 362
  - package 132
  - private 143
  - protected 141
  - public 139
  - static member classes 357
  - UML 139
- accessibility modifiers 40, 48
- activation record
  - see* method execution 235
- actual parameter 81
- actual parameter list 81, 323
- actual type parameters 665
- adding to class 284
- additive operators 180
- aggregation 10, 12, 287
  - hierarchy 287
- aliases 6, 192, 193
  - see also* references
- ambiguous call 324
- analyzing program code 854
- anchoring 557
- anchors 557
- annotations
  - @Override 289, 718
  - @SuppressWarnings("unchecked") 726
- anonymous arrays 74
  - [] 74
- anonymous classes 377, 773
  - event listeners 377
- anti-symmetric 766
- Apostrophe 24
- appending elements 805
- application 15, 563
- argument
  - see* actual parameter
- argument index 595
- arguments to main method 95
- ArithmeticException 241
- array creation expression 70, 74
- array initializer block 74
- array list 806
- array operations 790
- array store check 319, 726
- ArrayIndexOutOfBoundsException 73, 241
- arrays 69, 424
  - [] 70, 72
  - { } 71
  - anonymous 74
  - array creation expression 70
  - array initialize list 71, 74
  - array name 70
  - array size 71
  - array store check 319, 726
  - collection operations 790
  - declarations 70
  - default initialization 70, 71
  - dynamic 806
  - element default value 318
  - element type 70
  - elements 69, 72
  - index expression 72
  - initialization 71, 76
  - iterating over 220
  - length 69
  - multidimensional 76
  - objects 424
  - ragged 76
  - reference 70, 73, 319
  - searching 843
  - sorting 842
  - subtype covariance 317
- arrays of arrays 69, 76
  - multidimensional 76
- ArrayStoreException 319, 790
- ASCII 23, 29
- assert statement 265
  - see* assertions 265
- AssertionError 243, 265, 266
  - see also* assertions 265
- assertions 265, 376
  - assert statement 265
  - compiling 267
  - conditional compilation 275
  - control flow invariants 274
  - enabling and disabling 269
  - internal invariants 273
  - postconditions 275
  - preconditions 275
  - using 272
- assignable 164, 322
- assignment
  - widening reference 679
  - widening reference conversions 287
- assignment compatible 164, 322
- assignment conversions 164

- assignments
  - cascading 170
  - compound operators 182, 195
  - expression statement 170
  - implicit narrowing 171
  - multiple 170
  - numeric conversions 171
  - operator 167, 169
  - primitive values 169
  - references 169
- associativity 167, 168
- asynchronous 615
- attributes *see* properties
- autoboxing 430
- automatic garbage collection 6, 390
- automatic variables *see* local variables
- automaton 554

## B

- Backslash 24
- Backspace 24
- backtracking 559
- bags 796
- balanced tree 782, 783
- base 22, 431, 434
- base class 284
- basic collection operations 784
- basic map operations 821
- binary
  - numbers 1005
  - numeric promotion 166
  - operators 167
- binary files 480, 481
- binary numbers 1005
- binary representation
  - see* binary values
- binary search
  - arrays 843
  - lists 840
- binary values 480, 481
- bit patterns 1010
- blank final variable 89, 149
- blocked for I/O state 635
- blocked for join completion state 635
- blocked for lock acquisition state 635
- blocking operations 476
- blocks 44, 45, 131
  - catch 246
  - finally 246, 251
  - scope 131
  - try 245

- Boolean
  - conditional expression 204
- boolean conversion 598
- Boolean wrapper class 437
- booleans 28, 30
  - casting 165
  - expressions 190
  - literals 23
- boundary matchers 557
- bounded type parameters
  - constraint 684
- bounded wildcard 675
- bounds
  - constraint 684
- boxing
  - for(:) statement 221
- boxing conversions 162
- break statement 209, 224
- bridge method 716, 768
- BS *see* Backspace
- buckets 760
- buffering streams 479
- builder tools 41
- building abstractions 10
- bulk map operations 822
- bulk operations on collections 785
- byte 22, 28
- byte code 15
- byte streams 468

## C

- C 151
- C++ 151
- cache 153
- Calendar 536
- calendar 539
  - constants 538
  - factory methods 537
  - field numbers 537
  - field values 538
- call chaining 81
- call stack
  - see* runtime stack 235
- call-by-reference 86
- call-by-value 86
- callee 81
- caller 81
- canonical path 471
- capture conversion 163, 705
- capture of 703
- carriage return 24, 25

- cascading assignments 170
- cascading if-else statements 207
- case labels 207, 208
- case-sensitivity 20
- cast 730
- cast operator 161, 164, 166, 328
- casting 164
  - see also* conversions
- catch block 246
- catch-and-throw paradigm 235
- catching exceptions 235
- ceiling entry 828
- ceiling key 828
- certification 851, 959
- chained streams 468
- chaining 416
  - constructors 306, 416
  - finalizers 397
  - new operator 365
- changing list contents 841
- char 29
- character
  - non-delimiter 575
  - regular expression 554
- character class 555
- character conversion 598
- character encoding 488
  - see also* character sets
- character sequences
  - see* strings *and* string builders
- character set
  - ASCII 23, 29, 488
  - ISO Latin-1 23, 29
  - Unicode 23, 29, 488
- character streams 468, 488
- Character wrapper class 436
- character-based device
  - console 500
- characters 29
  - literals 23
  - searching for 448
- checked exceptions 243
- child class 284
- child threads 615, 619
- choosing between String and StringBuilder
  - class 456
- Class class 425
- class file 15
- class literals 733
- class lock 626, 631
- class method 10
- class modifiers 40
- class path
  - absolute pathnames 119
  - entries order 119
  - entry-separator character 119
  - fully qualified package name 119
  - path-separator character 119
  - relative pathnames 119
  - search in a named package 118
  - searching for classes 117
  - searching in JAR files 121
  - white space 120
  - wild card \* 121
- class path wildcard 121
- class search path
  - see* class path
- class variable 10
- ClassCastException 242, 329
- classes
  - abstract 135
  - accessibility 132, 353
  - adding to 284
  - anonymous 377
  - base 284
  - body 40
  - child 284
  - cohesion 346
  - constructors 48, 302
  - coupling 346
  - declaration context 353
  - declarations 40, 104
  - definitions 2, 5
  - derived 284
  - diagram 8, 9
  - encapsulation 345
  - extending 136, 284
  - file name 356
  - final 136
  - final vs. abstract 137
  - fully qualified name 117
  - fully qualified package name 106
  - generalized 286
  - grouping 105
  - header 40
  - initialization 412, 419
  - inner 352
  - instance members 40
  - instances 4
  - local 371
  - members 7
  - methods 147
  - modifiers 135

- name 105
- nested *see* nested classes
- non-static member 352, 359
- normal 135
- Object 424
- on-the-fly 352
- parent 284
- runtime 425
- scope 129
- searching for 117
- specialized 286
- static members 40, 355
- subclass 10, 284
- superclass 10, 284
- top-level 352
- variables 147
- wrappers 424, 428
- ClassLoader class 424
- ClassNotFoundException 241
- CLASSPATH environment variable
  - see* class path
- classpath option
  - see* class path
- cleaning up 392
- clean-up code 252
- client 15
- Cloneable interface 425
- cloning objects 425
- code optimizations 149
- code reuse 284, 344
- cohesion 346
  - coincidental 346
  - functional 346
  - high 346
- coincidental cohesion 346
- Collection interface 779
- collections 680, 777, 779
  - array operations 790
  - ArrayList 806
  - basic operations 784, 821
  - bulk operations 785, 822
  - Cloneable 780
  - customizing 838
  - data structures 782
  - duplicate elements 796, 804
  - elements 777
  - first element 800, 801
  - HashSet 796
  - implementations 778, 780, 782
  - interfaces 777, 778
  - iterators 785
  - last element 800, 801
  - LinkedList 806
  - List 806
  - lists 804
  - Map 821, 823
  - map views 822
  - maximum 841
  - minimum 841
  - NavigableSet 800
  - optional operations 784
  - ordered 780, 804
  - range-views 800, 801, 805
  - retaining elements 785
  - Serializable 780
  - shuffle elements 839
  - sorted 780
  - SortedSet 800, 801
  - subset 800, 801
  - textual representation 788
  - thread-safety 806, 823
  - TreeMap 828
  - TreeSet 802, 828
  - unordered 780
  - unsupported operations 821
  - utility methods 778
  - Vector 806
  - views 800, 801, 805, 822
- collections framework 777
- Collections utility class 838, 840
- command
  - java 17
  - javac 16
- Comma-Separated-Value 611
- comments 26
- common memory space 614
- communication 7, 81
- Comparable interface 765
  - compareTo 765
- comparable objects 765, 766
- comparator 765
- Comparator interface
  - compare 771
- comparing objects 425, 771
  - see also* equality
- comparison 190
- compilation unit 106
- compile pattern 554
- complete methods 149
- completes abruptly
  - see* exception propagation 238
- composite object 10
- compound assignment operators 182
- compound statement 45

- computerized scientific notation 599
- concatenation 558
- concatenation of strings 446
- concatenation operator 185
- concrete parameterized types 674
- concrete types 674
- concurrent activities 614
- ConcurrentModificationException 787, 788
- conditional 190
  - and 196
  - expressions 201, 204
  - operators 196, 201
  - or 196
  - statements 204
- conditional expression
  - Boolean 204
- conditions 190
- congruent 766
- connecting punctuation 20
- Console 594
- console 500
  - formatted prompting 502
  - formatted values 492
  - reading password 501
- const 21
- constant declarations 310
- constant values 21, 148
- constants 314
- constituent objects 10
- constituents 12
- constraints
  - forward reference 702
- constructor chaining 303, 306, 416
- constructors 3, 48, 302
  - accessibility 139
  - accessibility modifier 48
  - body 48
  - chaining 303, 306
  - class name 48
  - declaration 40
  - default 49
  - header 48
  - implicit default 49
  - local declarations 48
  - overloading 51
- constructs 20
  - high-level 20
  - loops *see* iteration statements
  - synchronized blocks 629, 630, 631
- containers 680
- context switching 614
- continue statement 226
- contract 2, 310, 312, 345
- contracting lists 805
- control flow
  - assert 265
  - break 209, 224
  - continue 226
  - do-while 217
  - for(:) 220
  - for(;;) 218
  - if 204
  - if-else 205
  - iteration *see* iteration statements
  - loops *see* iteration statements
  - return 228
  - statements 45, 204
  - switch 207
  - throw 255
  - transfer statements 223
  - while 217
- control flow invariants 274
- control transfer 223
- conversion categories 163
  - see* formatting conversions
- conversion contexts 163
- conversions 160, 319
  - assignment 164
  - contexts 163
  - identity 183
  - implicit narrowing 183
  - method invocation 164
  - narrowing reference 328, 679, 703
  - number systems 1006, 1008
  - numeric promotions 165
  - parameters 82
  - reference casting 327
  - string concatenation 185
  - to strings 450
  - truncation 172
  - type safe 323
  - unsafe casts 329
  - widening reference 287, 328, 679
- converting number systems 1006, 1008
- core interfaces 778
- Core Map Interfaces 782
- counter-controlled loops 218
- country codes 532
- coupling 346
  - loose 346
- covariant return 290
- cp option

- see class path*
  - CR *see* Carriage return
  - crab 220
  - creating
    - files and directories 473
    - objects 201
    - threads 615
  - CSV 611
  - currency 532, 546
    - formatting 546
  - currency formatter 546, 547
  - currency symbol 20
  - current directory 469
    - . 117
    - empty string 469
  - current object 45
  - cursor 806
- D**
- D option
    - see* properties
  - d option 115
  - daemon 615, 647
  - dangling references 390
  - data streams 477, 479, 480
  - data structures 782
  - data types *see* types
  - Date 535
  - date 532, 766
    - epoch 535
    - formatting 542
    - lenient 544
  - date formatting
    - static factory methods 541
    - styles 542
  - date/time composition conversions 600
  - date/time conversions 600
  - DateFormat 541
  - dates 532
  - dead state 635
  - deadlocks 651
  - decimal format 598, 599
  - decimal places 599
  - declaration statements 219
  - declarations
    - arrays 70, 201
    - classes 40, 104
    - enums 104
    - interfaces 104
    - local 45
    - main method 95
    - methods 44
    - packages 104, 106
    - statements 45
    - variables *see* variable declarations
  - declared type 288, 323
  - declaring *see* declarations
  - decoupling 341
  - decrement operator 186
  - deep copying 425
  - default
    - accessibility 132, 139, 142
    - constructor 49
    - exception handler 239
    - values 33, 406, 416
  - default
    - label 208
  - default locale 533, 547
  - default package 106
  - definitions
    - inheritance 313
    - interfaces 310
  - delegating requests 345
  - deleting files or directories 474
  - delimiter pattern 574
  - delimiters 547, 563, 571, 576
  - deque 779, 783
  - derived class 284
  - deserialization of objects 510
  - destination directory 115
  - destination object 593
  - destination of data 468
  - destroying objects 396
  - destructive collection operations 785
  - developer exam 856
  - diagrams
    - class 3
    - object 5
    - see also* UML
  - dictionary 780
  - dictionary order 445
  - difference 796
  - directories 470, 472
    - entries 470, 473
    - see also* files 474
  - directory permissions 472
  - disjunct collections 798
  - disposal of resources 393
  - dividend 178
  - division
    - floating-point 178
    - integer 178
  - division operator

- / 178
- divisor 178
- documentation 26
- documentation comment 26
  - tags 27
- documenting *see* documentation
- dot 105
- double 22, 29
- double quote 24
- do-while statement 217
- downcasting 161
- duplicate elements 796, 804
- duplicating objects 425
- dynamic method lookup 295, 340, 341
- dynamic type 288
- dynamically resizable arrays 806

## E

- element type 70
- elements 69, 777
- elements of collections 777
- eligible for garbage collection 391
- else clause matching 207
- empty set 796
- empty statement 45
- empty string 469, 558
- encapsulation 105, 345, 353
- encapsulation of implementation 345
- enclosing class 352, 360
- enclosing context 359, 363
- encoding of characters 488
- end of file 475, 478
  - see also* end of stream
- end of stream 482, 489, 496, 497, 510
- engine 554
- enhanced for loop 216
- entering shared resource 626
- entries 780, 821
- entries in directory 470, 473
- entry set 821
- enum type
  - finalization 397
  - static member type 355
- enum types
  - nested 355
- enumerated type
  - switch expression 207
- enumerated types 211
- EOFException 241
- epoch 535
- equality 191, 425
  - equals method 193, 425
    - object value 193
    - objects 193
      - primitive values 191
      - reference values 192
  - equals method 193, 425
  - equivalence relation 754
  - equivalent collections 798
  - erasure 714, 717, 719
    - of the supertype method 721
    - same 721
  - Error 242, 266
  - escape sequences 24
  - evaluation order 168, 197
    - arithmetic expressions 174
  - evaluation short-circuits 197
  - event 42
  - event listeners
    - anonymous classes 377
  - event model 42
  - exam 851
  - exam voucher 852
  - Exception class 241
  - exception handler 235
    - see also* exceptions
  - ExceptionInInitializerError 243
  - exceptions 235, 245
    - ArrayIndexOutOfBoundsException 73
    - ArrayStoreException 319, 790
    - ClassCastException 329
    - ClassNotFoundException 512
    - ConcurrentModificationException 787
    - default handler 239
    - EOFException 480
    - FileNotFoundException 478
    - handler 235
    - ignored 396
    - IllegalThreadStateException 615
    - IndexOutOfBoundsException 443, 805
    - IOException 475
    - NumberFormatException 430
    - situations 240
    - StringIndexOutOfBoundsException 449, 457, 458
    - throw 255
    - throwing *see* throwing exceptions
    - thrown by JVM 241
    - thrown by method 44
    - thrown programmatically 241
    - types 239
    - uncaught 239
    - unchecked 244



- UnsupportedOperationException 784
  - exchanging information 81
  - execution paths 614
  - existence of files 470, 472
  - exiting a shared resource 626
  - explicit
    - garbage collection 398
    - referencing 373
  - exponent 22
  - expression statements 45, 170, 219, 220
  - expressions 208
    - actual parameters 81
    - assert 265
    - boolean 190
    - case labels 208
    - conditional 201
    - deterministic evaluation 166
    - label 208
    - return 228
    - statements 45
  - extending
    - classes 284
    - interfaces 313
    - Thread 619
  - extends 675, 682, 685
    - bounded type parameters 684
    - constraint 684
  - extends clause 284
    - interfaces 310
  - extensions
    - .class 15
    - .java 15
  - external encoding 488
  - external libraries 410
  - extracting substrings 449
- F**
- fall through 208
  - false literal 23
  - FF *see* Form feed
  - field declarations 40
  - field initialization 416
  - field numbers 537
  - field values 538
  - fields 2, 611
  - file 766
  - File class 468
  - file name 104, 356, 468, 470
  - file permissions 472
  - file streams 477
  - file system 468
  - files 468, 470, 472
    - access 472
    - aliases 475
    - binary 480
    - creating 473
    - deleting 474
    - entries 473
    - existence 472
    - modification time 471
    - name 470
    - name filter 473
    - not found 478
    - path 470
    - renaming 474
    - size 471
    - streams 477
    - text 494
  - filter streams 479
  - filtering file names 473
  - filters 468
  - final
    - classes 136
    - members 148
    - parameters 89
  - finalization mechanism 391
  - finalize method 396
  - finalize method 425
  - finalizer 396
  - finalizer chaining 397
  - finally block 246, 251
  - first element 800, 801
  - first map entry 827
  - fixed arity method 90
  - fixed text 594
  - flags 595, 597
  - float 22, 29
  - floating-point 28
    - double 29
    - float 29
    - literals 22
  - floating-point arithmetic 176
    - strictfp 177
  - floating-point conversions 599
    - computerized scientific notation 599
    - decimal format 599
    - decimal places 599
    - general scientific notation 599
    - hexadecimal exponential format 599
    - significant digits 599
  - floating-point data types 22
  - floating-point division 178

- floating-point remainder 179
  - floor entry 828
  - floor key 828
  - flow control *see* control flow
  - flushing streams 476, 491
  - for(:) statement 220
  - for(;;) statement 218
    - collections 787
    - iterator 787
    - traverse collections 788
  - for-each loop 216
  - form feed 24, 25
  - formal parameter
    - modifier 44
    - name 44
    - type 44
  - formal parameters 44, 48, 81, 131, 323
  - formal type parameter 663
  - formal type parameter list 40, 44
  - format exceptions 601
  - format specifier 594, 595
    - argument index 595
    - flags 595, 597
    - formatting conversions 595
    - precision 596
    - width 596
  - format string 594
    - fixed text 594
    - format specifier 594
  - formatted prompting 502
  - formatted values 492
    - Console 501
  - Formatter 593
  - formatter 602
    - destination object 593
    - fixed text 594
    - format specifier 594
    - format string 594
  - formatting 26
  - formatting conversions 595, 596
    - boolean conversion 598
    - character conversion 598
    - date/time composition conversions 600
    - floating-point conversions 599
    - general conversions 597
    - hash code conversion 598
    - integral conversions 598
    - string conversion 598
  - forward reference 702
  - fractional signed numbers 28
  - framework for collections 777
  - fully qualified class name 117
  - fully qualified names 315
  - fully qualified package name 106
  - fully qualified type name 105, 109
  - functional cohesion 346
- ## G
- gain lock ownership 627
  - general abstractions 286
  - general contract
    - compareTo() 766
    - equals() 754
    - hashCode() 761
  - general conversions 597
    - boolean conversions 598
    - hash code conversions 598
    - string conversions 598
  - general loops 218
  - general scientific notation 599
  - generalization 10
  - generalized classes 286
  - generic class 663, 664
  - generic instance method 700
  - generic interface 666
  - generic method 699
    - calling 700
  - generic method call
    - inference 700
  - generic static method 700
  - generic subtype method 721
  - generic type 663
    - extending 668
    - implementing 695
  - generics
    - arrays 726
    - casting 724
    - class literals 733
    - collections 672
    - constructors 697
    - enums 733
    - erasure 714, 721
    - exception handling 730
    - inherited methods 721
    - instanceof 723
    - JVM 662
    - method signature 716
    - methods 697, 699
    - nested classes 731
    - non-reifiable types 723
    - overloading 716, 717
    - overriding 716, 718
    - reifiable types 722, 723

- subsignature 716
- type erasure 670, 714
- var-args 729
- wildcards 673
- get operation 680
- getter method 41
- goto 21, 223
- grammar rules 20
- greedy quantifiers 559
- grep 610
- group 554, 563
- group logical operator 558
- grouping 105
- groups of objects 777
- GUI applications 615

## H

- half-open interval 842, 844
- handles *see* references
- has-a relationship 287
- hash code conversions 598
- hash codes 424
- hash function 760
- hash table 782, 783
- hash value 760
- Hashing 760
- HashSet 796
- head map view 827
- head set view 800, 801
- heap 390
- hexadecimal exponential format 599
- hexadecimal format 598
- hexadecimal numbers 1005
- hiding internals 345
- high cohesion 346
- higher entry 828
- higher key 828
- Horizontal tab 24
- HT *see* Horizontal tab

## I

- I/O buffering 479
- I/O package 468
- identifiers 20
  - predefined 20
  - reserved 20
  - variable 31
- identity conversion 162, 183
- identity of object 5
- IEEE 754-1985 29

- if-else statement 205
- ignored exceptions 396
- illegal thread state 615
- IllegalArgumentException 242
- immediate superclass 305
- immediately enclosing instance 352
- implementation 2
- implementation inheritance 284
- implementations 286, 345
  - collections 778
  - inheritance hierarchy 137
- implementations of collections 780
- implementing
  - Comparable 765
  - Comparator 771
  - interfaces 312
- implements clause 312
- implicit
  - inheritance 284
  - narrowing conversions 183
- implicit default constructor 49
- import
  - declaration 108
  - single-type-import declaration 108
  - statement 104
  - type-import-on-demand declaration 108
  - see also* static import
- importing
  - enum constants 111
  - nested static types 113
  - nested types 357
  - reference types 107
  - static members 109
- increment expression 218
- increment operator 186
- independent lifetimes 287
- index 69
- index bounds 805
- index expression 72
- IndexOutOfBoundsException 443, 805
- individual array elements 72
- inequality 191
  - see also* equality
- infinite loop 220
- infinity 176
  - negative 176
  - positive 176
- information hiding 345
- inheritance 10, 287
  - hierarchy 286, 365
  - serialization 519
  - supertype-subtype relationship 287

- initial state of object 416
  - initialization
    - arrays 71, 76
    - code 72
    - default values 33
    - for statement 218
    - objects 5
    - references 32
    - variables 31
  - initializer expressions 406
  - initializers 406
    - declaration-before-reading rule 407
  - initializing *see* initialization
  - inner classes 352
    - synchronization 630
  - input 468
  - input stream 468
  - InputStream class 475
  - insertion order 797, 824
  - insertion point 840, 844
  - instance
    - members 10, 40
    - methods 10, 44, 45
    - variable initialization 33
    - variables 10, 35
    - see also* object
  - instance initializer block 40, 413
  - instance methods 6
  - instance variables 6, 416
  - instanceof operator 201, 327, 328
  - instantiation 4, 135
    - anonymous classes 380
    - local classes 374
    - non-static member classes 360
    - parameterized type 665
    - static member classes 355
  - int 22, 28
  - integer arithmetic 175
  - Integer data types 22
  - integer division 178
  - integer remainder operation 179
  - integers 28
    - byte 28
    - data types 29
    - int 28
    - literals 22
    - long 28
    - representation 1010
    - short 28
    - types 29
  - integral conversions 598
    - decimal format 598
    - hexadecimal format 598
    - octal conversions 598
  - integral types 28, 29
  - integrity (maintaining) 627
  - interface constant antipattern 110
  - interface-based 780
  - interfaces 309
    - abstract 135
    - accessibility 132
    - body 310
    - collection core 777
    - declarations 104
    - extending 313
    - header 310
    - implementing 379
    - initialization 419
    - references 314
    - top-level 352
    - variables 314
  - internal invariants 273
  - internationalization 532
  - interned 432
  - inter-package accessibility 346
  - interpreter 16
  - intersection 556, 796
  - intervals 556
  - intra-class dependencies 346
  - invocation
    - parameterized type 665
  - invocation stack
    - see runtime stack* 235
  - invoking garbage collection 398
  - IOException 243
  - IOException 241
  - is-a relationship 286, 287, 344
  - ISO Latin-1 23, 29
  - Iterable 788
  - iteration statements 216
    - next iteration 226
    - termination 216, 225
  - iterators 785
    - fail-fast 787
    - ordered collection 789
    - sorted collection 789
    - unordered collection 789
- J**
- JAR file 120
    - libraries 121
  - Java
    - Native Interface *see* JNI

java 17  
   -disableassertions or -da 269  
   -disablesystemassertions or -dsa 272  
   -enableassertions or -ea 269  
   -enablesystemassertions or -esa 272  
 Java byte code 15  
 Java compiler 16  
 Java Native Interface *see* JNI  
 Java Virtual Machine *see* JVM  
 JavaBeans 41  
   event model 42  
   events 42  
   getter methods 41  
   listener 42  
   naming patterns 41  
   properties 41  
   setter methods 41  
   source 42  
 JavaBeans Standard 41  
 javac 16  
 javadoc comment 26  
 javadoc utility 27  
 JDK 16  
 JNI 152  
 joining threads 637  
 JVM 16

## K

key insertion order 823  
 key objects 780  
 key sets 821, 828  
 key sorting 826  
 keys 821  
 keywords 20  
   abstract 135, 150  
   assert 265  
   boolean 30  
   byte 28  
   case 207  
   catch 246  
   char 29  
   class 40, 310  
   const 21  
   continue 226  
   default 208  
   do 217  
   double 29  
   else 205  
   extends 284, 675, 682, 684, 685  
   final 89, 136, 148  
   finally 251

float 29  
   for 218, 220  
   if 204  
   implements 312  
   import 108  
   instanceof 201, 327, 328, 723  
   int 28  
   interface 310  
   long 28  
   native 151  
   new *see* new operator  
   null 328  
   package 106  
   private 143  
   protected 141  
   public 139  
   reserved words 21  
   return 228  
   short 28  
   static 16, 109, 355, 410  
   strictfp 177  
   super 292, 295, 305, 366, 676  
   switch 207  
   synchronized 150, 627  
   this 45, 363  
   throw 255  
   throws 257  
   transient 152  
   try 245  
   unused words 21  
   void 16, 429  
   volatile 153  
   while 217

## L

labeled break statement 225  
 labels 223, 225  
   break 225  
   case 207  
   default 208  
   expressions 208  
   labeled statement 223  
   switch statement 207  
 language codes 532  
 last element 800, 801  
 last map entry 827  
 lazy quantifier 560  
 least significant bit 1010  
 left  
   associativity 167  
 left-most bit 1010

- legacy class 662
- legacy supertypes 719
- legal assignments 322
- length method 443
- length of file 471
- lenient 544
- lexical analyzer 571
- lexical tokens 20
- LF *see* Linefeed
- libraries 410
- lifetime 391
  - see* scope 35
- limit 564, 566
- line terminator 25
- linear implementation inheritance 286
- Linefeed 24
- line-oriented 581
- lines of text 494
- line-separator 491
- LinkageError 242
- linked data structures 662
- linked list 806
- listener 42
- listing directory entries 473
- lists 779, 804, 806
  - binary search 840
  - changing contents 841
  - iterators 806
  - positional 780
  - sublist 805
- literals 21
  - boolean 23
  - character 23
  - default type 22
  - double 22
  - escape sequences 24
  - false 23
  - float 22
  - floating-point 22
  - integer 22
  - null 21
  - predefined 21
  - prefix 22, 1006
  - quoting 23
  - scientific notation 22
  - string 25
  - suffix 22
  - true 23
- litmus test
  - design by inheritance 286
- live threads 636
- local 34
  - chaining of constructors 303, 416
  - classes 371
  - variables 36, 131
- local declarations 44, 45
- local variable declarations 48
- locale 446, 532, 576
  - country codes 532
  - country name 533
  - language codes 532
  - language name 533
  - predefined 533
- locale-sensitive 532
- localization 532
- locations
  - see* class path
- locks
  - class 626
  - object 626
- lockstep 567, 570, 572
- logical AND 558
  - & 194
- logical complement
  - ! 194
- logical exclusive OR
  - ^ 194
- logical inclusive OR
  - | 194
- logical operator 558
  - empty string 558
  - quantifiers 558
  - repetition operators 558
- logical OR 558
- logical XOR
  - ^ 194
- long 22, 28
  - suffix 22
- longest possible match 559
- lookahead 572
- lookahead methods 573
- lookahead-and-parse loop 572
- loop condition 216
- loops *see* iteration statements
- loose coupling 346
- loss of precision 160
- lower bound 675
- lower bounded wildcard
  - ? super T 676
- lower bounded wildcard references 683
- lower entry 828
- lower key 828

**M**

- magnitude 160
- main method 16, 17, 94
  - modifiers 95
- main thread 615
- maintaining integrity 627
- manifest constant 148
- manifest file
  - See* JAR utility 120
- manipulating 539
- many-to-one relation 821
- map views 822
- mappings 780
  - See* entries 780
- maps 779, 780, 821, 823
  - access order 823
  - entries 780
  - key insertion order 823
  - mappings 780
  - multimap 702
  - ordering mode 823
  - submap view 827
- marker interfaces 310, 511
- match-and-replace loop 570
- match-and-replace mode 567
  - lockstep 570
  - non-terminal append-and-replace step 570
  - terminal append-and-replace step 570
- Matcher 566
- matcher
  - lockstep 567
  - match-and-replace loop 570
  - match-and-replace mode 567
  - one-shot matching 566
  - reset 567
  - substrings 568
  - successive matching 567
- Math class 47
- mathematical set 796
- MAX\_VALUE constant 433
- member declarations 40, 310
- members 3, 129
  - access 45
  - accessibility 135, 138
  - default values 33
  - final 148
  - inheritance 284
  - instance 40
  - modified 284
  - modifiers 146
  - of objects 7
  - scope 129
  - short-hand 46
  - static 7, 40, 147
  - terminology 10
  - variables *see* fields
- memory management 390
- message
  - receiver 7
- metacharacters 555
  - escaping 561
- method
  - native 258
- method call 7, 44, 81
  - chaining 458, 459
  - non-varargs 93
  - varargs 93
- method declarations 40
- method header 150, 151
- method invocation conversions 164, 323
- method modifiers 44
- method overloading 47, 292
- method overriding 288, 292, 418
- method signature 44, 289
- methods 3
  - @Override 289
  - abstract 150
  - abstract method declarations 310
  - accessibility 44
  - advisory 638, 639
  - ambiguous call 324
  - automatic variables *see* local variables
  - blocks 44
  - body 44, 131
  - bridge 716
  - call chaining 81
  - call *see* method call
  - calling varargs method 91
  - chained 447
  - clone 425
  - declaration 44, 81
  - dynamic lookup 341
  - equals 193, 425
  - exceptions 44
  - finalize 396, 425
  - fixed arity 90
  - formal type parameter list 44
  - generic 699
  - getClass 425
  - header 44
  - implementation 150
  - invocation *see* method call
  - local declarations 44

- local variables
  - main *see* main method
  - method invocation conversions 323
  - modifiers 44
  - most specific 324
  - mutually exclusive 629
  - name 81
  - native 151
  - objects 45
  - overloaded resolution 324
  - overloading *see* method overloading
  - overriding *see* method overriding
  - overriding vs. overloading 292
  - parameters 44
  - recursive 243
  - return 228
  - return value 44
  - signature 44, 47, 292, 716
  - subsignature 718
  - synchronized 150, 627
  - termination 228
  - throws clause 257
  - toString 425
  - valueOf 450
  - variable arity 90
  - MIN\_VALUE constant 433
  - minimizing overhead 392
  - modification time 471
  - modifiers
    - abstract 135, 150
    - accessibility 132, 138
    - classes 135
    - final 148
    - members 146
    - native 151
    - static 147
    - strictfp 177
    - synchronized 150
    - transient 152
    - volatile 153
  - monitor 626
    - see* locks
  - most significant bit 1010
  - most specific method 324
  - multidimensional arrays 76
  - multi-line mode 572
  - multimap 702
  - multiple assignments 170
  - multiple bounds 686
  - multiple implementation inheritance 309
  - multiple interface inheritance 309
  - multiple-line comment 26
  - multiplication operator
    - \* 178
  - multiplicative operators 178
  - multisets 796
  - multitasking 614
  - multithreaded programming 614
  - mutable character sequences 456
  - mutex
    - see* locks 626
  - mutually comparable 838, 840, 844
  - mutually exclusive
    - actions 206
    - locks 626
    - operations 627
  - MVC 346
- N**
- name 20, 122
  - name clash 721
  - named constants 148
  - names of files 470
  - namespaces 48
  - naming conventions of files 468
  - naming patterns 41
  - NaN 177
  - narrower range 160
  - narrowing conversions
    - primitive 160
    - reference 161
  - narrowing reference conversions 328, 679, 703
  - native libraries 410
  - native methods 151, 258
    - header 151
  - natural order 765, 838, 842
  - navigable map 779
    - first entry 827
    - last entry 827
    - range-views 827
    - views 827
  - navigable set 779, 800
    - ceiling 802
    - floor 802
    - higher 802
    - iterator 802
    - lower 802
  - NavigableMap 827
  - NavigableSet 800
  - negation 556
  - negative zero 177
  - nested classes 352



- anonymous 377
    - full name 356
    - generic 731
    - local 371
    - non-static member 359
    - static members 355
    - synchronization 630
  - nested parameterization 670
  - nested parameterized types 706
  - nested types 40, 352
  - nested wildcards 678
  - new operator 4, 48, 70, 365, 377, 416
    - special form 361
  - Newline *see* Linefeed
  - NL *see* Newline
  - NoClassDefFoundError 243
  - node 662
  - non-associativity 167
  - non-metacharacters 555
  - non-reifiable types 722, 729
  - non-runnable states 635
  - nonstatic code
    - See* nonstatic context 40
  - nonstatic context 40
  - non-static field 10
  - non-static member classes
    - full name 356
  - non-static type 665
  - non-terminal append-and-replace step 570
  - non-varargs method call 93
  - normal class 135
  - normal execution 237
  - notifying threads 426, 640
  - null reference 21
    - casting 165, 328
  - nulling references 393
  - NullPointerException 242
  - Number class 433
  - number formatter 546, 547
    - currency 546
    - numbers 546
  - number formatting
    - factory methods 546
  - number systems
    - base 22
    - binary 1006
    - converting 1006, 1008
    - decimal 22, 1006
    - hexadecimal 22, 1006
    - octal 22, 1006
  - NumberFormat 546
  - NumberFormatException 242, 430
  - numbers 532, 546
    - formatting 546
  - numeric promotions 165
    - assignment 171
    - binary 166
    - unary 165
  - numeric wrapper classes 433
- O**
- object 4
  - Object class 286, 424
  - object creation 519
  - object creation expression 185
  - object hierarchy 287
  - object lock
    - see* locks 626
  - object references 4, 31
  - object state 6, 48, 86, 416
  - object-oriented design 345
    - cohesion 346
  - object-oriented programming 2
  - objects 13
    - alive 391
    - arrays 69
    - callee 81
    - caller 81
    - Class class 425
    - cleaning up 392
    - cloning 425
    - communication 81
    - comparing 425, 771
    - composite 391
    - constituent 12, 391
    - constructing 416
    - contract 345
    - creation 519
    - decoupling 341
    - deserialization 510
    - destroying 396
    - equality 193, 425
    - exchanging information 81
    - identity 5
    - immutable 428
    - implementation 345
    - initial state 416
    - initialization 5, 48
    - initializer block 413
    - internals 345
    - lifetime 391
    - members 7

methods 45  
 Object class 424  
 outer 362  
 persistence 510  
 persistent 152  
 reachable 391  
 reading 510  
 resurrection 391  
 serialization 510  
 services 345  
 state *see* object state  
 streams 477  
 value equality 193  
 writing 510  
 octal conversions  
   octal format 598  
 octal numbers 1005  
 one-dimensional arrays 69  
 one-shot matching 566  
 on-the-fly classes 352  
 OOP 2  
 open range-view operations 805  
 operands 164  
   evaluation order 168  
 operations 2  
 operators 166  
   - 180  
   -- 186  
   ! 194  
   != 191, 192  
   % 178  
   %= 182  
   & 194  
   && 196  
   &= 195  
   \* 174, 178  
   \*= 182  
   + 180, 185  
   ++ 186  
   += 182  
   . 7, 105  
   / 178  
   /= 182  
   < 190  
   <= 190  
   -= 182  
   = 169  
   == 191, 192  
   > 190  
   >= 190  
   ? : 201  
   ^ 194  
   ^= 195  
   | 194  
   |= 195  
   || 196  
   assignment 167, 169  
   associativity 166  
   binary 167  
   boolean 190, 191, 194  
   cast 166  
   comparisons 190  
   compound assignment 182, 195  
   conditional 196, 201  
   decrement 186  
   dot 7  
   equality 191  
   execution order 168  
   floating-point 176  
   floating-point division 178  
   floating-point remainder 179  
   increment 186  
   instanceof 201, 327, 328  
   integer arithmetic 175  
   integer division 178  
   integer remainder 179  
   logical 194  
   multiplicative 178  
   new *see* new operator  
   overflow 176  
   overloaded 175, 178  
   postfix 167  
   precedence 166  
   relational 190  
   short-circuited 196  
   string concatenation 185  
   ternary 166, 167  
   unary 166, 167, 177  
   unary - 177  
   unary + 177  
 optional operations 784  
 ordered collections 779, 804  
 ordering mode 823  
 out of bounds 805  
 outer object 362  
 output 468  
 output stream 468  
 OutputStream class 475  
 overflow 176, 1010  
 overloaded 175  
 overloaded method resolution 324  
 overloading 717  
   constructors 51  
   method resolution 324

- methods 47, 292
  - overloading vs. overriding 292
  - override 259
    - see* @Override
  - override-equivalent 717
  - overriding
    - @Override 718
    - equals 193, 751
    - finalizers 396
    - hashCode 760
    - methods 288, 292
    - override-equivalent 717
    - toString 185
  - overriding methods
    - covariant return 290
  - overriding vs. overloading 292
- P**
- package accessibility 132, 139
  - package directory 115
  - package statement 104, 106
  - packages 105
    - accessibility *see* package accessibility
    - declaration 104
    - definition 106
    - destination directory 115
    - hierarchy 105
    - java.io 468
    - java.lang 424
    - java.util 777
    - members 105
    - naming scheme 106
    - package directory 115
    - running code from 117
    - short-hand 108
    - statement *see* package statement
    - unnamed 106
    - using 107
  - parameter
    - varargs 90
  - parameter list *see* formal parameters
  - parameter passing
    - by value 81
    - variable arity 90
  - parameterized types 665, 705
    - as formal parameters 707
  - parameters 44
    - actual 81
    - array elements 87
    - final 89
    - fixed arity 90
    - formal *see* formal parameters
    - implicit 45
    - passing 81
    - primitives 82
    - references 84
    - this 45
    - variable arity 90
  - parent class 284
  - parent pathname 471
  - parentheses 166
  - parse method 572
  - parser 543, 571
  - parse*Type* method 434
  - parsing
    - date/time 543
    - numbers 547
    - primitive values 576
  - parsing numeric values 434
  - partial implementation 312
  - partly synchronizing a method 630
  - pass by value 81
  - passing
    - parameters 81
    - references 84
    - variable arity parameters 90
  - passwords
    - reading from console 501
  - path lists 469
  - path separator 468
  - pathname 121, 468, 470
  - paths
    - see* class path
  - paths of execution 614
  - path-separator character 119
  - Pattern 562
  - pattern 554
    - application 563
    - compile 554
    - creating matcher 563
    - engine 554
    - see also* regular expression
  - permutation 559
  - permute elements 839
  - persistence
    - see* serialization
  - persistent objects 152
  - polymorphic method 699
  - polymorphism 319, 340, 345, 374
  - polynomial 764
  - positive zero 177
  - possessive quantifiers 560
  - postconditions 275, 376

- postfix operators 167
- precedence rules 167
- precision 171, 596
- preconditions 275
- predefined identifiers 20
- predefined literals 21
- preempted threads 638
- preemptive scheduling 638
- prefix 22, 1006
  - 0 22, 1006
  - 0x 22, 1006
- primitive types 13
  - autoboxing 430
  - unboxing 431
    - see also* primitive values
- primitive values
  - assignment 169
  - binary representation 479
  - equality 191
  - passing 82
- print writers 490
- priorities 638
- priority heap 783
- priority of thread 638
- priority queue 783
- private 11
- private members 143
- process-based multitasking 614
- processes 614
- program
  - application 15
  - arguments 95
  - compiling 16
  - correctness 638
  - formatting 26
  - performance 638
  - running 17
- program arguments 95
- programmer exam 851
- promotion 165
- properties 2, 41, 122
  - D option 122
    - see also* class members
- property
  - name 122
  - value 122
- protected 11
- protected members 141
- public 16
- public members 139

**Q**

- quantifiers 558
- questions 854
- queue 779, 783
- quotation mark 24, 25
- quote ' 23
- quotient 178

**R**

- race condition 627
- radix
  - See* base 431
- ragged arrays 76
- random access 806
- random order 839
- range 556
  - character values 29
  - floating-point values 29
  - integer values 28
- range-view 800, 801, 805, 827
- raw types 670
  - unchecked warnings 670
- read access 470, 472
- readers 488
- reading bytes 475
- reading data 468
- Ready-to-run state 634
- reclaiming memory 390
- record 611
  - fields 611
- re-creating objects 510
- recursive bound 713
- reducing complexity 345
- reference types 32, 287
  - classes 40
  - enum types 55
- reference values 4
- reference variable 31
- reference variables 4
- references 4, 9, 31, 32, 81
  - abstract types 135
  - aliases 193
  - array 70, 73, 319
  - assignment 169
  - casting 165, 327
  - dangling 390
  - declared type 288
  - downcasting 161
  - dynamic type 288
  - equality 192

- interface type 314
  - narrowing conversions 161
  - null *see* null reference
  - parameterized types 665
  - passing 84
  - raw types 682
  - reachable 390, 391
  - super 295, 373
  - this 45, 373
  - upcasting 161
  - widening conversions 161
  - reflection 728
  - regular expression 554
    - anchoring 557
    - anchors 557
    - automaton 554
    - backtracking 559
    - boundary matchers 557
    - character 554
    - character class 555
    - concatenation 558
    - empty string 558
    - escaping metacharacters 561
    - greedy quantifiers 559
    - group 554, 563
    - group logical operator 558
    - intersection 556
    - intervals 556
    - lazy quantifiers 560
    - logical AND 558
    - logical operator 558
    - logical OR 558
    - longest possible match 559
    - metacharacters 555
    - negation 556
    - non-metacharacters 555
    - possessive quantifiers 560
    - precedence of logical operators 558
    - quantifiers 558
    - range 556
    - reluctant quantifiers 560
    - reluctantness/laziness 560
    - sequence of characters 554
    - shortest possible match 560
    - simple class 556
    - subtraction 556
    - union 556
  - reifiable types 722, 723
  - relational operators 190
  - relative pathnames 469
  - reluctant quantifiers 560
  - reluctantness/laziness 560
  - remainder 178
  - remainder operator
    - % 178
  - remove white space 449
  - renaming files or directories 474
  - repetition operators 558
  - replacing characters 448
  - reserved identifiers 20
  - reserved keywords 21
    - const 21
    - goto 223
  - reserved literals
    - false 23
    - null *see* null reference
    - true 23
  - reset 567
  - resizable arrays 783, 806
  - resources 392
  - resurrecting objects 391
  - retaining elements 785
  - return statement 228
  - return type
    - covariant
  - return value 7
  - reuse of code 284, 344
  - rhyming order 772
  - right associativity 168
  - right-most bit 1010
  - ripple effect 345
  - role relationship 344
  - Round-Robin scheduling 638
  - Runnable interface 615, 620
  - Running state 635
  - runtime checks 164
  - Runtime class 424
  - runtime class 425
  - runtime environment 390
  - runtime stack 235, 390
  - RuntimeException 241
- ## S
- scanner 571
    - delimiter pattern 574
    - delimiters 571, 576
    - exception handling 578
    - lexical analyzer 571
    - line-oriented 581
    - lockstep 572
    - lookahead 572
    - lookahead methods 573
    - lookahead-and-parse loop 572

- multi-line mode 572, 581
- parse method 572
- parser 571
- parsing next token 575
- parsing primitive values 576
- source 571, 573
- token pattern 574
- tokenization 571
- tokenizer 571
- tokenizing mode 572
- tokens 571
  - using delimiters and patterns 579
- Scanner class 571
- schedulers 638
- scientific notation 22
- scope 129
  - block 131
  - catch block 251
  - class 129
  - disjoint 132
- search key 751
- searching 778, 838
  - arrays 843
  - collections 840
- searching in string 448
- SecurityManager class 424
- selection statements 204
- self-referential data structure 662
- semantic definition 20
- semicolon 45
- separators 468
- sequence 779
- sequential I/O 468
- serialization
  - customizing 517
  - inheritance 519
  - transient fields 516
- serialization of objects 510
- services 345
- set 779
  - capacity 797
  - load factor 797
- set operation 680
- setter method 41
- shallow copying 425
- shared resource 626
- short 22, 28
- short-circuit 196
  - evaluation 197
- shortest possible match 560
- shuffle elements 839
- signature 47, 292, 716
- significant digits 599
- simple
  - assignment operator 169
  - if 204
  - statement 45
- simple class 556
- single implementation inheritance 286, 309, 313
- single quote 24
- single static import 109
- single-line comment 26
- single-quotes 23
- single-valued maps 821
- size of
  - file 471
- skeletal source file 104
- sleeping 640
- Sleeping state 635, 640
- sorted map 779, 780
  - comparator 827
- sorted set 779, 800, 801
  - comparator 801
  - half-open interval 801
- sorting 778, 838
- sorting arrays 842
- source 42, 571
  - file 15, 106
  - file name 104
  - file structure 104
- source of data 468
- spaces 25
- spawning threads 615
- special character values 24
- specialization 11
- specialized classes 286
- splitting 451, 563
  - delimiters 563
  - limit 564, 566
- stack 3
- stack frame
  - See method execution 236
- stack trace 239
  - See method execution 236
- StackOverflowError 243
- standard error 499
- standard input 499
- standard output 499
- starting threads 616
- state *see* object state
- statements 45
  - assert 265
  - break 224

- compound 45
- conditional 204
- continue 226
- control flow 45, 204
- control transfer 223
- declarations 45
- do-while 217
- empty 45
- expression 45
- for(:) 220
- for(;;) 218
- if 204
- if-else 205
- iteration 216
- labeled 223
- return 228
- selection 204
- simple 45
- simple if 204
- switch 207
- throw 255
- transfer 223
- while 217
- static
  - initializer block 410
  - members *see* static members
  - methods 7, 10, 44
  - variable initialization 33
  - variables *see* static variables
- static code
  - See* static context 40
- static context 40
- static field 10
- static import 109
  - conflicts 112
  - on demand 109
  - shadow static members 111
  - single static import 109
- static initializer block 40
- static initializer blocks 276
- static keyword 109, 410
- static member classes 355
  - full name 356
- static members 8, 9, 10, 40
- static variables 7, 10, 35
- storing objects 152
- streams 468
  - chaining 479
  - files 477
  - filters 468, 477, 479
  - of characters 488
  - streams of objects 477
- strict formatter 544
- strictfp 177
- string
  - splitting 563
- string builders 186
  - appending 458
  - capacity 456, 460
  - constructing 457
  - deleting 458
  - differences with strings 458
  - individual characters 457
  - inserting 458
  - length 457
  - thread-safety 456
- String class
  - see* strings
- string conversion 447
- string conversions 162, 598
- string literals 439
  - interned 440
- string pattern matching 554
- splitting 563
- StringBuffer class 456
  - see* string builders
  - thread-safe 456
- StringBuilder class 456
  - see* string builders
- StringIndexOutOfBoundsException 449, 457, 458
- strings
  - appending 458
  - buffers 456
  - builders 456
  - capacity 460
  - changing case 446
  - compareTo 445
  - comparing 445, 766
  - concatenation 185, 446
  - concatenation operator + 186, 447
  - constructing 457
  - copying characters 443
  - creating 439
  - deleting 458
  - differences with string builders 458
  - equals 445
  - extracting substrings 449
  - finding index 448
  - formatting of values 450
  - ignoring case in comparison 445
  - immutable 439
  - individual characters 443, 457
  - initializing 439

- inserting 458
  - interned 440
  - length 443, 457
  - lexicographical order 445
  - literals 25
  - mutable 456
  - natural order 766
  - pattern matching 451
  - read character at index 443
  - replacing 448
  - searching 448
  - splitting 451
  - substrings 449
  - trimming 449
  - strongly typed language 164
  - subclass 10, 11, 12, 284
  - subinterface 313
  - sublist 805
  - submap view 827
  - subpackage 105
  - subsequence 443
  - subset 796, 798, 800, 801
  - subsignature 718
  - substring searching 448
  - substrings 448, 449, 568
  - subtraction 556
  - subtype bivariance 675
  - subtype contravariance 675
  - subtype covariance 317, 318, 674, 675
  - subtype invariance 675
  - subtypes 312, 675, 686
  - subtype-supertype relationship 161
  - successive matching 567
  - suffix
    - D 22
    - F 22
    - L 22
  - Sun Educational Services 852
  - super
    - construct 305
    - keyword 292, 295
    - reference 295
    - special form 366
  - super keyword 305
  - superclass 10, 11, 284
  - superclass-subclass relationship 286
  - superinterfaces 313
  - superset 798
  - supertypes 312, 676
  - supertype-subtype relationship 287
  - supplementary characters 439
  - SuppressWarnings *see* @SuppressWarnings
  - switch statement 207
    - break 209
    - default clause 208
    - enum types 211
  - switching between threads 614
  - synchronization 615, 626
  - synchronized
    - blocks 629, 630, 631
    - keyword 627
    - methods 150, 627
  - syntactically legal 20
  - System class 424
  - system classes 269, 272
  - system properties 122
- T**
- TAB *see* Horizontal tab
  - tabs 25
  - tabulators 25
  - tagging interface
    - see* marker interfaces
  - tags 27
  - tail map view 827
  - tail set view 800, 801
  - telephone directory order 445
  - terminal append-and-replace step 570
  - terminating loops 224
  - ternary conditional operator 166, 167, 201
  - testing locations 853
  - text files 492, 494
  - text streams
    - see* character streams
  - textual representation 425
  - this
    - reference 45
    - special form 363
  - this() constructor call 302, 417
  - thread 390
    - live 390
  - Thread class 615
  - thread creation 615
  - thread of execution 614
  - thread safety 614
  - thread-based multitasking 614
  - ThreadDeath 242
  - threads 424, 647
    - alive 615, 636
    - blocked 626
    - Blocked for I/O state 635
    - Blocked for join completion state 635
    - Blocked for lock acquisition state 635



- child 615, 619
- class 615
- code executed 616
- creation 615
- daemon 615
- dead state 635
- death 239
- determining state 635
- exception propagation 239
- extending Thread 619
- IllegalThreadStateException 615
- implemented by inner classes 622
- joining 637
- main 615
- names 617
- non-runnable state 635
- notification 640, 642
- notifying 426, 640
- priorities 638
- Ready-to-run state 634
- Runnable 615, 620
- running 619
- Running state 635
- scheduler 638
- sleeping 640
- Sleeping state 635, 640
- spawning 615
- starting 616, 620
- states 634
- status 615
- switching 614
- synchronization 150, 626
- synchronized 627
- termination 650
- Thread class 615
- transitions 634
- user 615
- wait set for 641
- waiting 426
- waiting for notification state 635
- yielding 639
- thread-safe 439, 456, 784
- throw statement 255
- Throwable class 239, 424
- throwing exceptions 235
- throws clause 257
- time
  - see date and calendar*
- time of day 535
- time-sliced scheduling 638
- token pattern 574
- tokenization 571
- tokenize 563
- tokenizer 571
- tokenizing 572
- tokenizing mode 572
- tokens 20, 563, 571
- toString method 425, 431
- total order 765
- transfer statements 223
- transient fields 516
- transient variables 152
- transitive relation 287
- tree map 828
- tree set 802, 828
- trees 782
- trim 449
- true literal 23
- truth-values 23, 30
- try block 245
- try-catch-finally construct 245
- type cast 164
- type cast expression 327
- type declarations 104
- type erasure 670, 714
- type hierarchy 161, 287
- type import
  - see import*
- type information 662
- type instance 665
- type parameter constraints 684
- type parameters 310, 731
  - actual 665
  - bounds 684
  - formal 663
  - recursive bounds 712
  - restrictions 665
- type safe 323
- type variables 663
- types
  - boolean 28, 30
  - byte 22, 28
  - casting 164
  - char 29
  - classes *see classes*
  - comparing 328
  - compatibility 164
  - double 29
  - exceptions 239
  - float 29
  - floating-point 28, 29
  - int 22, 28
  - integers 28, 29
  - integral types 28

- long 22, 28
- nested 352
- parsing 434
- short 22, 28
- wrappers 428
- see also* classes

## U

### UML 2

- accessibility 139
- aggregation 12
- classes 3
- inheritance 10
- see also* diagrams

- unary numeric promotion 165

- unary operators 166, 167

- unbounded type parameter 675, 677

- unbounded type references 684

- unbounded wildcard

- ? 675, 677

- unbounded wildcard references 682

- unboxing 431

- assert statement 265

- do-while statement 217

- for(:) statement 221

- for(;;) statement 218

- if statement 204

- if-else statement 205

- switch statement 207

- while statement 217

- unboxing conversions 162

- uncaught exceptions 239

- unchangeable variables 149

- unchecked call warning 671, 682

- unchecked cast warning 724

- unchecked conversion warning 671

- unchecked conversions 162

- unchecked exceptions 244

- unchecked generic array creation warning 729

- unchecked warnings 161, 671

- @SuppressWarnings("unchecked") 726

- call warning 671, 682

- cast warning 724

- conversion warning 671

- generic array creation warning 729

- underflow 176, 1010

- Unicode 23, 29, 488

- Unified Modeling Language *see* UML

- union 556, 796

- unsafe casts 329

- unsupported operations 784, 821

- unused keywords 21

- upcasting 161

- upper bound 675, 685

- upper bounded wildcard

- ? extends T 675

- upper bounded wildcard references 682

- user threads 615

- using packages 107

- using variables 31

- UTF-16 439

- supplementary characters 439

## V

- value 122

- value collection 821

- value objects 780

- valueOf method 430, 431, 450

- values 821, 1010

- constants 21

- overflow 1010

- underflow 1010

- wrap-around 1010

- see also* variables

- varargs

- generic array creation 729

- varargs calls 93

- varargs parameter 90

- variable arity method 90

- variable declarations 4, 31, 131

- variable initialization 8, 34

- variables 31

- blank final 89, 149

- constant values 148

- declarations *see* variable declarations

- default values 33

- final 148

- identifiers 31

- in interfaces 314

- initialization *see* variable initialization

- lifetime 35

- local 131

- parameters 44, 81

- reference variable 32

- references 32

- static 7

- storing 152

- transient 152

- volatile 153

- vector 806

- Venn diagrams 785

- view of list 805
  - VirtualMachineError 242
  - void 228, 429
  - Void wrapper class 428
  - volatile variables 153
  - voucher 852
- W**
- wait set 641
  - waiting for conditions 641
  - waiting for notification state 635
  - waiting threads 426
  - while statement 217
  - white space 25, 449
  - whole-part relationship 287
  - widening conversions
    - primitive 160
    - references 161
  - widening reference conversions 287, 328, 679
  - wider range 160
  - width 596
  - wildcard parameterized types 678
    - type hierarchy 676, 678
  - wildcard types 675
    - see wildcards 675
  - wildcards 673, 675
    - ? 682
    - ? extends T 682
    - ? super T 683
  - assignment 679
  - bounded 675
  - capture 703
  - capture conversion 705
  - flexible comparisons 709
  - nested 705
  - parameterized types 678
  - restrictions 677
  - subtype relationships 673
  - type hierarchy 676, 678
- wrapper classes 28, 424, 428, 766
- wrapper types 162
- wrappers
  - interned 432
- write access 470, 472
- writers 488
- writing bytes 475
- Y**
- yielding 639
- Z**
- zero
    - negative 177
    - positive 177