

Index

Symbols

- 169
-- 176
^ 184, 189
^= 185
_ 32
; 50
: 110
! 184
!= 181, 342
?: 194
. 7, 72, 97, 108
... 81, 85
' 32, 33
" 34
[] 59, 61, 195
{ } 50, 60, 117
@FunctionalInterface 442, 443
@Override 270
@param 49, 56
@return 225
@throws 253
* 100, 163, 167
*= 172
/ 167
/* and */ 35
/** and */ 36
// 35
/= 172
\
& 184, 189
&& 186
&= 185
% 167, 168
%= 172
+ 169, 174

+ concatenation 364
++ 176
+= 172
< 180
<= 180
<> 414, 415, 416
-= 172
= 158
== 181, 342, 351, 359
-> 195, 439, 444
> 180
>= 180
| 184, 189
|= 185
|| 186
~ 189

A

ability interfaces
 see marker interfaces
abrupt method completion 232
absolute adjusters 470
abstract
 classes 120
 interfaces 290
 methods 136, 291, 442
abstract 120, 136, 290, 291, 442
abstract method declarations 442
 in interfaces 290, 291
abstraction 2, 10
accessibility 7, 17, 114
 default 118, 127
 members 114, 120, 123
 modifiers 118
 package 118

- private 128
- protected 126
- public 124
- UML notation 124
- accessibility modifiers 48, 53
- activation frame 384
 - see* method execution 230
- actual parameter 72
- actual parameter list 72, 315
- adding to class 264
- additive operators 169
- aggregation 10, 12, 267
 - hierarchy 267
 - versus inheritance 331
- aliases 6, 75, 182, 183
 - see also* references
- ambiguous call 316
- analyzing program code 512
- and operator 189
- annotations
 - @Override 270
- anonymous arrays 63, 66
 - [] 63
- anonymous classes 436
- anonymous functions 439
- API (application programming interface) 22
- apostrophe 33
- application 16
- architecture neutral 23
- argument
 - see* actual parameter
- arguments to main method 85
- arithmetic compound assignment
 - operators 172
- ArithmeticException 236
- arity 81
- array creation expression 59, 63
- array initializer 60, 63, 66
- array store check 311
- array types
 - see* arrays
- ArrayIndexOutOfBoundsException 61, 236
- ArrayList 366, 414
 - add collection 419
 - add element 417, 419
 - autoboxing 421
 - capacity 416
 - clear list 420
 - comparison with arrays 425
 - constructing 415
 - constructors 418
 - converting to array 424
 - element search 423
 - element type 415
 - filtering 434
 - import 415
 - inheritance hierarchy 415
 - initial capacity 417
 - insertion order 414
 - list of lists 417
 - membership test 422
 - modifying 419
 - nested lists 417
 - object value equality 422
 - open range-view operations 414
 - ordered 414
 - position-based access 415
 - positional index 422
 - positional order 414
 - positional retrieve 422
 - querying 422
 - references 415
 - remove collection 420
 - remove element 419
 - replace element 419
 - size 416, 422
 - sorting 425
 - subtype relationship 418
 - textual representation 417
 - traversing 423
 - trim to size 420
 - type-safety 416, 417, 418
 - unchecked conversion warning 416
 - zero-based index 414
- arrays 58, 342
 - [] 59, 61
 - { } 60
 - anonymous 63, 66
 - array creation expression 59
 - array initialize list 60, 63
 - array initializer 60, 66
 - array name 59
 - array size 60
 - array store check 311
 - ArrayIndexOutOfBoundsException 61
 - bounds 61
 - construction 59
 - declarations 59
 - default initialization 59, 60
 - dynamic 415
 - element access expression 61
 - element default value 310
 - element type 59

- elements 58, 61
- index 58
- index expression 61
- initialization 60, 65
- iterating over 217
- length 58
- multidimensional 63, 65
- objects 342
- ragged 65
- reference 59, 62, 311
- searching 69
- sorting 68
- subtype covariance 309
- traverse an array 62
- using 61
- arrays of arrays 59, 65
 - multidimensional 65
- ArrayStoreException 311, 418, 424
- arrow -> 195, 439, 444
- ASCII 32, 38
- AssertionError 237
- assignable 147, 314
- assignment compatible 148, 314, 416
- assignment conversions 147
- assignment operator 5
- assignments
 - arithmetic compound operators 172
 - bitwise 192
 - cascading 159
 - compound operators 185, 192
 - expression statement 159
 - implicit narrowing 160
 - multiple 159
 - numeric conversions 160
 - operator 151, 158
 - primitive values 159
 - references 159
 - widening reference conversions 267
- association 12
 - aggregation 267
 - composition 267
 - realization 296
- associativity 152
- atomic values 13
- attributes *see* properties
- autoboxing 68, 348
 - for(·) statement 218
- AutoCloseable 387
- automatic garbage collection 6, 384
- automatic variables *see* local variables

B

- backslash 33
- backspace 33
- base 30, 349, 352
- base class 264
- basic for statement
 - 215
- Before Current Era (BCE) 464
- behavior 433
- behavior parameterization 434, 441
- binary
 - numeric promotion 150
 - operators 151
- binary search
 - arrays 69
- bit mask 190
- bit patterns 154
- bitwise
 - and* operator 189
 - assignment 192
 - complement 189
 - compound assignment 192
 - operators 189
 - or* operator 189
 - xor* 189
- bitwise AND
 - & 189
- bitwise complement
 - ~ 189
- bitwise exclusive OR
 - ^ 189
- bitwise OR
 - | 189
- bitwise XOR
 - ^ 189
- blank final variable 80, 134
- blocks 49, 50, 117
 - scope 117, 448
 - try 240
- boilerplate code 436
- Boolean
 - condition 200
- Boolean wrapper class 355
- booleans 37, 39
 - casting 149
 - expressions 180
 - literals 32
- boxing conversions 145, 146
- break statement 205, 206, 221
- BS *see* backspace
- building abstractions 10
- byte 30, 38

bytecode 16, 23

C

C 137

C++ 23, 137

cache 139

call by reference 77

call by value 77

call chaining 72

call signature 316

call stack

see JVM stack 230

callee 72

caller 72, 224

capacity 416

carriage return 33, 35

cascading assignments 159

cascading if-else statements 203

case labels 203, 205

case sensitivity 28

cast operator 145, 148, 151, 162, 172, 182, 320

casting 147, 148, 149

see also conversions

catch clause 240

uni- 239

catching exceptions 230

catch-or-declare 251

CertView 509

chaining 406

constructors 287, 406

finalizers 391

char 38

character case 364

character sequences

see strings *and* string builders

character set

ASCII 32, 38

ISO Latin-1 32, 38

Unicode 32, 38

Character wrapper class 354

characters 38

literals 32

searching for 367

CharSequence interface 360, 365, 369

checked exceptions 237

child class 264

choosing between String and StringBuilder

class 374

Class class 343

class file 16

class hierarchy

see inheritance hierarchy

class inheritance

see implementation inheritance

class method 10

class modifiers 48

class path 107

absolute pathnames 110

entries order 110

entry-separator character 110

fully qualified package name 109

path-separator character 110

relative pathnames 110

search in a named package 109

searching for classes 107

whitespace 110

class search path

see class path

class variable 10

ClassCastException 236, 321

classes

abstract 120

accessibility 118

adding to 264

base 264

body 48

child 264

cohesion 335

concrete 121, 122, 436

constructors 53, 282

coupling 336

declarations 48, 96

definitions 2, 5

derived 264

diagram 8, 9

encapsulation 335

extending 122, 264

final 122

final vs. abstract 122

fully qualified name 107

fully qualified package name 98

generalized 266

grouping 97

header 48

implementing interfaces 291

initialization 409

instance members 48

instances 4

members 7

methods 132

modifiers 120

name 97

- normal 121
- Object 342
- parent 264
- runtime 343
- scope 114
- searching for 107
- specialized 266
- static members 48
- subclass 10, 264
- superclass 10, 264
- variables 132
- wrappers 342, 346
- ClassLoader class 342
- ClassNotFoundException 235
- CLASSPATH environment variable
 - see* class path
- classpath option
 - see* class path
- clauses
 - catch 240
 - extends 264
 - finally 240, 245
 - implements 291
 - throws 251
- cleaning up 386
- clean-up code 245
- client 7, 16
- Cloneable interface 343
- CloneNotSupportedException 343
- cloning objects 343
- code optimizations 134
- code reuse 23, 264, 334
- CodeRanch 508
- cohesion 335
 - coincidental 335
 - functional 335
 - high 335
- coincidental cohesion 335
- Collection 414
- collections 414
 - as single entity 414
 - elements 414
 - ordered 414
 - sorting 414
- command 17
 - java 17
 - javac 17
- command line 17, 86
- command prompt 17
- comments 35
- communication 7, 72
- Comparable interface 350, 363, 376, 425
- comparing objects 342
- comparing strings 363
- comparison 180
- compilation unit 98
- compiling Java source code 17
- complement
 - ~ 189
- completes abruptly
 - see* exception propagation 232
- composite object 10
- composition 12, 267
- compound statement 50
- concatenation of strings 364
- concatenation operator 174
- concrete classes 436
- concrete method 134
- ConcurrentModificationException 424
- condition
 - Boolean 200
 - expressions 200
- conditional 180
 - and* 186
 - operators 186, 194
 - or* 186
 - statements 200
- conditional expressions 194
 - associativity 195
 - nested 195
 - precedence 194
 - short-circuit evaluation 194
 - side effects 194
- conditions 180
- connecting punctuation character 28
- const 29
- constant declarations 290
- constant expression 147, 160, 161, 176
- constant field values
 - case labels 208
- constant string expressions 208
- constant values 30, 133
- constant variable 161
- constants 302
- constituent objects 10
- constituents 12
- constructing array 59
- constructor chaining 283, 287, 406
- constructors 3, 53, 282
 - accessibility 124
 - accessibility modifier 53
 - body 53
 - chaining 283, 287
 - class name 53

- declaration 48
- default 54
- header 53
- implicit default 54
- local declarations 53
- no-argument 53, 54, 283, 287
- non-zero argument 55, 287, 288
- overloading 56
- superclass constructor 54
- constructs 28
 - high-level 28
 - loops *see* iteration statements
- container
 - see* collections
- contains characters 368
- continue statement 223
- contract 2, 291, 293, 334, 335
- control flow
 - break 205, 221
 - continue 223
 - do-while 214
 - for(;;) 215
 - for(:) 217
 - if 200
 - if-else 201
 - iteration *see* iteration statements
 - loops *see* iteration statements
 - return 224
 - statements 50, 200
 - switch 203
 - throw 249
 - transfer statements 219
 - while 213
- control transfer 219
- conversion categories 147
- conversion contexts 147
- conversions 144, 311
 - assignment 147
 - contexts 147
 - identity 172
 - implicit narrowing 173
 - method invocation 148
 - narrowing reference 320
 - number systems 157
 - numeric promotions 149
 - parameters 73
 - reference casting 320
 - string concatenation 175
 - to strings 369
 - truncation 161
 - type-safe 315
 - unsafe casts 321

- widening reference 267, 320
- converting number systems 157
- converting values 348, 349, 350, 352, 353, 355, 369
- counter-controlled loops 215
- coupling 336
 - loose 336
- covariant return 269, 273
- cp option
 - see* class path
- CR *see* carriage return
- crab 217
- creating
 - objects 195
- criteria object 436
- currency symbol 28
- current directory
 - . 108
- Current Era (CE) 464
- current object 50

D

- d option 106
- dangling references 384
- data structures 414
- data types *see* types
- date
 - see* temporal objects
- date units 474
- date/time formatters
 - customized 486, 495
 - format styles 490
 - formatting 487
 - immutability 487
 - ISO-based default 486, 487
 - ISO-based predefined 486, 488
 - letter pattern 495
 - localized 486, 490
 - parsing 487
 - pattern letters 495, 496
 - thread-safety 487
- date-based values 462
- date-time
 - see* temporal objects
- DateTimeException 463
- DateTimeFormatter class
 - see* date/time formatters
- DateTimeParseException 477, 491
- DayOfWeek enum type 468
- declaration statement 4, 41, 171, 177, 187, 216

- declarations
 - arrays 59, 196
 - classes 48, 96
 - interfaces 96
 - local 50
 - main method 85
 - methods 49
 - multidimensional arrays 63
 - packages 96, 98
 - statements 50
 - variable arity method 81
 - declared type 268, 274, 275, 315
 - declared-type parameters 445
 - declaring *see* declarations
 - decoupling 330
 - decrement operator 176
 - deep copying 343
 - default
 - accessibility 118, 124, 127
 - constructor 54
 - exception handler 232
 - method 297
 - values 42, 400, 406
 - default 297
 - label 204
 - method 297
 - default constructor 54
 - default method 297, 442, 443
 - multiple inheritance 298
 - overriding 298
 - default package 98
 - deferred execution 451
 - definitions
 - inheritance 296
 - interfaces 290
 - delegating requests 334
 - derived class 264
 - destination directory 106
 - destination stream 18
 - destroying objects 390
 - diagrams
 - class 3
 - object 5
 - see also* UML
 - diamond operator (\diamond) 416
 - dictionary order 363
 - distributed 23
 - divide-and-conquer algorithm 69
 - dividend 168
 - division
 - floating-point 167
 - integer 167
 - division operator
 - / 167
 - divisor 168
 - documentation 35
 - documentation comment 35, 36
 - tags 36
 - documenting *see* documentation
 - dot 97
 - double 31, 39
 - double quote 33
 - do-while statement 214
 - downcasting 145
 - duplicating objects 343
 - Duration class 476
 - time-based 476
 - dynamic 23
 - dynamic arrays 415
 - dynamic binding
 - see* dynamic method lookup
 - dynamic method lookup 277, 329, 330
 - dynamic type 268, 274, 275
- ## E
- effectively final 448
 - element type 59, 415
 - elements 58, 414
 - eligible for garbage collection 385
 - ellipsis 81
 - else clause matching 203
 - embedded applications 22
 - empty statement 50
 - empty string 358
 - encapsulation 22, 97, 335
 - encapsulation of implementation 334
 - ends with characters 368
 - enhanced for loop 213
 - enterprise applications 22
 - enum constant 87
 - symbolic names 87
 - values 87
 - enum types 87, 103, 209, 303
 - declaring 87
 - finalization 391
 - named constants 87
 - natural order 90
 - ordinal value 90, 209
 - switch expression 204
 - using 88
 - enumerated types
 - see* enum types
 - EOFException 235

- equality 181, 342
 - equals method 183, 342
 - object value 183
 - objects 183
 - primitive values 181
 - reference values 182
 - equals method 183, 342
 - Error 237
 - escape sequences 33
 - evaluation order 152, 187
 - arithmetic expressions 164
 - evaluation short-circuits 187
 - exam 507
 - multiple-choice 513
 - program 510
 - questions 511
 - registration 508
 - result 511
 - testing locations 510
 - voucher 509
 - exam objectives
 - OCAJP8 515
 - Exception class 236
 - exception handler 230
 - see also* exceptions
 - exception handling
 - advantages 254
 - exceptions 230, 239
 - customized 238
 - default handler 232
 - handler 230
 - ignored 390
 - propagation 230
 - situations 235
 - throw 249
 - throwing *see* throwing exceptions
 - thrown by JVM 235
 - thrown by method 49
 - thrown programmatically 235
 - throws 251
 - types 233
 - uncaught 232
 - unchecked 237
 - exchanging information 72
 - explicit
 - garbage collection 393
 - explicit traversal 452
 - exponent 31
 - expression statements 50, 159, 177, 216, 217, 446
 - expressions 205
 - actual parameters 72
 - boolean 180
 - case labels 205
 - conditional 194
 - deterministic evaluation 150
 - label 205
 - return 224
 - statements 50
 - throw 249
 - extending
 - classes 264
 - interfaces 294
 - extends clause
 - see* extending
 - extensions
 - .class 16
 - .java 16
 - external libraries 403
 - extracting substrings 369
- ## F
- fall-through 204, 205
 - false literal 32
 - FF *see* form feed
 - field declarations 48
 - field hiding 275
 - field initialization 406
 - fields 2
 - file name 96
 - file path 105
 - separator character 105
 - filtering 434
 - final
 - classes 122
 - members 133
 - parameters 80
 - finalization 385
 - finalization mechanism 385
 - finalize method 343, 390
 - finalizer chaining 391
 - finalizer *see* finalize method
 - finally clause 240, 245
 - fixed arity method 81
 - fixed arity method call 84
 - float 31, 39
 - floating-point 37
 - double 39
 - float 39
 - literals 31
 - floating-point arithmetic 165
 - strictfp 166
 - floating-point data types 31

- floating-point division 167
- floating-point remainder 169
- flow control *see* control flow
- for(;;) statement 215
 - backward 216
 - forward 215
 - traverse array 62
- for(:) statement 217
 - traverse array 62
- for-each loop 213
- form feed 33, 35
- formal parameters 49, 53, 72, 117, 315
 - modifier 49
 - name 49
 - type 49
- formal type parameter 290
- format specifications 18, 370
- format specifier 19
- format styles 486, 490
- FormatStyle enum type 486, 490
- formatted output 18
 - format specifier 19
- formatted string 370
- formatting 35, 462, 486
- forward reference 400, 401, 403, 405, 406
- fractional signed numbers 37
- fully qualified class name 107
- fully qualified package name 97, 98, 100
- fully qualified type name 97, 101
- function 451
- function type 450
- functional cohesion 335
- functional interface 438
 - @FunctionalInterface 442, 443
 - abstract method 442
 - function type 450
 - functional method 442
 - general-purpose 443
 - generic 441
 - Predicate<T> 440, 451
 - primitive values 444
 - target type 450
 - see also* lambda expressions
- functional method 442
- functional programming 24
- functionality 433
- functional-style programming 433

G

- garbage collection 387, 389, 390, 393
 - automatic 384

- facilitate 387
- general abstractions 266
- general loops 215
- generalization 10
- generalized classes 266
- generic method 423
- generic type 414
- goto 29, 220
- grammar rules 28
- grouping 97

H

- handles *see* references
- has-a* relationship 267
- hash code 52, 343
- hash tables 52
- heap 384
- hiding internals 335
- high cohesion 335
- high-performance 24
- horizontal tab 33
- hotspots 24
- HT *see* horizontal tab

I

- IDE (integrated development environment) 508
- identifiers 28
 - predefined 29
 - reserved 29
 - variable 40
- identity conversion 146, 172
- identity of object 5
- IEEE 754-1985 38
- if block 200
- if-else statement 201
- ignored exceptions 390
- IllegalArgumentException 236, 495
- immediate superclass 285
- immutable 462
- immutable objects 346, 357
- immutable strings 357
- implementation inheritance 264
- implementations 2, 266, 335
 - inheritance hierarchy 122
- implementing
 - interfaces 291
- implements clause 291
- implicit
 - inheritance 264

- narrowing conversions 173
- implicit default constructor 54
- import
 - declaration 100
 - see also* static import
 - single-type-import declaration 100
 - statement 96
 - type-import-on-demand declaration 100
- importing
 - enum constants 103
 - reference types 99
 - static members 101
- increment operator 176
- index 58
- index expression 61
- IndexOutOfBoundsException 361, 369, 375, 376, 419, 422
- individual array elements 61
- inequality 181
 - see also* equality
- inferred-type parameters 445
- infinite loop 217
- infinity 165, 349
 - negative 165
 - positive 165
- information hiding 335
- inheritance 10, 267
 - hierarchy 266
 - supertype-subtype relationship 267
- initial capacity 417
- initial state of object 406
- initialization
 - arrays 60, 65
 - code 60
 - default values 42
 - for statement 215
 - objects 5
 - references 41
 - variables 41
- initializer 399
 - declaration-before-reading rule 401
 - static 400, 401, 405, 409
- initializer block
 - instance 404
 - static 402
- initializer expression 400
- initializers
 - non-static block 48
 - non-static field 48
 - static block 48
 - static field 48
- initializing *see* initialization
- insertion order 414
- insertion point 69
- instance
 - members 9, 48
 - methods 9, 49, 50
 - variable initialization 42
 - variables 9, 44
 - see also* object
- instance initializer block 404
- instance methods 6
- instance variables 6, 406
- instanceof operator 195, 320, 321
- instantiation 4
- int 30, 38
- integer arithmetic 165
- integer bitwise operators 189
- integer constant expressions 148
- integer data types 30
- integer division 167
- integer remainder operation 168
- integers 38
 - and* operator 189
 - byte 38
 - complement 189
 - data types 38
 - int 38
 - literals 30
 - long 38
 - or* operator 189
 - representation 154
 - short 38
 - types 38
 - xor* 189
- integral types 37, 38, 144
- interface constant antipattern 102
- interfaces 290
 - abstract 290
 - abstract methods 291
 - accessibility 118
 - body 290
 - constants 302
 - declarations 96
 - default methods 297
 - extending 294
 - header 290
 - implementing 291
 - initialization 409
 - marker 291
 - realization 296
 - references 296
 - static methods 300
 - subinterfaces 294

- superinterfaces 294
- UML 295
- variables 302
- internal traversal 452
- interned strings 358, 359
- interned values 351
- interpackage accessibility 335
- interpreter 17
- intraclass dependencies 336
- invocation stack
 - see* JVM stack
- invoker 224
- invoking garbage collection 393
- IOException 235
- is-a* relationship 266, 267, 334
- ISO Latin-1 32, 38
- ISO standard 486, 487, 488
- Iterable interface 366, 424
- iteration 215
- iteration statements 213
 - next iteration 223
 - termination 213, 222
- iterators 414, 424

J

- Java
 - Native Interface *see* JNI
- java 17
- Java bytecode 16
- Java Collections Framework 414
- Java compiler 17
- Java Development Kit (JDK) 21
- Java ecosystem 21
- Java EE (Enterprise Edition) 22
- Java ME (Micro Edition) 22
- Java Native Interface *see* JNI
- Java Platforms 22
- Java Runtime Environment (JRE) 22
- Java SE (Standard Edition) 22
- Java Virtual Machine *see* JVM
- java.time package 462
- java.time.format package 462
- java.util package 414
- java.util.function<T> package 444
- javac 17
- Javadoc comment 35
 - @param tag 49, 56
 - @return tag 225
 - @throws tag 253
- javadoc utility 36
- JDK 17, 508

- JNI 137
- just-in-time (JIT) 23
- JVM 17, 22, 384, 393
- JVM stack 230, 384

K

- key 69
- keywords 29
 - abstract 120, 136, 291, 442
 - boolean 39
 - break statement 221
 - byte 38
 - case 203
 - catch 240
 - char 38
 - class 48, 290
 - const 29
 - continue 223
 - default 204, 297
 - do 214
 - double 39
 - else 201
 - extends 264
 - final 80, 122, 133
 - finally 245
 - float 39
 - for 215, 217
 - if 200
 - implements 291
 - import 100
 - instanceof 195, 320, 321
 - int 38
 - interface 290
 - long 38
 - native 137
 - new *see* new operator
 - null 149, 183, 320
 - package 98
 - private 128
 - protected 126
 - public 124
 - reserved words 29
 - return 224
 - short 38
 - static 17, 101, 132, 300, 402
 - strictfp 166
 - super 54, 272, 276, 285, 299
 - switch 203
 - synchronized 136
 - this 50
 - throw 249

- throws 251
 - transient 138
 - try 240
 - unused words 29
 - void 17, 347
 - volatile 139
 - while 213, 214
- L**
- labeled break statement 222
 - labels 220, 222
 - break 222
 - case 203
 - default 204
 - expressions 205
 - labeled statement 220
 - switch statement 203
 - lambda body 439, 444, 445
 - lambda expressions 433, 438, 444
 - access class members 446
 - anonymous functions 439
 - arrow -> 439, 444
 - as values 439
 - blocks
 - scope 448
 - declared-type parameters 445
 - deferred execution 451
 - expression 445
 - expression statements 446
 - function 451
 - inferred-type parameters 445
 - lambda body 439, 444, 445
 - lambda parameters 445
 - non-void return 445
 - parameter list 439, 444
 - single expression 439, 445
 - statement block 439, 446
 - target type 450
 - target typing 451
 - type checking 450
 - variable capture 449
 - void return 445
 - lambda parameters 445
 - late binding
 - see* dynamic method lookup
 - least significant bit 155
 - left associativity 152
 - legal assignments 314
 - length method 361
 - letter pattern 495
 - lexical scope
 - see* blocks: scope
 - lexical tokens 28
 - lexicographical ordering 363, 425
 - LF *see* linefeed
 - libraries 403
 - lifetime 385
 - see* scope 44
 - line separator 19
 - line terminator 35
 - linear implementation inheritance 266
 - linefeed 33
 - LinkageError 237
 - LinkedList 417
 - List 414
 - lists
 - see* ArrayList
 - literals 30
 - boolean 32
 - character 32
 - default type 30, 31
 - double 31
 - escape sequences 33
 - false 32
 - float 31
 - floating-point 31
 - integer 30
 - null 30
 - predefined 29
 - prefix 30
 - quoting 32
 - scientific notation 31
 - string 34
 - suffix 30, 31
 - true 32
 - litmus test
 - design by inheritance 266
 - local 43
 - chaining of constructors 283, 406
 - variables 44, 117
 - local declarations 49, 50
 - local variables 53
 - blocks
 - scope 448
 - LocalDate class
 - see* temporal objects
 - LocalDateTime class
 - see* temporal objects
 - locale 364, 490, 492
 - localizing information 335
 - LocalTime class
 - see* temporal objects
 - locations

- see* class path
- logical AND
 - & 184
- logical complement
 - ! 184
- logical exclusive OR
 - ^ 184
- logical inclusive OR
 - | 184
- logical XOR
 - ^ 184
- long 30, 38
 - suffix 30
- loop body 213, 215
- loop condition 213, 215
- loops *see* iteration statements
- loose coupling 336
- loss of precision 144

M

- magnitude 144
- main method 17, 18, 85
 - arguments 86
 - modifiers 85
- manifest constant 134
- marker interfaces 291
- Math class 52
- MAX_VALUE constant 351
- member declarations 48, 290
- members 3, 114
 - access 50
 - accessibility 120, 123
 - default values 42
 - final 133
 - inheritance 264
 - instance 48
 - modified 264
 - modifiers 131
 - of objects 7
 - scope 114
 - short-hand 51
 - static 7, 48, 132
 - terminology 9
 - variables *see* fields
- memory management 384
- memory organization 384
- message
 - receiver 7
- method call 7, 49, 72
 - chaining 376, 378
 - fixed arity 84

- variable arity 84
- method chaining 471, 474, 479
- method declaration 48
- method header 136, 137
- method invocation conversions 148, 315
- method modifiers 49
- method overloading 52, 273
- method overriding 268, 273, 407
- method signature 49, 269
- method type 450
- methods 3
 - @Override 270
 - abstract 136, 291, 442
 - abstract method declarations 291
 - accessibility 49
 - ambiguous call 316
 - automatic variables *see* local variables
 - behavior 433
 - blocks 49
 - body 49, 117
 - call chaining 72
 - call *see* method call
 - calling variable arity method 82
 - chained 365
 - clone 343
 - concrete 134
 - declaration 49, 72
 - default 297
 - dynamic lookup 330
 - equals 183, 342
 - exceptions 49
 - final 134
 - finalize 343, 390
 - fixed arity 81
 - functional 442
 - getClass 343
 - header 49
 - implementation 136
 - invocation *see* method call
 - local declarations 49
 - local variables
 - main *see* main method
 - method invocation conversions 315
 - method type 450
 - modifiers 49
 - most specific 316, 422
 - name 72
 - native 137, 251
 - objects 50
 - overloaded resolution 316
 - overloading *see* method overloading
 - overriding *see* method overriding

- overriding vs. overloading 273
 - parameters 49
 - recursive 237
 - return 224
 - return value 49
 - signature 49, 52, 273
 - static 132, 300
 - synchronized 136
 - termination 224
 - throws clause 251
 - toString 343
 - variable arity 81
 - MIN_VALUE constant 351
 - minimizing overhead 386
 - mobile applications 22
 - modifiers
 - abstract 120, 136, 291
 - accessibility 118, 123
 - classes 120
 - default 297
 - final 133
 - members 131
 - native 137
 - static 132, 300
 - strictfp 166
 - synchronized 136
 - transient 138
 - volatile 139
 - Month enum type 465
 - most specific method 316, 422
 - multicore 441
 - multicore architectures 24
 - multidimensional arrays 63, 65
 - multiple assignments 159
 - multiple catch clauses 239
 - multiple implementation inheritance 290
 - multiple inheritance 298
 - multiple interface inheritance 290
 - multiple-line comment 35
 - multiplication operator
 - * 167
 - multiplicative operators 167
 - multithreaded 24
 - mutable character sequences 374
 - mutually comparable 68, 69
 - mutually exclusive
 - actions 202
 - MVC 335
- N**
- name 28
 - named constants 134
 - namespaces 53
 - NaN 166, 349
 - narrower range 144
 - narrowing conversions
 - primitive 144
 - reference 145
 - narrowing reference conversions 320
 - native libraries 403
 - native methods 137, 251
 - header 137
 - natural ordering 68, 69, 425
 - negative zero 165
 - nested lists 417
 - nested loops 66
 - new operator 5, 53, 59, 195, 406
 - newline *see* linefeed
 - NL *see* newline
 - no-argument constructor 53, 54, 283, 287
 - non-associativity 151
 - non-static code 48
 - see* non-static context 48
 - non-static context 48
 - non-static field 9
 - non-static field initializers 48
 - non-static initializer block 48
 - non-varargs call
 - see* fixed arity call
 - non-void return 445
 - non-zero argument constructor 55, 287, 288
 - normal class 121
 - normal execution 232
 - notifying threads 344
 - null reference 30
 - casting 320
 - null reference literal
 - casting 149
 - equality comparison 183
 - nulling references 387
 - NullPointerException 236
 - Number class 351
 - number systems
 - base 30
 - converting 157
 - decimal 30
 - hexadecimal 30
 - octal 30
 - radix 30
 - NumberFormatException 236, 347, 348
 - numeric promotions 149
 - assignment 160

- binary 150
- unary 149
- numeric wrapper classes 351
- numerical literals
 - using underscore 32

O

- object 4
- Object class 266, 342
- object hierarchy 267
- object references 4, 40
- object state 6, 53, 77, 406
- object-oriented design 334
 - cohesion 335
- object-oriented paradigm 22
- object-oriented programming 2
- objects 13
 - aggregate 12
 - alive 385
 - arrays 58
 - callee 72
 - caller 72
 - Class class 343
 - cleaning up 386
 - cloning 343
 - communication 72
 - comparing 342
 - composite 385
 - constituent 12, 385
 - constructing 406
 - contract 335
 - decoupling 330
 - destroying 390
 - eligible 387
 - equality 183, 342
 - exchanging information 72
 - finalization 385
 - garbage collection 384
 - identity 5
 - immutable 346
 - implementation 335
 - initial state 406
 - initialization 5, 53
 - initializer block 404
 - internals 335
 - lifetime 385
 - members 7
 - methods 50
 - Object class 342
 - persistence 138
 - reachable 384, 385
 - resurrection 385
 - services 335
 - state 133
 - see object state
 - value equality 183
- OCAJP8 507
 - exam objectives 515
 - exam question assumptions 511
- OCPJP8 507
- one-dimensional arrays 59
- operands 148
 - evaluation order 152
- operations 2
- operators 150
 - 163, 169
 - 176
 - ^ 184, 189
 - ^= 185, 192
 - ! 184
 - != 181, 182
 - ? : 194
 - . 7, 97
 - [] 61, 195
 - * 163, 167
 - *= 172
 - / 163, 167
 - /= 172
 - & 184, 189
 - && 186
 - &= 185, 192
 - % 163, 167, 168
 - %= 172
 - + 163, 169, 174
 - ++ 176
 - += 172
 - < 180
 - <= 180
 - 172
 - = 158
 - == 181, 182
 - > 195
 - > 180
 - >= 180
 - | 184, 189
 - |= 185, 192
 - || 186
 - ~ 189
 - arithmetic compound assignment 172
 - assignment 151, 158
 - associativity 150
 - binary 151
 - bitwise 189

- boolean 180, 181, 184
- cast 151
- comparisons 180
- compound assignment 185, 192
- conditional 186, 194
- decrement 176
- dot 7
- equality 181
- execution order 152
- floating-point 165
- floating-point division 167
- floating-point remainder 169
- increment 176
- instanceof 195, 320, 321
- integer 189
- integer arithmetic 165
- integer division 167
- integer remainder 168
- logical 184
- multiplicative 167
- new* *see* *new operator*
- overflow 165
- overloaded 164, 167
- postfix 151
- precedence 150
- relational 180
- short-circuited 186
- string concatenation 174
- ternary 151
- unary 150, 151, 167
- unary - 167
- unary + 167
- optimizations 24
- or* operator 189
- Oracle University 509
- ordinal value 90, 209
- OutOfMemoryException 395
- output 18
- overflow 155, 165
- overloaded 164
- overloaded method resolution 316
- overloading
 - constructors 56
 - method resolution 316
 - methods 52, 273
- overloading vs. overriding 273
- overriding 253
 - equals* 183
 - finalizers 390
 - methods 268, 273
 - toString* 175
- overriding methods

- covariant return 273
- overriding vs. overloading 273
- ownership 12

P

- package accessibility 118, 124
- package directory 106
- package statement 96, 98
- packages 97
 - accessibility *see* package accessibility
 - declaration 96
 - definition 98
 - destination directory 106
 - hierarchy 97
 - java.lang* 342
 - members 97
 - naming scheme 98
 - package directory 106
 - running code from 106
 - short-hand 100
 - statement *see* package statement
 - subpackages 97
 - unnamed 98
 - using 99
- palindromes 382, 434
- parallel code 441
- parameter
 - variable arity 81
- parameter list 439, 444
- parameter list *see* formal parameters
- parameter passing
 - by value 72
 - variable arity 81
- parameters 49
 - actual 72
 - array elements 78
 - final 80
 - fixed arity 81
 - formal *see* formal parameters
 - implicit 50
 - main method 86
 - passing 72
 - primitives 73
 - program 86
 - references 75
 - this 50
 - variable arity 81
- parent class 264
- parentheses 150
- parseType* method 352
- parsing 462, 486

- parsing numeric values 352
 - partial implementation 293
 - pass by value 72
 - passing
 - parameters 72
 - references 75
 - variable arity parameter 81
 - paths
 - see* class path
 - path-separator character 110
 - pattern letters 486, 495, 496
 - Pearson VUE 509
 - performance 24
 - period 462, 476
 - creating 476
 - date-based 476
 - equality 478
 - get methods 478
 - immutable 476
 - normalization 479
 - parsing 477
 - period-based loop 481
 - plus/minus methods 479
 - querying 478
 - temporal arithmetic 479
 - textual representation 477
 - thread-safe 476
 - with methods 479
 - Period class
 - see* period
 - persistent objects 138
 - polymorphism 311, 329, 334
 - portability 23
 - positional order 414
 - positive zero 165
 - postfix operators 151
 - precedence rules 151
 - precision 160
 - predefined identifiers 29
 - predefined literals 29
 - predicate 436
 - Predicate<T> 440, 451
 - prefix 30
 - 0 30
 - 0x 30
 - primitive data types
 - see* primitive types
 - primitive types 13, 144
 - autoboxing 348
 - unboxing 350
 - see also* primitive values
 - primitive values
 - assignment 159
 - equality 181
 - passing 73
 - printing values 18
 - private 11
 - private members 128
 - process of elimination 510
 - program
 - application 16
 - arguments 86
 - command line 86
 - compiling 17
 - formatting 35
 - running 17
 - program arguments 86
 - program output 18
 - programming to an interface 417
 - proleptic year 464
 - promotion 149
 - properties 2
 - see also* class members
 - protected 11
 - protected members 126
 - public 17
 - public members 124
 - punctuators 29
- ## Q
- quotation mark 33, 34
 - quotient 168
- ## R
- radix
 - prefix 30
 - see* base 349
 - ragged arrays 65
 - range
 - character values 38
 - floating-point values 39
 - integer values 38
 - range of date-based values 464
 - range of time-based values 464
 - ranking criteria 414
 - realization 296
 - reclaiming memory 384
 - reducing complexity 335
 - reference types 41, 267
 - classes 48
 - enum types 87
 - reference values 4

- reference variables 40
- references 4, 9, 40, 41, 72
 - abstract types 121
 - aliases 75, 183
 - array 59, 62, 311
 - assignment 159
 - casting 149, 320
 - dangling 384
 - declared type 268
 - downcasting 145
 - dynamic type 268
 - equality 182
 - field 385
 - interface type 296
 - local 384
 - narrowing conversions 145
 - null *see* null reference
 - passing 75
 - reachable 384, 385
 - super 276
 - this 50
 - upcasting 145
 - widening conversions 145
- relational operators 180
- relative adjusters 474
- reliability 24
- remainder 168
- remainder operator
 - % 168
- remove whitespace 369
- replacing characters 367
- reserved identifiers 29
- reserved keywords 29
 - const 29
 - goto 220
- reserved literals
 - false 32
 - null *see* null reference
 - true 32
- resources 387
- resurrecting objects 385, 391
- return statement 224
 - @return tag 225
- return type
 - covariant
- return value 7
- reuse of code 264, 334
- right associativity 152
- rightmost bit 155
- ripple effect 334
- robustness 24, 254
- role relationship 334

- root
 - see* inheritance hierarchy
- running a Java application 17
- runtime
 - bounds checking 61
- runtime checks 148, 418
- Runtime class 342, 393
- runtime class 343
- runtime environment 384
- runtime stack
 - see* JVM stack
- RuntimeException 236

S

- scientific notation 31
- scope 114
 - block 117
 - catch clause 244
 - class 114
 - disjoint 118
- searching
 - arrays 69
- searching in string 367
- secure 24
- SecurityManager class 342
- selection statements 200
- semantic definition 28
- semicolon 50
- separators 29, 151
- serialization 138
- services 335
- shadowing 446
- shallow copying 343
- short 30, 38
- short-circuit 186
 - evaluation 187
- signature 52, 273
- simple
 - assignment operator 158
 - if 200
 - statement 50
- simple type name 97
- simplicity 23
- single expression 439
- single implementation inheritance 266, 290, 296
- single quote (') 32, 33
- single static import 101
- single-line comment 3, 35
- skeletal source file 96
- sorting arrays 68

- source
 - file 15, 98
 - file name 96
 - file structure 96
- spaces 35
- special character values 33
- specialization 10
- specialized classes 266
- stack 3
- stack frame
 - see* method execution
- stack trace 232, 235
 - see* method execution
- StackOverflowError 237
- standard error stream 235
- standard out 18
- starts with characters 368
- state *see* object state
- statement block 439, 446
- statements 50
 - break 221
 - compound 50
 - conditional 200
 - continue 223
 - control flow 50, 200
 - control transfer 219
 - declaration 171, 177, 187
 - declarations 50
 - do-while 214
 - empty 50
 - expression 50, 177
 - for(;;) 215
 - for(:) 217
 - if 200
 - if-else 201
 - iteration 213
 - labeled 220
 - return 224
 - selection 200
 - simple 50
 - simple if 200
 - switch 203
 - throw 249
 - transfer 219
 - try 240
 - while 213
- static
 - members *see* static members
 - methods 7, 10, 49
 - variable initialization 42
 - variables *see* static variables
- static 101, 132, 300
- static code
 - see* static context 48
- static context 48
- static field 10
- static field initializers 48
- static import 101
 - conflicts 104
 - on demand 101
 - shadow static members 103
 - single static import 101
- static initializer block 48, 137, 402
- static keyword 402
- static members 7, 9, 10, 48
- static type
 - see* declared type
- static variables 7, 10, 44
- storing objects 138
- strictfp 166
- string builders 176
 - appending 376
 - capacity 374, 378
 - constructing 374
 - deleting 376
 - differences with strings 376
 - individual characters 375
 - inserting 376
 - joining 366
 - length 375
 - thread-safety 374
- String class
 - see* strings
- string conversion 146, 175
- string conversions 370
- string literal pool 358
 - interned 358
- string literals 357
 - case labels 208
 - hash value 208
 - interned 358
- StringBuffer class 374
 - see* string builders
 - thread-safe 374
- StringBuilder class 374, 434
 - see* string builders
- strings
 - appending 376
 - buffers 374
 - builders 374
 - capacity 378
 - changing case 364
 - compareTo 363
 - comparing 363

- concatenation 174, 364
- concatenation operator + 176
- constructing 374
- contains 368
- conversions 370
- convert to character array 361
- copying characters 361
- creating 357
- deleting 376
- differences with string builders 376
- empty 358
- ends with 368
- equals 363
- extracting substrings 369
- finding index 367
- formatted 370
- ignoring case in comparison 363
- immutable 357
- individual characters 361, 375
- initializing 357
- inserting 376
- interned 358
- joining 365, 366
- length 361, 375
- lexicographical ordering 363
- literals 34, 357
- mutable 374
- read character at index 361
- replacing 367
- searching 367
- starts with 368
- string literal pool 358
- substrings 369
- trimming 369
- strongly typed language 148
- subclass 10, 11, 264
- subinterface 294
- subpackages 97
- subsequence 361
- substring searching 367
- substrings 367, 369
- subtype covariance 309, 310
- subtype relationship 418
- subtypes 293
- subtype–supertype relationship 145
- suffix
 - D 31
 - F 31
 - L 30
- super 299
 - construct 285
 - keyword 272, 276

- reference 276
- superclass 10, 11, 264
- superclass constructor 54
- superclass–subclass relationship 266
- superinterfaces 294
- supertypes 293
- supertype–subtype relationship 267
- supplementary characters 357
- suppressed exceptions 235
- switch statement 203
 - break 205, 206
 - default clause 204
 - enum types 209
 - using strings 208
- synchronized
 - methods 136
- syntactically legal 28
- System
 - out 18
- System class 342
- system clock 466

T

- TAB *see* horizontal tab
- tabs 35
- tabulators 35
- tags 36
- target type 450
- target typing 451
- telephone directory order 363
- temporal arithmetic 474, 479
- temporal objects
 - before/after methods 469
 - combining date and time 466
 - common method prefix 463
 - comparing 470
 - creating with factory methods 464
 - date 462
 - date units 474
 - date-based values 462
 - date-time 462
 - formatting 486
 - get methods 468
 - immutable 462
 - method naming convention 463
 - parsing 486
 - plus/minus methods 474
 - querying 468
 - range of date-based values 464
 - range of time-based values 464
 - temporal arithmetic 474, 479, 480

- temporal values 464
 - thread-safe 462
 - time 462
 - time units 474
 - time-based values 462
 - with methods 470
 - temporal values 464
 - TemporalAmount interface 479
 - terminating loops 221
 - ternary conditional expressions
 - see also* conditional expressions 194
 - ternary conditional operator 151, 194
 - textual representation 343
 - this
 - reference 50
 - this() constructor call 282, 406
 - ThreadDeath 237
 - threads 24, 342, 384
 - death 232
 - exception propagation 232
 - JVM stack 385
 - live 384
 - notifying 344
 - synchronization 136
 - waiting 344
 - thread-safe 357, 374, 415, 462
 - throw statement 249
 - Throwable 233, 342
 - throw-and-catch paradigm 230
 - throwing exceptions 230
 - throws clause 251
 - time
 - see* temporal objects
 - time units 474
 - time-based values 462
 - tokens 28
 - toString method 343, 349
 - transfer statements 219
 - transient variables 138
 - transitive relation 267
 - trim method 369
 - true literal 32
 - truth-values 32, 39
 - try block 240
 - try-catch-finally construct 238
 - two's complement 154
 - type
 - declared 274
 - dynamic 274
 - type cast 148
 - type cast expression 320
 - type checking 450
 - type declarations 96
 - type hierarchy 145, 267
 - type import
 - see* import
 - type parameter 290, 414, 441
 - types
 - boolean 37, 39
 - byte 30, 38
 - casting 148
 - char 38
 - classes *see* classes
 - comparing 321
 - compatibility 148
 - double 39
 - exceptions 233
 - float 39
 - floating-point 37, 38
 - int 30, 38
 - integers 38
 - integral types 37
 - interface 290
 - long 30, 38
 - parsing 352
 - short 30, 38
 - wrappers 346
 - see also* classes
 - type-safe 315
 - type-safety 416, 417, 418
 - typeValue method 350, 352
- ## U
- UML 2
 - accessibility 124
 - aggregation 12
 - associations 12
 - classes 3
 - composition 12
 - inheritance 10
 - see also* diagrams
 - unary arithmetic operators 167
 - unary numeric promotion 149
 - unary operators 150, 151
 - unboxing 350
 - do-while statement 214
 - for(;;) statement 215
 - for(:) statement 218
 - if statement 200
 - if-else statement 202
 - switch statement 204
 - while statement 213
 - unboxing conversions 145, 146

- uncaught exceptions 232
- unchangeable variables 134
- unchecked conversion warning 416
- unchecked conversions 146
- unchecked exceptions 237
- unchecked warnings 145
- underflow 155, 165
- uni-catch clause 239
- Unicode 32, 38, 354, 357, 363, 434
- Unified Modeling Language *see* UML
- unreachable code 244
- unsafe casts 321
- unsigned integer 353
- UnsupportedTemporalTypeException 480, 490
- unused keywords 29
- upcasting 145
- update expression 215
- using arrays 61
- using packages 99
- using variables 41
- UTF-16 357
 - supplementary characters 357

V

- valueOf method 348, 349, 369
- values 155
 - constants 30
 - overflow 155
 - underflow 155
 - wrap-around 155
 - see also* variables
- varargs 81
- variable arity call 84
- variable arity method 81
- variable arity parameter 81
- variable capture 449
- variable declarations 41, 117
- variable initialization 8, 43
- variables 4, 41
 - blank final 80, 134
 - constant values 133
 - default values 42
 - effectively final 448
 - final 133
 - identifiers 40
 - in interfaces 302
 - initialization *see* variable initialization
 - lifetime 44
 - local 117
 - parameters 49, 72
 - reference variable 41

- references 41
 - static 7
 - storing 138
 - transient 138
 - volatile 139
- virtual method invocation
 - see* dynamic method lookup
- VirtualMachineError 237
- void 17, 224, 347
- void return 445
- Void wrapper class 346
- volatile variables 139
- voucher 509

W

- waiting threads 344
- while statement 213
- whitespace 35, 369
- whole-part relationship 267
- widening conversions
 - primitive 144
 - references 145
- widening reference conversions 267, 320
- wider range 144
- withers 470
- wrapper classes 38, 342, 343, 346, 347
 - interned values 351
- wrapper type 146, 164, 177

X

- xor 189

Z

- zero
 - negative 165
 - positive 165
- zero-based index 414