

INDEX

A

ABS, 428
abstract class, 112, 252
abstract data types, 104
abstract entities, 315
accessor method, 159, 238
ACOS, 428
Add, 256, 259, 260, 262
aggregate objects, 34
aggregation, 126
A-Kind-Of, 126
ALGOL, 13
ALGOL 60, 17
Allocated-Occurrences, 428
American National Standards Institute
 (ANSI), 5
analysis, 310, 313, 314
Annuity, 428
ANSI/ISO, 349
Aristotle, 109
Array, 253, 254
ArrayedCollection, 253
As, 213, 214
AsArray, 257, 259, 263
asBag, 257, 259, 280, 281, 299
As Universal, 215

ASIN, 428
association, 126, 128, 252, 293, 294, 296,
 315
associationAt, 301, 302
AT&T Bell Labs, 15
at, 283, 300
ATAN, 428
atPut, 256, 259, 278, 297
automata theory, 149
automatic garbage collection, 153, 350

B

Bag, 252–254
Base, 37, 125, 154, 253, 353, 361
behaviors, 33
Binary, 441
Binding, 204
Booch, G., 77, 311
By Content, 138
By Reference, 138

C

C, 13
C++, 5, 13, 15, 143, 153, 155
Call, 137
Call ... by Content, 70, 231

498 INDEX

- Call by Reference, 70
- Call Of Self, 217
- Callback, 252, 271, 273
- Cancel, 105
- candidate classes, 310, 314, 315, 328, 329
- capacity, 257, 259, 265, 266, 268, 282, 300,
- categories of collection classes, 256
- CBL-Base, 352
- CBL-New, 353
- Char, 428
- Char-National, 428
- CharacterArray, 252–254, 276, 279, 291
- Class, 350
- Class clause, 438
- class data, 31, 95
- class definition, 45, 47, 319, 329
- class diagrams, 314, 315, 319, 323, 312, 320, 324
- Class-Id, 31, 32, 44, 49, 71, 100
- class interface, 66
- class libraries, 36
- Class Library, 125
- class method data, 95
- class methods, 31, 94, 113
- class-name, 49, 72
- Class-Object, 351
- class program, 44
- class relationship diagrams, 126
- class visibility, 319
- classes, 29, 75, 350
- client, 7, 9, 26, 136
- Coad, P., 312
- Coad/Yourdon, 311
- COBOL, 4, 104
- COBOL 68, 13
- COBOL 74, 13
- COBOL 85, 13, 350, 433
- COBOL 89 addendum, 433
- COBOL 97, 5, 13, 339
- COBOL for MVS, 356
- COBOL Reserved Words, 419
- COBOL Set for AIX, 356
- CobolPicX, 293, 294, 296
- CODASYL (Committee on Data Systems Language), 3
- cohesion, 24
- collaborate, 24
- collaborating objects, 331
- Collection, 101, 152, 252, 253, 333
- collection classes, 166, 252, 304
- Communication Section, 57
- Comp, 441
- Comp-3, 441
- Comp-5, 254
- Comp-x, 254
- composite, 34
- Composition, 126
- concrete class, 112
- Configuration Section, 50, 61, 71, 73, 91
- confluence, 87
- Conformance, 199, 201, 202
- conformance checking, 155, 168
- constructor operation, 156
- container class, 252
- Continue, 440
- continuity, 310
- contract, 22
- conversion methods, 264
- Copy, 257, 259, 264
- copyEmpty, 257, 259, 281

CORBA (Common Object Request Broker
 Architecture), 11, 346, 356
 COS, 428
 coupling, 24
 CRC (Class, Responsibility, Collaborator),
 76, 106
 CRC cards, 312, 317, 318, 320, 324
 CRC sessions, 183
 Creating Bag Instances: ofReferences, 258
 creating instances, 259
 Creating Objects, 154
 Current-Date, 428, 439

D

Dahl, J., 17
 data dictionary, 320, 324
 Data Division, 25, 54, 72, 74, 75
 data-flow diagrams (DFDs), 146
 Data is Protected, 359
 data items, 116
 data name conflicts, 70
 Data Records clause, 58
 data values, 116
 DataBase Manager Class, 329
 Date is Private, 359
 Date-Of-Integer, 428, 439, 440
 Day-Of-Integer, 428, 439, 440
 Declaratives, 73, 74
 deepCopy, 299
 deepFinalize, 257, 301–303
 deferred classes, 112
 Design, 310
 design patterns, 311
 Destroying Objects, 175
 Dictionary, 252, 253, 256, 292, 296–298,
 302

Dijkstra, E., 109
 Display-Of, 428
 Distributed System Object Model (DSOM),
 346
 do, 257, 259, 269, 271
 Do-Until, 436
 Do-While, 436
 DSOM (Distributed SOM), 356
 DynamicArrayedCollection, 253
 dynamic binding, 204, 350
 dynamic objects, 153

E

E, 428
 EDVAC, 12
 Eiffel, 143
 elements, 254
 emergent behavior, 87
 encapsulate, 22
 encapsulation, 350
 End Class, 31, 75
 End Factory, 74
 End Method, 25, 74
 End Object, 25, 75, 90
 ENIAC, 12
 Environment Division, 47, 50, 54, 55, 71–73,
 75
 equal, 257, 259, 285
 equalByLengthValue, 259, 285, 288
 equalsIgnoreCase, 257, 259, 285–287
 Evaluate, 278
 evolution of programming languages, 103
 Exception-File, 428
 Exception-Location, 428
 Exception-Statement, 429
 Exception-Status, 429

500 INDEX

Exit Method, 65, 74
 EXP, 429
 EXP10, 429
 External, 435
 external system view, 316

F

Factorial, 429, 438
 Factory, 72
 factory data, 31
 Factory Data Division, 57
 Factory Definition, 45, 54
 Factory Environment Division, 55
 factory methods, 31, 113
 Factory Object, 31, 54
 Factory Of Self, 217
 Factory Procedure Division, 58
 FD description entries, 72, 74
 File Section, 57, 72, 74
 File-Control, 72, 73, 91
 Filler, 441
 Finalize, 176, 257, 259, 302, 361
 FORTRAN, 13, 104
 Fraction-Part, 429
 frameworks, 311
 Function, 438

G

Gamma, E., 311
 Gang of Four, 311
 generalization/specialization, 126
 GetClass, 219
 getValue, 257, 259, 283, 284
 Global, 435
 grow, 256, 259, 279, 298
 growTo, 256, 259, 279, 280

H

Has-A, 126, 127
 hierarchies, 109
 High-Level design, 313, 317
 Hitachi, 349, 350
 Hitachi's Object-Oriented COBOL, 340
 HP UX, 350
 hybrid object-oriented language, 15

I

IBM, 11, 349, 356
 IBM 702, 104
 IBM 704, 104
 IBM's SOM (System Object Model), 356
 IBM's VisualAge for COBOL, 340
 ICON, 13
 Identification Division, 32, 48, 71–74
 IdentifyDictionary, 253, 256
 IdentifySet, 253, 256
 impedance mismatch, 37
 In-line invocation, 221, 222, 225
 includes, 259, 274
 includesKey, 301, 303
 inheritance, 32, 110, 114, 115, 212, 350
 Inherits, 34, 110, 122
 Inherits Base, 172
 Inherits with Data, 359
 Initial, 105
 Initialize, 441
 I-O-Control, 72, 73, 91
 Input-Output Section, 56, 61, 72, 73, 91
 instance, 29
 instruction formats, 491
 Integer, 429
 Integer-Of-Date, 429, 439, 440
 Integer-Of-Day, 429, 440

Integer-Part, 429
 interface, 354
 International Standards Organization (ISO), 5
 Intrinsic, 293
 intrinsic classes, 252
 intrinsic data, 252, 254
 Intrinsic Function, 427
 Invariant, 354
 invariant methods, 354
 Invoke, 5, 8, 27, 119, 137, 154
 Is-a, 126, 127
 is at, 301
 isEmpty, 257
 Iterating in Phase I, 317
 Iteration, 312

J

Jacobson, I., 312, 316
 Justified, 441

K

Kay, A., 149
 key, 294
 key/component pair, 294
 key value, 297, 298

L

Label Records clause, 58
 legacy code, 223, 337, 341
 Length, 429
 Length-An, 429
 levels of abstraction, 82
 Linkage Section, 57, 64, 74, 145
 Local-Storage Section, 360
 Log, 429
 Log10, 429

low-level design, 313, 319
 Lower-Case, 429

M

Max, 429
 Mean, 429
 Median, 429
 memory leakage, 153
 messages, 7, 26, 136
 Message-Trace Diagram, 321, 324
 method data, 59
 Method Data Division, 63
 Method Definition, 45, 60, 75, 89, 97
 Method Environment Division, 61
 Method-Id, 25, 61, 73, 98, 121
 Method Procedure Division, 65
 method prototype, 354
 method resolution, 119
 methods, 7, 23, 25, 59, 94, 350
 Methods—invariant, 350
 Methods—prototype, 350
 Micro Focus, 349, 357
 Micro Focus class library, 252
 Micro Focus' Visual Object COBOL, 340
 Micro Focus Workbench, 357
 Midrange, 430
 Min, 430
 Mod, 430
 Modeling, 314
 modern languages, 235
 multiple inheritance, 122, 350
 MVS, 10

N

National-Of, 430
 nested, 433

502 INDEX

Nested Properties, 233
Nesting In-Line Invocations, 227
New, 31, 154, 361
NewClass, 294, 295
New COBOL Reserved Words, 425
No Get, 230, 232
No Set, 230, 231
Norwegian Computing Centre, 17
Not On Exception, 140
Null, 216
Null object, 176
numeric key, 297
Numval, 430
Numval-C, 430
Nygaard, K., 17

O

object, 6, 22, 25, 29, 74, 350
Object COBOL, 357
Object-Computer, 50, 72, 73
object creation, 186
object data, 95
Object Data Division, 25, 92
Object Definition, 45, 89, 90
Object Environment Division, 91
object handle, 35, 98, 206
object instance, 29
object interface, 6
Object Life Span, 152
Object Management Group (OMG), 11
object-message diagram, 146, 310, 312, 320, 324
object method data, 95
object methods, 94, 158
object notation, 102
object-oriented analysis, 310

Object-Oriented COBOL (OOCOBOL), 5, 10, 153, 155, 169, 349
object-oriented design, 146, 310
object-oriented programming, 22
Object paragraph, 90, 159
Object Reference, 98, 217
Object-Relation Conditions, 198
Object sequences chart, 310
Object-Storage Section, 360
Object technology, 5
object visibility, 323
Object Procedure Division, 93
Objects in Tables, 163
occurencesOf, 257, 259, 303, 304
Occurs clause, 163
ofAssociations, 295
ofReferences, 257
ofValues, 257
OMG (Object Management Group), 356
OMT (Object Modeling Technique), 77
On Exception, 140
Only, 168, 204
Opler, A., 191
Ord, 430
OrderedCollection, 253, 254
Ord-Max, 430
Ord-Min, 430
OS/2, 10
Override, 115, 121
Overriding Inheritance, 121

P

Packed-Decimal, 441
Parameterized classes, 350, 362

parameters, 138
 Pascal, 13
 patterns, 311
 Perform, 435
 Persistence, 105
 Persistent objects, 153, 177
 Phase I, 314, 316, 325
 Phase II, 317, 328
 Phase III, 319
 Pi, 430
 PL/I, 13
 Policy classes, 332
 Polymorphism, 27, 28, 200
 predefined classes, 254
 Present-Value, 430
 Problem Description, 314
 problem domain, 314
 problem statement, 324
 Procedure Division, 54, 55, 73, 74, 145
 Property clause, 229, 230, 236

R

Random, 430
 Range, 430
 rapid application development (RAD), 5
 recursive, 438
 reference modification, 437
 Rem, 430
 Remove, 257, 259, 273, 301
 removeAll, 259
 removeAssociation, 301, 302
 Remove-Name, 179
 Report Section, 57, 73, 74
 Repository, 50–52, 72, 110, 260, 296
 Repository paragraph, 157
 reserved words, 49

responsibility-driven design, 20, 21, 106,
 149, 183
 Returning, 139
 Reuses, 361
 Reverse, 430
 role name, 128
 Rolling Lifecycle, 310, 324
 Rolling Lifecycle Perspective (RLP), 313
 Rolling Lifecycle phases, 314
 rules of inheritance, 119
 Rumbaugh, J., 77, 311

S

scenarios, 310, 316, 324, 326, 333
 scope, 70
 scope of variables, 192
 Select, 257
 Self, 141, 142, 172, 174, 175
 SequencedCollection, 253
 server, 26, 136
 service request, 6
 Set, 176, 210, 275
 Sign, 430
 simple aggregate data types, 104
 Simple class, 180, 239, 244, 350
 Simula, 11, 14, 17, 143
 Simula 67, 151
 SIN, 430
 single inheritance, 350
 size, 257, 259, 265, 268, 282, 300
 Smalltalk, 5, 14, 81, 143, 149, 153, 155
 SNOBOL, 13
 software architectures, 311
 SOMClass, 357
 SOMFree, 357
 SOMNew, 357

504 INDEX

SOMObject, 357
 SortedCollection, 253, 254
 Source-Computer, 50, 71, 73
 source program, 46
 source unit, 46
 Special-Names, 50, 51, 72, 73
 spiral model, 20
 SQRT, 431
 Standard-Deviation, 431
 Standish, 104
 state, 147
 state-transition diagrams, 147
 static binding, 204
 static objects, 180, 181, 239, 245
 stereotype, 75
 Strachey, 104
 strong typing, 5
 Stroustrup, B., 15
 structured analysis, 146
 subclass, 33, 110, 114
 subtypes, 200
 Sum, 438
 Super, 174
 superclass, 33, 110
 superclass-defined data, 118
 system identifier, 30
 System-Object, 177
 System Object Model (SOM), 346

T

Tables of Objects, 101
 TAN, 431
 templates, 362
 Test After, 435
 Test Before, 435
 Tool Class Library, 351

Turing, A., 149
 type checking, 155
 typed objects, 155, 168, 350
 typing, 204

U

Unified Method, 77, 102, 146, 186, 312
 UNIVAC, 12
 Universal, 214
 Unix, 10, 350
 untyped objects, 166, 168, 350
 Upper-Case, 431
 Usage clause, 99
 Usage phrase, 155
 usage scenarios, 314
 use-case scenarios, 316
 user-defined key, 297
 user-defined words, 49
 User Interface Manager, 330
 Using, 139

V

Value Of clause, 58
 ValueSet, 253, 256
 Variance, 431
 VisualAge for COBOL, 356
 von Neumann, J., 12, 191

W

waterfall model, 20
 When-Compiled, 431
 whole/part, 126, 127
 Windows 95, 10, 350
 Windows NT, 10, 350
 Wirfs-Brock, R., 21, 312
 withLengthValue, 277

Cambridge University Press
978-0-132-61140-4 - Object-Oriented COBOL
Edmund C. Arranga and Frank P. Coyle
Index
[More information](#)

Working Draft 1.1, 349
Working-Storage Section, 57, 72, 74
Wrapping Programs, 342
Wrapping Strategies, 343

X
Xerox PARC (Palo Alto Research Center), 14

'\$' symbol, 78