



Symbols

- # Not equal to 87
- \$ String selector 68, 333
- & And 88
- * Export mark 36
- * Multiplication 51, 54, 58
- + Addition 51, 54, 58
- + Concatenation 66
- + UML public 540, 563
- Read-only export mark 82
- Subtraction 51, 54, 58
- / Division 51, 58
- := Assignment 50
- := GCL assignment 69
- := Import abbreviation 544
- < Less than 87
- <= Less than or equal to 87
- = Equal to 87
- > Greater than 87
- >= Greater than or equal to 87
- ^ Pointer dereference operator 468
- | Case separator 106
- ~ Not 89
- UML private 540, 563

Numerics

- 0X String sentinel 65, 333–335
- 100 Invariant violation 157
- 20 Precondition violation 120

A

- ABS 58
- ABSTRACT
 - method attribute 553
 - objects and methods 578
 - record attribute 553
- Abstract data structure 113, 129
 - list 130
 - stack 119
- Abstract data type 113, 129
 - list 138
 - stack 124

- Abstraction 6, 30, 113, 533–552
 - and nesting 151
 - behavior 551
 - class 538
 - computation 536
 - data 313, 533
 - procedure 537
 - program 313
 - statement 536
 - structure 535
 - type 534
- Actual parameter 37, 51
- ADS. See Abstract data structure
- ADT. See Abstract data type
- Algol 2
- Alphabet 15
 - closure of 16
 - nonterminal 17
 - terminal 17
- Alternation, EBNF operation 25
- Analysis
 - versus design 321
- ANYREC 476
- API. See Application programming interface
- Application programming interface 4
- ARRAY 65
 - one-dimensional 313–336
 - open 319
 - See also specific type, e.g. CHAR, ARRAY OF
 - two-dimensional 381–392
- ASSERT 144, 156, 157–163
 - purpose of 160
- Assertion 156
 - See also ASSERT
- Assignment
 - class 474, 554
 - formal definition of 146
 - in GCL 69
 - integer to real 53
 - multiple, in GCL 70

- pointer 465, 470
- record 490
- rule for extensible records 575
- rule for parameter 557
- statement 50
- symbol 50
- Associative operator 52
- Attribute. See Unified Modeling Language
- Automatic garbage collection 473
- Average
 - computing from window 211

B

- Backus, John 2
- Binary search
 - iterative 357–360
 - recursive 433–435
- Binary tree 515–528
 - abstract 515
 - depth of node 516
 - height 516
 - inorder traversal 518
 - insert operation 519
 - internal node 519
 - leaf 515
 - left child 515
 - postorder traversal 518
 - preorder traversal 517
 - right child 515
 - root 515
 - search tree 516, 581–592
- Binomial coefficient 409
- Bisection algorithm 214–217, 277–279
- BlackBox framework 1–6
- Bohm, Corrado 222
- BOOLEAN 87
 - expression 87
 - variable 88, 96
- Bulletproof 144, 261
- Button

- command control 79
 - commander 40
 - dialog box 76
 - radio 103
- C**
- Calling mode 126
 - by constant reference 126, 326–328
 - by reference 126, 263–265
 - by result 66, 101, 120, 126, 265–267
 - by value 126, 252–256, 261–262
 - default, See by value
 - for function procedure 287
 - summary of by value, by constant reference, by result, and by reference 327
 - summary of by value, by result, and by reference 267
 - Caption control 79
 - CASE statement 106
 - CHAR 62
 - ARRAY OF 65, 333
 - number line 63
 - ordinal value 63
 - output of array to Log 66
 - output to Log 63
 - Check box 97
 - CHR 63
 - Class 175
 - assignment rule 474, 554
 - assignment rule for extensible records 575
 - assignment rule for parameters 557
 - Class composition 175, 191, 564–565
 - symbol for 182, 564
 - versus inheritance 573
 - Client module 30
 - Closure
 - of an alphabet 16
 - of the derivation operation 19
 - Code 101
 - Code folder 8
 - Commander button 40
 - Comments 34
 - Compilation unit 29
 - Compilers 31–32
 - Compiling a program 38
 - Component Pascal 2–3
 - calling mode 126
 - comments 34
 - identifier 34–36
 - program 32
 - reserved words 34
 - semicolon 38
 - statement 37
 - syntax 26–27, 603–604
 - Concatenation 16, 66
 - Constant 97
 - Context sensitive grammar 21
 - Control
 - caption 79
 - command button 79
 - dialog box 79
 - radio button 103
 - text field 79
 - Control guard 546
 - Controller 180
 - CP. See Component Pascal
 - Cross-platform 3–4
- D**
- Dead code 161
 - DEC 56
 - Default calling mode. See Calling mode, by value
 - Delegation 584, 594
 - Derivation
 - of a valid sentence 19
 - Design
 - by contract 144, 577
 - by contract rule 370, 577
 - versus analysis 321
 - Dialog
 - .Par 548
 - .Update 83
 - interface 548
 - Dialog box 75–84
 - button 76
 - control 79
 - from program 195
 - tool 353–355
 - I/O, See specific type e.g. String, output to dialog box
 - Directory 184
 - DIV 54, 58
 - do statement, GCL 227
 - Docu folder 8, 40, 81
 - Documentation files 40
 - Drag and drop 9
 - Dynamic storage allocation 463
- E**
- EBNF. See Extended Backus-Naur form
 - ELSE 94
 - ELSIF 153, 164
 - Empty string 16
 - Enc folder 8
 - Encoding BlackBox documents 9–13
 - ENTIER 59, 258
 - eot technique 202
 - Error
 - logical 39
 - syntax 38–40
 - Evaluation
 - full 350
 - short-circuit 351
 - Execution count
 - best-case versus worst-case 329, 351
 - nested loop 240–242, 243–245
 - single loop 204–208
 - Export
 - read-only 82, 191, 563
 - read/write 83, 563
 - Exporting procedures 36
 - Expression
 - BOOLEAN 87
 - infix 114
 - INTEGER 54
 - mixed 57
 - postfix 114
 - prefix 114
 - REAL 51, 213
 - Extended Backus-Naur form 24–25
 - EXTENSIBLE 575

F

Factorial
 iterative 281
 recursive 400
 Factory design pattern 184
 Field 82
 in trap window 124
 Flowchart 94
 CASE 107
 IF with ELSE 96
 IF without ELSE 94, 95
 nested IF 153
 REPEAT 298
 sequential vs nested IF 157
 WHILE 201
 FOR statement 222–227
 Formal parameter 37, 51
 Formal specification 145
 for array 337
 Formatter 182
 called by constant reference 255
 Full evaluation 350
 Function procedure 60–62, 277–288

G

GCD. See greatest common divisor
 GCL. See Guarded command language
 Global data 29
 Global variable 76, 269–272
 in trap window 124
 when to use 272
 Grammar 17–22
 context sensitive 21
 for algebraic expressions 23
 for identifiers 18
 for signed integers 19
 four parts of 17
 Graphical user interface 5
 Greatest common divisor 407
 Guard
 control 546
 Guarded command language 68
 assignment statement 69
 do statement 227
 if statement 108, 165
 multiple assignment 70

skip statement 108
 GUI. See Graphical user interface

H

HALT 161
 has-a relationship 564
 Heap 463
 Hoare triple 145, 156, 338
 Hollywood Principle 548

I

Identifier
 Component Pascal 34–36
 grammar for 18
 Identity element 16
 IF statement 92–103
 nested 151–165
 if statement, GCL 108, 165
 Import abbreviation 544
 Importing procedures 36
 IN. See Calling mode, by constant reference
 INC 56
 Infix expression 114
 Inheritance 175, 474, 552–554
 symbol for 181, 564
 versus class composition 573
 Input
 See specific type, e.g. REAL, input from dialog box
 Insertion sort 438
 INTEGER 53
 ARRAY OF 331
 expression 54
 input from dialog box 76
 output to dialog box 81
 output to Log 53
 range of 59
 to string conversion 66
 Interactor 83
 Interface 30–31
 See Module for interfaces of specific modules
 Invariant 156–163
 implemented with ASSERT 160
 strong versus weak 159
 Inverse operator 90

is-a relationship 564
 Iterator 182

J

Jacopini, Guisepppe 222

L

Language 15–17
 definition of 17
 Largest
 computing from array 328
 computing from window 212
 LargestLast 330
 Layout mode 79, 80, 83
 Left-to-right rule 52
 Lehmer generator 301
 LEN 321, 325, 334
 with \$ string selector 335
 with two-dimensional array 383
 Length
 of ARRAY OF CHAR 334
 of ARRAY OF REAL 321, 325, 334
 of string 16
 lg 360, 458
 List 130
 abstract data structure 130
 abstract data type 138
 circular doubly-linked 488
 circular linked 477
 linked 469–508
 linked class 492, 592–601
 Loading modules 76
 Local data 29
 Local variable 76
 Logarithm 360, 458
 Logical error 39
 LONGINT
 range of 59
 Loop invariant 208–209, 505

M

Mask mode 79
 Math
 .Exp 60
 .IntPower 61, 214
 .Ln 60

- .Log 60
- .Pi 60
 - interface 61
- Matrix multiplication 387–391
- MAX 60
- Menu selections 194
- Merge sort
 - concept 436
 - implementation 448–449
 - in-place 452–457
- Merritt sort taxonomy 435
- Merritt, Susan 436
- Method 177
- MIN 60
- Mixed expression 57
- MOD 55, 58
- Mod folder 8
- Model 179
- Model view controller
 - design pattern 179–197
- Modula-2 2
- Module 29, 33
 - Dialog 83, 548
 - Math 60
 - PboxCListADT 477
 - PboxListADS 130
 - PboxListADT 138
 - PboxLListObj 492
 - PboxLListSta 592
 - PboxMappers 182–183
 - PboxPizza 569
 - PboxRandom 293
 - PboxShapeAbs 552
 - PboxShapeObj 541
 - PboxStackADS 119
 - PboxStackADT 125, 176
 - PboxStackObj 176
 - PboxStrings 64
 - PboxTreeADT 520
 - PboxTreeObj 521
 - PboxTreeSta 582
 - StdLog 31
 - TextControllers 190–191
 - TextModels 184
 - TextViews 185
 - unloading 508
 - Views 186
- Mössenböck, H. P. 3

- Mutual recursion 422
- MVC. See Model view controller

N

- Name
 - of variable 49
- Naur, Peter 2
- Nesting 151, 239
- NEW 465, 470
 - method attribute 553
- NIL 464, 469
- Nonterminal alphabet 17

O

- Oberon 2
- Oberon microsystems 3
 - URL 7
- Object 175
- Object program 31
- Object-oriented language 6
- Object-oriented programming 29
 - primary benefit of 562
 - shift in focus 540
 - terminology 179, 540
- ODD 88
- OOP. See Object-oriented programming
- Open array 319
- Operation. See Unified Modeling Language
- Operator
 - associative 52
 - inverse 90
 - precedence 91
- Optional, EBNF operation 25
- OR 89
- ORD 63
- Ordinal value of CHAR 63
- OUT. See Calling mode, by result
- Output
 - See specific type, e.g. REAL, output to Log

P

- Parameter
 - actual 37, 51
 - class assignment rule 557

- formal 37, 51
- Parsing 22
- Pascal, Blaise 2
- Pascal's triangle 409
- Pbox project 113
 - URL 64
- PboxCListADT
 - .Clear 478, 486
 - .Empty 478, 486
 - .GoNext 479, 486
 - .Insert 479, 486
 - .NodeContent 480, 486
 - implementation 485
 - interface 477
- PboxListADS
 - .Clear 131
 - .Display 134
 - .GetElementN 133
 - .InsertAtN 131
 - .Length 131
 - .RemoveN 132
 - .Search 133
 - .T 131
 - interface 130
- PboxListADT
 - .InsertAtN 139
 - .List 139
 - .RemoveN 145
 - implementation 375
 - interface 138, 375
- PboxLListObj
 - documentation 493–494
 - implementation 498–499
 - interface 492, 593
- PboxLListSta
 - .EmptyNode.DisplayN 598
 - .EmptyNode.InsertAtN 600
 - .List.Display 598
 - .List.DisplayN 598
 - .List.InsertAtN 598
 - .List.Search 600
 - .List.SearchN 600
 - .NonEmptyNode.DisplayN 598
 - .NonEmptyNode.InsertAtN 599
 - implementation 595
 - interface 592
 - UML class diagram 594
- PboxMappers

- .Formatter.ConnectTo 187
 - .Formatter.WriteReal 189
 - .Formatter.WriteRealMatrix 383
 - .Formatter.WriteRealVector 326
 - .Formatter.WriteString 187
 - .Scanner.ConnectTo 192
 - .Scanner.eot 202
 - .Scanner.ScanInt 210
 - .Scanner.ScanReal 193, 202, 210
 - .Scanner.ScanRealMatrix 382
 - .Scanner.ScanRealVector 318
 - interface 182–183
 - PboxPizza
 - alternate design 572
 - implementation 569
 - UML class diagram 568
 - PboxRandom
 - .Int 298
 - .Randomize 294
 - .Real 296
 - .SetSeed 294
 - implementation 302
 - interface 293
 - PboxShapeAbs
 - .Circle 554
 - .Circle.SetIDString 561
 - .Rectangle 553
 - .Rectangle.SetIDString 554, 561
 - .Rectangle.SetState 554
 - .Shape 553
 - .Shape.SetIDString 553
 - implementation 559
 - interface 552
 - PboxShapeObj
 - implementation 548
 - interface 541
 - PboxStackADS
 - .capacity 119
 - .Clear 119
 - .NumItems 120
 - .Pop 120, 371
 - .Push 120, 369
 - implementation 370
 - interface 119, 368
 - PboxStackADT
 - .Clear 126
 - .Pop 126
 - .Stack 125
 - interface 125, 176
 - PboxStackObj
 - implementation 372
 - interface 176, 371
 - PboxStrings
 - .IntToString 66
 - .Lower 64
 - .RealToString 84
 - .ToLower 101
 - .Upper 64
 - interface 64
 - URL 64
 - PboxTreeADT
 - .Insert 520
 - .T 520
 - .Tree 520
 - interface 520
 - PboxTreeObj
 - implementation 525
 - interface 521, 582
 - PboxTreeSta
 - .EmptyNode.Insert 590
 - .EmptyNode.PreOrder 588
 - .Node.Insert 589
 - .NonEmptyNode.Insert 589
 - .NonEmptyNode.PreOrder 589
 - .Tree.Clear 587
 - .Tree.Insert 589
 - .Tree.PreOrder 588
 - implementation 585
 - interface 582
 - UML class diagram 583
 - Performance. See Execution count
 - Permutation 340, 417
 - Plato 533
 - POINTER 463
 - assignment 465, 470
 - legal operations 468
 - period abbreviation for \wedge . 468
 - versus record 574
 - Polymorphic dispatch 558
 - Polymorphism 475, 555–558, 585
 - Pop operation 114
 - Postcondition 120, 145
 - Postfix expression 114
 - Precedence of operators 91, 116
 - Precondition 120, 145, 156
 - Prefix expression 114
 - Primitive type 125
 - Private 563
 - versus public 576
 - Procedure
 - function 60–62, 277–288
 - proper 42–45, 251–272
 - signature 37
 - Production
 - rules of 17, 18
 - Program 32
 - compiling 38
 - style 41
 - Programmer-defined type 125
 - Project folders 6–8
 - Proper procedure 42–45, 251–272
 - Public 563
 - versus private 576
 - Push operation 114
- Q**
- Queue 143
 - Quick sort
 - concept 437
 - correctness of 446
 - implementation 439–446
- R**
- Radio button 103
 - Random
 - integers 296
 - number generators 300
 - reals 294
 - Range
 - of INTEGER 59
 - of LONGINT 59
 - of REAL 60
 - Read-only export 82, 191, 563
 - Read/write export 83, 563
 - REAL 49
 - ARRAY OF 314
 - expression 51, 213
 - input from dialog box 84
 - input from window 190
 - input of matrix from window 382
 - input of vector from window 318
 - output of matrix to window 383
 - output of vector to window 326

610 Computing Fundamentals

- output to Log 50, 52
- output to window 189
- range of 60
- Receiver 176
 - restrictions on 500
- RECORD 82
 - assignment 490
 - versus pointer 574
- Recursion 399–423
 - cost of 423
 - macroscopic viewpoint 405
 - microscopic viewpoint 404
 - mutual 422
- REPEAT statement 298
- Repetition, EBNF operation 25
- Republic 533
- Reserved words 34
- Return address 251, 280, 284
- RETURN statement 278
 - in proper procedure 358
- Rigid variable 146, 338
- Rsrc folder 8, 80
- Rules of production 17, 18
- Run-time stack
 - for function procedures 280, 401
 - for proper procedures 251

S

- Scanner 182
- Schrage's algorithm 305
- Searching
 - iterative 349–360
 - recursive 433–435
- Selection sort 360–362
 - concept 438
- Selector, \$ 68, 333
- Semantics 15
- Semicolon
 - rules for 96
- Semicolon separator 38
- Sequential search 350–356
- Server module 30
- SHORT 59, 257
- Short-circuit evaluation 351
- Signature of procedure 37
- skip statement, GCL 108
- Socrates 533

- Sorting
 - complexity 457
 - iterative 360–362
 - recursive 435–458
- Source program 31
- Specification, formal 145
 - for array 337
- Stack 114
 - abstract data structure 119
 - abstract data type 124
 - class 176
 - See also Run-time stack
- Stack frame 255
- Start symbol 17, 18
- State design pattern 581–601
- Statement
 - See specific statement, e.g. IF statement
- StdCmds
 - .OpenAuxDialog 81
 - .OpenToolDialog 354
- StdLog
 - .Char 63
 - .Real 51
 - .String 37, 66
 - interface 31
- Stepwise refinement 217–222, 245–247
- String
 - alphabetic order 99
 - empty 16
 - length of 16
 - output to dialog box 84
 - output to Log 37
 - output to window 186
 - selection with 99
 - selector \$ 68, 333
- Structured programming theorem 222
- Stub 497
- Style
 - precondition 120
 - program 41
- Subclass 474, 553
- Superclass 474, 553
- Sym folder 8
- Syntax 15
 - Component Pascal 26–27, 603–

- 604
- error 38–40

T

- Terminal alphabet 17
- Text field control 79
- Text subsystem 8–9
- TextControllers 181
 - .Controller 191
 - .Controller.text 192
 - .Focus 191
 - interface 190–191
- TextModels 181
 - .dir 185
 - .dir.New 187
 - .Directory 184
 - .Directory.New 184
 - .Model 184
 - interface 184
- TextViews 181
 - .dir.New 187
 - .Directory.New 185
 - interface 185
- Tool dialog box 353–355
- Towers of Hanoi 419
- Trap 122, 260, 317
- Type
 - base 552
 - extension 552
 - guard 475
 - in GCL 69
 - of variable 49
 - primitive 125
 - programmer-defined 125
 - static versus dynamic 475, 558
 - See also specific type, e.g. REAL

U

- UML. See Unified modeling language
- Unified Modeling Language 181, 474, 562–565
 - attribute 540
 - class diagram 539, 563
 - operation 540
- URL
 - Oberon microsystems 7

PboxStrings 64

V

Value

of variable 49

VAR. See Calling mode, by reference

Variable

global 76

local 76

name of 49

rigid 146, 338

three attributes of 49

type of 49

value of 49

See also specific type, e.g. REAL

View 179

Views 181

.OpenView 186, 187

.View 186

interface 186

W

WHILE statement 201–222

Window 179–197

I/O, See specific type e.g. String,
output to window

Wirth, Niklaus 2