

Index

- abstract class, 34
- abstract data type, 11
- activation record, 90
- ADT. *See* abstract data type
- ako*. *See* specialization
- association, 4
- asymptotic algorithm analysis, 38
- attribute, 2

- big-Oh. *See* big-Omicron notation
- big-Omicron notation, 42
- binary search algorithm, 45
 - implementation, 47
- binary search tree, 143
 - ADT, 145
 - applications, 158
 - AVL tree, 158
 - implementation, 149
 - insertion, 146
 - red-black tree, 158
 - removal, 148
 - searching, 148
 - splay tree, 158
 - time analysis, 157
 - total ordering, 145
- binary tree, 52, 131
 - ADT, 133
 - applications, 138
 - balanced, 157
 - complete, 52
 - depth of a node, 52
 - edge, 52
 - full, 143
 - height, 52
 - implementation, 134
 - internal node, 52
 - leaf node, 52
 - level, 52
 - traversal, 133
- BST. *See* binary search tree
- bubble sort algorithm, 39
 - implementation, 47
 - time analysis, 41

- call frame, 90
- class inheritance, 28
- classifier, 1
- collection, 12
- computer simulation, 117
- constant folding, 75
- container, 12

- data structure, 12
 - dynamic, 25
- data structures, 11
- dependency, 6
- deque, 127
 - ADT, 129
 - applications, 131
 - implementation, 129
- divide-and-conquer algorithm, 46
- dynamic array, 25

- expression evaluation, 76
- expression tree, 142

- factorial function, 96
- Fibonacci numbers, 96
- field shadowing, 122
- FIFO. *See* first-in-first-out
- first-in-first-out, 113

- generalization, 3
- growth rate, 38
 - constant-time, 45
 - exponential, 39
 - linear, 39
 - logarithmic, 39
 - quadratic, 39
- has-a*. *See* association
- heap, 51
 - ADT, 53
 - applications, 63
 - implementation, 54, 122
 - insertion, 57
 - partial ordering, 53
 - removal, 60
- heap sort algorithm, 64
 - implementation, 64
 - time analysis, 65
- Huffman coding, 138
- infix expression, 77
- infix-to-postfix conversion, 78
- inheritance
 - First Principle of, 29
 - Second Principle of, 29
- inner class, 125
 - anonymous, 125
 - local, 125
 - member, 125
- interface, 46
- iteration, 97
- Java Virtual Machine, 89
- JVM. *See* Java Virtual Machine. *See* Java Virtual Machine
- last-in-first-out, 69
- least-integer function, 65
- LIFO. *See* last-in-first-out
- linear search algorithm, 44
 - implementation, 47
 - time analysis, 44
- linked list, 70
 - doubly linked, 115
 - insertion, 71
 - removal, 73
 - singly linked, 115
- max-heap, 53
- min-heap, 53
 - implementation, 61
- model, 1
- multi-tasking operating system, 64
- nested type, 125
- nesting, 119
- operation, 2
- postfix evaluation, 86
- postfix expression, 77
- precondition/postcondition, 19
- print queue, 117
- priority queue, 64
- queue, 113
 - ADT, 115
 - applications, 117
 - implementation, 116
- random-access, 17
- realization, 7
- recursion, 87
 - pros and cons of, 97
- recursive call, 87
- recursive definition, 87
- recursive method, 87
- return address, 90
- run-time stack, 88
 - storing arguments on, 90
- selection sort algorithm, 42
 - implementation, 47
 - time analysis, 43
- sequential-access, 70
- set, 39
- specialization, 3

- stack, 67
 - ADT, 69
 - applications, 75
 - implementation, 74
 - overflow, 95
- stack frame, 90
- static array, 28

- type independence, 25

- UML. *See* Unified Modeling Language

- Unified Modeling Language, 1
 - and Java code, 8
 - class diagram, 1
 - static-structure diagram, 1
 - stereotyping, 7
- upper bound, 42
- uses-a*. *See* dependency

- vector, 17
 - ADT, 18
 - implementation, 19, 26, 34

- wrapper, 25