

Second AP* Edition
— with GridWorld

Java

Methods

Object-Oriented Programming
and
Data Structures

Maria Litvin

Phillips Academy, Andover, Massachusetts

Gary Litvin

Skylight Software, Inc.

Skylight Publishing
Andover, Massachusetts

Skylight Publishing
9 Bartlet Street, Suite 70
Andover, MA 01810

web: <http://www.skylit.com>
e-mail: sales@skylit.com
support@skylit.com

**Copyright © 2011 by Maria Litvin, Gary Litvin, and
Skylight Publishing**

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the authors and Skylight Publishing.

Library of Congress Control Number: 2010915303

ISBN 978-0-9824775-7-1

* AP and Advanced Placement are registered trademarks of The College Board, which was not involved in the production of and does not endorse this book.

The names of commercially available software and products mentioned in this book are used for identification purposes only and may be trademarks or registered trademarks owned by corporations and other commercial entities. Skylight Publishing and the authors have no affiliation with and disclaim any sponsorship or endorsement by any of these product manufacturers or trademark owners.

Oracle, Java, and Java logos are trademarks or registered trademarks of Oracle Corporation and/or its affiliates in the U.S. and other countries.

SCRABBLE® is the registered trademark of HASBRO in the United States and Canada and of J.W. Spear and Sons, PLC, a subsidiary of Mattel, Inc., outside the United States and Canada.

1 2 3 4 5 6 7 8 9 10 16 15 14 13 12 11

Printed in the United States of America

☆ Appendices ☆

The following appendices are online at

<http://www.skylit.com/javamethods>

- Appendix A: The 17 Bits of Style
- Appendix B: Common Syntax Error Messages
- Appendix C: GUI Examples Index
- Appendix D: The `EasyReader`, `EasyWriter`, `EasySound`,
and `EasyDate` Classes

☆ Solutions to Selected Exercises ☆

Solutions to exercises marked with a ✓ are downloadable from <http://www.skylit.com/javamethods>. Click on the “Download *Student Files*” link, download and unzip the `studentfiles.zip` file, then open `JM\SolutionsToExercises.pdf`.

Index of Tables and Figures

Tables:

- 1-1. Memory units 1★-6
- 6-1. The primitive data types 133
- 11-1. Similarities and differences between a class and an interface 321
- 18-1. `KeyEvent`'s symbolic constants for virtual keys 18★-5
- 19-1. Best, average, and worst case for sorting and searching 460
- 20-1. Average time big-O for some methods of `ArrayList` and `LinkedList` 476
- 20-2. The Java collections framework summary 490
- 20-3. Structural design decisions for *SafeTrade* 495

Figures:

- 1-1. AND, OR, and NOT gates 3
- 1-2. XOR circuit made of AND, OR, and NOT gates 3
- 1-3. Software functional layers 1★-9
- 1-4. Excel-97 "Easter Egg." 1★-12
- 1-5. IEEE standard representation of 32-bit floating-point numbers 1★-18
- 1-6. ASCII code used in personal computers and printers 1★-20
- 1-7. TCP/IP protocols in layered network architecture 1★-21

- 2-1. Two technicians wiring the right side of ENIAC 12
- 2-2. Software development cycle for a compiled program: edit-compile-link-run 16
- 2-3. Java software development and distribution through the Internet 18
- 2-4. The term "bug" was popularized by Grace Hopper 19
- 2-5. A hierarchy of classes that represent compact disks with different content 29
- 2-6. `JM\Ch02\HelloGui\HelloGui.java` 31
- 2-7. `JM\Ch02\HelloGui\HelloGraphics.java` 32
- 2-8. `JM\Ch02\HelloGui\HelloApplet.java` 34
- 2-9. `JM\Ch02\HelloGui\TestApplet.html` 35
- 2-10. `JM\Ch02\HelloGui\Banner.java` 36

-
- 3-1. A window from the *BugRunner* program in GridWorld 46
 - 3-2. A CRC card for the class Bug 49
 - 3-3. Class vs. object 50
 - 3-4. A schematic view of a class's source code 52
 - 3-5. Fragments from BugRunner.java 55
 - 3-6. `JM\Ch03\GridWorld\UTurnBug.java` 65
 - 3-7. Actor has a `Grid<Actor>`, `Location`, and `Color`; `Rock`, `Flower`, and `Bug` extend `Actor`; 68

 - 4-1. Brute-force “non-algorithm” for calculating $1^2 + 2^2 + \dots + 100^2$ 81
 - 4-2. Pseudocode and flowchart for an algorithm that calculates $1^2 + 2^2 + \dots + n^2$ 83
 - 4-3. Python, Java, and C/C++ implementations of the sum-of-squares algorithm 84
 - 4-4. A version of the sum-of-squares algorithm without multiplications 85
 - 4-5. Some of the graphics elements are “primitives”; others are “pictures” 87
 - 4-6. Recursive method in Java for computing $1^2 + 2^2 + \dots + n^2$ (not recommended for this task) 89
 - 4-7. Euclid's Algorithm for finding the greatest common factor of two positive integers 90
 - 4-8. Iterative implementation of Euclid's Algorithm in Java 91
 - 4-9. Recursive implementation of Euclid's Algorithm in Java 91
 - 4-10. Binary Search 94
 - 4-11. A snapshot of the *Windows Explorer* screen 95

 - 5-1. Java reserved words 111
 - 5-2. Reserved words in the `HelloGui` class 112
 - 5-3. `HelloGui.java`: compiles with no errors 115
 - 5-4. Nested blocks, marked by braces, within a class 118
 - 5-5. `JM\Ch05\Syntax\MovingDisk.java` (with syntax errors) 120

 - 6-1. A fragment from the `Actor` class 128
 - 6-2. The *Poll* program 147

 - 7-1. Dots configurations on a die 173
 - 7-2. The *Craps* program 174
 - 7-3. GUI classes in the *Craps* program 175
 - 7-4. *Craps* classes and their relationships 176
 - 7-5. Task assignments in the *Craps* project 179
 - 7-6. A preliminary program for testing `CrapsGame` 181
 - 7-7. Temporary “stub” class `CrapsTable.java` 187
 - 7-8. A fragment from `JM\Ch07\Craps\RollingDie.java` 189

-
- 9-1. `JM\Ch09\Fraction\Fraction.java` 221
 - 9-2. Copying references vs. copying objects 229
 - 9-3. The *Snack Bar* program with three vending machines 242
 - 9-4. *Snack Bar* program's objects 243
 - 9-5. `JM\Ch09\SnackBar\Vendor.java` 245

 - 10-1. A new value assigned to a `String` variable 269
 - 10-2. Assigning references vs. copying strings 269
 - 10-3. Commonly used `String` methods 271
 - 10-4. Converting input into an `int` with exception handling 281
 - 10-5. The *Lipogrammer* program 283
 - 10-6. Examples of common `StringBuffer` methods 285

 - 11-1. Inheritance terminology and notation in class diagrams: a subclass extends superclass 294
 - 11-2. A hierarchy of classes 295
 - 11-3. A fragment of Java library GUI class hierarchy (abstract classes are boxed) 301
 - 11-4. A constructor invokes its superclass's constructor, all the way up to `Object` 304
 - 11-5. `GridWorld` "dance" 308
 - 11-6. `GridWorld` Dance project: a `Dancer` "learns" a `Dance`, then performs the necessary steps in its `act` method 311
 - 11-7. `YMCA`Number extends `DanceNumber`; each has a `main` method and can be used as a runnable class 317
 - 11-8. A sketch of a concrete class that implements the `Dance` interface 320
 - 11-9. `GridWorld`'s `Grid` interface and the classes that implement it 324

 - 12-1. The *Fortune Teller* applet 339
 - 12-2. The *Chomp* game program 343
 - 12-3. The classes in the *Chomp* program 344
 - 12-4. Inserting an element in the middle of an array 351

 - 13-1. The capacity and size of an `ArrayList` 363
 - 13-2. Commonly used `ArrayList<E>` methods 366
 - 13-3. (a) A list with references to different objects; (b) A list with references to the same object 370
 - 13-4. A sample text file and its index 371
 - 13-5. *IndexMaker* classes 371

- 14-1. `JM\Ch14\Compare\Country.java` 384
- 14-2. `JM\Ch14\Compare\PopulationComparator.java` 388
- 14-3. `JM\Ch14\Compare\ComparatorTest.java` 389
- 14-4. `JM\Ch14\BinarySearch\BinarySearch.java` 391
- 14-5. `JM\Ch14\Benchmarks\SelectionSort.java` 395
- 14-6. `JM\Ch14\Benchmarks\InsertionSort.java` 396
- 14-7. `JM\Ch14\Benchmarks\Mergesort.java` 399
- 14-8. Array partitioning algorithm 400
- 14-9. `JM\Ch14\Benchmarks\Quicksort.java` 402
- 14-10. The *Benchmarks* program 402

- 15-1. A subset of `java.io.File`'s methods 415
- 15-2. A `JFileChooser` example 416
- 15-3. A subset of `java.util.Scanner`'s methods 418
- 15-4. Reading a file character by character using a `BufferedReader` 419
- 15-5. Creating a text file 420
- 15-6. Appending data to a text file 421
- 15-7. The output file format in the *Dictionary Maker* program 422

- 16-1. Graphics coordinates: y-axis points down 16*-7
- 16-2. A filled oval inscribed into a filled rectangle 16*-9
- 16-3. Positioning of ovals, arcs, and rounded rectangles 16*-9
- 16-4. The drawing methods of the `Graphics` class 16*-11
- 16-5. Font metrics 16*-14
- 16-6. The *Puzzle* program 16*-15
- 16-7. `pickPiece` and `dropPiece` methods in the `Puzzle` class
(`JM\Ch16\Puzzle\Puzzle.java`) 16*-17
- 16-8. State machine diagram for moving pieces in the *Puzzle* program 16*-18
- 16-9. The offset of relative coordinates for a piece of the puzzle 16*-19

- 17-1. An `ActionListener` implemented as an inner class 17*-7
- 17-2. The border layout 17*-12
- 17-3. The *Ramblecs* game 17*-14
- 17-4. Classes in the *Ramblecs* program 17*-15

- 18-1. Adding a `MouseListener` to a panel using `MouseAdapter` 18*-3
- 18-2. `&` operator used to test a particular bit in an integer 18*-6
- 18-3. `|` operator is used to combine bits in two integers 18*-7
- 18-4. `&` operator is used to “cut out” mask bits from an integer 18*-7
- 18-5. Drawing an image in an applet 18*-10
- 18-6. Drawing an `ImageIcon` 18*-11
- 18-7. The *BalloonDraw* program 18*-12
- 18-8. Classes in *BalloonDraw* 18*-13

- 19-1. (a) Linear growth (b) Logarithmic growth
(c) Logarithmic growth is slower than linear 449
- 19-2. Rates of growth 454
- 19-3. A sorting algorithm as a decision tree (for three values) 456

- 20-1. Part of the Java collections framework 467
- 20-2. Some of the methods of `java.util.Collection<E>` interface 468
- 20-3. The methods of `java.util.Iterator<E>` interface 470
- 20-4. Some of the methods of the `java.util.List<E>` interface 472
- 20-5. Methods of the `java.util.ListIterator<E>` interface 473
- 20-6. Logical “cursor” positioning for a `ListIterator` 474
- 20-7. A doubly-linked list 475
- 20-8. Additional methods in `java.util.LinkedList<E>` 477
- 20-9. Merging two sorted `LinkedLists` 478
- 20-10. Methods of `java.util.Stack<E>` class 479
- 20-11. An example of a method that uses a stack 480
- 20-12. Methods of `java.util.Queue<E>` interface 481
- 20-13. Quicksort algorithm implemented with queues 482
- 20-14. A simplistic version of Heapsort 484
- 20-15. Some of the methods of `java.util.Set<E>` interface 485
- 20-16. An example of using a `TreeSet` 486
- 20-17. A mapping from a set of keys onto a set of values 487
- 20-18. Some of the methods of `java.util.Map<K,V>` interface 488
- 20-19. An example of code that uses a map 489
- 20-20. Class diagram for the *SafeTrade* program 496
- 20-21. A trader window 497

- 21-1. A class that represents a node in a singly-linked list
(`JM\Ch21\Lists\ListNode.java`) 509
- 21-2. A linked list with only one node 511
- 21-3. The second node is appended to the list 511
- 21-4. The constructors of the `SinglyLinkedList` class
(`JM\Ch21\Lists\SinglyLinkedList.java`) 512
- 21-5. A singly-linked list 513
- 21-6. A new node inserted at the head of a linked list 513
- 21-7. `JM\Ch21\Lists\SinglyLinkedListIterator.java` 515
- 21-8. A singly-linked list with a tail 517
- 21-9. A class that represents a node of a doubly-linked list
(`JM\Ch21\Lists\ListNode2.java`) 518
- 21-10. A doubly-linked circular list 519
- 21-11. A circular doubly-linked list with a header node 520
- 21-12. A snapshot from the *Teletext* program 521
- 21-13. Class diagram for the *Teletext* program 522

- 22-1. “Our own” `Stack` interface (`JM\Ch22\Stack.java`) 529
- 22-2. An implementation of stack based on `ArrayList`
(`JM\Ch22\ArrayStack.java`) 529
- 22-3. An implementation of stack based on `LinkedList`
(`JM\Ch22>ListStack.java`) 529
- 22-4. An array implementation of stack 530
- 22-5. *Line Cruiser* classes 531
- 22-6. Frames on the system stack after a few method calls 535
- 22-7. Ring-buffer implementation of a queue 536
- 22-8. Classes, objects, and public methods in the *Three Kittens* program 539

- 23-1. A “composite list”: some nodes hold strings, others hold lists 553
- 23-2. A 5 by 5 *Hex* board 565
- 23-3. A *Hex* board represented as a 2-D array 567
- 23-4. *Hex*: detecting a win for black stones 568

- 24-1. A tree structure 578
- 24-2. Common uses of tree structures 579
- 24-3. Arrangement of tree nodes in levels 580
- 24-4. A shallow tree can hold many nodes 580
- 24-5. A tree with a nearly linear shape — each level holds only a few nodes 581
- 24-6. Each node in a tree is the root of its own subtree 582
- 24-7. `TreeNode` represents a node in a binary tree (`JM\Ch24\TreeNode.java`) 583
- 24-8. The ordering property of a binary search tree 587
- 24-9. Locations of the smallest and largest values in a BST 588
- 24-10. Public methods from `MyTreeSet` (`JM\Ch24\BST\MyTreeSet.java`) 589
- 24-11. Recursive helper methods in `JM\Ch24\BST\MyTreeSet.java` 591
- 24-12. Removing the root node from a subtree in a BST 592
- 24-13. *Java Messenger* classes 595

- 25-1. A lookup table for zip codes in the United States 607
- 25-2. Resolving collisions in a hash table by chaining 613
- 25-3. Resolving collisions in a hash table by linear probing 614

- 26-1. The shapes of *full* and *complete* binary trees 628
- 26-2. Numbering of the nodes in a complete tree 629
- 26-3. Representation of a complete binary tree in an array 629
- 26-4. A fragment from the `HeapPriorityQueue` class
(`JM\Ch26\Heap\HeapPriorityQueue.java`) 631
- 26-5. Removing the root node from a heap 632
- 26-6. `add` and `remove` methods in `HeapPriorityQueue` 633

- 27-1. The Decorator design pattern 27*-11
- 27-2. The Decorator design pattern with the intermediate abstract decorator class 27*-13
- 27-3. *Sphere* model with text and mouse input 27*-17
- 27-4. “Totalitarian” control of the model and the views 27*-19
- 27-5. The MVC design pattern 27*-19
- 27-6. A sketch for the *Sphere* classes with `Observer/Observable` 27*-22

Index

- decrement operator, 143
- ! (*not*) operator, 164
- != (is not equal) operator, 162
- % (modulo division) operator, 139
- && (*and*) operator, 164
- || (*or*) operator, 164
- + and += operators for strings, 134, 272, 144
- ++ increment operator, 143
- += and -= arithmetic operators, 142
- = assignment operator, 127
 - vs. equals method, 163
- == (is equal) operator, 162
- == and != operators
 - for enum types, 185
 - for objects, 163
- <, <=, >, >= relational operators, 162

- A/D (analog to digital) converter, 1★-8, 1★-14
- abs method of Math, 253
- absolute pathname, 414
- abstract class, 299, 326
- abstract keyword, 299
- abstract methods, 299, 326
- abstraction, 320
- access rules in classes, 222
- accessor methods, 66, 71, 223
- ActionListener interface, 17★-6
- actionPerformed method, 35, 17★-17
- adapter class, 16★-15, 18★-3, 18★-18

- add method of
 - Collection, 468
 - ListIterator, 474
 - Queue, 481
 - Set, 485
- addActionListener method, 17★-8
- addFirst method of LinkedList, 477
- addLast method of LinkedList, 477
- addMouseListener method, 18★-2
- addObserver method, 27★-20
- addSeparator method, 17★-13
- Alexander, Christopher, 27★-2, 27★-24, 640, 641
- algorithm, 1★-11, 13, 80, 98
- Alt key, 18★-5
- anagram, 425
- analog electronics, 2
- analog-to-digital converter, 1★-8
- AND gate, 2
- API (Application Programming Interface), 39, 72
- appending text to a file, 421
- applet, 17, 33
- Applet Viewer*, 19, 22
- area fill, 568
- args (in main), 25
- arguments (parameters), 58
 - of a method, 229
 - of primitive data types, 236
 - passed as references, 237, 257
 - passed by reference, 236
 - passed by value, 236
- arithmetic expressions, 138, 149, 1★-14

- array, 334
 - 2-D, 340
 - as object, 335
 - default values for elements, 336
 - finding largest/smallest element, 349
 - indices, 334, 337, 352
 - inserting an element, 351
 - length, 334, 337
 - one-dimensional, 335
 - passed to a method, 338
 - searching, 389
 - size, 334
 - sorting, 394
 - subscript, 334, 337, 352
 - two-dimensional, 340
- ArrayIndexOutOfBoundsException, 338
- ArrayList class, 362
 - capacity vs. size, 362
 - methods, 366
 - no-args constructor, 365
 - toString method, 367
 - type of elements, 364
- Arrays class, 404
 - asList method, 405
 - binarySearch method, 404
 - fill method, 405
 - sort method, 404
 - toString method, 405
- artificial intelligence (AI), 537
- ASCII, 1^{*}-19
 - control codes, 1^{*}-19
 - file, 1^{*}-19
- asList method in Arrays, 405
- assembly language, 14
- assignment operation, 127
- asymptotic behavior, 447, 450
- autoboxing (autowrapping), 365, 469
- autounboxing, 469
- backslash in literal strings, 266
- base case in recursion, 88, 397, 551
- base class, 64, 71
- benchmark, 402
- Benchmarks program, 402
- big-O notation, 390, 450
- binary arithmetic, 1^{*}-16
- binary files, 413, 1^{*}-19
- binary numbers, 1^{*}-15, 1^{*}-16
- binary operator, 167
- Binary Search, 93, 390, 407, 448
- binary search tree (BST), 485, 586
 - add method, 591
 - do-it-yourself, 588
 - inorder traversal, 587
 - remove method, 591
 - smallest and largest values, 587
- binary tree, 583
 - and recursion, 581
 - children nodes, 578
 - complete, 628
 - full, 628
 - inorder traversal, 585
 - leaves, 578
 - left and right child, 583
 - node, 578
 - nodeCount method, 584
 - non-linked representation, 627
 - parent node, 578
 - perfect, 628
 - postorder traversal, 585
 - preorder traversal, 585
 - root, 578
 - traversal, 582
- binarySearch method in Arrays, 404
- binary-to-decimal conversion, 1^{*}-17
- binomial coefficients, 104, 356
- BIOS, 1^{*}-9
- bit, 3, 1^{*}-24
- bit-wise logical operators, 168, 18^{*}-6
- BMI program, 153

- Boole, George, 2
- boolean data type, 132, 161
- Boolean expressions, 159, 161
- Boolean operations, 2
- BorderLayout, 17*-12, 17*-18
- borders in GUI components, 438, 439
- Box class, 17*-10
 - createHorizontalBox method, 17*-13
 - createVerticalBox method, 17*-13
- BoxLayout, 17*-13, 17*-18
- braces, 118
- break
 - in loops, 205, 341
 - in switch, 183, 184
- browser, 1*-9, 1*-22, 1*-24
- BST. *See* binary search tree
- BufferedReader class, 419
- bugs, 19, 117
- bus, 1*-5
- byte, 3, 1*-24
- byte data type, 132
- bytecode, 17, 18, 37

- Calendar class, 642, 643
- case in switch, 183
- cast operator, 139
- Central Processing Unit (CPU), 2
- chaining (in hash table), 613
- Change of Base Theorem, 451
- ChangeListener interface, 17*-6
- char array, 426
- char data type, 132

- Character static methods
 - isDigit, 282
 - isLetter, 282
 - isLetterOrDigit, 282
 - isLowerCase, 282
 - isUpperCase, 282
 - isWhitespace, 282
 - toLowerCase, 282
 - toUpperCase, 282
- Character wrapper class, 277, 281
- charAt method of String, 268, 270
- CharMatrix class, 345
- charValue method of Character, 277
- checked exceptions, 417
- Chomp game, 342
- circular doubly-linked list, 519
- class, 47
 - abstract, 299, 326
 - concrete, 300, 326
 - definition, 49
 - inline, 18*-3
 - inner, 470, 515
 - private, 17*-8
 - vs. interface, 320
 - vs. object, 50
- class (static) methods, 250, 254
- class (static) variables, 249, 254
- ClassCastException, 385
- client of a class, 218, 223
- clipping rectangle, 16*-7
- close method for files, 420
- cloud computing, 1*-23
- cohesion, 177
- Collection interface, 466, 468
- Collections class, 404
 - methods, 405
- collections framework, 466
 - summary, 490
- collisions (in a hash table), 612
- Color class, 16*-10
- color constants, 16*-10

- Command Prompt*, 23, 24
- command-line arguments, 25
- command-line tools, 22
- commenting out code, 110
- comments, 109, 121
- Comparable interface, 386
- Comparator interface, 387
 - compare method, 387
- compareTo method, 386
 - and equals, 387
 - of `String`, 275
- compiler, 16, 37
- complete binary tree, 635
- complete tree, 628
- Composite design pattern, 556, 641, 27*-3, 27*-15, 27*-24
- compound assignment operators, 142
- compound statement, 119
- computational complexity, 446
- concat method of `String`, 272
- concatenating strings, 134, 144, 272
- concrete class, 300, 326
- conditional branching, 82, 158
- conditional branching instructions, 4
- console application, 22, 1*-10
- constant, 131, 134
- constructor, 30, 51, 70, 224, 256
 - calling another constructor, 227
 - copy, 225
 - default, 226
 - no-args, 225
 - private, 224
- Container class, 17*-10, 17*-17
- contains method of `Collection`, 468
- containsKey method of `Map`, 488
- converting into strings
 - numbers, 277
 - objects, 146
 - other types, 144
- converting strings into numbers, 280
- Cookie Monster* program, 503, 574
- Cooney* program, 291
- coordinates in graphics, 16*-7
- copy constructor, 225
- copying references, 229
- coupling, 177, 344
- CPU, 2, 1*-5, 1*-23
 - instructions, 4, 12, 1*-24
 - registers, 1*-5
- Craps* program, 173
- CRC card, 48, 70
- Cryptogram Solver* program, 609
- Ctrl key, 18*-5
- currentTimeMillis method of `System`, 254
- D/A (digital to analog) converter, 1*-8
- “dangling else” bug, 172
- data file, 412, 422
- data structure, 446
- data type, 127
- De Morgan’s Laws, 165, 191
- debugger, 20
- DecimalFormat class, 278, 286
- decimal-to-binary conversion, 1*-17
- decision tree, 456
- declarations, 128
 - inside nested blocks, 138
- Decorator adapter, 27*-13
- Decorator design pattern, 641, 27*-3, 27*-10, 27*-23
- decrement operator `--`, 143
- default in switch, 183
- default layouts, 17*-11
- default no-args constructor, 226
- default values for fields, 59, 70, 129, 224
- derived class, 64, 71
- design patterns, 347, 640, 27*-2
- device drivers, 1*-8
- digital electronics, 2

- digital-to-analog converter, 1*-8
- divide-and-conquer algorithms, 390, 454, 459, 586, 587
- division of integers, 139
- documentation comments, 110
- double data type, 132
- Double wrapper class, 277
 - doubleValue method, 277
 - parseDouble method, 281
- doubly-linked list, 475, 508, 519, 523
- do-while loop, 200, 204, 212
- drawImage method of Graphics, 18*-10
- Drawing Editor* prigram, 18*-12
- drawing methods of Graphics, 16*-11
 - drawLine, 16*-11
 - drawPolygon, 16*-12
 - drawPolyline, 16*-12
- dynamic memory allocation, 250

- E constant in Math, 253
- EasyDate class, 643, D-4
- EasyReader and EasyWriter classes, D-1
- EasySound class, 124, D-4
- editor, 15
- Eight Queens* problem, 432
- embedded class, 17*-8
- empty method, 298
- empty string, 134, 266, 267, 268
- encapsulation, 61, 71, 223
- endsWith method of String, 289
- ENIAC computer, 12
- Enigma machine, 609
- enum reserved word, 184
- enumerated data type, 184

- equals method, 383
 - and compareTo, 387
 - of Object, 383
 - of String, 274
 - overriding, 384
- equalsIgnoreCase method of String, 274
- escape characters, 134, 266, 286
- Euclid's GCF Algorithm, 80, 90
- event listener, 17*-17
- event types, 17*-9
- event-driven programs, 28, 241, 536, 17*-2
- exception, 20, 417
- exception handling, 280
 - try-catch-finally, 280, 417
- exclusive OR operation, 2, 192
- executable program, 16, 1*-7
- exit method of System, 254
- exponent (in floating-point numbers), 1*-18
- exponential growth, 454, 552
- expression tree, 600
- extends keyword, 63
- extracting numbers from a string, 280
- extracting words from a string, 373, 619

- Facade design pattern, 641, 27*-3, 27*-23
- factorial, 203
- factorial method and recursion, 558
- fibonacci method and recursion, 559
- Fibonacci numbers, 216, 559
- fields, 51, 56, 70, 128, 149
 - accessing, 234
 - declaration, 128
 - default values, 59, 129, 224
 - private, 129, 222
 - public, 222
 - scope, 137
 - static, 249, 250, 257

- FIFO (First-In-First-Out), 480
- file, 412, 1*-7, 1*-10
- binary, 413
 - closing, 420
 - pathname, 412
 - random-access, 412, 413
 - reading, 416
 - text, 413
 - writing, 419
- File class, 414
- methods, 415
- File Transfer Protocol (FTP), 1*-22
- FileNotFoundException, 417, 419
- FileReader class, 419
- FileWriter class, 421
- fill method in Arrays, 405
- fillPolygon method of Graphics, 16*-12
- final class, 260
- final variables, 131, 135
- float data type, 132
- floating-point arithmetic, 1*-18
- floating-point numbers, IEEE standard, 1*-18
- flowchart, 82
- FlowLayout, 17*-11, 17*-17
- fonts, 16*-13
- attributes, 16*-13
 - defining, 16*-13
 - Monospaced, 16*-13
 - names, 6*-13
 - SansSerif, 16*-13
 - Serif, 16*-13
 - size, 16*-13
- “for each” loop, 349, 470
- for loop, 200, 202, 212
- formal parameters, 130
- format method of String, 280
- formatting numbers, 278
- FORTRAN programming language, 15
- Fortune Teller* applet, 339
- frames on stack, 534
- framework, 466
- FTP (File Transfer Protocol), 1*-22
- full binary tree, 628
- fully-qualified name, 54
- garbage collection, 59
- gate, 2
- GB. *See* gigabyte
- generics, 466
- get method
- of ArrayList 366
 - of LinkedList, 476
 - of Map, 488
- getActionCommand method, 17*-10
- getAllFonts method, 16*-13
- getAudioClip method, 18*-8
- getContentPane method, 17*-10
- getDocumentBase method, 18*-9
- getFirst method of LinkedList, 477
- getHeight method, 16*-8
- getImage method, 18*-9
- getKeyText method, 444
- getLast method of LinkedList, 477
- getModifiers method, 18*-5
- getSelectedIndex method, 17*-9
- getSelectedItem method, 17*-9
- getText method, 17*-9
- getWidth method, 16*-8
- gif files, 18*-9
- gigabyte, 4, 1*-6, 1*-24
- Giggle* search engine, 617, 638
- GIMPS project, 211
- Goldbach conjecture, 208
- Goofenspiel game, 643
- Gosling, James, 241
- GradientPaint class, 429, 16*-3

- graphical user interface (GUI), 19, 37,
 - 1*-10, 1*-24, 17*-2
- Graphics class, 428, 16*-2
 - drawing methods, 16*-11
- graphics coordinates, 16*-7
- Graphics2D package, 428, 16*-2
- Greco-Roman square, 9
- grep program, 424
- GridLayout, 17*-11, 17*-17
- GridWorld case study, 45, 128, 192, 216,
 - 263, 296, 299, 307, 374
- GUI (Graphical User Interface), 19, 37,
 - 1*-10, 1*-24

- Hangman* program, 292
- hardware, 4
- hardware interface, 1*-8
- hardware stack, 533, 534, 550
- HAS-A relationship, 57
- hash function, 612
- hash table, 485, 606, 612
 - chaining, 613
 - collisions, 612
 - load factor, 615
 - probing, 613
 - traversal, 615
- hashCode method, 485, 616, 617
- hashing, 612
- HashMap class, 606, 615
- HashSet class, 484, 606, 615
 - vs. TreeSet, 486
- hasNext method of Iterator, 470
- hasPrevious method of
 - ListIterator, 473
- heap, 483, 627, 630
- HeapPriorityQueue class, 630, 631
- Heapsort, 484, 634
- HelloGraphics.java, 32
- HelloGui.java, 30
- HelloWorld.java, 23

- Hex game, 565
- hexadecimal (hex) numbers, 1*-16
- hierarchy of classes, 295, 326
- high-level programming language, 15, 37
- horizontal and vertical boxes, 17*-13
- horizontal strut, 17*-13
- Horstmann, Cay, 299
- host, 1*-21
- HTML (Hypertext Mark-Up Language),
 - 34, 110, 121

- IDE (Integrated Development Environment), 19, 37
- IEEE standard for floating-point numbers,
 - 1*-18
- if-else statement, 160
 - nested, 171
- if-else-if sequence, 170
- IllegalArgumentException, 226,
 - 280, 393
- ImageIcon class, 18*-11
- immutability of strings, 286
- immutable objects, 237, 257, 268
- implements keyword, 319
- import statement, 53, 70
- increment operator ++, 143
- indentation, 119
- index in array, 334
- Index Maker* program, 371
- indexOf methods of String, 273
- IndexOutOfBoundsException, 365
- induction hypothesis, 562
- infix notation, 601
- information hiding, 61, 71, 222
- inheritance, 28, 63, 71, 294
- inheritance hierarchy, 295, 326
- init method of JApplet, 33
- inline class, 18*-3
- inner class, 470, 515
- input stream, 413

- InputStream class, 414
- Insertion Sort, 383, 396, 407, 457
- instance of a class, 47
- instance variables (fields), 51, 56, 70, 249
- instanceof operator, 385
- int data type, 132
- integer division, 139
- Integer wrapper class, 277, 286
 - parseInt method, 280
- integrated development environment (IDE), 19, 37
- interface, 320, 327
 - vs. class, 320
- interface reserved word, 319
- Internet Protocol (IP), 1^{*}-22
- interpreter, 16, 37
- intValue method of Integer, 277
- IOException, 417, 419, 421
- IS-A relationship, 67, 71, 294
 - with interfaces, 323
- isAltDown method, 18^{*}-5
- ISBN check digit, 291
- isControlDown method, 18^{*}-5
- isDigit method of Character, 282
- isEmpty method of Collection, 468
- isLetter method of Character, 282
- isLetterOrDigit method of Character, 282
- isLowerCase method of Character, 282
- isShiftDown method, 18^{*}-5
- isUpperCase method of Character, 282
- isWhitespace method of Character, 282
- Iterable interface, 514, 593
- iterations, 83, 86, 200
 - and arrays, 348
- Iterator interface, 469
 - methods, 470
 - iterator method of Collection, 468
 - iterators: in nested loops, 471
- jar file, 53
- Java Development Kit (JDK), 19, 21
- Java Virtual Machine (JVM), 17, 37
- java.awt package, 54
- java.io package, 414
- java.lang package, 133, 277, 514
- java.util package, 466
 - Scanner class, 26, 416
- Javadoc, 110
- javax.swing package, 30, 434, 17^{*}-2
- JButton class, 435, 17^{*}-3, 17^{*}-6
- JCheckBox class, 435, 17^{*}-3, 17^{*}-6
- JCheckBoxMenuItem class, 17^{*}-13
- JColorChooser class, 16^{*}-10, 18^{*}-13
- JComboBox class, 435, 17^{*}-3, 17^{*}-6
- JDK (Java Development Kit), 21
- JFileChooser class, 415
- JFrame class, 30
- JIT compiler. *See* Just-In-Time compiler
- JLabel class, 435, 17^{*}-3, 17^{*}-6
- JMenu, JMenuBar, and JMenuItem classes, 17^{*}-3, 17^{*}-6, 17^{*}-13
- JPasswordField class, 435, 17^{*}-3, 17^{*}-6
- jpg file, 18^{*}-9
- JRadioButton, 435, 17^{*}-3, 17^{*}-6
- JRadioButtonMenuItem, 17^{*}-13
- JSlider class, 435, 17^{*}-3, 17^{*}-6
- JTextArea class, 435, 16^{*}-13, 17^{*}-3, 17^{*}-6
- JTextField class, 435, 16^{*}-13, 17^{*}-3, 17^{*}-6
- JToggleButton class, 435, 438, 17^{*}-3, 17^{*}-6
- Just-In-Time compiler, 18
- JVM (Java Virtual Machine), 17

- KB. *See* kilobyte
- keyboard focus, 442, 18*-2, 18*-4
- KeyEvent class, 18*-4
- KeyListener interface, 442, 17*-6, 18*-2
- Keypad* program, 439
- keySet method of Map, 488
- kilobyte, 1*-6, 1*-24
- Knapsack* problem, 576
- LAN. *See* local area network
- large integer, 358
- lastIndexOf methods of String, 273
- layout manager, 17*-10, 17*-17
- layout types, 17*-11
- leaves (of a tree), 578
- left and right subtrees, 586
- length field in an array, 337
- length method of String, 270
- LIFO (Last-In-First-Out), 478
- linear growth, 449
- LineCruiser* program, 531
- linked list, 446, 508
 - circular, 518
 - compared to array, 475
 - doubly-linked, 518
 - singly-linked, 508
 - traversal, 476
 - with a header node, 520
 - with a tail, 517, 522
- LinkedList methods, 477
- lipogram, 282
- list data structure, 92
- List interface, 472
- list traversal, 92, 473
- ListIterator interface, 473
 - methods, 473
- ListNode class, 509
- literal constant, 134
- literal string, 134, 266, 286
- load factor, 615
- local area network (LAN), 1*-8
- local variables, 128, 129, 149, 534
 - default values (no), 130
 - in a nested block, 138
 - scope, 137
 - vs. fields, 132
- Location class in GridWorld, 345
- logarithmic growth, 449, 451
- logic errors, 19
- logical expressions, 159
- logical operators &&, ||, and !, 164
- long data type, 132
- look and feel, 17*-3
- lookup table, 606
- loop invariant, 87
- loops, 200
 - do-while, 204
 - for, 202
 - “for each”, 349, 470
 - nested, 207
 - while, 200
- Lucas, François Edouard Anatole, 565
- machine code, 15
- Mad Libs* program, 426, 440
- magic square, 9
- main method, 24, 30, 55, 252
- mainframe, 1*-8, 1*-9
- mantissa, 1*-18
- Map interface, 487
 - methods, 488
- Math class, 253
 - abs method, 253
 - E constant, 253
 - max method, 253
 - min method, 253
 - PI constant, 253
 - pow method, 141, 253
 - random method, 253, 339
 - round method, 263
 - sqrt method, 253
- mathematical induction, 562

- max-heap, 627
- MB. *See* megabyte
- megabyte, 1^{*}-6, 1^{*}-24
- megahertz, 1^{*}-5
- memory, 1^{*}-14
- Mergesort, 89, 383, 397, 407, 458, 464
- merging sorted arrays, 399
- merging sorted lists, 477
- Mersenne primes, 210
- method, 28, 37, 51, 60, 70, 71, 229, 256
 - accessor, 223
 - arguments, 256
 - array passed to, 338
 - body, 232
 - calling, 230
 - class (static), 250
 - defining, 230
 - modifier/mutator, 223
 - naming, 230, 247
 - overloaded, 247, 257
 - parameters, 229, 235, 236
 - private, 222, 256
 - public, 222, 256
 - recursive, 397
 - return statement, 257
 - return type, 230, 238
 - signature, 230
 - static, 250
 - syntax for calling, 232
 - void, 230, 256
- min-heap, 627, 635
- model-view-controller (MVC) design
 - pattern, 347, 641, 27^{*}-3, 27^{*}-17, 27^{*}-24
- modifier keys, 18^{*}-5
- modifier method, 223
- modulo division operator %, 139
- Monospaced font, 16^{*}-13
- Morse, Samuel, 593
- motherboard, 3, 1^{*}-8
- mouse coordinates, 442, 18^{*}-2
- MouseListener interface, 442, 16^{*}-15, 18^{*}-2
- MouseEvent interface, 18^{*}-3
- multi-dimensional arrays, 342
- multiple inheritance, 296
- mutator method, 223
- names of methods, 230
- natural ordering, 386
- negative numbers, representation, 1^{*}-17
- nested blocks, 119, 122
- nested “for each” loops, 471
- nested for loops, 341
- nested if-else statements, 171
- nested loops, 207
- network adapter, 1^{*}-21
- network protocol, 1^{*}-21
- new operator, 57, 71, 127, 226, 256
 - for arrays, 336
- newline character, 135, 266
- next method of
 - Iterator, 470
 - Scanner, 418
- nextDouble method of Scanner, 418
- nextInt method of Scanner, 418
- nextLine method of Scanner, 418
- Nim game, 8
- no-args constructor, 58, 70, 225
- node of a linked list, 475, 508
- NoSuchElementException, 530
- NOT gate, 2
- notifyObservers method of
 - Observable, 27^{*}-20
- null, 59, 111, 129, 224, 228, 268, 336, 365, 508
- NullPointerException, 131, 228, 268, 274
- NumberFormatException, 280

- O(...)* notation, 390
- object, 45
- Object class, 64, 301
- object module, 16
- object-oriented design (OOD), 48
- object-oriented programming (OOP), 28, 37, 44
- Observable class, 646, 27*-20
- Observer interface, 646, 27*-20
- operating system, 412, 1*-6, 1*-7, 1*-9
- operations research, 446
- operators == and != vs. equals method, 163
- OR gate, 2
- order of growth, 450
- order of operators, 166
- output stream, 413
- OutputStream class, 414
- overloaded methods, 247, 257
- overriding (redefining) a method, 64, 71, 113, 231, 294, 298, 301, 306

- Paint interface, 429, 16*-3
- paint method, 33, 16*-4
- paintComponent method, 16*-5
- palindrome, 290
- parameters (arguments), 58, 130
 - of a method, 229
 - of primitive data types, 236
 - passed as references, 237, 257
 - passed by reference, 236
 - passed by value, 237, 257
- parseDouble method of Double, 281
- parseInt method of Integer, 280
- partitioning in Quicksort, 400, 458
- partitions, 380
- Pascal programming language, 15
- Pascal's Triangle, 356
- passing parameters to methods, 235, 257
 - by value vs. by reference, 236

- path environment variable, 22
- pathname, 412, 414
- pattern recognition, 382
- PC board. *See* printed circuit board
- peek method of PriorityQueue, 483
- perfect number, 210
- perfect tree, 628
- peripheral devices, 4
- permutations, 557
- permutations method, 563
- petabyte, 1*-6
- PI constant in Math, 253
- pivot in Quicksort, 400, 458
- pixel, 428, 1*-15, 16*-2
- PLAF (Pluggable Look and Feel), 17*-3, 17*-17
- play method of AudioClip, 18*-9
- Polish notation, 602
- Poll program, 147
- polymorphism, 97, 232, 316, 322, 326, 429, 16*-3
 - for interfaces, 322
- polynomial growth, 451, 454
- pop method of Stack, 479
- postfix notation, 602
- pow method of Math, 253
- prefix notation, 602
- preorder traversal, 585
- previous method of ListIterator, 473
- primitive data types, 127, 132
- printed circuit board, 3
- printf method, 279
- PrintWriter class, 419, 421, 422
- priority queue, 482, 496, 626, 635
- PriorityQueue class, 627
 - methods, 630
- private, 222, 256
 - class, 17*-8
 - constructor, 224
 - field, 256
 - method, 256

- private inner class, 470, 515
- probing (in hash table), 613
- procedural programming, 27
- programming style, 108
- promotion of operands, 140
 - int to double, 140
- prompt, 26, 1*-10
- pseudocode, 81
- pseudo-random numbers, 403
- public, 222, 256
 - field, 256
 - method, 256
- punch cards, 12
- push and pop CPU instructions, 534
- push method of `Stack`, 479
- put method of `Map`, 488
- Puzzle* program, 429, 16*-3, 16*-14
- Pythagorean triple, 354
- Python programming language, 17, 37, 80, 84, 89
- QA (Quality Assurance), 493
- quadratic growth, 451
- “quality without a name”, 641, 27*-24
- Queue interface, 481
 - add and remove, 528
 - methods, 481
- queues 480, 528
 - and events handling, 480
 - as a ring buffer, 536
- Quicksort, 383, 400, 458, 482, 526
- Radix Sort, 455, 621
- Rainbow* program, 155
- RAM, *See* random-access memory
- Ramblecs* program, 421, 17*-14
- random access, 377, 475
- Random class, 403
 - nextDouble method, 404
 - random method of `Math`, 253, 339
 - random numbers, 403
 - random-access file, 412, 413
 - random-access memory (RAM), 4, 1*-6, 1*-23
 - ranks of operators, 138
 - Reader class, 414
 - reading a text file, 416
 - character by character, 419
 - read-only memory (ROM), 1*-6, 1*-9
 - recursion, 87, 88, 289, 550
 - base case, 88
 - recursive
 - algorithm, 106, 397
 - helper method, 380, 588
 - method, 397, 533
 - procedure, 88
 - structure, 88
 - redundancy, 115
 - reference, 127, 228
 - initializing, 228
 - regular expression, 373, 424, 444
 - reheap down / reheap up, 630
 - relational operators ==, !=, <, <=, >, >=, 162
 - relative pathname, 415
 - remove method of
 - `ArrayList`, 366
 - `Collection`, 468
 - `Iterator`, 470
 - `Map`, 488
 - `PriorityQueue`, 483
 - `Set`, 485
 - removeFirst method of `LinkedList`, 477
 - removeLast method of `LinkedList`, 477
 - repaint method, 16*-5, 16*-6
 - replace method of `String`, 276
 - requestFocus method, 18*-4
 - reserved words, 111, 121
 - return address, 230

- return statement, 238, 240, 257
 - in loops, 206
 - in void method, 240, 257
- return type of a method, 230, 238
 - reference, 239
- return value, 60
- reusable software, 20
- Reverse Polish Notation (RPN), 602
- ring buffer, 536
- ROM. *See* read-only memory
- rounding a double value, 141
- router, 1^{*}-21

- SafeTrade* program, 492
- SansSerif font, 16^{*}-13
- Scanner class, 26, 416
 - methods, 418
- scope, 137
 - of a field, 137
 - of a local variable, 137
 - of a variable, 131, 137
- search engine, 617, 1^{*}-23, 1^{*}-24
- searching, 382
- Selection Sort, 94, 383, 394, 407, 457
- Sequential Search, 93, 389, 407, 448
- Serif font, 16^{*}-13
- Set interface, 484
- set method of ArrayList and LinkedList, 476
- setActionCommand method, 17^{*}-10
- setBackground method, 16^{*}-10
- setBorder method, 438
- setCharAt method of StringBuffer, 270, 285
- setClip method of Graphics, 16^{*}-8
- setColor method, 16^{*}-10
- setFirst method of LinkedList, 477
- setFont method, 16^{*}-13, 16^{*}-14
- setLast method of LinkedList, 477
- setLayout method, 17^{*}-11
- setPreferredSize method, 439, 17^{*}-12
- setResizable method of JFrame, 16^{*}-8
- setting the path, 24
- Shape interface, 428, 16^{*}-3
- Shift key, 18^{*}-5
- short data type, 132
- short-circuit evaluation, 167, 18^{*}-6
- signature of a method, 230
- Singleton design pattern, 641, 27^{*}-3, 27^{*}-7, 27^{*}-23
- singly-linked list, 508
- SinglyLinkedList class, 512
- size method of
 - ArrayList, 366
 - Collection, 468
- Smalltalk, 28, 537
- SMTP (Simple Mail Transfer Protocol), 1^{*}-22
- Snack Bar* program, 241
- software, 4, 1^{*}-9
 - reusability, 20
- software engineering, 4
- software environment, 1^{*}-9
- software package, 21
- sort method of Arrays, 404
- sorting, 94, 382, 455
 - algorithms summary, 407
 - by counting, 457
- source code, 15
- split method of String, 619
- sqrt method of Math, 253
- stack, 478
 - and hardware interrupts, 534
 - and nested structures, 478, 528
 - implemented using a list, 528
 - methods, 479
 - push and pop, 478, 528
 - using ArrayList, 529
 - using LinkedList, 529
- Stack class, 478
 - methods, 479
- stack pointer, 530, 533, 534
- standard input stream, 413

- standard output stream, 413
- startsWith method of `String`, 289
- state machine, 16*-16
- static, 55, 249
 - fields, 249, 250, 257
 - methods, 250, 257
 - syntax for calling, 233
- stepwise refinement, 188
- Strategy design pattern, 347, 641, 27*-3, 27*-5, 27*-23
- strategy stealing, 343
- stream, 412
- string
 - comparisons, 274
 - concatenation, 272
 - conversions, 276
- `String` class, 133, 267
 - + and += operators, 272
 - charAt method, 270
 - compareTo method, 275
 - concat method, 272
 - constructor from `char[]`, 426
 - constructors, 267
 - endsWith method, 289
 - equals method, 274
 - equalsIgnoreCase method, 274
 - format static method, 280
 - hashCode method, 616
 - immutability, 268
 - indexOf methods, 273
 - lastIndexOf methods, 273
 - length method, 270
 - methods, 270
 - replace method, 276
 - split method, 619
 - startsWith method, 289
 - substring methods, 272
 - toCharArray method, 426
 - toLowerCase method, 276
 - toUpperCase method, 276
 - trim method, 276
 - valueOf method, 278
- `StringBuffer` class, 270, 284, 286, 287, 557
 - append and insert methods, 285
 - capacity, 284
 - constructors, 284
 - methods, 285
- `StringIndexOutOfBoundsException`, 280
- `Stroke` interface, 428, 16*-3
- stub class, 187, 498
- stub method, 611
- style in programs, 108
- style in programs, 15
- subclass, 29, 48, 64, 71, 294
- subinterface, 321
- subscript in array, 334
- substring methods of `String`, 272
- subtree, 581, 586
- super keyword, 302
- superclass, 29, 48, 64, 71, 294
 - calling a constructor of, 302
 - calling a method of, 306
- super-dot prefix, 306
- superinterface, 321
- Swing* package, 30, 434, 17*-2
- switch statement, 182, 191
- symbolic constants, 134, 135, 149
 - reasons for using, 136
- syntax, 12, 15, 37, 108, 115, 11*-12
 - errors, 19, 116, B-1
- `System` class, 254
- system stack, 137, 230
- `System.currentTimeMillis` method, 254
- `System.exit` method, 254
- tail recursion, 561
- taxonomy, 295
- TCP. *See* Transmission Control Protocol
- TCP/IP, 1*-22, 1*-24

- Teletext* program, 521
- terabyte, 1★-6
- text file, 413
 - reading, 416
 - writing, 419
- this reference, 227, 233, 234, 235
 - not in a static method, 251
- Three Kittens* program, 538
- throwing an exception, 417
- throws keyword, 417
- time sharing, 1★-9
- Timer class, 35
- toCharArray method of String, 426
- toLowerCase method of Character, 282
- toLowerCase method of String, 276
- top-down programming, 188
- toString method, 146, 231, 256
 - of Arrays, 405
 - of Object overriding, 146
- total ordering, 455
- toUpperCase method
 - of Character, 282
 - of String, 276
- Tower of Hanoi puzzle, 564
- transistor, 2
- translate method of Graphics, 16★-7
- traversal, 348
 - of a hash table, 615
 - of a singly-linked list, 514
 - of a tree, 582
- tree, 578
 - and recursion, 581
 - binary. *See* binary tree
 - traversal, 582
- TreeMap class, 487
- TreeNode class, 583
- TreeSet class, 484
- TreeSet vs. HashSet, 486
- trim method of String, 276
- try-catch statement, 280, 417
- Turing, Alan, 609
- two-dimensional array, 340
 - number of rows and columns, 340
- types of the operands in expressions, 139
- UIManager class, 17★-4
- UML (*Unified Modeling Language*), 57
- unary operators, 167
- Unicode, 414, 1★-19, 1★-24
 - for letters, 275
- unreachable code, 240
- UnsupportedOperationException, 298, 515
- update method of Observer, 27★-21
- valueOf method of String, 278
- variables, 81, 126, 149
- vertical strut, 17★-13
- video adapter, 1★-8
- video memory (VRAM), 428, 1★-8, 16★-2
- virtual code, 18★-4
- virtual key, 18★-4
- VM (virtual machine), 442, 18★-2
- void method, 61, 71, 230
- von Neumann, John, 2, 12
- while loop, 200, 211
- whitespace, 240, 276, 282, 418, 444
- wrapper class, 277, 296, 27★-10
- Writer class, 414
- writing to a text file, 419
- XOR circuit, 2
- XOR operation, 192

