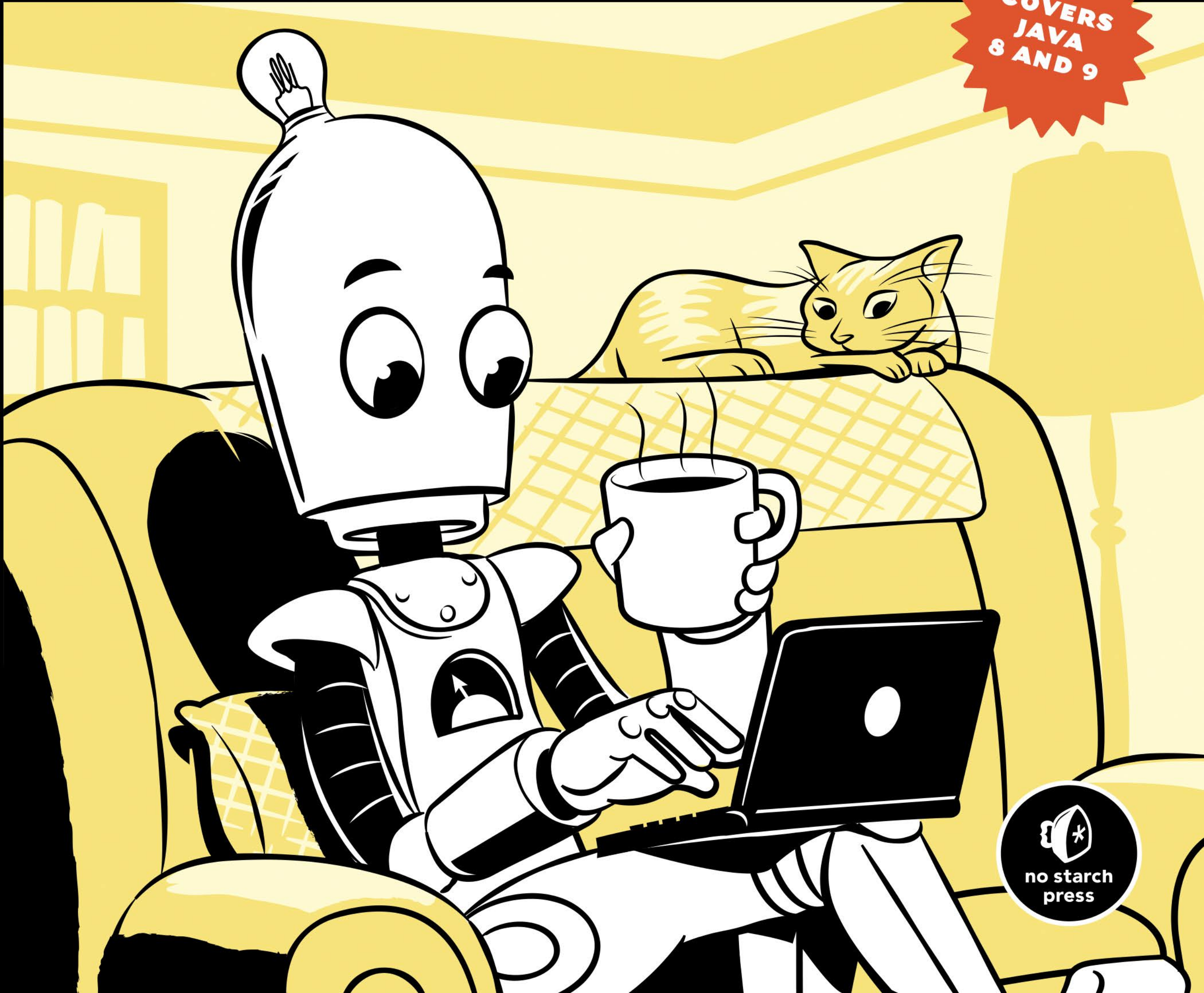
LEARN JAVA THE EASY WAY

A *HANDS-ON* INTRODUCTION TO PROGRAMMING

BRYSON PAYNE

COVERS JAVA 8 AND 9



no starch press

INDEX

Symbols

+ (addition operator), 12
= (assignment operator), 13–15, 245, 276
{ } (braces), 23
+ (concatenation operator), 12–13, 27, 30, 130–131
-- (decrement operator), 130
. (dot operator), 59, 68
" (double quotation marks), 27
== (equal-to operator), 30, 276
// (forward slashes), 23, 28
> (greater-than operator), 30, 60
++ (increment operator), 129
< (less-than operator), 30, 60
% (modulo operator), 139
!= (not-equal-to operator), 29
; (semicolon), 24
~ (tilde), 16
* (wildcard character), 46

A

About screen, 113–114
AbsoluteLayout value, 240
ActionListener class, 63–64, 226
action listeners, 46–47, 68, 234
actionPerformed() method, 63, 154–156, 227–228, 234
addActionListener() method, 63
addMouseListener() method, 213
addition operator (+), 12
AlertDialog class, 113–114
alert dialog pop-up, 113–114
Android devices
  connecting to computer, 101
  Developer mode, 100–101
  preparing to run apps, 100
  testing apps on, 100–102
Android emulator
  BubbleDraw app, 260–261
  Hi-Lo guessing game, 94–99
  Secret Messages app, 188–189
Android package files (APK), 101–102
Android Studio, 1. *See also* BubbleDraw app (Android mobile version); Hi-Lo guessing game (Android mobile version); Secret Messages app (Android mobile version)
  activity selection, 78–79
  Android emulator, 94–99
  code completion, 91–92, 278
  correcting spelling errors, 275
  default project view, 80
  Design tab, 80–81
  grouping symbols, 277–278
  importing classes automatically, 86
  installing, 8–9
  SDK or API levels, 78
  starting projects, 76–80
  Target Android Devices window, 78
  updating, 176
  widgets, 81
animation, 225–229. *See also* BubbleDraw app (Android mobile version); BubbleDraw app (GUI version)
  adding timer, 226
  frame rate, 225, 241, 253
  preparing, 227–228
  setting timer, 226–227
  starting timer, 229
  threading, 249
anonymous inner classes, 63, 91, 155, 167, 184, 210
APK (Android package files), 101–102
application program interface (API) levels, 78
apply() method, 119
arguments, 155
arithmetic operators (+, -, *, /), 12
ArrayList data structure, 205–210, 253

assignment operator (=), 13–15, 245, 276
attributes, 26

B

binary search, 40
bin folder, 1, 11–12, 141–143
Boolean expressions, 65
braces ({ }), 23
break statement, 113
Bubble class, 200–205, 255–257
- attributes for, 201
- defining bubbles, 200–202
- methods for, 202–205
- modifying, 255–257

BubbleDraw app (Android mobile version)
- adding animation variables, 252–253
- adding testBubbles() method, 258–259
- BubbleView() constructor, 254
- custom app launcher icon, 268–270
  - adding, 268–269
  - changing app name, 270
  - creating, 268
  - displaying, 269–270
- drawing with onDraw() method, 257–258
- handling OnTouchListener events, 259–260, 264-265
- modifying Bubble class, 255–257
- running in Android emulator, 260–261
- setting up project, 250–252
- testing, 258–261, 266–267
- threading and multitasking, 261–264
- touch event handler code, 264–265
  - combining single-pointer and multitouch events, 271–272
  - multitouch, 266–267

BubbleDraw app (desktop version)
- building frame, 199
- creating Bubble class, 200–205
  - attributes for, 201
  - defining bubbles, 200–202
  - methods for, 202–205
- creating project files, 198–199
- mouse events, 210–217
  - clicks, 212–214
  - creating reusable event listener, 211–212
  - drags, 214–215
  - wheel events, 215–217
- pixelated effect, 219–220
- size variable limits, 218
- storing bubbles in ArrayList, 205–210
  - adding constructor, 206–207
  - adding method, 207–208
- testing BubblePanel class, 208–210

BubbleDraw app (GUI version)
- adding transparency, 224–225
- animation, 225–229
  - adding timer, 226
  - frame rate, 225, 241
  - preparing, 227–228
  - setting timer, 226–227
  - starting timer, 229
- building GUI, 232–235
  - event handlers, 234–235
  - setting up panel and buttons, 232–233
- collision detection, 235–239
  - hard bounces, 238–239
  - soft bounces, 236–238
- copying project and renaming Java file, 222–223
- fixed speed, 244–245
- pixelated effect, 245–247
- preventing stationary bubbles, 244
- random speed and direction, 229–232
- sharing app as runnable JAR file, 243
- speed control slider, 240–243
  - customizing, 241
  - event handler, 242–243

BubbleDraw class, 199
BubbleListener class, 212–214, 227
Bubble() method, 202–204
BubblePanel class, 199, 201, 203, 206–209, 214–215
BubbleView class, 251–255
BubbleView() constructor, 254
Button component, 82, 84, 87, 180
buttons
- adding, 82
- adding event listeners, 62–64, 91–94

  - adding labeled, 46–47, 54
  - Clear button, 232–235
  - Click me! button, 46–47
  - Encode/Decode button, 152, 154–156, 159–164, 167, 179–180, 182–186
  - floating action button (fab), 84–85, 190–193
  - Guess! button, 54–55, 62–63, 66–69, 82, 93, 99
  - Move Up ^ button, 172–173, 196
  - OK button, 114
  - Pause/Start button, 232–235
  - Play Again button, 66, 73–74
  - showing and hiding, 73–74

C

- Caesar cipher, 125–126
- camel case, 14, 21
- Canvas class, 256
- case sensitivity, 14, 21, 274
- casts (type casts), 24, 132
- chained assignment, 245
- charAt() method, 130–132
- char data type, 132
- checkGuess() method, 58–59, 65, 68–69, 73, 88–89, 106, 116, 120
- child classes, 49
- classes, 21, 86–87
  - anonymous inner classes, 63, 91, 155, 167, 184, 210
  - creating, 21–23, 200–204
  - extending, 49
  - importing, 25
  - public, 23
  - superclasses, 48
- Clear button, 232–235
- Click me! button, 46–47
- close() method, 37, 135
- code completion, 91–92, 278
- code recommenders, 59, 274–275
- collision detection, 235–239
  - hard bounces, 238–239
  - soft bounces, 236–238
- Color Chooser dialog (in Eclipse), 160, 165
- Color class, 201, 203, 224, 255–256
- command line apps, running without Eclipse, 141–144
  - opening command line window, 142–144
  - workspace folders, 141–142
- comments, 23, 28
- comparison operators
  - equal to (==), 30, 276
  - greater than (>), 30, 60
  - less than (<), 30, 60
  - not equal to (!=), 29
- compiler directives, 91
- concatenation operator (+), 12–13, 27, 30, 130–131
- conditional expressions, 30
- conditions, 28
- confirm dialogs, 163
- content assist, 59, 274–275
- content panes, 52

D

- data structures, 205
- debugging, 28
  - case sensitivity, 274
  - code completion, 278
  - commenting out, 28
  - grouping symbols, 276–278
  - spelling errors, 274–276
  - statement completion, 278
  - syntax coloring, 273
  - USB debugging, 100–102, 188, 267
- decrement operator (--), 130
- defaultCloseOperation property, 51–52
- Dimension class, 64–65
- dismiss() method, 114
- dot notation, 26
- dot operator (.), 59, 68
- double quotation marks ("), 27
- double variable type (double-precision floating point), 14
- do-while loops, 33–35
- draw() method, 204–205, 247, 256
- drawOval() method, 250, 256–257
- drawRect() method, 204

E

- Eclipse, 1. *See also* BubbleDraw app (GUI version); Hi-Lo guessing game (GUI version); Secret Messages app (GUI version)
  - automatic indentation clean-up, 35
  - content assist and code recommenders, 59, 274–275

Eclipse (*continued*)
- correcting spelling errors, 274–275
- creating classes, 21–23
- creating new Java project, 20–21
- customizing appearance of, 7–8
- grouping symbols, 277
- installing, 3–4
- perspectives, 21
- saving code, 25
- setting up
  - default workspace storage location, 4–5
  - installing WindowBuilder Editor, 6–7
  - modifying to run with Java 9, 5
- setting up GUI projects, 48–49
- themes, 7
- WindowBuilder Editor, 50
  - Design tab, 50
  - installing, 6–7
  - Palette pane, 52–54
  - Properties pane, 51–52
  - Source tab, 56
- workspace folders, 4–5, 141–142

/edit command, 16
editor action, 103
EditText component, 82, 84, 87, 102, 107, 178, 185
else-if statements, 138–139
else statements, 31
encapsulation, 201
Encode/Decode button, 152, 154–156, 159–164, 167, 179–180, 182–186
encode() method, 151–155, 182–185
equalsIgnoreCase() method, 34
equals() method, 34, 276
equal-to operator (==), 30, 276
escape character sequence (\n), 107, 178
event listeners, 62, 117. *See also* mouse events; touch events
- for buttons, 62–64, 91–94
- creating reusable, 211–212
- for ENTER key, 103–104

exception handlers, 70–72
exceptions, 70
/exit command, 16
expressions, evaluating in JShell, 12–13
extending classes, 49

F

fab (floating action button) icon, 84–85, 190–193
fillOval() method, 204–205
fillRect() method, 220, 247
finally block, 71–72
findViewById() method, 87
floating action button (fab) icon, 84–85, 190–193
floating-point numbers, 14
FlowLayout value, 240
focus, 69
Font Chooser dialog (in Eclipse), 159
for-each loops, 228, 263
for loops, 129–134
forward slashes (//), 23, 28

G

game loops (do-while loops), 33–35
getHeight() method, 236–238
getInt() method, 120
getMenuInflator() method, 111
getPointerCount() method, 266, 271
getProgress() method, 186
getSource() method, 235
getStringExtra() method, 194
getText() method, 59, 155, 185
getUnitsToScroll() method, 216
getValue() method, 167–168, 243
getWidth() method, 236–238
Graphics class, 204
greater-than operator (>), 30, 60
grouping symbols, 276–278
Guess! button, 54–55, 62–63, 66–69, 82, 93, 99

H

Handler class, 253, 262
"Hello, Java!" test code, 9
/help command, 16
HiLo class, 21–23
Hi-Lo guessing game (Android mobile version)
- adding method to begin new game, 90–91
- adding method to check player's guess, 88–89
- building GUI layout, 80–83

- changing guessing range, 114–117
  - adding variable, 115
  - dialog to allow user to select range, 116–117
  - modifying behavior of app to use variable, 115–116
- changing number of tries, 122–123
- event listeners, 91–94
- options menu, 109–114
  - adding items to XML file, 110–111
  - alert dialog pop-up for About screen, 113–114
  - displaying, 111–112
  - responding to user selections, 112–113
- ratio of wins to losses, 123
- running in Android emulator, 94–99
- running on real Android device, 100–102
- setting up project, 76–80
- storing game stats, 120–122
- storing user preferences, 118–120
- Toast messages, 106
- UX improvements, 102–105
  - aligning text field, 104
  - centering input in text field, 103
  - event listener for ENTER key, 103–104
- visual improvements, 107
- wiring up, 83–88

Hi-Lo guessing game (command line version)
- completed code for, 37–38
- counting tries, 40–41
- creating `HiLo` class, 21–23
- creating new Java project, 20–21
- generating random numbers, 23–25
- getting keyboard input, 25–27
- increasing range of numbers, 40
- loops, 28–35
  - `if` statements, 30–32
  - play again feature, 33–35
- making program print output, 27–28
- overview of, 19–20
- planning basic version of, 20
- prompting for user input, 26
- testing, 36–37
- thank-you for playing, 36–37

Hi-Lo guessing game (GUI version)
- adding method to check player's guess, 58
  - converting strings to numbers, 60–61
  - getting text from text field, 59–60
- designing GUI, 50–58
  - aligning elements, 54–55
  - customizing components, 52–54
  - setting properties, 51–52
  - wiring up, 55–58
- event listeners, 62–64
- handling bad input, 69–72
- new game (picking new random number), 61–62
- play again feature, 66–67, 73–74
- setting up GUI window, 64–65
- setting up project, 48–49
- showing and hiding Play Again button, 73–74
- showing number of tries, 73
- testing, 66
- UX improvements, 67–70
  - automatically removing old guesses, 68–69
  - pressing ENTER to guess, 68

`/history` command, 16

I

`if-else` statements, 31, 135

`if` statements, 30–32, 134

image editors, 268

`ImageView` class, 251, 253

`/imports` command, 16

increment operator (`++`), 129

indentation, 30, 35, 277

indexes, 129

`inflate()` method, 111

inheritance, 48

input dialogs, 163

`Integer` class, 60

`int` variable type (integer), 13

`invalidate()` method, 259, 263

`isSpace()` method, 135

`<item>` tag, 111

iteration, 128

J

JAR (Java archive) files, 169–171, 243
Java, 1
  as first programming language, xx
  modifying Eclipse to run with Java 9, 5
  as object-oriented programming language, 21
  printing output in, 15
  testing setup using JShell, 9–17
Java archive (JAR) files, 169–171, 243
Java Development Kit (JDK), 2–3, 10
Java Perspective (in Eclipse), 21
Java Runtime Environment (JRE) software, 1–2, 141
Java Virtual Machine (JVM) software, 1–2, 141
JButton component, 46, 52, 54, 63, 74, 151, 232
JCheckBox component, 52
JDK (Java Development Kit), 2–3, 10
JFrame class, 44, 48–49, 199, 223
JLabel component, 52–54, 57, 61, 74, 151, 240
JOptionPane component, 164
JPanel component, 46, 198–199, 207–208, 232, 240
JRE (Java Runtime Environment) software, 1–2, 141
JScrollPane component, 173–174
JShell, 9
  commands, 16–17
  creating GUIs in, 44–47
  declaring Java variables in, 13–15
  evaluating Java expressions in, 12–13
  "Hello, Java!" test code, 9
  opening code in, 16
  printing output, 15
  running, 9–11
    from command line, 10–11
    confirming installed version of JDK, 10
    from shortcut, 11–12
  saving code created in, 16
  starting new code snippets, 17
  testing Java setup, 9–17
JShell Edit Pad, 46–47
JSlider component, 164–167
JTextArea component, 150–152
  scroll bars for, 173–174
  wrapping text, 162
JTextField component, 52–53, 56–57, 59–60, 68–69, 151–152
JVM (Java Virtual Machine) software, 1–2, 141

K

key/value pairs, 118

L

labels
  adding, 52–54, 81–82
  alignment, 53
  editing, 53, 81–82
  font, 53
  naming for coding, 56
length() method, 129–131
less-than operator (<), 30, 60
lineWrap property, 162
/list command, 16
logarithms, 123
log() method, 123
loops, 28–35, 144–145
  do-while loops, 33–35
  for-each loops, 228, 263
  for loops, 129–134
  nested, 38
  while loops, 28–30

M

MadLibs class, 41
MadLibs game
  Android mobile version, 107
  command line version, 41
  GUI version, 74
main() method
  BubbleDraw app
    desktop version, 210
    GUI version, 222–223
  Hi-Lo guessing game
    command line version, 22–23, 25, 29, 35
    GUI version, 48, 64–65
  Secret Messages app
    command line version, 127–128, 135–136
    GUI version, 148, 156–157
majorTickSpacing property, 165, 241
makeText() method, 106
manifest file, 193

Math class, 23
math expressions, 12
MenuInflator class, 111
<menu> tag, 110–111
message dialogs, 163–164
methods, 22, 200–210
  designing, 202–205, 207–208
  displaying available methods with content assist, 68–69
method signature, 207
minorTickSpacing property, 165, 241
modulo operator (%), 139
MouseAdapter class, 212
mouseClicked() method, 212
mouseDragged() method, 214–215
MouseEvent class, 212–213
mouse events, 210–217
  clicks, 212–214
  creating reusable event listener, 211–212
  drags, 214–215
  wheel events, 215–217
mouseMoved() method, 214
mousePressed() method, 212–214
mouseReleased() method, 212
MouseWheelEvent class, 215–216
mouseWheelMoved() method, 216, 218
Move Up ^ button, 172–173, 196
multitasking, 249, 261–264
multitouch, 249, 266–267

N

\n (escape character sequence), 107, 178
nested loops, 38
newGame() method, 61–62, 64, 66–67, 90, 113–115, 120
New Java Class dialog (in Eclipse), 22–23, 49, 148
New Java Project dialog (in Eclipse), 20–21
newline character, 107, 178
nextInt() method, 25, 27–28, 203–204, 224, 231
nextLine() method, 25, 128
not-equal-to operator (!=), 29
numeric variables, 13–14

O

object-oriented programming, 21, 198
objects, 21
OK button, 114
OnClickListener event handler class, 91–92, 184
onClick() method, 93, 114, 184–185, 191
onCreate() method, 87, 90–91, 115, 120, 182, 191
onCreateOptionsMenu() method, 111
onDraw() method, 257–258, 263
online resources, xxi–xxii
onOptionsItemSelected() method, 112, 116, 118–119, 121
onProgressChanged() method, 188
OnSeekBarChangeListener event handler, 187–188
onStartTrackingTouch() method, 188
onStopTrackingTouch() method, 188
OnTouchListener interface, 251–252, 259–260
onTouch() method, 259–260, 264–265, 271
/open command, 16
operators
  arithmetic (+, -, *, /), 12
  assignment (=), 13–15, 245, 276
  comparison
    equal to (==), 30, 276
    greater than (>), 30, 60
    less than (<), 30, 60
    not equal to (!=), 29
  concatenation, 12–13, 27, 30, 130–131
  decrement (--), 130
  dot (.), 59, 68
  increment (++), 129
  modulo (%), 139
options menu, adding in Android, 109–114
  adding items to XML file, 110–111
  alert dialog pop-up for About screen, 113–114
  displaying, 111–112
  responding to user selections, 112–113

P

package declarations, 85–86
Paint class, 253
paintComponent() method, 207
parameters, 153
parent classes (superclasses), 48
parseInt() method, 155, 185

- parsing, 60
- Pause/Start button, 232–235
- persistent information, storing, 118–121
- pixelated effect, 219–220, 245–247
- Play Again button, 66, 73–74
- pointers, 266
- `postDelayed()` method, 263
- `println()` method, 15, 26–27
- `private` modifier, 57
- procedural programming, 198
- `public` modifier, 58

R

- `Random` class, 203
- `random()` method, 24, 203
- range settings dialog, 116–117
- refactoring, 223
- `RelativeLayout` value, 255
- `repaint()` method, 209, 213, 215, 228, 234
- `requestFocus()` method, 164
- `/reset` command, 16
- resource leaks, 37, 135
- `return` statement, 113
- `run()` method, 262
- `Runnable` interface, 262

S

- `/save` command, 16
- `Scanner` class, 25, 37
- scrolling
  - different operating systems and, 216–217
  - mouse wheel events, 215–217
  - scroll bars, 173–174
- SDK (software development kit) levels, 78
- Secret Messages app (Android mobile version)
  - connecting Encode button to `encode()` method, 182–185
  - customizing fab icon, 190–193
  - designing GUI, 177–181
  - Move Up ^ button, 196
  - receiving secret messages from other apps, 193–195
  - running in Android emulator, 188–189
  - running on real Android device, 188–190
  - `SeekBar` component, 187–188, 196
  - setting up project, 176–177
  - testing, 185–187
  - wiring up, 182–188
- Secret Messages app (command line version)
  - adding loops to, 144–145
  - Caesar cipher, 125–126
  - character values, 132–133
  - closing all input/output resources, 135–136
  - custom key values, 137–138
  - encoding digits, 138–141
  - encoding only letters, 133–135
  - full version of, 140–141
  - languages other than English, 135
  - reversing and encoding, 145
  - setting up project, 126–131
    - asking user for input, 128
    - creating project in Eclipse, 127
    - message reverser, 128–131
    - strings, 128–131
  - `try-catch` blocks, 145
- Secret Messages app (GUI version)
  - building `main()` method, 156–157
  - creating `encode()` method, 151–154
  - designing GUI, 148–151
  - event handling, 154–156
  - full source code, 157–159
  - handling bad input, 156, 162–164
  - improving GUI, 159–169
  - line and word wrap, 161–162
  - Move Up ^ button, 172–173
  - numeric slider
    - adding, 164–166
    - code cracking with, 166, 168–169
    - moving when value in text field changes, 174
  - scroll bars, 173–174
  - setting up project, 148
  - sharing as runnable JAR file, 169–171
  - wiring up, 148–151
- `SecretMessagesGUI` class, 152
- `SeekBar` component, 179–180, 187–188, 196
- `selectAll()` method, 69, 164

- Select Deployment Target dialog (in Android Studio), 98, 198
- semicolon (;), 24
- `setBackground()` method, 207
- `setButton()` method, 114
- `setColor()` method, 204
- `setItems()` method, 117
- `setOnClickListener()` method, 183–184, 191
- `setOnSeekBarChangeListener()` method, 187
- `setOnTouchListener()` method, 265
- `setSize()` method, 64
- `setText()` method, 61, 107, 156, 167
- `setVisible()` method, 45, 65
- shared preferences, 118–121
- `showMessageDialog()` method, 163
- `show()` method, 106
- sliders
  - numeric
    - adding, 164–166
    - code cracking with, 166, 168–169
    - moving when value in text field changes, 174
  - speed control, 240–243
    - customizing, 241
    - event handler, 242–243
- smart completion, 278
- Snackbar pop-up messages, 191
- software development kit (SDK) levels, 78
- spelling errors, 274–276
- *src* (source code) folder, 22, 48, 127, 141, 148
- `start()` method, 229
- `stateChanged()` method, 166–167, 242
- statement completion, 278
- `stop()` method, 229
- `storeRange()` method, 119
- strings, 12
- `String` variable type, 14–15, 33
- stubs, 22
- superclasses (parent classes), 48
- `super()` method, 253
- `switch` statement, 113
- syntax coloring, 273
- `System` class, 26
- `System.in` object, 25
- `System.out` object, 26

T

- temporary variables, 12
- Terminal, 142
- `testBubbles()` method, 208–210, 213, 258–259, 265
- `Text` class, 88
- text fields
  - adding, 82
  - aligning, 104
  - centering input in, 103
  - declaring at top level of class, 56–57
  - getting text from, 59
  - sizing, 83
  - wiring up, 55–56, 83–84
- `TextView` component, 81–82, 84, 87, 104, 107, 114–116, 177
- `Thread` class, 262
- threading, 249, 261–264
- threads, 253
- tilde (~), 16
- `Timer` class, 225–227
- `title` property, 51
- Toast messages, 106
- `toString()` method, 88, 185
- touch events, 251–252, 259–260, 264–267, 271–272
- transparency, 224–225
- `try-catch` statements, 71–72, 145, 156, 162–163
- type casts (casts), 24, 132

U

- undo, 55
- Unicode, 132
- `update()` method, 228, 237–238, 255
- USB debugging in Android, 100–101
- user experience (UX), 67

V

- variables
  - declaring and initializing, 26
  - declaring at top level of class, 56–58
  - declaring in JShell, 13–15
  - temporary, 12
- `/vars` command, 16
- `void` modifier, 58

W

what-you-see-is-what-you-get (WYSIWYG) interfaces, 50
while loops, 28–30
wildcard character (*), 46
WindowBuilder Editor (in Eclipse), 50–52
  Design tab, 50
  installing, 6–7
  Palette pane, 52–54
  Properties pane, 51–52
  Source tab, 56
windows
  close operation, 51–52
  content panes, 52
  creating, 44–45
  sizing, 64
  titles, 51
wrapStyleWord property, 162
WYSIWYG (what-you-see-is-what-you-get) interfaces, 50