



Symbols

A

of an object 11, 12
 automobile as an object 11
 average 105, 108
 average calculation 112

B

backslash (\) 45
 banker's rounding 189
 bar chart 192
 behavior
 of a class 11
 big data 8
 BigDecimal class 116, 164, 186, 186
 add method 188
 multiply method 188
 ONE constant 188
 pow method 188
 rounding policy 186, 187
 setScale method 189
 significant digits 186
 TEN constant 188
 TWO constant 188
 ZERO constant 188
 BigInteger class 128
 add method 131
 divide method 137
 mod method 137
 multiply method 131
 pow method 137
 subtract method 131
 valueOf method 130
 binary digit (bit) 6
 binary integer 136
 binary number system 6
 binary operator 51, 54, 178
 binary search 22
 bit ("binary digit") 6
 Bitcoin 128
 bitwise operators 175
 blank line 40, 112
 block 60, 100
 blockchain 128
 body
 class declaration 41
 of a loop 103
 of a method 42
 of an if statement 57

Bohm, C. 92
 boolean logical AND & 175, 177
 boolean logical exclusive OR ^ 175, 178
 boolean logical inclusive OR | 177
 boolean primitive type 40, 95
 braces ({ and }) 60, 100, 101
 not required 170
 break statement 40, 170, 173, 194
 building block 90
 appearance 183
 building your own compiler case study 9
 byte 6
 byte primitive type 40, 167
 bytecode 14, 43
 bytecode verifier 15

C

calculations 93
 camel case 41
 case keyword 40, 170
 case sensitive 41
 Java commands 18
 cast operator 67, 115
 catch keyword 40
 cd to change directories 43
 char
 keyword 40
 primitive type 50, 167
 character 6
 constant 172
 set 6
 character set 67
 character string 42
 checkerboard pattern 67
 circumference 67
 class 11
 declaration 40
 file 43
 instance variable 12
 name 40
 class-average problem 105, 106, 113

class declaration
 body 41
 .class file 14, 44
 class keyword 40, 40
 class loader 15
 classes
 186
 BigDecimal 116, 164
 BigInteger class (package java.math 128
 Long 130
 Scanner 49
 String 61, 63
 CLASSPATH problem 18
 code 3, 12
 comma (,) formatting flag 163
 comma in an argument list 47
 comma-separated list 158
 of arguments 47
 command line 42
 Command Prompt 42
 command window 17, 42
 comment
 end-of-line (single-line), // 39, 42
 Javadoc 39
 single line 42
 commission 133
 compilation error 40, 43
 compile 43
 compile a program 14
 compiler 10
 component 11
 compound assignment operators 122, 125
 compound interest 161, 192, 193, 194
 Compound-Interest Program: Nested for Loops-Triangle Printing Program
 Nested for Loops 192
 condition 57, 165
 conditional AND && 176, 177
 conditional expression 102

conditional operator `?:`:
101, 126
conditional OR `||` 175, **176**
const keyword 40
constant **172**
Math.PI 67
constant integral expression
167, 172
constant variable **172**
contextual keywords 40
continue statement 40, 173,
174, 175, 194
control statement **91**, 93, 94,
96
for 157
nesting **95**, 184
stacking **95**, 183
control statement stacking
114
control variable **105**, 154,
155, 157
controlling expression of a
switch **170**
core package 44
correct in a mathematical
sense 181
counter **105**, 106, 111, 118
counter-controlled iteration
105, 106, 114, 117,
118, 154, 155, 156
counting loop 155
credit limit on a charge ac-
count 132
cryptocurrency 128
Ctrl key 169
 `<Ctrl>-d` 169
 `<Ctrl>-z` 169
cursor 42, 45
custom precision for for-
matted floating-point
values **115**

D

dangling-else problem **134**
data hierarchy **5**
data mining **8**
database **7**
decimal

digit **6**
decimal integer formatting
51
decision **57**, 96
symbol in the UML **96**
declaration
class **40**
import **49**
method **42**
decrement a control vari-
able **158**
decrement operator `--` **123**
decrypt 199
deeply nested statement 184
default case in a switch
170, 171, 172
default keyword 40
definite iteration **105**
dependent condition **177**
design process **13**
diameter 67
diamond in the UML **93**,
193
decision symbol **96**
merge symbol **103**
digit 50
digit separator (`_`) **50**
display a line of text 42
divide by zero 16, **112**
divide method of the
BigInteger class **137**
division 53, 54
compound assignment
operator, `/=` 123
do...while iteration state-
ment 94
do...while repetition state-
ment 40, **165**, 185
document a program 39
dollar signs (\$) 41, 49
dot (`.`) separator 130, 163
dot operator (`.`) **63**
dotted line in the UML **94**
(double) cast **115**
double-precision floating-
point number **112**
double primitive type 40,
50, 112

double quotes, " 42, 45, 46
double selection 185
double-selection statement
94, 118
driver program **62**

E

Eclipse (www.eclipse.org)
14
Eclipse IDE for Java De-
velopers 3
edit a program 14
editor 13
else keyword 40, 97
emacs 14
employee identification
number 7
empty statement (a semico-
lon, `;`) 60, 101, 158
empty String 63
encapsulation **12**
encryption 198
“end of data entry” 110
end-of-file (EOF)
indicator **169**
end-of-line (single-line)
comment, `//` **39**, 42
endsWith method of class
String 64
Enforcing Privacy with
Cryptography exercise
198
English-like abbreviations 9
Enter (or Return) key 42
entry point of a control
statement 95, 181
enum
keyword 40
environment variable
PATH 14
equality operators **57**
equals method of class
string **64**
escape character **45**
escape sequence **45**, 50
 `\` backslash 46
 `\"` double-quote 46
 `\n` newline 45, 46

- `\t` horizontal tab 46
- execute **16**, 43
- executing an application 18
- execution-time error **16**
- Exercises
 - Another Dangling-`else` Problem 135
 - Arithmetic 66
 - Arithmetic, Smallest and Largest 66
 - Bar-Chart Printing Program 192
 - `BigInteger` class: Other arithmetic operations 137
 - Calculating Sales 192
 - Calculating the Product of Three Integers 66
 - Calculating the Value of π 193
 - Checkerboard Pattern of Asterisks 67, 136
 - Comparing Integers 66
 - Compound-Interest Program (modified) 193
 - Compound-Interest Program: Nested for Loops 192
 - Compound-Interest Program: Solving without `Math.pow` 192
 - Dangling-`else` Problem 134
 - De Morgan's Laws 193
 - Diameter, Circumference and Area of a Circle 67
 - Diamond-Printing Program 193
 - Diamond-Printing Program (modified) 194
 - Displaying Data 66
 - Displaying Integers 191
 - Displaying Shapes with Asterisks 66
 - Exceeding a Credit Limit 132
 - Factorial 137
 - Factorials 191
 - Find the Largest Number 133
 - Find the Smallest Value 191
 - Find the Two Largest Numbers 133
 - Gas Mileage 132
 - Integer Value of a Character 67
 - Largest and Smallest Integers 66
 - Multiples 66
 - Odd or Even 66
 - Palindromes 136
 - Peter Minit problem 194
 - Powers of 2 137
 - Printing the Decimal Equivalent of a Binary Number 136
 - Product of Odd Integers 191
 - Pythagorean Triples 193
 - Removing `break` and `continue` 194
 - Salary Calculator 133
 - Sales Commission Calculator 133
 - Separating the Digits in an Integer 68
 - Sides of a Right Triangle 137
 - Sides of a Triangle 137
 - Square of Asterisks 136
 - Table of Squares and Cubes 68
 - Tabular Output 133
 - "The Twelve Days of Christmas" Song 194
 - Triangle-Printing Program (modified) 192
 - Validating User Input 134
 - What Does This Code Do? 191
 - Write a Line of Code 65
 - exit point 181
 - of a control statement 95
 - exiting a `for` statement 173
 - explicit conversion **115**
 - exponentiation operator 163
 - `exports` keyword 40
 - expression **51**
 - `extends` keyword 40
- F**
 - factorial 137, 191
 - fair rounding algorithm 186
 - `false` keyword **57**, **95**
 - fatal error **101**
 - fatal logic error **101**
 - runtime error **16**
 - fault tolerant 50
 - field **7**, **7**
 - field width **163**
 - file **7**
 - `final`
 - keyword 40, 172
 - final state in the UML **94**, 183
 - final value 155
 - `finally`
 - clause 40
 - first refinement in top-down, stepwise refinement 110, 118
 - fixed text 51
 - in a format string **47**
 - flag value **110**
 - `float`
 - primitive type 40, **50**, 112
 - floating-point literal **113**
 - `double` by default 113
 - floating-point number **108**, 112, 114, 115
 - division 115
 - double precision **112**
 - single precision **112**
 - 67
 - `%f` format specifier **115**
 - floating-point value
 - custom precision for formatting **115**

flow of control **93**, 114
 flow of control in the `if` statement 96
 flow of control in the `if...else` statement 97
 flow of control in the `while` statement 103
 for iteration statement 94, 157
 header **157**
 for repetition statement 40, **156**, 163, 185
 format specifier
 %**f** **115**
 format specifiers **47**
 %.2**f** for floating-point numbers with precision 115
 %**b** **64**
 %**b** for boolean values 179
 %**c** 67
 %**d** **51**
 %**f** 67
 %**n** (line separator) **47**
 %**s** **47**
 format string **47**
 formatted output
 , (comma) formatting flag **163**
 - (minus sign) formatting flag **163**
 boolean values 179
 comma (,) formatting flag **163**
 field width **163**
 grouping separator **163**
 left align **163**
 minus sign (-) formatting flag **163**
 right align **163**
 formatting
 display formatted data 46
 formulating algorithms 105
 fundamental type
 int 123

G
 general class average problem 110
 generative AI 43, 44, 46, 48, 52, 57, 61, 92, 96, 98, 100, 101, 102, 105, 110, 117, 122, 123, 127, 144, 156, 159, 160, 161, 165, 167, 173, 175, 181, 190
 Global Warming Facts Quiz exercise 202
`goto`
 elimination 92
 keyword 40
 statement **92**
`goto` elimination 92
 graph 192
 gross pay 133
 group
 left to right 126
 right to left 126
 group digits for readability 50
 grouping of operators **55**, 125
 grouping separator (formatted output) **163**
 guard condition in the UML 96

H
 half even rounding **189**
 half up rounding **189**
 hardware **3**, 9
 hasNext method
 of class Scanner **170**
 high-level language **10**
 horizontal tab 46
 hot spots in bytecode 16
 HTTPS protocol 128

I
 IDE (integrated development environment) **14**
 Eclipse IDE for Java Developers 3
 JetBrains IntelliJ IDEA 3
 Visual Studio Code 3

identifier **41**, 49
`if` single-selection statement 40, **57**, 94, 96, 185
`if...else` double-selection statement 94, **97**, 97, 114, 185
 implements 40
 implicit conversion **115**
 import declaration 40, **49**
 increment 158
 a control variable 155
 expression 174
 of a `for` statement 158
 increment a control variable **154**
 indentation 97, 99
 infinite loop **103**, 158, 160
 infinite series 193
 information hiding **12**
 inheritance **12**
 initial state 183
 initial state in the UML **93**
 initial value of control variable **154**
 initialization phase **110**
 initialize a variable in a declaration **49**
 input/output operation 93
 instance **11**
 instance main 41
 instance variable **12**
 instanceof operator 40
`int` data type 123
`int` primitive type 40, **50**, 167
 integer **48**
 division **108**
 quotient 54
 super-sized 127
 value 50
 integer division **54**
 integral expression 172
 integrated development environment (IDE) **14**
 IntelliJ IDEA (www.jetbrains.com) 14
 interest on deposit 194
 interest rate 162

interface keyword 40
 interpreter **10**
 isEmpty member function
 of String **64**
 iteration **107**
 counter controlled 117,
 118
 definite **105**
 of a loop 103, **154**, 174
 sentinel controlled 110,
 111, 113
 iteration statement 93, **94**,
 94, 102, 111
 do...while 94
 for 94
 while 94, **102**, 107
 iteration terminates 103

J

Jacopini, G. 92
 Java API
 documentation 51
 String class **61**, 63
 Java API (Application Pro-
 gramming Interface **38**
 Java Application Program-
 ming Interface (Java API)
 48
 java command **15**, 17, 18
 Java compiler **14**
 Java Development Kit
 (JDK) 43
 Java Enterprise Edition (Ja-
 va EE) **5**
 .java extension **14**
 Java HotSpot compiler **16**
 java interpreter 43
 Java Language Specification
 55
 Java Micro Edition (Java
 ME) **5**
 Java SE 22 (Java Standard
 Edition 22) **5**
 Java SE 8 Development Kit
 (JDK) 13
 Java Standard Edition (Java
 SE) **5**

Java Standard Edition 22
 (Java SE 22) **5**
 Java Virtual Machine
 (JVM) **14**, 17
 java.lang package **50**, 63
 java.math package 116, **187**
 BigInteger class **128**
 java.util package **49**
 javac compiler **14**, 43
 Javadoc comment 39
 javadoc utility program
 39
 JDK 13, 43
 JetBrains IntelliJ IDEA 3
 just-in-time (JIT) compiler
 16
 JVM (Java Virtual Ma-
 chine) **14**

K

keyboard 48
 keywords **40**, 94
 break **170**
 case **170**
 char **50**
 class **40**
 contextual 40
 continue **174**
 default **170**, 171
 do 94, **165**
 double **50**, 112
 else 94
 false **95**, 97
 final 172
 float **50**, 112
 for 94, **156**
 if 94
 import **49**
 int **50**
 new 49
 public **40**
 reserved but not used by
 Java 40
 static **130**
 switch 94
 true **95**, 97
 void **41**
 while 94, **165**

L

L for long integer literal **130**
 label in a switch **170**
 left align output **163**
 left brace { **41**, 42
 left-to-right evaluation 56
 length method of class
 string **63**
 letter **6**
 level of indentation 97
 lexicographical
 comparison 64
 Linux operating system 42
 literals
 floating point **113**
 loading **15**
 local variable **107**, 107
 location of a variable in the
 computer's memory 52
 logic error **14**, 50, 101, 157,
 160
 logical AND (&&) operator
 177
 logical complement opera-
 tor ! **178**
 logical negation ! **178**
 logical operators **175**, 178
 Long class **130**
 MAX_VALUE constant **130**
 long primitive type 40, 128,
 167
 L for literals **130**
 loop 107
 body 165
 continuation condition
 94, 157
 counter 154
 infinite **103**
 nested within a loop 118
 statement **94**
 loop-continuation condi-
 tion **154**, 154, 165, 174
 looping **107**
 lowercase letter 7, 40

M

machine dependent 9
 machine language **9**

programming 9
 macOS 42
 main method **41**, 42
 instance main 41
 Math class **163**
 pow method **163**, 163
 Math.PI constant 67
 MAX_VALUE constant
 Long class **130**
 memory location 52
 method **11**, 41
 argument **42**, 63
 call **12**
 static 130
 Microsoft Windows 169
 mileage obtained by automobiles 132
 minus sign (-) formatting flag **163**
 mobile application 4
 mod method of the BigInteger class **137**
 module keyword 40
 monetary calculations 164
 BigDecimal class **186**, 186
 multiline string (text block) 42, **169**
 multiple-selection statement **94**
 multiplication
 * **54**, 54
 compound assignment operator, *= 123
 multiply method of class BigDecimal **188**
 multiply method of the BigInteger class **131**

N

%n format specifier (line separator) **47**
 name of a variable **49**, 52
 native keyword 40
 nested building block 185
 nested control statements **117**, 184, 185

Examination-results problem 120
 nested if...else selection statement **98**, 135
 nested parentheses **54**
 nesting rule **184**
 NetBeans (www.netbeans.org) 14
 new keyword 40, **49**
 new Scanner(System.in) expression 49
 newline character **45**
 newline escape sequence \n 45, 46
 nonfatal logic error **101**
 nonfatal runtime error **16**
 non-sealed keyword 40
 Notepad 14
 notes (in the UML) **94**

O

object 4
 of a class 11, 12
 object oriented
 analysis and design (OOAD) **13**
 programming (OOP) **13**
 object-oriented programming (OOP) **4**
 Objects-Natural Approach 3, 38
 Objects Natural Case Study
 Precise Monetary Calculations with Java API
 Class BigDecimal 186
 Super-Sized Integers 127
 objects-natural case study
 Creating and Using Objects of Standard-Library Class string 61
 objects-natural approach 38, 128, 129
 Objects-Natural Case Study 61, 116
 off-by-one error **157**
 ONE constant of class BigDecimal **188**

OOAD (object-oriented analysis and design) 13
 OOP (object-oriented programming) **13**
 opens keyword 40
 operand **51**, 115
 operator **52**
 operator precedence 54
 rules **54**
 Operators
 ?: ternary conditional operator 101, 126
 boolean logical exclusive OR \wedge **178**
 operators
 -- predecrement/post-decrement **123**
 -- prefix decrement/postfix decrement 124
 ! logical NOT 175, **178**
 ?: ternary conditional operator 101, 126
 *= multiplication assignment operator 123
 /= (division assignment) 123
 /= division assignment operator 123
 & boolean logical AND 175, **177**
 && conditional AND **176**, 177
 %= remainder assignment operator 123
 \wedge boolean logical exclusive OR 175, **178**
 ++ prefix increment/postfix increment 124
 ++ preincrement/postincrement 123
 += addition assignment operator **122**
 = **52**
 -= subtraction assignment operator 123
 -= (subtraction assignment) 123

- | boolean logical inclusive OR 175, [177](#)
- || conditional OR 175, [176](#)
- arithmetic [53](#)
- binary [51](#), 54
- boolean logical AND & 175, [177](#)
- boolean logical exclusive OR ^ 175, [178](#)
- boolean logical inclusive OR | [177](#)
- cast [115](#)
- compound assignment 122, 125
- conditional AND && [176](#), 177
- conditional operator ?: [101](#), 126
- conditional OR || 175, [176](#), 177
- decrement operator -- 123, 124
- dot (.) [63](#)
- logical AND, && 177
- logical complement ! [178](#)
- logical negation ! [178](#)
- logical operators [175](#), 178, 179
- multiplication * [54](#)
- postfix decrement [124](#)
- postfix increment [124](#)
- prefix decrement [124](#)
- prefix increment [124](#)
- remainder 136
- remainder % [54](#), 54
- subtraction - 54
- optimizing compiler 163
- order 92
- order in which actions should execute [91](#)
- output 42
- output cursor 42, 45
- P**
- package [48](#)
- package keyword 40
- Packages
 - java.lang [50](#), 63
 - java.math 116, [187](#)
 - java.util [49](#)
- pair of braces {} 60
- palindrome 136
- parentheses [41](#), 54
 - nested [54](#)
 - redundant 56
- parentheses to force order of evaluation 126
- PATH environment variable 14
- pattern of 1s and 0s 6
- perform an action 42
- permits keyword 40
- Peter Minuit exercise 194
- phase 110
- “pick off” each digit 68
- platform independent 15
- polymorphism 172
- polynomial 56
- portable [15](#)
- postfix decrement operator [124](#)
- postfix increment operator [124](#)
- postincrement 125
- pow method of class BigDecimal [188](#)
- pow method of class Math [163](#), 163
- pow method of the BigInteger class [137](#)
- power of 2 larger than 100 102
- precedence [54](#), 60, 125
 - chart 55, 126
- precise monetary calculations 186
 - BigDecimal class [186](#), 186
- precision
 - BigDecimal 186
 - format of a floating-point number 115
- precision for formatted floating-point values [115](#)
- prefix decrement operator [124](#)
- prefix increment operator [124](#)
- preincrement 125
- Preincrementing and post-incrementing 124
- primitive type [50](#)
 - boolean [95](#)
 - byte 167
 - char 50, 167
 - double [50](#), 112
 - float [50](#), 112
 - int 50, 51, 167
 - long 128, 167
 - short 167
- principal in an interest calculation 162, 194
- print method of System.out [45](#)
- print on multiple lines 44
- printf method of System.out [46](#)
- println method of System.out 42, 45
- privacy 127
- private
 - keyword 40
- procedure for solving a problem 90
- processing phase [110](#)
- product of odd integers 191
- program control [91](#)
- program development tool 112
- promotion [115](#)
 - rules 115
- prompt message [49](#)
- protected access modifier 40
- provides keyword 40
- pseudocode [91](#), 97, 117
 - first refinement [110](#), 118
 - second refinement [111](#), 119
- public
 - keyword [40](#), 40

public-key cryptography
128

R

range checking 109
readability 39, 119
real number 50
record 7
redundant parentheses 51, 56
refinement process 110
reinventing the wheel 48, 186
relational database 7
relational operators 57
remainder 54
remainder compound assignment operator, %= 123
remainder operator % 54, 54, 136
repetition 185
 counter controlled 114
 sentinel controlled 114
repetition statement 94
 do...while 165, 185, 185
 for 185
 while 114, 155, 185
representational error 186, 187
 in floating point 164
requirements 13
requires keyword 40
return keyword 40
reusable software components 11
reuse 12, 48, 63
right align output 163
right brace (}) 41, 42, 107, 114
rise-and-shine algorithm 91
robust 50
round a floating-point number for display purposes 115
round half to even 189
rounding 54
 banker's 189

 fair rounding algorithm 186
 half even 189
 policy of a BigDecimal 186, 187
 policy Rounding-Mode.HALF_EVEN 189
 round half to even 189
rounding a number 108, 164
RoundingMode.HALF_EVEN
 rounding policy 189
RSA Public-Key Cryptography algorithm 128
rules of operator precedence 54
running an application 18
running total 111
runtime error 16

S

savings account 162
Scanner class 49, 49
 hasNext method 170
scope of a variable 158
screen cursor 46
sealed keyword 40
second-degree polynomial 56
second refinement in top-down, stepwise refinement 119
security 15, 127
selection 94, 185
selection statement 93, 94, 111
 if 94, 96, 185
 if...else 94, 97, 97, 114, 185
 switch 94, 167, 171, 185
self-documenting code 49
semicolon (;) 42, 49, 60
send a message to an object 12
sentinel-controlled iteration 111, 113, 114, 193
sentinel value 110, 111, 114
sequence 94, 185

sequence structure 93
sequential execution 92
setScale method of class BigDecimal 189
shell 42
shopping list 102
short-circuit evaluation 177
short primitive type 40, 167
side effect 177
signal value 110
significant digits of a BigDecimal 186
simple condition 175
simplest activity diagram 183
single-entry/single-exit control statements 95, 181, 183
single-line (end-of-line) comment 42
single-precision floating-point number 112
single-selection statement 94, 96, 185
 if 96
size of a variable 52
small circles in the UML 93
smallest of several integers 191
smartphone 4
software 2
software engineering
 reuse 63
solid circle
 in the UML 93
 surrounded by a hollow circle in the UML 93
source code 10, 14
space character 40
special character 50
special symbol 6
stack control statements 114
stacked building blocks 185
stacking control statements 185
stacking rule 184

- standard input stream
(`System.in`) 49
- standard library
 - class `String` 62
- standard output stream
(`System.out`) 42
- `startsWith` method of class `String` 64
- statement 42, 111
 - `break` 170, 173, 194
 - `continue` 174, 175, 194
 - control statement 91, 93, 94, 96
 - control-statement nesting 95
 - control-statement stacking 95
 - `do...while` 94, 165, 185
 - double selection 94, 118
 - empty 60, 101
 - `for` 94, 156, 163, 185
 - `if` 57, 94, 96, 185
 - `if...else` 94, 97, 97, 114, 185
 - iteration 93, 94, 102
 - looping 94
 - multiple selection 94
 - nested 117
 - nested `if...else` 98, 135
 - selection 93, 94
 - single selection 94
 - `switch` 94, 167, 171, 185
 - `while` 94, 102, 107, 114, 155, 185
- static
 - keyword 40
 - method 130
- straight-line form 54
- `strictfp` keyword 40
- string 42
 - literal 42
 - of characters 42
- `String` class 61, 63
 - `endsWith` method 64
 - `equals` method 64
 - `isEmpty` method 64
 - `length` method 63
 - `startsWith` method 64
- String concatenation 64
- string literal 63
- structured programming 92, 154, 174, 183, 184, 185
 - summary 181
- `subtract` method of the `BigInteger` class 131
- subtraction 53
 - operator - 54
- subtraction compound assignment operator, `--` 123
- `super` keyword 40
- super-sized integers 90, 127
- switch logic 172
- switch multiple-selection
 - statement 40, 94, 167, 171, 185
 - case label 170, 171
 - controlling expression 170
 - default case 170, 171, 172
- synchronized
 - keyword 40
- syntax error 40
- `System.out` 42
 - `print` method 45
 - `printf` method 46
 - `println` method 42, 42, 45
- T**
- tab character `\t` 46
- Tab* key 41
- tab stops 46
- Tax Plan Alternatives exercise 202
- temporary 115
- TEN constant of class `BigDecimal` 188
- terminal (window) 42
- terminate a loop 111
- termination phase 110
- ternary operator 101
- text block 42
- text block (multiline string) 42, 169
- text editor 42
- `TextEdit` 14
- The “FairTax” 202
- The Java Language Specification 55
- `this`
 - keyword 40
- `throw` keyword 40
- `throws` keyword 40
- `to` keyword 40
- top 110
- top-down, stepwise refinement 110, 111, 112, 118, 119
- total 106, 111
- traditional comment 39
- transfer of control 92
- transient keyword 40
- transition arrow in the UML 93, 97, 104
- transitive keyword 40
- translation 9
- translator program 9
- `true` 95
- `true` reserved word 57, 97
- truncate 54
- truncate fractional part of a calculation 108
- truth tables
 - for operator `^` 178
 - for operator `||` 177
- truth tables for logical operators 176
- `try` keyword 40
- TWO constant of class `BigDecimal` 188
- two largest values 133
- type 49
- type casting 115
- type of a variable 49, 52
- U**
- UML
 - activity diagram 93, 93, 97, 103, 183
- UML (Unified Modeling Language) 13

- activity diagram **93**, 93, 97, 165
- arrow 93
- diamond 96
- dotted line **94**
- final state **94**
- guard condition **96**
- merge symbol **103**
- note **94**
- solid circle **93**
- solid circle surrounded by a hollow circle 93
- UML (www.uml.org) 94
- unary operator 178
 - cast **115**
- underscore
 - in numbers for readability 50
- underscores 50
- Unicode character set **6**, 67, 127, 172
- Unified Modeling Language (UML) 13
- uppercase letter 41
- uses keyword 40

V

- valid identifier 49
- validate data 109
- value of a variable **52**
- valueOf method of class BigInteger **130**
- var keyword 40
- variable 48, **49**, 49
 - name **49**, 52
 - size **52**
 - type **49**, 52
 - value **52**
- variable declaration statement **49**
- variable scope 158
- vi editor 14
- virtual machine (VM) **14**
- Visual Studio Code 3
- void keyword 40, **41**
- volatile keyword 40

W

- when keyword 40
- while iteration statement 94, **102**, 107, 114
- while repetition statement 40, 155, 185
- whitespace **40**, 42, 60
- Windows operating system 169
- with keyword 40
- workflow **93**

Y

- yield keyword 40

Z

- zero-based counting 157
- ZERO constant of class BigDecimal **188**