

INDEX

A

- abstract classes, 125-134
 - base class nature of abstract classes, 127-128
 - concrete classes, 127-128
 - implementation of abstract classes, 126-131
 - instantiation of abstract classes, 127-128
 - polymorphism, 131-133
- abstract data types, 22, 74, 402-403
- Abstract Windowing Toolkit (AWT) (*See also* graphical user interface (GUI) development), 108, 317-393
 - AWT component classes, 318-319
 - AWT foundation classes, 318
 - broadcasting of events, 328
 - event handling, GUI, 326-336, **328**
 - four steps to creating a GUI, 319-326
 - GUI creation, 317-393
 - inner classes, 334-335
 - interfaces, 121
 - Microsoft Foundation Classes (MFC) vs., 318
 - source-event-listener model event handling, 329
 - summary event handling, 336-337
 - Swing vs. AWT, 386-387
 - window-closing events, 333-334
- abstraction, 22, 74, 96, 98, 125-134, 402-403
- access specifiers, 92-94, **93**, 100-101, 408
 - Private access specifier, 85, 88, 92-94, **93**, 144, 408
 - Protected access specifier, 85, 88, 90-91, 92-94, **93**, 100-101, 408
 - Public access specifier, 14, 85, 88, 92-94, **93**, 100-101, 408
- Accessing the AS/400 System with Java*, 357
- ADD, 37
- adding records to databases, 182-184
- AND operator, 28
- API for class, DB2/400 encapsulated into Java, 245-247, 248-249
- applets, 49-55, 50, 56
 - APPLET class, 50
 - choosing when to use applets, 62-63
 - functions of APPLET class, 51-52
 - initializing an applet, 50-52, **53**
 - invokation of applet by browser, 50-51, 54
 - passing HTML applet files from AS/400, 55
 - playing of applets in browser sandbox, 49-50
 - sandbox of browsers, 49-50
 - serving up applets from AS/400, 54-55
 - storage of applets on AS/400, 54
- applications, Java (*See also* Java Virtual Machine), 55-58, 58
 - choosing when to use applications, 62-63
 - executing an application, 56-58
 - Java Virtual Machine (JVM) interpretation & execution, 56-58
- arrays, 32-34, 212
 - compile-time arrays, 33-34
 - creating array for parameter list, 212-217
 - dynamic arrays, 46-48
 - VECTOR, 46-48

INDEX

AS/400 Internet Expert, 401
association of classes, 77-78, 77, 409-410
attributes, 35, 69, 82, 241-245, 404, 405
 DB2/400 encapsulated into Java, 241-245
AWT foundation classes, 318

B

base classes, 100, 127-128
 factoring out code to base classes, 267-268
 functions in base classes, 176-178
batch processing
 Java Toolbox for AS/400, enhancing
 performance, 164
 SQL statements, 202-203
behavior of objects, 69, 73, 403, 404
big loop, 19
BigDecimal, 39-41
BigInteger, 37-39
bind-by-copy, 133
binding, 133-134
 bind-by-copy, 133
 late binding, 143
Boolean values, in loops, 27-28
BorderLayout manager, 322-323, **323**
BREAK, 28
broadcasting of events, AWT, 328
BYTE data type, 36

C

callable statements to access data, 195-196
cardinality of classes, 77
case statements, 30-31
casting references, 119-120, 179
 downcasting references, 120
 upcasting references, 179
CATCH, 148, 152, 154-155
CD-ROM disk, loading the software, 395-399
CHAR data type, 36
CLASS class, 143-144
class fields, 251, 273, 404, 405
class methods, 403
class paths, 17
classes, 3-6, 7, 19, 20, 24, 34-36, 70-73, 75-76,
 96-98, **97**, 402-403

abstract classes, 125-134
access specifiers, 93-94, 100-101
AS/400 data types/ Java classes, 179, 242
association of classes, association of classes,
 77-78, 77, 409-410
AWT component classes, 318-319
AWT foundation classes, 318
base class, 100, 127-128
cardinality of classes, 77
CLASS class, 143-144
concrete classes, 127-128
construction of classes, 105-107
default constructors, 107
derived classes, 85, 283-288, 405
descendant classes, 97-98
enforced function implementation, 288-290
explicit class construction, 107
extending a class with inheritance, 89-90,
 98-101
frameworks for Java classes, 269-270
helper classes, 342-343
implicit class construction, 106-107
inner classes, 334-335
interfaces vs. classes, 110
Java Foundation Classes (JFC), 385-386
JavaDoc documentation of class, 254-257,
 256, 257
Microsoft Foundation Classes (MFC), 318
modifying objects and classes through
 polymorphism, 139-143
parent class, 100, 405
protected access specifier, 100-101
public access specifier, 100-101
singleton classes, 275
subclass, 85, 100, 405
SUPER keyword, 106-107
superclass, 100, 405
command calls, 161
comments in code, 258
common gateway interface (CGI), 58-59
common object request broker architecture
 (CORBA), 60-62, **61**
compilation, 133-134
 bind-by-copy, 133
 binding, 143, 133-134

- compile-time arrays, 33-34
- Java source files on sample CD-ROM, 397-399
- late binding, 143
- RPG programs vs. Java classes, 5
- compile-time arrays, 33-34
- component classes, AWT, 318
- composition of classes, 410
- concrete classes, 127-128
- Connection component, in JDBC, 186, 188
- constants, 31-32
- construction of classes, 105-107
- construction of objects, 74-75, 408-409
- constructors, 14, 15-16, 20, 70, 105-107
- containment, containers, 320-322, 410
 - nesting containers and objects, 325-326
- CONTINUE, 29
- conversion classes, data conversion, 161, 215-217
- creating objects, 79, 408-409
- curly brace to delimit Java code, 7-8
- cursor, dynamic cursor, 205-206
- Customer class sample code, 258-265
- CustomerDataModel listing, 387-389
- CustomerSubfile listing, 390-391

- D**
- data areas vs. STATIC, 35-36
- data conversion, 161, 215-217
- data members, 88, 404
- data queues, 161
 - integrating Java& AS/400 apps, 227-229
- data types, 10-11, 36-37
 - abstract data types, 22, 74, 402-403
 - AS/400 data types/ Java classes, 242
 - BigDecimal, 39-41
 - BigInteger, 37-39
 - BYTE data type, 36
 - CHAR data type, 36
 - converting values to Java data types, 219-220
 - FLOAT, 37-39
 - INTEGER, 37-39
 - numeric data types, 36, 37-39
 - string handling, 14, 41-44
- databases (*See* DB2/400 entries, below)

- DB2/400 data access encapsulated into Java classes, 239-265, 267-293
 - API for class, 245-247, 248-249
 - AS/400 data types/ Java classes, 179, 242
 - class attributes, 35, 69, 82, 241-245, 404, 405
 - class fields, 251, 273
 - comments in code, 258
 - creating classes, 281
 - Customer class sample code, 258-265
 - Customer class use, 291-292
 - deleting records, 252-253, 281-283
 - derived classes, 283-288
 - designing to an interface, 270-273
 - enforced function implementation, 288-290
 - enhancing the process, 292-293
 - entity encapsulation, 240-241
 - factoring out code to base classes, 267-268
 - frameworks for Java classes, 269-270
 - getter functions, 242-245
 - INIT functions, 251
 - interfaces, additional, 291
 - iterator functions, 279-280
 - JavaDoc documentation of class, 254-257, **256, 257**
 - lazy initialization, 252-253
 - object fields, 273
 - persistence, 273
 - processing sets, 247-249, 277-280
 - RDBObject base class, 268, 270, 271-274
 - retrieving records, 251-252
 - SEARCH, 251, 277-278
 - setter functions, 242-245
 - singleton classes, 275
 - STATIC, 251, 273, 275-277
 - static initializer function, 275-277
 - unit testing, 253-254
 - updating records, 252-253, 281-283
 - WRITE, 282
 - writing to records, 252-253, 281-283
- DB2/400 record-level access, 161, 167-184
 - adding records, 182-184
 - base class functions, 176-178
 - casting references, 179
 - classes used for record-level access, 168, **168**
 - connecting to AS/400, 169-170, **169**

DB2/400 record-level access, *continued*
 data types
 record-level database access, AS/400 to
 Java classes, 179
 AS/400 to Java classes, 179
 deleting records, 182-184
 file location specification, 170-171
 get and set record format, 171-172
 GetField function, 178-179
 Java database connectivity (JDBC), 167
 keyed access, KeyedFile class, 175
 opening the file, 172-173
 reading a record, 173-178
 retrieving field values, 178-179
 sample listing, 180-182, **182**
 sequential access, SequentialFile Class, 174
 steps to record-level access, 169
 upcasting references, 179
 updating records, 182-184
 DDMIterator class, 295-314
 debugging
 polymorphism for debugging, 138-139, 142
 unit testing, 253-254
 decrement math operator, 36
 default constructors, 107
 default output, 44-46, **45**
 deleting records
 DB2/400 record-level access, 182-184
 DB2/400 encapsulated into Java, 252-253,
 281-283
 delimiters, curly braces vs. op codes, 7-8
 deprecation errors, companion software, 399
 derivation, 85
 derived classes, 85, 283-288, 405
 descendant classes, 97-98
 designing to an interface, 270-273
 discovery of objects, 68-70
 DisplayCustomer application, GUI, 340-342
 DisplayCustomer listing, GUI, 347-349
 distributed common object model (DCOM),
 60-62, **61**
 distributed objects, 60-62, **61**
 Do loop, 24
 Do Until (DOU) loops, 28
 Do/while loops, 27-30

Document/View, JTable grids, 359
 documentation, JavaDoc documentation of
 class, 254-257, **256, 257**
 DOUBLE, 37-39
 downcasting references, 120
 DriverManager component, JDBC, 186, 188
 D-specs, 10-11, 31-32, 33
 dynamic arrays, 46-48
 dynamic cursor, 205-206

E

encapsulation, 80, 81-94, 95
 access specifiers, 92-94, **93**
 AS/400 program call encapsulation, 222-224
 benefits of encapsulation, 82
 DB2/400 data access encapsulated into Java
 classes, 239-265
 encapsulated business objects to develop
 GUI, 339-354
 entity encapsulation, 240-241
 friendly functions, 92
 GET functions, 86
 information hiding, 86, 89
 interfaces, 83-85
 packages, 92
 plausible deniability, 85-86
 Private, 85, 88, 92-94, **93**
 Protected, 85, 88, 90-91, 92-94, **93**
 Public, 85, 88, 92-94, **93**
 SET functions, 86-88
 threads, 89-90, 89
 UML notation showing encapsulation, 82-83,
 82
 Enterprise JavaBeans, 60
 entity encapsulation, DB2/400 encapsulated into
 Java, 240-241
 entry points to program, 13-14
 Enumeration interface, 296
 environment variables, 17
 EQUAL operator, 28
 error handling, 147-157
 CATCH, 148, 152, 154-155
 categories of errors, 149
 describing errors, 148-150
 enforcement of error handling, 153-156

- EXCEPTION objects, 148, 154
- exceptions are errors, 149, 153
- GUIs, 344-346, 344
- invocation stack, 152
- JTable grids, 364
- message generation, 151
- MONMSG vs. CATCH, 148, 152
- MSGDTA vs. THROW, 151
- PRINT function to view errors, 156
- signatures, 154
- THROW, 151, 156-157
- try/catch blocks, 152, 156
- E-specs, 32
- event handling, GUI, 326-336, **328**, 342-346
- event-driven programming, 19-20
- EXCEPTION objects (*See also* error handling), 148, 154
- exceptions are errors, 149, 153
- expanding subfiles, 381-385, 382, 391-393
- explicit class construction, 107
- extending a class with inheritance, 89-90, 98-101
- extensible markup language (EXL), 229

- F**
- factoring out code to base classes, 267-268
- FINAL/FINALIZE, 32, 114, 144, 305
- fixed decimal arithmetic, 39-41
- FLOAT, 37-39
- FlowLayout manager, 323-324, **324**
- For loops, 27-30, 311-312
- foundation classes, 318, 385-386
- frameworks for Java classes, 269-270
- friendly functions, 92, 298
- F-specs, 12, 142
- function members, 403
- function prototypes, 11
- functions, 8-11, 76, 402, 403
 - enforced function implementation, 288-290
 - friendly functions, 92, 298
 - getter functions, 86, 242-245
 - overloading functions, 104-105, 304-305
 - members, function members, 403
 - prototypes, function prototypes, 11
 - setter functions, 17, 86-88, 242-245
 - signatures, 104, 154, 405-407
- GetField function, record-level database access, 178-179
- getter functions, 86, 242-245
- global variables, 12-13, 73
- glossary of Java terms, 401-410
- graphical user interface (GUI) development (*See also* JTable grids), 62-63, 108, 317-393
 - Application Foundation Classes (AFC), 358
 - AWT component classes, 318-319
 - AWT foundation classes, 318
 - BorderLayout manager, 322-323, **323**
 - broadcasting of events, AWT, 328
 - buttons, 319
 - CardLayout manager, 325
 - components in AWT, 319
 - containers for components, 320-321
 - CustomerDataModel listing, 387-389
 - CustomerSubfile listing, 390-391
 - DisplayCustomer application, 340-342
 - DisplayCustomer listing, 347-349
 - encapsulated business objects to develop GUI, 339-354
 - error handling, 344-346
 - event handling, 326-336, **328**, 342-346
 - expanding subfiles, 381-385, **382**, 391-393
 - extending classes to create GUI, 339-340
 - FlowLayout manager, 323-324, **324**
 - four steps to creating a GUI, 319-326
 - grid choices (*See also* JTable grid), 357
 - GridBagLayout manager, 325
 - GridLayout manager, 324, **324**, **325**
 - helper classes, 342-343
 - inner classes, 334-335
 - Java Foundation Classes (JFC), 358, 385-386
 - Java Native Interface (JNI), grid support, 358
 - jt400 tables, 375-376
 - JTabbedPane, 376-378, **377**
 - JTable grid, 358-370
 - JTree, 378-381, **378**
 - Label component, 346, **346**
 - layout managers, 322-325
 - layout or arrangement of components in containers, 321-322
 - ListCustomer application, 349-354, **352**

INDEX

graphical user interface development, *continued*
 listener interfaces, 329-333
 load-all subfiles, 381
 nesting containers with components in them,
 325-326
 page-at-a-time subfiles, 381
 persistence, ISPersistent function, 344
 remote method invocation (RMI), grid
 support, 358
 scroll bars, 372
 source-event-listener model, AWT, 329
 subfiles vs. grids, 355-357, **356**
 summary event handling, 336-337
 Swing vs. AWT, 386-387
 tabbed dialogs, 325, 376-378
 tool tips associated with panels, 373-374
 tree hierarchies, 378-381, **378**
 window-closing events, 333-334, 342
GridLayout manager, 324, **324**, **325**
grids, subfiles vs. grids (*See also* JTable grids),
 355-357, **356**

H

HasNext, 300, 304
helper classes, 342-343

I

implementation of a function, 407
implementation of abstract classes, 126-131
implementation of interface, 111-114, **113**
implicit class construction, 106-107
Import statements, 6-7
increment math operator, 36
information hiding, in encapsulation, 86, 89
inheritance, 80, 85, 95-108, 405
 abstraction, 22, 74, 96, 98, 125-134, 402-403
 access specifiers, 92-94, **93**, 100-101, 408
 base class, 100, 127-128
 classes, 96-98, **97**
 constructors, 14, 15-16, 20, 70, 105-107
 default constructors, 107
 descendant classes, 97-98
 explicit class construction, 107
 extending a class with inheritance, 98-101
 implicit class construction, 106-107

 interfaces vs. inheritance, 115, 121-123
 IS A KIND OF, 102
 overloading functions, 104-105, 304-305
 overriding, 101-104, 105
 parent class, 100, 405
 Protected, 100-101
 Public, 100-101
 subclass, 85, 100, 405
 SUPER keyword, 106-107
 superclass, 100, 405
initialization
 initialization subroutine (INZSR), 15-16, 20,
 22
 lazy initialization, 252-253
 static initializer function, 275-277
inner classes, 334-335
Inprise, 318, 395
instantiation, 33, 127-128, 408-409
INTEGER, 37-39
Integrated File System (IFS), 161
 applet storage, 54
 Java package storage, 5-6
integrating Java and AS/400 applications,
 211-236
 array creation for parameter list, 211,
 212-217
 call object creation, 212, 217
 converting values to Java data types, 219-220
 data conversion classes, 215-217
 data queues, 227-229, 227
 encapsulating AS/400 program calls, 222-224
 error handling, 212, 218-219
 establishing AS/400 connection, 211, 212
 executing the program, 212, 218
 input parameters list, 214-217
 Java Native Interface (JNI), 225-227
 OS/400 command invocation, 224-225
 output parameters list, 213-214
 path creation to program, 211, 217
 PLIST, 212-217
 program call markup language (PCML)
 program calls, 229-235
 program parameters list, 212-213
 remote program calls, 211-222
 sample listing, 220-222

- interfaces, 83-85, 109-123, 405-407
 - abstract classes, 125-134
 - Abstract Window Toolkit (AWT) interfaces, 121
 - benefits of interfaces, 115-120
 - casting references, 119-120
 - classes vs. interfaces, 110
 - designing to an interface, 270-273
 - downcasting references, 120
 - extensibility of interface, 115, 116
 - FINAL, 114
 - implementation of interface, 111-114, **113**
 - inheritance vs. interfaces, 115, 121-123
 - Java Native Interface (JNI), 225-227
 - listener interfaces, GUI, 329-333
 - pluggability of interfaces, 116-117
 - polymorphism and interfaces, 117, 118
 - remote method invocation (RMI), 121
 - STATIC, 114
- Internet Inter-ORB Protocol (IIOP), 62
- invocation stack, 152
- IS A KIND OF inheritance, 102
- I-specs, 9, 142
- iterator functions, 29, 279-280, 295-314
 - DDMIterator class, 295-314
 - FINALIZE, 305
 - HASNEXT, 300, 304
 - logical file support, 302-308
 - looping, 311-312
 - NEXT, 299-301, 302, 304, 312
 - overloaded functions, 104-105, 304-305
 - PREVIOUS, 304
 - processing set of entities with DDMIterator, 308-313
 - processing subset of entities with DDMIterator, 308-313
 - REMOVE, 300
 - SQLIterator class, 295-314
 - THIS keyword, 304
- J**
- Java database connectivity (JDBC) (*See also* DB2/400 entries), 161, 167, 185-209, 339, 395-396
 - accessing data with SQL, 193-196
 - batch processing SQL statements, 202-203
 - callable statements to access data, 195-196
 - Connection component, 186, 188
 - DriverManager component, 186, 188
 - dynamic cursor, 205-206
 - JDBC 2.0 enhancements, 196-202
 - metadata, 196
 - Open Database Connectivity (ODBC), 186
 - prepared statements to access data, 193-195
 - reading backward, 203-205
 - ResultSet component, 186, 190-193
 - retrieving values of records, 190-193
 - sample application, 187-193
 - Statement component, 186, 188-190
 - updates, RDB UPDATE vs. SQL statements, 206-208
- Java Development Kit (JDK), 395-396
- Java Foundation Classes (JFC), 385-386
- Java Integrated Development Environment (IDE), 395
- Java Native Interface (JNI), integrating Java and AS/400 applications, 225-227
- Java Runtime Environment (JRE), 58
- Java Toolbox for AS/400, 161-165, 395, 396-397
 - batch processing to enhance performance, 164
 - command calls, 161
 - data conversion, 161
 - data queues, 161, 227-229
 - DB2/400 record-level access, 161, 167-184
 - installing the toolbox, 162-163
 - integrated file system (IFS), 161
 - integrating Java and AS/400 applications, 211-236
 - JAR files for toolbox, 162
 - Java database connectivity (JDBC), 161, 185-209
 - Java Native Interface (JNI), 225-227
 - program call markup language (PCML)
 - program calls, 229-235
 - program calls, 161
 - spool files, 161
 - visual components of toolbox, 162

INDEX

Java Virtual Machine (JVM), 16
 application execution, 56-58
 platform independence of JVM, 56
JavaBeans, JTable grids, 358-370
JavaDoc documentation of class, 254-257, **256**,
257
JBuilder, 395
job control, 17
jt400 tables, 375-376
JTabbedPane, 376-378, **377**
JTable grids (*See also* graphical user interface
(GUI) development), 360-393, **356**
 adding components, 360
 code implementation, minimum
 requirements, 365-366
 column attributes, 366
 document in Document/View, 361
 Document/View, 359
 error handling, 364
 expanding subfiles, 381-385, **382**, 391-393
 headers, 362, 365
 jt400 tables, 375-376
 populating the grid, 363-364
 read-only attribute for columns, 367-370
 scrolling through the database, 372
 tabbed dialogs, JTabbedPane, 376-378, **377**
 title of frames, 360
 tool tips associated with panels, 373-374
 tree hierarchies, JTree, 378-381, **378**
 updating column data, 367-370
 updating DB2/400 data through grid, 370-372
 validation of data, 368-370
 views in Document/View, 361
JTree, 378-381, **378**

K

keyed access, KeyedFile class, record-level
 database access, 175

L

Label component, GUI, 346, **346**
late binding, 143
layout managers, 322-325
lazy initialization, 252-253

LEAVE, 28
libraries, 3-6
ListCustomer application, GUI, 349-354, **352**
listener interfaces, GUI, 329-333
load-all subfiles, 381
local variables, 9
logical file support, iterators, 302-308
LONG, 37-39
looping, 19-20, 24, 27-30, 311-312

M

MAIN function, 14
 unit testing, 253-254
mathematical operators, 36
 fixed decimal arithmetic, 39-41
messages 403, 404
metadata, 196
methods, 402
Microsoft Foundation Classes (MFC), 318
modeling of objects, 69-70, **69**, **70**
modifying objects and classes through
 polymorphism, 139-143
modular vs. object-oriented programming, 8-9
MONMSG vs. CATCH, 148, 152
MSGDTA vs. THROW, 151

N

naming conventions, classes vs. RPG programs,
 7
native, Java Native Interface (JNI), 225-227
NEW, 22, 24, 32, 34, 35, 79, 105-107
NEXT, 23-24, 299-301, 302, 304, 312
numeric data types, 36, 37-39
numerical classes, 37-39

O

object fields, 273
Object Windows Library (OWL), 318
objects, 402
 abstract data types, 22, 74, 402-403
 attributes, 35, 69, 82, 241-245, 404, 405
 behavior of objects, 69, 73, 403, 404
 cardinality of classes, 77

- casting references, 119-120
 - class vs. object, 75-76
 - classes, 70-73
 - construction of objects, 74-75
 - constructors, 14, 15-16, 20, 70, 105-107
 - creating objects, 79, 408-409
 - discovery of objects, 68-70
 - distributed objects, 60-62, **61**
 - downcasting references, 120
 - functions, 76
 - global variables, 12-13, 73
 - instantiation, 33, 127-128, 408-409
 - modeling of objects, 69-70, **69, 70**
 - modifying objects and classes through polymorphism, 139-143
 - references or handles for objects, 136
 - relationships of objects, 73, 77-78, **77**
 - time sequence diagrams, 79, **79**
 - use cases, 67-68
 - variables, 76-77
 - Open Database Connectivity (ODBC), 186
 - opening files, 19, 24-25
 - operations, 403
 - operators, 36
 - OR operator, 28
 - OS/400 command invocation, integrating Java & AS/400 applications, 224-225
 - O-specs, 142
 - output, default, 44-46, **45**
 - overloading functions, 104-105, 304-305
 - overriding inheritance, 101-104, 105
- P**
- packages, 3-6, 92, 298
 - page-at-a-time subfiles, 381
 - parent class, 100, 405
 - paths, 17
 - persistence, 247, 273, 410
 - ISPersistent function, 344
 - plausible deniability, 85-86
 - playing of applets in browser sandbox, 49-50
 - PLIST, 14, 212-217
 - polymorphism, 80, 110, 117, 118, 135-145
 - abstract classes, 131-133
 - CLASS class, 143-144
 - debugging with polymorphism, 138-139, 142
 - FINAL, 144
 - late binding, 143
 - modifying objects and classes through polymorphism, 139-143
 - Private, 144
 - references or handles for objects, 136
 - STATIC, 144
 - stopping polymorphism, 144-145
 - uses of polymorphism, 137-139
 - VIRTUAL keyword, 144
 - prepared statements to access data, 193-195
 - PREVIOUS, 304
 - primitives, Java, 36, 38-39
 - PRINT function to view errors, 156
 - Private access specifier, 85, 88, 92-94, **93**, 144, 408
 - procedures, 8-9
 - program call markup language (PCML) program calls, 229-235
 - program calls, 161
 - program entry points, 13-14
 - Protected access specifier, 85, 88, 90-91, 92-94, **93**, 100-101, 408
 - prototypes, function prototypes, 11
 - Public access specifier, 14, 85, 88, 92-94, **93**, 100-101, 408
- R**
- RDBObject base class, DB2/400 encapsulated into Java, 268, 270, 271-274
 - reading backward through records, JDBC, 203-205
 - reference list, further reading, 411-412
 - references or handles for objects, 136
 - relational databases, 19
 - relationships of objects, 73, 77-78, **77**
 - remote method invocation (RMI), 60-62, **61**, 121
 - remote program calls, 211-222
 - REMOVE, 300
 - ResultSet component, JDBC, 186, 190-193
 - retrieving records, DB2/400 encapsulated into Java, 251-252
 - Rogue Wave, 357

- RPG vs. Java programming, 3-18
 - class paths, 17
 - constructors, 14, 15-16, 20, 70, 105-107
 - data types, 10-11
 - delimiters, curly braces vs. op codes, 7-8
 - Denoncourt's RPG improvements, 9-10
 - D-specs, 10-11
 - environment variables, 17
 - F-specs, 12
 - function prototypes, 11
 - functions, 8-11
 - global variables, 12-13, 73
 - IBM's RPG, 10-12
 - import statements, 6-7
 - initialization subroutine (INZSR), 15-16
 - I-specs, 9
 - Java Virtual Machine (JVM), 16
 - job control, 17
 - libraries vs. packages, 3-6
 - local variables, 9
 - MAIN function, 14
 - modular vs. object-oriented programming, 8-9
 - naming conventions, classes vs. programs, 7
 - paths, 17
 - PLIST, 14
 - procedures, 8-9
 - program entry points, 13-14
 - programs vs. classes, 3-6
 - Public keyword, 14
 - SET command, 17
 - STATIC keyword, 14
 - string array parameters, 14
 - subroutines, 8-9
 - VOID keyword, 14
- RPG++ as prototype to Java programming, 20-23
- S**
 - sandbox of browsers, 49-50
 - scroll bars, 372
 - SEARCH, 251, 277-278
 - SELECT, 31-32
 - sequential access, SequentialFile Class,
 - record-level database access, 174
 - serialization, 142, 410
 - servlets, 58-60, **59**, 62-63
 - setter functions, 17, 86-88, 242-245
 - set processing, DB2/400 encapsulated into Java, 247-249, 277-280
 - signatures, 104, 154, 405-407
 - singleton classes, 275
 - source-event-listener model event handling,
 - AWT, 329
 - spool files, 161
 - SQL access to DB2/400, 185-209
 - accessing data with SQL, 193-196
 - batch processing SQL statements, 202-203
 - callable statements to access data, 195-196
 - dynamic cursor, 205-206
 - JDBC 2.0 enhancements, 196-202
 - metadata, 196
 - Open Database Connectivity (ODBC), 186
 - prepared statements to access data, 193-195
 - reading backward, 203-205
 - updates, RDB UPDATE vs. SQL statements, 206-208
 - SQLIterator class, 295-314
 - Statement component, JDBC, 186, 188-190
 - static initializer function, 275-277
 - STATIC keyword, 14, 34-36, 114, 144, 251, 273, 275-277
 - Stingray, 357
 - string array parameters, 14
 - string handling, 14, 41-44
 - SUB, 37
 - subclasses, 85, 100, 405
 - subfiles
 - expanding subfiles, 381-385, **382**, 391-393
 - grids vs. subfiles, 355-357, 356
 - load-all subfiles, 381
 - page-at-a-time subfiles, 381
 - subroutines, 8-9, 22
 - summary event handling, GUI, 336-337
 - SunSoft, 357
 - SUPER keyword, 106-107
 - superclass, 100, 405
 - Swing, 386-387, 395, 396
 - SWITCH, 30-31
 - Symantec, 395

T

tabbed dialogs, GUI development, 325
 JTabbedPane, 376-378, **377**
tables (*See also* JTable grid)
 jt400 tables, 375-376
Taligent, 357
thin clients, 61
THIS keyword, 90, 304
threads, 89-90
THROW, 151, 156-157
time sequence diagrams, 79, **79**
tool tips associated with panels, 373-374
tree hierarchies, JTree, 378-381, **378**
try/catch blocks, 152, 156

U

unified modeling language (UML), 67, 70, **69**,
70
 cardinality of classes, 77
 discovery of objects, 68-70
 modeling of objects, 69-70, **69**, **70**
 record-level database access, 82-83, **82**
 time sequence diagrams, 79, **79**

unit testing, 253-254
upcasting references, 179
updating records,
 DB2/400 encapsulated into Java, 252-253,
 281-283
 record level access, 182-184
use cases, 67-68

V

variables, 22, 76-77
 environment variables, 17
 global variables, 12-13, 73
 local variables, 9
VECTOR, 46-48
VIRTUAL keyword, 144
visibility, 408
Visual Cafe, 395
VisualAge for Java, 395
VOID keyword, 14

W

While loops, 27-30
WRITE, 282-283
writing to records, DB2/400 encapsulated into
 Java, 252-253, 281-283

