

Index

A

- Action pattern, 299
- Actions, 264–265
- Apache Struts framework, 295
- Application Assembly Tool (AAT)
 - function of, 356
- Application state
 - action pattern, 299
 - controller layer and, 248
 - interrogation of, 249
 - managing in Web applications, 228–229
 - session lifetime and, 234–235
 - state marker, 249
 - state pattern, 299
 - using HttpSession interface, 229–233
 - writing stateless servlets, 228

B

- Breakpoints
 - conditional, 98–101
 - line, 79–82
 - Method, 105–**106**
 - setting in JavaServer Pages, **275–276**
 - single-stepping and, 87
 - using Breakpoints view, **81–82**
 - Watchpoints, 104–105

C

- Callable statements, 330
- Cascading Style Sheet (CSS), 151–152

- Certification test objectives, 6–7
- Classloaders, 359
 - configuring policies, 360
- Classpaths, 359–360
- Cloudscape, 217
- Conditional breakpoints
 - enabling hit count, 98–**99**
 - Java Exception, 101–104
 - setting the logic condition, **100–101**
- Controller layer
 - defined, 248
 - form beans and, 248

D

- Database brokers, 337
- Data Definition Language (DDL) file
 - creating, 343–344
 - defined, 343
- Data perspective
 - Data Definition view, 338, 342
 - DB Output view, 339
 - DB Servers view, 338, 341, 342
 - opening, 338
- DataTruncation exceptions, 335, 336
- DDL file. *See* Data Definition Language (DDL) file
- Debugging practices
 - changing variable values, 91
 - checking variable values, 90 (*See also* Variables view)

Note: Boldface numbers indicate illustrations.

Debugging practices (*Continued*)

- displaying output in Console view, 77
 - evaluating a Java expression, 95–96
 - fixing code with hot code replace, 110–113
 - implementing the toString() method, 77
 - inspecting variables, 93–94
 - Java Exception breakpoints, 101–104
 - JavaServer Pages and, 274–276
 - managing processes and threads, 89
 - Method breakpoints, 105–106
 - Scrapbook pages, 107–110
 - setting hit count, 98–99
 - setting line breakpoints, 79–82
 - single-stepping through code, 87–88
 - step filtering, 88–89
 - using logic condition breakpoints, 100–101
 - using print statements, 76
 - Watchpoints, 104–105
- Debug perspective, 76
- Breakpoints view, 81
 - Debug view, 84
 - description of, 82–84
 - Display view, 86–87 (*See also* Display view)
 - Editor view, 85–86
 - Outline view, 85
 - Pause function, 89
 - removing views, 87
 - Resume function, 89
 - Run to Line action, 89–90
 - Servers view, 84
 - Terminate function, 89
- Declarations
- defined for JavaServer Pages, 261–262
- Deployment Descriptor Editor. *See* Web Deployment Descriptor editor
- Display view, 86–87
- evaluating a Java expression, 95–96
- Document Type Definition (DTD), 137
- doGet() method, 178. *See also* GET method
- doPost() method, 178. *See also* POST method

E

- EAR files. *See* Enterprise archive (EAR) files
- EAR project. *See* J2EE Enterprise Application project
- Eclipse, 23–24
 - architecture, 25
 - JDT and, 26–27 (*See also* Java Development Tools [JDT])
 - Platform, 24–25
 - Plug-in Development Environment (PDE), 28
 - products built on, 28
 - tools built for, 32
 - websites for, 32
 - Workbench, 25–26
- Enterprise archive (EAR) files
 - contents of, 180
- Error handling
 - precondition check, 298
 - validating data with JavaScript, 286, 287–289
 - validating FormBeans, 286–287
- Explicit scripting objects, 262
- Expressions, 263–264
- Expressions view, 93–95

F

- Form beans, 196. *See also* Forms, encapsulating data with JavaBeans
 - creating, 214
 - validating input data, 286–287, 290–295
- Forms
 - encapsulating data with JavaBeans, 196–201
 - <FORM> tag, 140, 142
 - validating input data with JavaScript, 201–202, 286, 287–289
 - validating input data with servlets, 290–294
 - widgets for, 140

G

- GET method, 129
 - request and response interaction, **132**
- Getters, 191
- Go to File for Breakpoint button, 103

H

- Hot code replace, 110–113
- HTTP. *See* HyperText Transfer Protocol (HTTP)
- HTTP requests, 189
 - components of, 128
 - debugging input streams, 194–195
 - encapsulating with JavaBeans, 196–201
 - GET method, 129
 - headers, 129–130
 - HttpServletRequest interface, 191–193, 194
 - message stream contents, 190–**191**
 - POST method, 129
 - reading input streams, 194
 - server responses to GET, **132**
 - server responses to POST, **132–133**
 - servlet controllers and, 248–249
 - URLs and, 128
 - validating with JavaScript, 201
- HTTP responses
 - components of, 130
 - GET requests and, **132**
 - headers, 131
 - message body, 132
 - POST requests and, **132–133**
 - status line, 130–131
- HttpServletRequest interface, 191–193
 - getters for, 192
 - lazy instantiation, 194
 - request scope and, 265
 - sharing object references, 251
- HttpSession interface, 229–233
 - configuration settings for, 235
 - controlling session size, 234
 - determining session lifetime, 234

- disadvantage of using as scope object, 251
- invalidation of expected sessions, 235
- Serializable objects and, 235
- session persistence and, 235
- session scope and, 265
- storing application state, 234–235
- HyperText Markup Language (HTML)
 - description of, 133
 - disadvantages of creating with text editors, 133–134
 - <FORM> tag, 140–142
 - <LINK> tag, **152**
 - using tags, 136–139
 - using WebSphere Studio Page Designer (*See* Page Designer)
 - website resource, 133
- HyperText Transfer Protocol (HTTP)
 - description of, 128
 - GET method, 129
 - HttpServletRequest interface, 191–193, 194
 - HttpSession interface, 229–233
 - interaction of requests and responses, 132–133
 - POST method, 129
 - producing response streams, 178
 - requests (*See* HTTP requests)
 - responses (*See* HTTP responses)
 - Secure Sockets Layer (SSL) and, 128
 - status line, 130–131
 - website resource, 127

I

- IBM Agent Controller, 14
- IBM Certification tests
 - website for, 2
- IDE. *See* Integrated Development Environment (IDE)
- Implicit scripting objects, 262
- include directive, 260
- Initialization parameters
 - accessing in servlets, **185**

Initialization parameters (*Continued*)
displaying, 186
setting in servlets, 183–**184**

Integrated Development Environment (IDE), 2

J

Jacl. *See* Java Application Control language (Jacl)

Jakarta tag libraries, 303

JAR files. *See* Java Archive files

Java Application Control language (Jacl)

Java Archive (JAR) files
utility JAR files, 361

JavaBeans
encapsulating request parameters, 196–201
form beans and, 196
value beans and, 196

Java Database Connectivity (JDBC)
connecting to databases, 340–341
connection pooling, 331–332
controlling transactions, 336
function of, 327
managing with JNDI, 332, 333–334
mapping data types, 334–335
processing errors, 335–336
using database brokers, 337
version 2.0 Core API, 328–331
version 2.0 Standard Extension API, 331

Java Development Tools (JDT)
Core, 26
Debug, 27
user interface (UI), 26–27

Java editor
displaying errors, **55–57**
functions of editor view, 47–49
functions of outline view, 51–**54**
using code assist, **49–51**

Java Exception breakpoints, 101–104
Caught option, 103
Uncaught option, 103

Java Exception mechanism, 296

Java Management Extensions (JMX) Architecture, 358

Java Naming and Directory Interface (JNDI), 332, 333–334

Java perspective, 30, **31**, 64
creating Scrapbook pages, **107–110**
using, **41–42**

Java Platform 2, Enterprise Edition. *See* J2EE

Java programs
changing variable values, 91
checking variable values, 90 (*See also* Variables view)
compatibility with SQL, 334–335
debugging, 77–79
debugging with Scrapbook pages, 107, 110
evaluating a Java expression, **95–96**
inspecting variables, 93–94
Method breakpoints and, 105–**106**
running in WebSphere Studio, 62–64
setting conditional breakpoints, 98–101
setting line breakpoints in, 79–82
step filtering, 88–**89**
stepping through, 87–88
using hot code replace, 110–113
using Java Exception breakpoints, 101–104

Java projects
compiler and, 54
creating packages, 42–**43**
creating types, 44
creating with WebSphere Studio, **38–40**
error messages, **55–57**
New Java Class wizard, **44–46**
New Java Interface wizard, **46–47**
using Java editor (*See* Java editor)
using Java perspective, **41–42**

JavaScript, validating form data with, 201–202, 286, **287–289**

- JavaServer Pages (JSPs)
 - actions, 264–265
 - building in Page Designer, 271–273
 - building servlets, 258
 - compilation validator, 274
 - creating, **266–271**
 - declarations, 261–262
 - debugging, 275–276
 - directive syntax, 259
 - document structure, **258**
 - editing in Page Designer, 274
 - explicit scripting objects, 262
 - expressions, 263–**264**
 - function of, 258
 - implicit scripting objects, 262
 - include directive, 260
 - _jspService() method, 259
 - <jsp:useBean> tag, 265
 - page compilation, 258–259
 - page directive, 260
 - scriptlets, **263**
 - taglib directive, 260
 - tag libraries (*See* JSP tag libraries)
 - typeSpec attribute set, 265–266
 - URL Mappings for, 182–183
 - Web components of, 182
 - Java virtual machine (JVM)
 - classloaders and, 359
 - classpaths and, 359–360
 - JDT Debug and, 27
 - JDBC. *See* Java Database Connectivity (JDBC)
 - JDT. *See* Java Development Tools (JDT)
 - JMX Architecture. *See* Java Management Extensions (JMX) Architecture
 - JNDI. *See* Java Naming and Directory Interface (JNDI)
 - JSP. *See* JavaServer Pages (JSPs)
 - _jspService() method, 259
 - explicit scripting objects, 262
 - implicit scripting objects, 262
 - JSP Standard Tag Library (JSTL), 303
 - defining in JSP pages, 304
 - expression language (EL) versions, 304
 - runtime (RT) versions, 305
 - website for, 305
 - JSP tag libraries
 - creating custom action tags, 312
 - creating custom tag libraries, 312
 - defining, 306–308
 - Jakarta tag libraries, 303
 - reasons for using, 302
 - selecting, **309**
 - standard tag library (JSTL), 303–305
 - <taglib> element and, 308
 - using custom tags, 310
 - <jsp:useBean> tag, 265, 266, 277
 - JSTL. *See* JSP Standard Tag Library (JSTL)
 - J2EE
 - Web applications and, 3–4
 - J2EE Enterprise Application project
 - associating with a Web project, 167–169
 - modules of, 180
 - packaging for, 179–180
 - JVM. *See* Java virtual machine (JVM)
- L**
- Line breakpoints, 79–82
 - <LINK> tag, **152**
- M**
- Management beans (MBeans), 358
 - Mapper classes, 219
 - Method breakpoints, 105–**106**
 - Model layer
 - defined, 248
 - domain objects and, 248
 - validating business rules in, 295

Model-View-Controller (MVC) design
 principle, 285
 controller layer, 248, 285–286
 model layer, 248, 295
 view layer, 247, 285–286

N

New Java Class wizard, **44–46**
New Java Interface wizard, **46–47**
New JSP File wizard, **266–271**
New Servlet wizard, **170–172**

P

Package Explorer, 41–42
 comparing to Outline view, 53
 error indication, **56**
 running Java programs, 62
Page compilation, 258–259
page directive
 attributes of, 260–261
Page Designer
 Attributes view, **147**
 building JSP pages, 271–273
 Cascading Style Sheets and,
 151–152
 changing style sheets, 161–162
 Colors view, 150–**151**
 Design view, 135, **136**
 editing JSP pages, 274
 formatting a table, 160–161
 functions of, 143
 Gallery view, 145
 inserting a heading, 159–160
 inserting a link, **160**
 inserting scriptlets, 272–273
 Library view, **145–146**
 Links view, **148**
 Outline view, **147**
 Preview view, **135**
 Source view, 135
 Styles view, **150**
 Thumbnail view, 145, 148–**149**

Pause function, 89
POST method, 129
 request and response interaction,
 132–**133**
Precondition check, 298
Prepared statements, 330

R

RequestDispatcher interface, 250
Resume function, 89
Run to Line action, 89–90

S

Schemas, 341
Scrapbook pages, 107–110
Scriptlets, **263**
 inserting with Page Designer, 272–273
Secure Sockets Layer (SSL), 128
ServletContext interface, 250
Servlets
 accessing initialization parameters, **185**
 acting as controllers for HTTP requests,
 248–249
 catching exceptions, 296
 creating, **170–172**
 defined, 165
 determining lifecycles of, 173–174
 HTTP response streams and, 178
 JavaServer Pages and, 258
 lifecycle of, 166–167
 overriding methods, **173–174**
 passing object references between, 251
 populating FormBeans, 290
 RequestDispatcher interface, 250
 ServletContext interface, 250
 setting initialization parameters, 183–**184**
 testing lifecycles of, 174–178
 URL Mappings for, 182–183
 using HttpSession interface with, 230–233
 validating form data, 292–294
 Web components of, 182
 writing thread-safe servlets, 228

ServletException class, 296
SessionContext interface, 265
Show Breakpoints Supported by Selected
 Target button, 103
Silent warnings, 336
SQL. *See* Structured Query Language (SQL)
SQLException, 335
SQLWarning exceptions, 335, 336
SSL. *See* Secure Sockets Layer (SSL), 128
State marker
 defined, 249
State pattern, 299
Structured Query Language (SQL)
 callable statements, 330
 compatibility with Java, 334–335
 prepared statements, 330

T

Tag handlers, 312
Tag library descriptor (TLD) file, 312
taglib directive, 260
Tcl. *See* Tool Control Language (Tcl)
TCP/IP monitoring server
 setting up, 211
Terminate function, 89
Tool Control Language (Tcl)
Transactions
 controlling with JDBC, 336–337
 defined, 336
typeSpec attribute set, 265–266

U

Unit of work (UoW), 336
Universal Resource Locator (URL), 128
URL. *See* Universal Resource Locator
 (URL)
Utility JAR files, 361

V

Value beans, 196, 219, 249
 class diagrams for, **220**
Variables view

 changing variable values, 91
 checking variable values, 90
 displaying information, 92–93
 inspecting variables, 94
View layer, 247

W

WAR files. *See* Web archive (WAR) files
Watchpoints, 104–105
Web applications
 assembling, 356
 defined, 3
 deploying, 356
 layers of, 247–248
 managing application state, 228–233
 resolving runtime references, 361
Web archive (WAR) files
 dependence on utility JAR files, 361
 determining classloader policy settings,
 360
 contents of, 180
Web Container
 function of, 166
Web Deployment Descriptor editor
 Environment tab, 186
 Filters tab, 186
 Listeners tab, 186
 Overview tab, **181**
 Pages tab, 186
 Parameters tab, 186
 References tab, 186
 Security tab, 186
 setting error pages, **297**
 Source tab, 181
Web projects
 associating with Enterprise Application
 project, 167–169
 creating a J2EE project, 167–170
 creating a static project, 153–**154**
WebSphere Application Server
 configuring classloader policies, 360
 differences from Express edition, 357–358

- WebSphere Application Server (*Continued*)
- wsadmin command line interface tool, 358–359
- WebSphere Application Server – Express
- adding data sources, **368–370**
 - adding Enterprise Applications, 371
 - configuring classloader policies, 360
 - configuring the test environment, 238–241
 - creating remote server configurations, **362–367**
 - creating the Test Environment Server, 237–**238**
 - installing, 11–12
 - Linux versions supporting, 11
 - restrictions of, 357–358
 - running launchpad.exe, 12
 - starting remote servers, 375–376
 - starting the test environment, 241–242
 - system requirements, 10–11
 - testing applications on remote servers, 377
 - using Installer, **15–22**
 - verifying Agent Controller installation, 14
 - verifying Application Server installation, 14
 - verifying installation of directories, 13
 - verifying Site Developer installation, 14
 - Windows versions supporting, 11
 - wsadmin command line interface tool, 358–359
- WebSphere Studio
- Application Developer, 29
 - Application Developer, Integration Edition, 29
 - Asset Analyzer, 29
 - Cascading Style Sheet (CSS) editor, 151–152
 - connecting to databases, 340–341
 - creating HTML pages, **157–158**
 - creating Java classes, **44–46**
 - creating Java interfaces, **46–47**
 - creating JSP pages, **266–271**
 - creating J2EE Web projects (*See* Web projects: creating a J2EE project)
 - creating servlets, **170–172**
 - creating shortcuts to, 65–66
 - creating static Web projects (*See* Web projects: creating a static project)
 - creating Java projects, **38–40** (*See also* Java projects)
 - Data perspective, 338–**339** (*See also* Data perspective)
 - debugging (*See* Debug perspective)
 - defining tag libraries, 306–308
 - Device Developer, 28
 - differences between editions, 29
 - displaying errors in Java projects, 55–57
 - Enterprise Developer, 29
 - exporting a Java Archive file, 70–72
 - functions of Local History, 57–62
 - Homepage Builder, 28
 - HTML editor (*See* Page Designer)
 - Integrated Development Environment (IDE), 2
 - Java editor (*See* Java editor)
 - Java perspective, **41–42**
 - JSP Compilation validator, 274
 - methods for starting, 67
 - perspectives, 30–32
 - product editions, 2
 - role-based development, 30
 - running Java programs in, 62–64
 - Server perspective, 175
 - Site Developer, 29
 - Web perspective, **144**
 - Workbench (*See* WebSphere Workbench)
 - working with perspectives, 36–38
- WebSphere Workbench
- compare function, 57, **59–60**
 - configuring Local History, 61–**62**
 - Eclipse and, 2
 - replace function, 57, **59–60**
 - restore function, 57, **60–61**

- web.xml file
 - editors for opening, 180–181
 - functions of, 180
 - location of, 180
 - specifying Web components with, 182
- Wizards,
 - New Java Class, **44–46**
 - New Java Interface, **46–47**
 - New JSP File, **266–271**
 - New Servlet, **170–172**
- wsadmin command line interface tool, 358–359