

Index

SYMBOLS AND NUMERICS

2-byte Unicode strings, 433
 200 (HTTP response code), 532
 403 (HTTP response code), 532
 404 (HTTP response code), 532
 500 (HTTP response code), 532
 @AroundInvoke annotation, 481, 484
 @AttributeOverride annotation, 490
 @Column annotation, 479
 @deprecated annotation, 27
 @Discriminator annotation, 488
 @EJB annotation, 490
 @Entity annotation, 487
 @Id annotation, 478
 @Interceptors annotation, 481
 @MappedSuperclass annotation, 489
 @OneToMany annotation, 478
 @overrides annotation, 27
 @PersistenceContext annotation, 487
 @PersistenceContexts annotation, 487
 @PersistenceUnit annotation, 487
 @PersistenceUnits annotation, 487
 @Remote annotation, 483
 @Resource annotation, 380–381
 @Stateless annotation, 497
 @TransactionAttribute marking, 508
 @TransactionManagement annotation, 508
 @Transient annotation, 512

A

abstraction, 126
access
 to databases, 311
 to fields, 442–445
 in JNI, 442–445
access control, 585–586
accessor methods, 104
Action classes. See also specific types, e.g.: XWork
 Action
 IoC and, 401
 in WebWork, 399
 of XWork, 399, 412–415
ActionListener interface, 163, 205, 217–218
actionPerformed method, 163, 205, 213
Adapter pattern
 using, 126
 Adaptee interface in, 133
 Adapter interface in, 133
 Client interface in, 132
 discussed, 131–132
 Target interface in, 132
addActionListener method, 205
addFolder method, 466
adding data
 in EJB database, 476, 477
 in EntityManager API, 477–478
 in Hibernate, 412
 to Model 2 system, 417–419
 web application visualization for, 416
 XWork Action for, 413–414

addMouseListener method, 183

addTableData method, 185

addTree method, 185

AdjustmentListener, 234

advanced programming, 455–460

Agent layer, 601

Aggregate, 621

agility, 81–82

AJAX (Asynchronous JavaScript and XML), 381–384

AJAXTags library, 382–384

algorithm(s)

in JCA, 640–641

management of, 640–641

algorithm method, 635

AlgorithmParameter object, 640

AlgorithmParameterGenerator engine class, 627, 640, 641

AlgorithmParameterSpec interface, 640

ALL logging level, 38

Ambler, Scott, 81

AnnotatedElement interface, 31–32

annotation(s)

@deprecated, 27

@overrides, 27

custom, 27

defined, 26

discussed, 292

doclet API for, 28–29

functions of, 292–293

for Java classes, 290–292

in JDBC 4.0, 333–334

in metadata, 26–27

in Query interface, 333

for resource injection in EJB, 487

at runtime, 31

source-level, 27

usage of, 293–295

XML schema and, 301–302

XmlAccessorType, 296–297

XmlAttribute, 297

XmlElement, 297

XmlElementWrapper, 297–298

XmlJavaTypeAdapter, 298–300

XmlRootElement, 295

XmlTransient, 300–301

XmlType, 295–296

AnnotationDesc.ElementValuePair method, 29

AnnotationTypeDoc method, 29

AnnotationTypeElementDoc method, 29

AnnotationValue method, 29, 30

Ant (Apache)

development scenarios with, 94–101

discussed, 93–94, 125

Hibernate build file for, 408–409

installing, 709–710

in JMeter, 118

Maven 2 and, 106

project building with, 710–713

TestNG and, 108, 109

Web ARchive use with, 709–713

Apache Ant. See Ant

Apache Axis, 538

Apache Derby. See Derby

Apache TCPMon. See TCPMon

API (Application Program Interface). See also specific types, e.g.: Java API for XML Binding

for Hibernate, 340–341

in Java programming, 124

for Service Oriented Architecture, 600

applet(s)

discussed, 691–692

in JAR, 685

packaging, for execution, 693–694

security analysis for, 694

structure of, 692–693

APPLET tags, 693

appletviewer command, 693

Application Component, 401

application data

for configuration, 239–241

discussed, 237–239

saving, 239

application development (Model 2), 396

Application Program Interface (API). See also specific types, e.g.: Java API for XML Binding

for Hibernate, 340–341

in Java programming, 124

for Service Oriented Architecture, 600

applicationScope implicit object, 366

@AroundInvoke annotation, 481, 484

array(s)

functions of, 437–442

of generic types, 14

in JNI, 436–442

length property of, 18

native code for, 438, 440

of objects, 436, 437

of primitive types, 436, 437–442
usage of, 436–437

ArrayList class, 7, 8

assertions, in JMeter, 119

assignment conversion, 21

asterisk, 60, 61

ASyncHandler, 583

asynchronous invocation, 581, 583–584

Asynchronous JavaScript and XML (AJAX), 381–384

@AttributeOverride annotation, 490

authentication

checks for, 668
on client, 588–589
codes for, 665
configuration for, 670
credentials for, 669
defined, 666
execution of, 668
HTTP, 586–588, 588–589
in JCE, 665–673, 667–672
LoginContext for, 671–672
of messages, 665–666
principals in, 668–669
with security checks, 668
on server, 586–588
of subject, 670
of user identity, 667

authorization

defined, 666
in JCE, 672–673

autocommitting, 5

Axis (Apache), 538

B

back-end

passing data to, 494–495
queries in, 505

bank applications

communication for, 520
EJBs for, 520
Java EE for, 520

Basic Profile (WS-I), 567–568

batch updates

using PreparedStatement, 327–328
using Statement object, 326–327
using statements, 326–328

BatchUpdateException, 326–327

bean classes, 482

Beck, Kent, 91

binding, 568

body (SOAP), 570

BorderFactory class, 186

BorderLayout manager, 158–164

BorderLayoutPanel, 159–160

bound type parameters, 12–13, 299

boundary meta-characters, 62–63

bounded type variables, 12–14

bounds, 12–13

boxing conversions

context for, 21
defined, 7, 19
discussed, 19–20
with generics, 21

BoxLayout manager, 164–172

browsing data

in Hibernate, 412
in Model 2 system, 416, 419–422
web application visualization for, 416
XWork Action for, 412–413

Buest, Cedric, 106

buf byte array, 629

buffers, direct byte, 456

bugs (defect), 80, 87–88

building, with design patterns, 127–131

building process, 84

bundling, resource, 44

business tier (J2EE), 93

ButtonGroup component, 161

buttonPanel, 226–227

ButtonText variable, 160

bytecode, 7, 8

bytes array, 641

C

C (programming language)

data types, 432
Java objects in, 442–449
strings in, 432
variable arguments in, 18

Call Level Interface, X/Open SQL (CLI), 312

Call method, 448

callable statements (JDBC 4.0), 318, 324–326

CallMethod functions, 445–447

CallNonVirtual functions, 447, 448

CallNonVirtual method, 448

CallNonVirtualVoid function, 449

CallVoidMethod, 448, 449, 470

canImport method, 221–222

Canonicalization class

Canonicalization class, 644–645

CardLayout manager, 202–207

Cartesian join, 353

cascade property, 350

case implementation, 412–415

CDO (Collaborative Data Objects), 460

CenterPanel, 138

certificate(s)

for digital signature verification, 632

in JCA, 654–656

management of, 654–656

trusted, 638

type parameters for, 656

certificate path, 654

Certificate Revocation List (CRL), 626, 654, 656

CertificateFactory engine class, 627, 654–655

CertPathBuilder engine class, 627, 655

CertPathValidator engine class, 627, 654, 655

CertStore engine class

defined, 627

discussed, 654–656

Chain of Responsibility pattern, 173–175, 177

ChainingInterceptor, 400

ChangeListener, 135

character classes

in meta-characters, 63

in regular expressions, 63

checkAll function, 386, 389

checks, authentication, 668

ChildLogger, 58, 60

cipher, 656

Cipher engine class

data encryption/decryption with, 658

in JCE, 656–665

key wrapping/unwrapping in, 658–662

sealing objects in, 663–665

CipherInputStream class, 659–662

CipherOutputStream class, 659–662

CLASS policy, 26

ClassCircularityError, 459

classes

defining, 407–408

in generics, 14

in Hibernate, 407–408

in Java Serialization API, 242–243

JAXB, 280–281

for XML Digital Signatures, 643–645

ClassFormatError, 459

Class.forName method, 676–677

classpaths, 675–680

clazz, 457

clean target, 111–112

clearFolderList method, 466

cleartext (plaintext), 656

CLI (X/Open SQL Call Level Interface), 312

client

HTTP authentication on, 588–589

for JMS, 613–614

for MBeans, 607–609

remote, 607–609

for Web Services, 580–589

writing, 580–589

client layer

in three-tier model, 314

in two-tier model, 313

client programming, 524–525

client tier (J2EE), 93

closing, result sets, 332

CloudScape, 3

CMP (Container-Managed Persistence), 496–497

CMT. See *under* Contact Management Tool

code(s)

for authentication, 665

executing, 431–432

Java, 428–429

for JNI, 428–429, 431–432

code phase (Waterfall methodology), 88

code reuse

in JSP 2.0, 361–362

with .tag files, 361–362

with .tagx files, 361–362

CodeTag annotation, 27

coding

in Extreme Programming, 91

during software development, 83

cohesion, high, 126

Collaborative Data Objects (CDO), 460

collection classes, 7

collection mapping, 348–350

CollectTask, 96

color definition, 274, 300

Colors enum, 24

@Column annotation, 479

Command interface, 142–143, 162, 164, 211–213

Command pattern, 142–146, 190, 203

CommandManager interface, 142, 143

Common Object Request Broker Architecture (CORBA)

using, 553–554

classes in, 558

- for communication, 547–563
- COS Naming for, 550–551
- discussed, 547–548
- for distributed file system notifications (example), 554–563
- IDL of, 548–550
- Internet InterORB Protocol for, 551
- in JDK, 547
- Object Request Broker in, 550
- RMI compatibility with, 551
- RMI-IOP for, 551–553
- Common Object Service (COS) Naming, 550–551, 559–560**
- communication**
 - during software development, 81
- communication, component-to-component. See component-to-component communication**
- compatibility issues, of Model 2, 403–405**
- compile target, 112**
- complex types, 274**
- complexity, 396**
- component(s)**
 - defined, 519
 - discussed, 519
 - mapping, in Hibernate, 347–348
- component class, 151**
- Component injecting, 401**
- componentPanel, 226**
- component-to-component communication**
 - for bank applications, 520
 - CORBA for, 547–563
 - discussed, 519, 520
 - interprocess, 521–522
 - network architecture supporting, 521–522
 - RPC/RMI for, 542–547
 - scenarios for, 520–521
 - sockets usage for, 522–541
 - for web browsing, 520
 - for web portals, 521
 - for Web Services, 563–596
- composite class, 152–154**
- Composite pattern**
 - component class in, 151
 - composite class in, 152–154
 - discussed, 150–151
 - leaf class in, 151–152
- concurrency, 329**
- Concurrent Versioning System (CVS), 85**
- CONFIG logging level, 38**
- config target, 116**
- configuration**
 - application data for, 239–241
 - for authentication, 670
 - in Hibernate, 339–340, 409–410
 - of Hibernate, 409–410
 - internal data changes in, 266–268
 - in JMeter, 119
 - loading, 248–249
 - management of, 84–85
 - in Model 2 architecture, 422–424
 - reading, from disk, 246
 - saving, 248
 - verification and validation for, 258–259
 - writing, to disk, 245–246
- configuration object**
 - deserialization of, 258
 - in Enterprise JavaBeans, 240
- ConfigurationType, 277, 282**
- Connection class, 611**
- Connection interface, 315–316**
- connection management, 316–318**
- ConnectionFactory class, 611**
- ConsoleCorbaServer, 562–563**
- ConsoleHandler, 48–49**
- constants, in enumerations, 24**
- construction phase (UP), 91**
- constructors, 25**
- contact management system, 406**
- Contact Management Tool (CMT), 366–375, 370–375**
- ContactMgmtTool, 390–391**
- ContactMgmtTool POJO, 390–391**
- ContactMgmtToolDAO JavaBean, 388, 390–391**
- Container-Managed Persistence (CMP), 496–497**
- Context class, 229**
- context-root element, 700**
- continuous integration, 85**
- control**
 - of access to Web Services, 585–586
 - IoC, 397–399
- control gates (Waterfall methodology), 88**
- controller (MVC), 394**
- controller component (MVC), 140–142**
- cookie implicit object, 366**
- CORBA. See Common Object Request Broker Architecture**
- COS (Common Object Service) Naming, 550–551, 559–560**
- coupling, low, 126**

C++ programming

- data types, 432
- Java objects in, 442–449
- javah for, 429–430
- strings in, 432
- variable arguments in, 18

crashing, in Java virtual machine, 432

createMenuBar method, 218–219

createReference method, 646

credentials, 669

Criteria interface, 340, 350–352

CRL (Certificate Revocation List), 626, 654, 656

cryptographic service providers, 626

CustomHolder class, 11–12

CVS (Concurrent Versioning System), 85

D

data

- adding, 417–419
- browsing, 419–422
- in C/C++, 432
- changing, 424–426
- decryption, 658
- for digital signing, 632–634
- encryption, 658
- insertion of, 477–478
- integrity of, 629
- in JNI, 432
- in Model 2, 417–422, 424–426
- removing, 412
- types of, 432
- verification of, 631–632, 632–634

data model, 238

DatabaseMetaData, 325

databases

- access to, 311
- in Derby, 4–5
- discussed, 311–312
- with Hibernate, 335–353
- ij creation of, 4
- with JDBC API, 312–335

databases, persisting applications with

- discussed, 311–312
- with Hibernate, 335–353
- with JDBC API, 312–335

DatagramSocket class, 523

DataSource interface, 317–318

DBPanel constructor method, 184

Decorator class, 170

Decorator pattern, 165–166

decryption

- with Cipher engine class, 658
- defined, 656

DECRYPT_MODE, 657

defaultReadObject method, 254

DefaultWorkflowInterceptor, 401

deleteEntry method, 639

DeleteGlobalRef function, 454

DeleteLocalRef, 442, 451

DeleteWeakGlobalRef function, 454

deleting data. *See* removing data

Dependency Injection (DI), 474, 487. *See* Inversion of Control

dependency mediation, 102

dependency scope, 103

deployment, of Web Services (WS), 575–580

deployment descriptor

- for Interceptors, 482
- for WAR files, 696–698
- for WebWork framework, 422–423

@deprecated annotation, 27

Derby (Apache)

- using, 4–7
- benefits of, 3
- defined, 3
- development of, 3
- discussed, 4, 7
- for Hibernate configuration files, 339
- ij use in, 4–5
- location of, 317, 341
- in network mode, 6

descriptors, method, 445–446

deserialization

- of Configuration, 258
- defined, 241
- discussed, 252
- inside Swing actions, 246–247
- by value, 304

design, 126–127

design patterns

- Adapter pattern, 131–134
- building with, 127–131
- Command pattern, 142–146
- Composite pattern, 150–154
- defined, 124
- discussed, 123–124
- importance of, 124–127
- for inheritance loops, 129–131

- for interfaces, 129
 - MVC pattern, 134–142
 - single class design, 127
 - Strategy pattern, 146–150
 - TeacherResponsibilities and, 128
 - design phase (Waterfall methodology), 88**
 - Destination class, 611**
 - destroy method, 669, 692, 693**
 - detached signature**
 - defined, 643
 - discussed, 646–649
 - in XML Digital Signatures, 646–649
 - DHPrivateKey interface, 635**
 - DHPublicKey interface, 635**
 - DI (Dependency Injection), 474, 487**
 - Dialect class, 339**
 - Dialog class, 215–216**
 - digest method, 629**
 - DigestMethod class, 645**
 - DigestValue class, 645**
 - digital keys**
 - creation of, 634–638
 - JCA for, 634–640
 - management of, 634–640
 - storing, 638–640
 - digital signature**
 - as byte array, 632
 - detached, 646–649
 - enveloped, 649–652
 - on JAR files, 681
 - types of, 643
 - validating, 652–653
 - XML Digital Signatures, 649–653
 - Digital Signature Algorithm (DSA), 626, 630**
 - Digital Signature Standard (DSS), 630**
 - digital signing**
 - data verification for, 632–634
 - with detached signatures, 646–649
 - of documents, 645
 - with enveloped signatures, 649–652
 - in JAR, 686–690
 - JCA for, 630–634, 642–654
 - process for, 645–646
 - validating signatures in, 652–653
 - with XML Digital Signatures, 645–646
 - direct byte buffers, 456**
 - Direct Web Remoting (DWR), 384–391**
 - directories, endorsed, 680–681**
 - discipline, 82**
 - @Discriminator annotation, 488**
 - displayMessage method, 192, 193**
 - distributed file system notifications (example), 554–563**
 - distributed objects, 545–547**
 - DLL files, 430–431**
 - doAs method, 668**
 - doclet API (Javadoc API), 28–29, 30–31**
 - Document class, 651**
 - Document Object Model (DOM), 238**
 - document signing. See digital signing**
 - document type definition (DTD), 338**
 - doGet method, 253**
 - DOM (Document Object Model), 238**
 - domain model**
 - for contact management system, 406
 - defined, 336
 - in Model 2 architecture, 405–412
 - DOMSignContext, 647**
 - doTag method, 364**
 - dragEnter method, 168–169**
 - dragExit method, 168–169**
 - driver(s)**
 - in JDBC 4.0, 312–313, 327
 - in JDBC API, 327
 - DriverManager, 316–317**
 - DropMode, 221**
 - DropTargetListener interface, 167**
 - DSA (Digital Signature Algorithm), 626, 630**
 - DSAPrivateKey interface, 635**
 - DSAPrivateKeySpec, 634**
 - DSAPublicKey interface, 635**
 - DSAPublicKeySpec, 634**
 - DSS (Digital Signature Standard), 630**
 - DTD (document type definition), 338**
 - DWR (Direct Web Remoting), 384–391**
 - dwReplacement method, 359–361**
 - dwr.xml file, 390**
 - dynamic registration, 456–459**
- ## E
- EAR (Enterprise ARchives). See also specific types, e.g.: Web Archive**
 - deployment of, 700–701
 - descriptor file for, 699–700
 - inspecting, 699
 - EAR descriptor file, 699–700**
 - eastPanel component, 161**
 - echo server, 526–530**

education, for software development, 84

EIS (Enterprise Information System), 93

EJB. See Enterprise JavaBeans

EJB 3 (Enterprise JavaBeans 3), 474

@EJB annotation, 490

EJBContext interface, 487

ejb-jar.xml file, 699

EL. See Expression Language

elaboration phase (UP), 91

elements

initial setting for, 437

in native arrays, 439

ElementType, 26

Ellipse2D class, 180

ellipses, in variable arguments, 18, 19

email client

development of, 460–471

JNI for, 460–471

system design for, 460–461

user interface of, 461–471

EncodedKeySpec, 634

encryption

with Cipher engine class, 658

defined, 656

ENCRYPT_MODE, 657

ENCTYPE_FORM attribute, 379–380

enctype tag, 379–380

endorsed directories, 680–681

Endorsed Standard Override Mechanism, 680

endpoints, 579

engine classes, 626–628

engine.js script, 385

enhanced for loop. See for loop

EnsureLocalCapacity function, 453

Enterprise ARchives (EAR). See also specific types, e.g.:

Web ARchive

deployment of, 700–701

descriptor file for, 699–700

inspecting, 699

Enterprise Information System (EIS), 93

Enterprise JavaBeans (EJBs)

for bank applications, 520

configuration object in, 240

discussed, 473–474

EL features in, 359

entities in, 475

features of, 474

inspecting, 699

interceptor classes in, 481–485

JPA of. See Java Persistence API

for many-to-many relationships, 506–517

Object-Relational Mapping vs., 473

for one-to-many relationships, 496–506

packaging, 698–699

persistence capabilities of, 485–496

problems with, 473, 480

serialization, 262–263

session beans and, 480–481

usage of, 240

XMLEncoder/Decoder API for, 262–263

Enterprise JavaBeans 3 (EJB 3), 474

entities

in EJBs, 475, 478

in JPA, 475

of Plain Old Java Object, 475

@Entity annotation, 487

entity beans, 480

EntityManager API

acquisition of, 476

adding data in, 477–478

defined, 475

discussed, 476–480

persistence in, 476

Query method for, 478

enumerations

constants in, 24

defined, 24

discussed, 24–26

with fields and methods, 25–26

in JDK 5, 24

with methods, 25–26

EnumMap, 25

enums, 25

EnumSet, 25

envelope (SOAP), 570

enveloped signature

defined, 643

discussed, 649–652

in XML Digital Signatures, 649–652

equals method, 668

eraseItems method, 390

ErrorManager, 56

EventPanel method, 196–197

exception(s), 15–16

exception handling, 449–451

ExceptionCheck function, 449

ExceptionClear function, 450

ExceptionDescribed function, 450

ExceptionOccurred function, 449

execMethods, 448
executable JAR, 691
execute method, 200, 205, 211–213, 227–228
Executor interface, 15–16
executeTests method, 34
execution (code)
 of authentication, 668
 using JNI, 431–432, 445–449
 of JSP page, 376
 of methods, 445–449
existing protocols, 541
ExitAction, 140–141
Expression Language (EL)
 in JSP 2.0, 359–361, 365–366
 in JSTL 1.1, 374–375
 in Model 1 Architecture, 357
eXtensible Markup Language. *See under XML*
Externalizable classes, 243
Externalizable interface, 243, 259–260
Extreme Programming (XP), 87, 91–92

F

FatalError function, 450, 452
field(s)
 access to, 442
 enumerations with, 25–26
field access, 442–445
field descriptors, 446
file manipulation (JAR), 681–684
file naming, 50
file system
 for Maven 2, 102
 notification events, 554, 556, 558–561
 POM files on, 102
FileHandler, 49–50
FileItem class, 379–380
FileManager bean, 378, 380–381
fill method, 181
fillTable function, 386–387
Filter interface, 55–56
findClass method, 678–679
findHelper method, 677–678
finding data (EJB), 476
FINE logging level, 38
FINER logging level, 38
FINEST logging level, 38
FinishButton class, 228
fireTableDataChanged method, 187
FLAG, 670

flexibility, 395, 426
FlowLayout manager, 173–177, 226–227
folders, storing and retrieving, 464–465
for loop, enhanced
 defined, 7
 improvements on, over JDK5, 17–18
 syntax for, 16–17
foreach, 16
formal type parameters, 8, 9
Formatter class, 52–55
formPanel method, 197–198, 200
FROM clause, 475
FromReflectedMethod function, 459–460
front-end
 with AJAX, 382
 DWR library for, 391
 Java EE for, 520
function signatures (prototypes), 430
Function Tag Library, 366–368

G

Gang of Four (GoF) design patterns, 158
generatePanel method, 233
generateParameters, 640
generatePrivate method, 636
generatePublic method, 636
generate-web target, 112
generation (sequence) pattern, 49
generics (parameterized types)
 using, 14
 arrays of, 14
 bounded type variables, 12–14
 boxing with, 21
 class instances in, 14
 defined, 7
 discussed, 7–8
 exceptions and, 15–16
 in JDK 5, 7
 methods for, 15
 type erasure, 8–11
 wildcards, 11–12
Getahead, 384
GetArrayElements, 438
GetArrayLength, 437, 442
GetArrayRegion, 439
getCertificateAlias method, 639
getCertificateChain method, 639
getColor method, 226–227, 229
getConnection method, 388

- getContactData method, 388–389**
- getContactMgmtTool method, 389**
- getDate method, 201–202**
- getDeclaredAnnotations, 32**
- getElementById method, 389**
- getEncoded method, 635**
- getFieldID, 443–444, 446**
- getFolderList function, 467**
- getFormat method, 635**
- getInstance method**
 - for Cipher objects, 657
 - for engine classes, 627, 628
 - for KeyFactory, 635
- getKeySpec method, 636, 639**
- getLocalAddr method, 358**
- getLocalName method, 358**
- getLocalPort method, 358**
- getMarker method, 706**
- GetMethodID, 446**
- getName method, 669**
- GetObjectArrayElement, 437, 442**
- GetObjectClass function, 447, 459**
- getPanel method, 229**
- getParameter method, 694**
- getProcedures method, 326**
- getRandomNumbers, 105**
- getRecentFiles, 267**
- getRemotePort method, 358**
- getSplashScreen method, 216–217**
- GetStaticField, 444**
- GetStaticMethodID function, 447**
- GetStringChars, 434, 436**
- GetStringLength, 434**
- GetSuperClass, 459**
- getValues method, 231–232**
- getWinner method, 707–708**
- getXturn method, 708**
- global references**
 - using, 453
 - creating and destroying, 454
 - defined, 451
 - local references with, 455
 - weak, 453
- GoF (Gang of Four) design patterns, 158**
- Graphics2d method, 180**
- greedy operators, 64–65**
- GridBagConstraints class, 189, 192–193**
- GridLayout manager, 189–194**
- GridLayout class, 161**

- GridLayout manager, 169–171, 177–189**
- GROUP clause, 475**
- GroupLayout manager, 207–214**

H

- handleNext method, 230–233**
- handlePrevious method, 230–233**
- Handler class**
 - methods for, 46–48
 - stock handlers for, 48–51
 - use of, 46
- hashCode method, 9, 669**
- hashtableQuestions collection class, 162**
- HAVING clause, 475**
- header files, javah for, 429–430**
- header implicit object, 366**
- headers (SOAP), 570**
- headerValues implicit object, 366**
- Hibernate**
 - API, 340–341
 - components of, 336–337
 - configuration of, 409–410
 - Criteria interface in, 350–352
 - discussed, 335–336
 - many-to-one relationship in, 345–346
 - mapping files in, 338–339, 347–350, 424
 - Model 2 architecture support for, 402–403
 - persistent objects in, 337–338, 342–344, 407
 - persisting applications with, 335–353
 - Plain Old Java Object vs., 337
 - Query interface in, 352–353
 - setup for, 341, 409–410
 - usage example for, 341–353
 - utility classes in, 341–342
 - WebWork support for, 402–403
 - in XDoclet, 113, 115–116, 115–117
- Hibernate API**
 - many-to-many relationships in, 345–346
 - role of, 337
- Hibernate configuration file**
 - discussed, 339–340
 - role of, 337
- Hibernate mapping file**
 - defined, 338
 - discussed, 338–339
 - role of, 337
- Hibernate Query Language (HQL), 340, 352–353**
- HibernateAction class, 404–405**
- HibernateFactory object, 402**

HibernateInterceptor, 403–404
holdability, of result sets, 329–330
Holtz, Lou, 82
horizontal components, 397
HQL (Hibernate Query Language), 340, 352–353
HTML files, 530
HTMLTableFormatter, 58
HTTP 1.0, 533
HTTP authentication
 on client, 588–589
 on server, 586–588
HTTP GET
 background on, 532–533
 implementation with, 533–538
 for protocols, 532–538
HTTP Input tag, 379–380
HTTP protocol
 elements of, 531–532
 in JDK, 541, 588
 response codes in, 532
 in web applications, 521, 522
 for Web Services, 571

I

IBM, 3
@Id annotation, 478
id property
 in Hibernate database, 339, 407
 for mapping, 339
IDE (integrated development environment), 84
identification, user, 667
IDL (Interface Definition Language), 548–550
ignoreflag, 387–388
IIOB. See Internet InterORB Protocol
ij (interactive JDBC scripting tool), 4–5
impl package classes, 277
implementation
 with HTTP GET, 533–538
 in Web Services, 579
implicit objects, 366
importData method, 221–222
IN parameters
 pitfalls of, 322–323
 in prepared statements, 320–323
 setting, 320–322
inception phase (UP), 91
index option (JAR), 690–691
indirect measurement, 87
InetAddress class, 523, 524
INFO logging level, 38
ignoreflag attribute, 385–386
inheritance, 126
inheritance loops, 129–131
init method, 692, 693
initComponents () method, 160
initialElement, 437
initialize, 636
initParam implicit object, 366
Injector interface, 398
INOUT parameter, 324
InputScreen, 249
insensitive result sets, 329
insideCircle method, 182
inspecting
 Enterprise Archives, 699
 Enterprise JavaBeans, 699
 WAR files, 699
installation wizards (Swing), 225–234
installing
 Ant (Apache), 709–710
 JDBC API, 313
Instrumentation layer, 601
integrated development environment (IDE), 84
integration
 service-oriented, 599
 during software development, 85
 systems, 599
interactive JDBC scripting tool (ij), 4–5
interceptor classes
 deployment descriptor for, 482
 in EJBs, 481–485
 InvocationContext for, 481–482, 484–485
 life cycle of, 485
 StatelessSession interface for, 482–484
 styles of, 482
 in WebWork, 400–401
@Interceptors annotation, 481
Interface Definition Language (IDL), 548–550
interface design, 129
International Organization for Standards (ISO), 312
internationalization, of strings, 433
Internet InterORB Protocol (IIOB)
 for CORBA, 547
 discussed, 551
 for objects passed by value, 551
 for RMI, 545
interprocess communication, 521–522
InterruptedException, 588

InvalidJarIndexException, 690

inverse attribute, 350

Inversion of Control (IoC, dependency injection)

defined, 396

discussed, 146

in Model 2 architecture, 397–399, 426

Plain Old Java Object for, 398

invocation protocol

asynchronous, 581, 583–584

in JSP 2.0, 363–365

method, 21

synchronous, 582–583

of Web Services, 581–584

InvocationContext, 481–482, 484–485

InvocationContext reference, 481, 484

Invoker interface, 142, 144–145

IoC. See **Inversion of Control**

IOException, 15–16

IsAssignableFrom function, 459

isCertificateEntry method, 639

isCurrent method, 669

isDesktopSupported method, 222–223

isDestroyed method, 669

isKeyEntry method, 639

ISO (International Organization for Standards), 312

IsSameObject function, 455

Iterable interface, 17

iterator(s), 16

iterator tags, 421

J

J2EE web application

architecture of, 93

defined, 125

design patterns in, 125

WebWork as, 422

JAAS (Java Authentication and Authorization Service), 625, 666

JAddEventButton class, 200

Jakarta Commons Net Package, 541

Jakarta Commons Upload, 378

JAR. See **JAVA ARchive**; **Java ARchive**

jar tool, 681–684

jarsigner, 686

Java 5 Tiger release, 106

Java API for XML Binding (JAXB)

advantages to using, 307

disadvantages to using, 307

discussed, 270–271

in JDK, 308

object graphs, 277–280, 283

pitfalls of, 302–307

runtime compatibility for, 277

serialization in, 290

usage of, 307–308

value serialization in, 302–305

version compatibility for, 277

WSDL for, 308

XML Schema Definition in, 271

Java API for XML Binding classes, 280–281

annotations, 292–301

usage of, 283–290

from XML schema, 276–277

Java ARchive (JAR)

applets in, 685

creation of executable JAR, 691

discussed, 681

and endorsed directories, 680–681

exploring, 681

file manipulation in, 681–684

index option for, 690–691

and Java classpaths, 675–680

for license files, 253, 254

manifest files for, 684–685

signing, 686–690

Java Authentication and Authorization Service (JAAS), 625, 666

Java class annotations, 290–292

Java code

discussed, 428–429

native methods for, 427

strings in, 432

variable arguments in, 18

Java Cryptography Architecture (JCA)

algorithms in, 640–641

architecture of, 626

certificate management in, 654–656

design of, 626

for digital keys, 634–640

for digital signing, 630–634, 642–654

discussed, 625

engine classes for, 626–628

MessageDigest class in, 628–630

RNG in, 641–642

security and, 625–626

XML Digital Signatures in, 642–654

Java Cryptography Extension (JCE)

authentication in, 665–673, 667–672

authorization in, 672–673

Cipher class in, 656–657, 656–665
 discussed, 625
 KeyGenerator engine class in, 662
 for license files, 254
 message authentication code in, 665–666
 sealing objects with, 663–665
 SecretKeyFactory in, 662–663
 security and, 625, 656
 services of, 656

Java DataBase Connectivity. See under JDBC

Java Development Kit. See under JDK

Java drivers, 312

Java EE. See Java Enterprise Edition

Java Enterprise Edition (Java EE)

for bank applications, 520
 discussed, 519
 for RMI, 547

Java Foundation Classes (JFC)

discussed, 157–158
 layout managers, 158–214
 Mustang release, 214–225

Java logging

defined, 34
 discussed, 35–36
 ErrorManager, 56
 examples of, 56–60
 Filter interface, 55–56
 Formatter class, 52–55
 Handler class, 46–51
 Level class, 45–46
 Logger class, 38–42
 LogManager class, 36–42
 LogRecord class, 42–45

Java Management Extensions (JMX)

architecture for, 601
 discussed, 600
 importance of, 601
 MBeans in, 602–610
 relevance of, 600
 for SOA, 600–610

Java Messaging Service (JMS)

client for, 613–614
 discussed, 610
 importance of, 610
 JMSAgent in, 617–618
 JMSWorker MBean in, 614–616
 message functions in, 611–612
 queues in, 610–611
 receiving messages with, 611–612
 relevance of, 600

sending messages with, 611–612
 server configuration for, 612–613
 for SOA, 610–618
 topics in, 611

Java meta-characters, 63, 64

Java Naming and Directory Interface (JNDI), 317–318, 380, 474

Java Native Interface (JNI)

using, 428
 advanced programming with, 455–460
 arrays in, 436–442
 code for, 428–429
 creating, 429–431
 data types in, 432
 discussed, 427, 432
 dynamic registration with, 456–459
 for email client development, 460–471
 exception handling using, 449–451
 executing code using, 431–432
 field access using, 442–445
 local reference in, 452
 method execution using, 445–449
 NIO support in, 456
 object references in, 451–455
 program writing with, 427
 reflection functions in, 459–460
 strings in, 432–436
 threading using, 455–456

Java Network Launch Protocol (JNLP), 702–704

Java objects. See object(s)

Java Persistence API (JPA)

discussed, 474–475
 entities in, 475
 entity manager of, 476–480
 features of, 474, 475
 for Plain Old Java Object, 474
 query language of, 475–476
 select statements in, 475–476

Java persistence query language, 475–476

Java preferences. See Preferences class

Java Remote Method Protocol (JRMP), 545

Java Runtime Environment, 680

Java SDK, 626–627

Java Serialization API

classes in, 242–243
 discussed, 241–242
 extending and customizing, 257–261
 format customization, 258–260
 implementation, into applications, 253–257
 steps for, 243

Java Serialization API (continued)

- strengths and weaknesses in, 261
- time-based licensing using, 249–253
- transient keyword for, 257
- usage of, 261–262
- versioning in, 260–261
- XMLEncoder/Decoder API vs., 269

Java Server Page (JSP)

- for EJB 3 components, 513
- EL features in, 359

Java Socket API, 523

Java Standard Template Library. See JSTL 1.1

Java strings, UTF-8 format for, 433

Java virtual machine

- crashing, 432
- string usage in, 432

Java Web Start applications

- discussed, 702, 709
- TicTacToe example for, 702–709

Java2DPanel class, 179–180, 182–183

Java2DPanelMouseClickedPrint class, 188–189

JavaBeans (Enterprise). See Enterprise JavaBeans

Javadoc API (doclet API), 28–29, 30–31

javah, 429–430

javap utility, 446

JavaScript

- AJAXTags libraries as alternative to, 382
- and DWR, 384

javascript function, 385

JAXB. See Java API for XML Binding

JAXB 1.0, 271, 277

JAXB 2.0, 271, 277, 280

JAXB classes (JDK 6), 276

JAXBContext, 281

JAXBElement, 280–281

JAXBException, 287, 289

JAX-WS, 575–577, 579

JBoss, 612–613

JButton component, 160, 174, 205

JButton setText() method, 162

JButtonCoins method, 176

JButtonSave component, 200–201

JButtonStrategy1 class, 203–204, 205–206

JButtonStrategy2 button, 204

JButtonStrategy2 class, 206

jbyte, 434

JCA. See Java Cryptography Architecture

JCE. See Java Cryptography Extension

jchar, 432, 434

JConsole, 604–605

JDBC 3.0, 335

JDBC 4.0

- using, 315–316
- annotations in, 333–334
- connection management in, 316–318
- DataSource interface in, 317–318
- discussed, 311
- DriverManager in, 316–317
- drivers in, 312–313, 327
- result sets in, 328–333
- statements in, 318–328
- transaction management in, 334–335

JDBC API (Java DataBase Connectivity API)

- classes in, 243
- in Derby, 5–6
- discussed, 312–313
- drivers in, 327
- establishing data source for Contact Management Tool, 373
- installing, 313
- parameters in, 353
- persisting applications with, 312–335
- setObject method for, 323–324
- three-tier model of, 314–315
- two-tier model of, 313–314

JDBC-net pure Java driver, 312

JDBC-ODBC bridge driver, 312

JDK (Java Development Kit)

- CORBA support in, 547
- dependencies in, 103
- HTTP support in, 541, 588
- JAXB in, 308
- manipulating JAR files, 681–682
- versioning in, 260
- XMLEncoder/Decoder API in, 269

JDK 5

- enumerations in, 24
- features of, 3
- for loop enhanced from, 17–18
- generics in, 7

JDK 6

- changes in, 3
- Derby in, 3
- JAXB classes in, 276
- WS deployment with, 577–578

JFC (Java Foundation Classes)

- discussed, 157–158
- layout managers, 158–214
- Mustang release, 214–225

JFormattedTextField class, 174
JFrame (BorderLayout) container, 158
JKS, 638
JMeter, 117–120
JMeter 2.1, 119
JMS. See Java Messaging Service
JMSAgent component, 611, 617–618
JMSWoker component, 611, 614–616
JMX. See Java Management Extensions
JNDI (Java Naming and Directory Interface), 317–318, 380, 474
JNextButton component, 228
JNI. See Java Native Interface
JNIMailBridge, 460, 465
JNINativeMethod structure, 457, 459
JNLP (Java Network Launch Protocol), 702–704
JPA. See Java Persistence API
JPanel, 138
JPanel (FlowLayout) container, 158
JPreviousButton component, 227–228
JQuestionButton, 162
JRadioButtonAnswer class, 162
JResetButton button, 164
JRMP (Java Remote Method Protocol), 545
JSP (Java Server Page)
 for EJB 3 components, 513
 EL features in, 359
JSP 2.0
 code reuse in, 361–362
 discussed, 357–358
 Expression Language in, 359–361, 365–366
 invocation protocol in, 363–365
 .jspx page extensions in, 362–363
 for Model 1 architecture, 357–365, 376–380
 Servlet 2.4 support in, 358
 web application visualization with, 376–380
JSP custom tags (WebWork), 417
.jspx page extensions, 362–363
JSR 105, 642
JSTL 1.1
 discussed, 366
 Function Tag Library in, 366–368
 for Model 1 architecture, 366–375
 SQL transactions in, 368–370
 web application visualization with, 370–375
jstring data type (C/C++), 432, 433
JTextArea component, 233
JTextField, 221, 231–232
JTree object, 208
JUnit, 85

K

key agreement, 656
key entry, 638
key interface, 634–635
key unwrapping, 658–662
key wrapping, 658–662
KeyFactory engine class
 defined, 627
 for representation conversion, 635
KeyGenerator engine class, 662
KeyInfo class, 645
KeyPair class, 636
KeyPairGenerator engine class
 defined, 627
 for key management, 636
KeySelectorResult interface, 653
keysize parameter, 636
KeySpec interface, 634
keystore, 638–639
KeyStore engine class
 defined, 627
 discussed, 638
 for key management, 636
King, Gavin, 402

L

language features
 boxing conversions, 19–21
 discussed, 7
 enumerations, 24–26
 for loop, enhanced, 16–18
 generics, 7–16
 metadata, 26–34
 static data importing, 21–23
 unboxing conversions, 20–21
 variable arguments, 18–19
Larman, Craig, 90
layout managers
 BorderLayout manager, 158–164
 BoxLayout manager, 164–172
 CardLayout manager, 202–207
 discussed, 158
 FlowLayout manager, 173–177
 GridBagLayout manager, 189–194
 GridLayout manager, 177–189
 GroupLayout manager, 207–214
 SpringLayout manager, 194–202
lazy attribute, 350
lazy (reluctant) operators, 65

leaf class, 151–152

learning curve (Model 2), 395

len parameter, 628, 629

Level class, 45–46

license files
JAR for, 253, 254
JCE for, 254

life cycle
of interceptor classes, 485
of `JMSWorker`, 614
with Maven, 101, 106

listeners, in JMeter, 119

listkey, 419

listvalue, 419

load method, 638

loading, 248–249, 287–290

load-on-startup attribute, 697

local references
creating and deleting, 451
defined, 451
global references with, 455
management of, 452–453
memory issues of, 452

localized strings, 433

Logger class
levels of, 38
methods for, 38–42
usage of, 38

Logger objects, 34–35, 39–42

logging. See **Java logging**

LoggingInterceptor, 400

logic controllers, in JMeter, 119

login method, 223

LoginAction, 140

LoginContext, 670, 671–672

LogManager class
configuration of, 37
control methods in, 37–38
properties of, 36

logout parameter, 671

LogRecord class, 42–45
defined, 42
methods for, 43
for origination, 43–44
for resource bundling, 44
for setting information, 44–45

loops
for checking, 506
inheritance, 129–131

loose-coupling, 128, 619

M

MAC (message authentication code)
defined, 656
in JCE, 665–666

mail messages, 464–466

MailClient, 460

MailFolder class, 463

MailMessage class, 463

main method, 679

manifest files (JAR), 681, 684–685

manipulating result sets, 331–332

many-to-many relationships
EJBs for, 506–517
in Hibernate API, 345–346
one-to-many relationships vs., 507
web components for, 506–517

many-to-one relationship, 345–346

MAPI routines, 465–466

@MappedSuperclass annotation, 489

mapping
of collections, 348–350
components of, 347–348
files for, 424
in Hibernate, 338–340, 347–350
properties for, 350
property-to-column, 339
XDoclet for, 110–111

marshaller
in RMI, 544
XML, 281–282

Maven 1, 101

Maven 2
Ant and, 106
archetypes in, 102
discussed, 101–102
transitive dependencies in, 102–106

MBeans
creation of, 602–606
in `JConsole`, 605
`JMSWorker`, 614–616
in `JMX`, 602–610
in `JMX` architecture, 601
management of, 602–606, 606
remote client for, 607–609
`RemoteAgent` for, 606–607
`WorkMonitor` for, 609–610

MBeanServer, 603, 607

MD4 algorithm, 628

MD5 algorithm, 628

- MD5 algorithms, 626**
- MDI (multiple document interface), 239**
- member variables, 442**
- memory, 452**
- MemoryHandler, 50–51**
- message authentication code (MAC)**
 - defined, 656
 - in JCE, 665–666
- Message class, 611**
- MessageClient component, 611**
- MessageConsumer class, 611**
- MessageDigest engine class**
 - calculating, 628–629
 - defined, 627
 - in JCA, 628–630
 - verifying, 629–630
- Message-Oriented Middleware (MOM), 600, 610, 618**
- MessageProducer class, 611**
- messages**
 - in JMS, 611–612
 - receiving, 611–612
 - sending, 611–612
- meta-characters**
 - boundary, 62–63
 - character classes in, 63
 - discussed, 60
 - Java, 63, 64
 - POSIX, 63–64
 - predefined types, 61–63
 - in regular expressions, 61–63
 - types of, 60
- metadata**
 - AnnotationDesc, 29
 - AnnotationDesc.ElementValuePair, 29
 - annotations in, 26–27
 - AnnotationTypeDoc, 29
 - AnnotationValue, 30
 - AnnotationTypeElementDoc, 29–30
 - defined, 7
- META-INF directory, 681, 683–684, 686, 689, 690**
- method descriptors, 445–446**
- method execution, 445–449**
- method invocations, 21**
- methodology**
 - discussed, 88
 - Extreme Programming, 91–92
 - observations on, 92–93
 - Unified Process, 90–91
 - Waterfall methodology, 88–89
- methods, 25–26**
- methods parameter, 457**
- mgResponseBody object, 386–387**
- Microsoft .NET framework, 554, 596**
- middle layer, in three-tier model, 314**
- middleware**
 - message-oriented, 600, 610, 618
 - for RMI, 546
- mime-mapping element, 698**
- Mine, Philip, 269**
- model(s), 406**
- model (MVC), 394**
- model component (MVC), 136–137**
- Model 1 architecture**
 - discussed, 355–357
 - EL for, 359–361, 365–366
 - JSP 2.0 specification for, 357–365, 376–380
 - JSTL 1.1 for, 366–375
 - Model 2 architecture vs., 396
- Model 2 architecture**
 - adding data to, 417–419
 - advantages of, 395
 - application development with, 396
 - browsing data in, 419–422
 - case implementation in, with actions, 412–415
 - changing data in, 424–426
 - compatibility issues of, 403–405
 - configuring, 422–424
 - disadvantages of, 395–396
 - discussed, 393–395, 426
 - domain model definition in, 405–412
 - Hibernate support in, 402–403
 - Inversion of Control in, 397–399
 - Model 1 architecture vs., 396
 - MVC in, 393–394
 - need for, 393
 - Plain Old Java Object in, 426
 - problems with, 403–405
 - process of, 394–395
 - scope in, 395
 - security in, 395
 - use of, 395–396
 - web application visualization in, 415–422
 - with WebWork, 396, 399–402
- ModelDrivenInterceptor, 400**
- mode-less, 215**
- modeling, 81**
- Model-View-Controller (MVC) pattern**
 - application initialization with, 136
 - changes to, 135
 - components of, 394

Model-View-Controller (MVC) pattern (continued)

- controller component of, 140–142
- discussed, 134–142, 393
- in Model 2 architecture, 393–394
- model component of, 136–137
- purpose of, 134
- view component of, 137–139

modularity

- in Model 2 architecture, 426
- of WebWork, 426

module element, 699–700

MOM (Message-Oriented Middleware), 600, 610, 618

MonitorEnter function, 456

mousePressed method, 181

MS Outlook, 460

multiple document interface (MDI), 239

Mustang release, 214–225

MVC pattern. See Model-View-Controller pattern

MyAdjustmentListener method, 234

MyTableModel class, 186–187

N

name parameter, 579, 671

named queries, 475

NamedQuery annotation, 506

naming (COS), 550–551, 559–560

naming files, 50

native API/part Java driver, 312

native arrays, 439

native code

- for arrays, 438, 440
- Java objects in, 437
- sort routine in, 4389

native keyword, 428

native libraries, 431

native methods

- using, 428–429
- for Java code, 427
- registering, 456–459
- unregistering, 457

native-protocol pure Java driver, 312

navigating result sets, 330–331

.NET framework (Microsoft), 554, 596

NetBeans, 207

network architecture, 521–522

NewArray, 438

NewGlobalRef function, 453, 454

NewLocalRef function, 451

NIO direct buffers, 456

NIO support, 456

NoClassDefFoundError exception, 450

nodes, in Preferences class, 70, 72, 73

nonscrollable result sets, 329

non-static fields, 442

non-static methods, 445

NULL character, 433

NumberFormat class, 174

numBytes array, 641

O

OASIS, 596

object(s)

- in ArrayList, 7
- arrays of, 436, 437
- in C/C++, 442–449
- distributed, 545–547
- point, 275
- in RMI, 545–547
- sealing, 663–665

object graph, 239

Object Graph Navigation Language (OGNL), 401

object graphs (JAXB), 277–280

object graphs serialization, 290

Object Management Group (OMG), 547

Object method, 30

object references

- comparing, 455
- global, 453–455
- global references, 453–455
- in JNI, 451–455
- local, 451–453
- local references, 451–453

Object Request Broker (ORB), 550

ObjectFactory class, 280

ObjectInputStream, 249, 265

object-oriented (OO) design, 126, 127, 234, 311

ObjectOutputStream, 249, 265

Object-Relational Mapping (O/RM)

- in @Column annotation, 479
- compatibility issues of, 403
- EJBs vs., 473
- solutions of, 344
- tools for, 403

object-to-object communication, 548

OFF logging level, 38

OFX (Open Financial Exchange), 271

OGNL (Object Graph Navigation Language), 401

OMG (Object Management Group), 547

omitCheckedItems function, 385–386

@OneToMany annotation, 478

one-to-many relationships

- EJBs for, 478–479, 496–506
- many-to-many relationships vs., 507
- web components for, 496–506

OO design. See **object-oriented design**

opaque representations, 634, 635

Open Financial Exchange (OFX), 271

Open Systems Interconnection (OSI), 521

OperatingSystems enum, 24

operations, in Preferences class, 70–71, 74

opMode parameter, 657

Optional FLAG, 670

O/R mappings, standardized, 475

ORB (Object Request Broker), 550

ORDER BY clause, 475

O/RM. See **Object-Relational Mapping**

OSCAR, 540

OSI (Open Systems Interconnection), 521

OutOfMemoryError, 459

@overrides annotation, 27

OwnerSession interface, 507–508

P

package target, 112

packaging

- for applet execution, 693–694
- of EJBs, 698–699

pageContext implicit object, 366

pageScope implicit object, 366

paint method, 692, 693

paintComponent (Graphics g) method, 166, 180, 183

panelTable method, 185, 197

param implicit object, 366

PARAM tag, 694

parameterized types, 8. See also **generics**

params parameter, 657

paramSpec object, 640

paramValues implicit object, 366

parentheses, inside regular expressions, 65

ParentLogger, 58

parse method, 649

parsing, file formats, 530

Password-Based Encryption (PBE), 656

password-based encryption (PBEKey), 635

Pattern-Matcher model, 61, 68

patterns. See **design patterns**

PBE (password-based encryption), 656

PBEKey interface, 635

persistence query language, 475–476

@PersistenceContext annotation, 487

@PersistenceContexts annotation, 487

PersistenceDelegates, 269

@PersistenceUnit annotation, 487

@PersistenceUnits annotation, 487

persistent objects

- in Hibernate, 337–338, 342–344, 407
- libraries for, 497
- properties of, 496
- role of, 337

pigLatin method, 359–361

PKCS (Public Key Cryptography Standard), 638

Plain Old Java Object (POJO)

- ContactMgmtTool, 390–391
- entities of, 475
- Hibernate vs., 337
- for IoC, 398
- JPA for, 474
- in Model 2 architecture, 426
- session beans as, 474

plaintext (cleartext), 656

platform-independent RPCs, 566–567

POA (Portable Object Adapter), 559

point objects, 275

POJO. See **Plain Old Java Object**

polymorphism, 126, 130, 149

POM (project object model), 103, 106

Popescu, Alexandru, 106

PopLocalFrame function, 452, 453

populateTable method, 184, 185, 187

PopupMenu class, 217–218

port types, 568

Portable Object Adapter (POA), 559

portlet, 361–362

POSIX meta-characters, 63–64

PostPersist, 501

post-processor tests, in JMeter, 119

PostRemove, 501

Preferences class

- using, 75–77
- discussed, 70
- events in, 73
- exporting, to XML, 74
- nodes in, 70, 72, 73
- operations in, 70–71, 74
- retrieving values for, 72–73
- setting values for, 73

prepared statements (JDBC 4.0)

- batch updates using, 327–328
- discussed, 318
- IN parameters in, 320–323
- setObject method for, 323–324

PreparedStatement, 327–328

pre-processor tests, in JMeter, 119

primary key, 500, 511

primitive types

- array type counterpart for, 436
- arrays of, 436, 437–442
- conversion of, to reference type, 19
- references to, 20–21
- types of, 438

principals, in authentication, 668–669

printf (), 432

PrintWriter class, 524

private key, 632

PrivateKey interface, 635

processing chain pattern, 619

programming

- client, 524–525
- scripting vs., 396
- sockets for, 524–525, 525–526

project building, with Apache Ant, 710–713

project object model (POM), 103, 106

PropertyChangeListener interface, 139, 174–175

property-to-column mapping, 339

propOrder value, 295–296

proprietary protocols, 540–541

protocol(s)

- defined, 521
- discussed, 530
- existing, 541
- HTTP GET for, 532–538
- HTTP specification for, 531–532
- implementation with, 530–541
- proprietary, 540–541
- reverse engineering with, 540–541
- for RPCs, 544–545
- with sockets, 530–541
- sockets and, 530–541
- specification for, 531–540
- TCPMon testing for, 538–540
- utilizing, 541

prototypes (function signatures), 430

provided scope, 103

providers (cryptographic service providers), 626

pseudo-random random number, 640

Public Key Cryptography Standard (PKCS), 638

public key object, 632

PublicKey interface, 635

PushLocalFrame, 453

Q

quality measures, 80

Query interface

- annotations in, 333
- for EntityManager API, 478
- in Hibernate, 352–353
- in Hibernate API, 340
- of JPA, 475–476

queues (JMS), 610–611

R

random number generation, 641–642

random number generator (RNG), 641

RDBMS (Remote Database Management System), 324

readObject method, 254

read-only result sets, 329

refactoring, 83

reference, 543

reference(s)

- conversion of, from primitive types, 19
- object. See object references
- to primitive types, 20–21

Reference class, 643

reflection API, 32–33

reflection functions, 459–460

refresh method, 669

RegisterNatives function, 457, 459

registerOutParameter, 324

registration

- dynamic, 456–459
- with JNI, 456–459
- for native methods, 456–459

registry (RMI), 545

regular expressions

- character classes in, 63
- defined, 34, 60
- Matcher class, 66–68
- MatchResult interface, 68
- meta-characters in, 61–63
- parentheses inside, 65
- Pattern class, 65–66
- repetition operators in, 64
- usage examples for, 68–70

relational database, 315–316

ReleaseStringChars, 434, 436

reluctant (lazy) operators, 65

@Remote annotation, 483

remote client (MBeans), 607–609

Remote Database Management System (RDBMS), 324

Remote Management layer, 601

Remote Method Invocation (RMI)

CORBA compatibility with, 551

defined, 542

defining, 543

distributed objects in, 545–547

IIOB for, 545

in Java EE, 547

marshalling/unmarshalling in, 544

middleware for, 546

objects in, 544

principles of, 542–543

registry for, 545

RMI-IIOP objects from, 551–553

serialization in, 249

remote procedure calls (RPC)

defined, 542

platform-independent, 566–567

principles of, 542–543

protocols for, 544–545

in Web Services, 566–567

remote reference, 542

RemoteAgent (MBeans), 606–607

removeAll method, 200, 227–228

removing data

in EJB database, 476

in Hibernate, 412

renewItems method, 390

renewResults function, 385–388

repetition operators, 64

replaceString, 435–436

Request Component, 401

request-reply pattern

discussed, 619–621

for SOA, 619–621

request-response flow, 399–400

requestScope implicit object, 366

Required FLAG, 670

Requisite FLAG, 670

@Resource annotation, 380–381

resource bundling, 44

Resource Injection, 486

restoreLogPanel method, 201–202

result sets

using, 330

closing, 332

concurrency of, 329

discussed, 328

holdability of, 329–330

in JDBC 4.0, 328–333

manipulating, 331–332

navigating, 330–331

row insertion/deletion in, 332

types of, 329

ResultSet interface, 315–316

retention, 26, 27

RetentionPolicy enumeration, 26

reuse

code, 361–362

at code-level, 124

at design-level, 124

in Model 2 architecture, 395

reverse containment, 130

reverse engineering, 540–541

RMI. See Remote Method Invocation

RMI-IIOP, 551–553

RNG (random number generator), 641

rootElement object, 280

RPC. See remote procedure calls

RSAMultiPrimePrivateCrtKey interface, 634, 635

RSAPrivateCRTKey interface, 634, 635

RSAPrivateKey interface, 634, 635

RSAPublicKey interface, 634, 635

RulesButton class, 164

runtime

annotations at, 31

JAXB version compatibility for, 277

RUNTIME policy, 26

S

sampler plans, in JMeter, 119

savepoint, 335

saving

action implementation for, 284–287

of application data, 239

configuration, 248

scalability, 395

SchemaExport (Hibernate), 408, 409, 425

scope, 395

scripting, programming vs., 396

scrollable result sets, 329, 330

sealing objects, 663–665

searchClassPath method, 678

searchJarFile method, 677

SecretKey interface, 635

SecretKeyFactory, 662–663

Secure Hash Algorithm (SHA-1), 626, 628, 630

SecureRandom engine class

defined, 627

use of, 642

security

of Applets, 694

discussed, 625

using JAAS, 666

with JCA, 625–626

with JCE, 625, 656

in Model 2 architecture, 395

user identification and, 667

security checks, 668

security-constraint element, 698

security-role element, 698

Select statements, 475

select tag (WebWork), 419

sendMail method, 464

sensitive result sets, 329

sequence (generation) pattern, 49

Serializable interface, 243, 249

serialization. See also Java Serialization API

classes for, 243

defined, 241

inside Swing actions, 246–247

in Java API, for XML Binding, 302–305

JAXB, 290

for object graphs, 290

in RMI, 249

strengths of, 261

as temporary solution, 262

by value, 302–305

weaknesses of, 261

XmlJavaTypeAdapter as root of, 306

serialVersionUID, 260

server

HTTP authentication on, 586–588

JMS configuration of, 612–613

programming, 525–526

sockets for programming of, 525–526

server layer

in three-tier model, 314

in two-tier model, 313

ServerSocket class, 523

Service Oriented Architecture (SOA)

APIs for, 600

discussed, 599–600

for JMS, 610–618

JMX for, 600–610

processing chain pattern for, 619

request-reply pattern for, 619–621

Split-Aggregate pattern for, 621–623

system integration patterns for, 619–623

for Web Services, 563

Service Provider Interface (SPI), 626, 627

service-oriented integration, 599

servlet, for timeserver, 252

Servlet 2.4 support, in JSP 2.0, 358

servlet element, 697

session bean, 474, 480–481

Session class, 611

Session Component, 401

Session object

in Hibernate API, 340, 402, 403

transaction support in, 611

SessionFactory, 340–342

sessionScope implicit object, 366

sessionStatelessLocal, 494–495

SetArrayRegion, 439

setBackground method, 226–227

setCertificateEntry, 639

setEnabled method, 226–227

SetField, 443–444

setKeyEntry method, 639

setLayout() method, 158

setListData method, 222

setMarker method, 706, 708

setMinimumFractionDigits method, 174

setObject method

for JDBC API, 323–324

for prepared statements, 323–324

setPreferredSize method, 179–180

setRecentFiles, 267

setSuccessor method, 175–176

setTabComponent method, 220–221

setText method, 231–232

setValueAt method, 186–187

setValues method, 230–232

setVerticalGroup method, 211

SEVERE logging level, 38

SHA-1 (Secure Hash Algorithm), 626, 628, 630

short iterations, 86

Show Tables command, 412

- SIGN signature state, 630**
- Signature engine class**
 - defined, 627, 643
 - states of, 630
 - for verification of data, 631–632
- signatures. See digital signature**
- SignedInfo class, 643**
- signing. See digital signing**
- Simple Object Access Protocol (SOAP)**
 - and WAR files, 695
 - in WS, 567, 570–571, 642
 - in WSDL, 567–568
- SimpleFormatter, 52, 58**
- SimpleKeySelectorResult, 653**
- SimpleTagSupport class, 364**
- single class design, 127**
- SOA. See Service Oriented Architecture**
- SOAP. See Simple Object Access Protocol**
- socket(s)**
 - classes of, 523
 - for client programming, 524–525
 - for communication, 522–541
 - communication with, 522–541
 - defined, 521, 522
 - in echo server, 526–530
 - Java Socket API, 523
 - protocol implementation with, 530–541
 - protocols and, 530–541
 - for server programming, 525–526
 - types of, 523
- Socket class, 523, 524**
- SocketEcho class, 526–529**
- SocketHandler, 49**
- software development principles, 80–88**
- software estimation, 87**
- source code, 83**
- source control, 85**
- SOURCE policy, 26, 27**
- source-level annotations, 27**
- specification, for protocols, 531–540**
- SPI (Service Provider Interface), 626, 627**
- spiral methodology, 89**
- SplashScreen class, 216–217**
- Split, 621**
- Split-Aggregate pattern, 621–623**
- SpringLayout manager, 194–202**
- SQL transactions, in JSTL 1.1, 368–370**
- stack trace, 450**
- standard transactions, 335**
- standardized O/R mappings, 475**
- start method, 692, 693**
- State class, 225, 229–234**
- State pattern, 203, 225**
- stateful session bean, 480**
- state-full filter, 622**
- @Stateless annotation, 497**
- stateless session bean, 480**
- StatelessSession interface, 482–484**
- statement(s) (JDBC 4.0)**
 - batch updates using, 326–328
 - callable, 324–326
 - interface for, 318–319
 - prepared, 320–324
- Statement interface, 315–316**
- Statement object**
 - batch updates using, 326–327
 - execution methods for, 319
- static data importing, 21–23**
- static fields, 442**
- static importing**
 - defined, 7
 - discussed, 21–23
 - syntax for, 22
- static methods, 445**
- StaticParametersInterceptor, 400**
- stock formatters**
 - creating, 54–55
 - discussed, 52
 - SimpleFormatter, 52
 - XMLFormatter, 52–54
- stock handlers**
 - ConsoleHandler, 48–49
 - FileHandler, 49–50
 - for Handler class, 48–51
 - MemoryHandler, 50–51
 - SocketHandler, 49
 - StreamHandler, 48
- stop method, 692, 693**
- storage, of digital keys, 638–640**
- store method, 638**
- Strategy pattern, 126, 146–150, 203–204**
- StreamHandler, 48**
- string(s)**
 - in C/C++, 432
 - functions of, 433–434
 - internationalization of, 433
 - in Java code, 432
 - in Java virtual machine, 432
 - in JNI, 432–436

string(s) (continued)

- localized, 433
- in native code, 435
- replacing, 435
- storage of, 432
- types of, 432

String method, 30

Struts, 396

stubs

- defined, 543
- for Web Services, 581

sub-elements

- in JAXB, 277
- in WAR file attributes, 696

subject authentication, 670

Subject class, 667, 668

subtyping, 147

Sufficient FLAG, 670

SUN package, 626, 628, 630, 638

sun-jaxws.xml, in Tomcat, 579–580

superclass constructors, 25, 459

supportsStoredProcedures, 326

Swing API, 262

Swing applications

- in GroupLayout manager, 208–210
- layout managers in, 158, 190, 191
- and Mustang release, 223
- navigation flows in, 225–234
- in SpringLayout manager, 196–197, 199

switch statements, 24–25

synchronizing objects, 455

synchronous invocation, 582–583

system design, 460–461

system integration patterns

- processing chain pattern, 619
- request-reply pattern, 619–621
- for SOA, 619–623
- Split-Aggregate pattern, 621–623

System.load, 455

systems integration, 599

T

tabTest, 219–220

.tag files

- code reuse with, 361–362
- conversion into Java code, 364

tag library (WebWork), 401, 417

.tagx files, code reuse with, 361–362

target, 26

TCP (Transmission Control Protocol), 523

TCPMon (Apache)

- using, 539–540
- acquiring, 539
- discussed, 538–539
- protocol testing with, 538–540

TeacherResponsibilities, 128

teamwork, 84

testing methods, 32–33, 85

TestNG, 85, 106–110, 125

TestParameters annotation, 34

TestStrategy interface, 207

testStrategy method, 205

THREAD tag, 386–387

thread synchronization, 455

ThreadGroups, 117, 119

threading, using JNI, 455–456

three-tier model, 314–315

Throw function, 450

ThrowNew function, 450

throws, 15

TicTacToe Java Web Start application example, 702–709

time-based licensing

- discussed, 249–250
- implementing, 250–252
- using Java Serialization API, 249–253
- timeserver in, 252–253

TimerInterceptor, 400

timers, in JMeter, 119

timeservers, 252–253

Tomcat

- sun-jaxws.xml in, 579–580
- Web service deployment on, 578
- web.xml configuration in, 579
- WS deployment using, 578–580

ToReflectedField function, 460

ToReflectedMethod function, 460

toString method, 669

traceability, 82–83

tracking bugs, 87–88

transaction(s)

- in Hibernate API, 340
- in JDBC 4.0, 334–335
- management of, 334–335
- SQL, in JSTL 1.1, 368–370
- standard, 335

@TransactionAttribute marking, 508

@TransactionManagement annotation, 508

TransferHandler class, 221–222

Transform class, 644

@Transient annotation, 512

transient keyword, 257
transition phase (UP), 91
translation, of JSP page, 376
Transmission Control Protocol (TCP), 523
transparent representations, 634, 635
transport protocols, 571
TrayDemo constructor, 216–217, 221
trayIcon object, 217–218
treePanel method, 186
trusted certificate entry, 638
Twain, Mark, 87
two-tier model, 313–314
type erasure, 8–11
type parameters. *See also* generics
 bound, 12–13, 299
 for certificates, 656
 formal, 8, 9
types section, 570
type-safe, 7, 8, 24

U

UDP (User Datagram Protocol), 523
UDT (User Defined Types), 323
UML (Unified Modeling Language) training, 81
unboxing conversions
 context for, 21
 defined, 7
 discussed, 20–21
uncheckAll function, 386, 389
Unicode strings, 2-byte, 433
Unified Modeling Language (UML) training, 81
Unified Process (UP), 90–92
UNINITIALIZED signature state, 630
unmarshaller
 creation of, 281–282
 in RMI, 544
 XML, 281–282
unregistering native methods, 457
UnregisterNatives function, 457
UNWRAP_MODE, 657
unwrapped keys, 658–662
UP (Unified Process), 90–92
updatable result sets, 329, 331
update clause, 16
URL class, 533, 534, 579
User Datagram Protocol (UDP), 523
User Defined Types (UDT), 323
user identity authentication, 667
user interface, of email client, 461–471

UTF-8 format, for Java strings, 433
utility classes (Hibernate), 341–342
utility libraries
 discussed, 34–35
 Java logging, 35–60
 Preferences class, 70–77
 regular expressions, 60–70
util.js script, 385

V

validate method, 201–202
validation
 of configuration, 258–259
 of signatures, 652–653
 of XWork, 415
value deserialization, 304
value serialization, 302–305
ValueStack, 401
variable arguments
 in C/C++, 18
 defined, 7
 ellipses in, 18, 19
 in Java code, 18
velocity, 87
verification
 of configuration, 258–259
 of data, 632–634
 for digital signing, 632–634
 of MessageDigest, 629–630
VERIFY signature state, 630, 632
versioning, 260–261
vertical components, 397
view (MVC), 394
view component (MVC), 137–139
Visitor pattern, 190, 193–194
Visual Studio 2005, 430
volatility, in application, 126

W

WAR files. *See* Web Archive
WARNING logging level, 38
Waterfall methodology, 88–89, 92–93
weak global references, 451, 453, 454
weather application example (WS), 564–566, 572–575, 589–596
Web application(s), 694–695
web application deployment descriptor (web.xml), 422–423

web application visualization

- for adding data, 416
- for browsing data, 416
- with JSP 2.0, 376–380
- with JSTL 1.1, 370–375
- in Model 2 architecture, 415–422

Web ARchive (WAR files)

- with Ant (Apache), 709–713
- deployment descriptor for, 696–698
- directory structure, 695
- discussed, 694–695
- inspecting, 699
- package target in, 112
- SOAP and, 695
- structure of, 695

web browsers, 238

web browsing, 520

web component (EJB 3)

- construction of, with tables, 496–506
- deployment of, 506–517
- for form usage, 485–496
- for many-to-many relationships, 506–517
- for one-to-many relationships, 496–506

web portals, 521

Web Services (WS)

- access control in, 585–586
- asynchronous invocation of, 581, 583–584
- calling, 581–582
- communication for, 563–596
- deployment of, 575–580
- discussed, 563–564
- enabling example for, 564–566, 572–575, 589–596
- endpoints attributes for, 579
- invocation of, 581–584
- of JAX-WS, 575–577, 579
- JDK 6 for deployment of, 577–578
- platform-independent RPCs in, 566–567
- relevance of, 600
- SOAP in, 570–571
- stubs for, 581
- synchronous invocation of, 582–583
- Tomcat for deployment of, 578–580
- transport protocols for, 571
- weather application example for, 564–566, 572–575, 589–596
- writing client for, 580–589
- wsgen and, 577
- wsimport tool for, 580–581

Web Services Description Language (WSDL)

- discussed, 568–570

- for JAXB, 308
- location of, 585
- SOAP in, 567–568

Web Services Interoperability Organization (WS-I)

- Basic Profile of, 567–568
- defined, 564

Web Services Interoperability Technologies Project (WSIT), 596

web sites

- automated browsing on, 520
- server-side applications for, 525
- Web Services on, 563

Web tier (J2EE), 93

WebMethod annotation, 575

WebParam annotation, 575

WebResult annotation, 575

WebService annotation, 575

WebServiceContext, 587

WebWork

- Actions in, 399
- architecture of, 399–400
- components in, 401
- defined, 396
- deployment descriptor for, 422–423
- Hibernate support by, 402–403
- interceptors in, 400–401
- as J2EE web application, 422
- Model 2 architecture with, 396, 399–402
- modularity of, 426
- OGNL in, 401
- request-response flow in, 399–400
- tag library, 401, 417
- ValueStack in, 401
- XWork as component in, 399
- XWork in, 144
- XWork wrapped to, 423

WebWork JSP custom tags, 417

WebWork2, 396

web.xml, 422–423, 579

welcome-file-list element, 698

WHERE clause, 475

wildcards, 11–12

WorkMonitor, 609–610

WorkPanel, 138, 139

World Wide Web Consortium, 270

WRAP_MODE, 657

wrapped keys, 658–662

wrapper, 423

writeObject method, 254

writing, configuration, 245–246

WS. See **Web Services**

WSDL. See **Web Services Description Language wsgen**, 577

WS-I (Web Services Interoperability Organization)
Basic Profile of, 567–568
defined, 564

wsimport tool, 580–581, 585, 596

WSIT (Web Services Interoperability Technologies Project), 596

X

XDoclet, 110–117

- clean target in, 111–112
- as code generation engine, 113–114
- compile target in, 112
- config target in, 116
- discussed, 125
- generate-web target in, 112
- Hibernate in, 113, 115–117
- for mapping, 110–111
- package target in, 112
- usage of, 110

xjc command, 276–277

XML

- annotations generating, 301–302
- content creation, 283
- for content creation, 283
- data types in, 568
- format for, 273–280
- format of, 273–280
- JAXB classes from, 276–277
- marshaller, 281–282
- marshalling/unmarshalling, 281–282
- relevance of, 600
- unmarshaller, 281–282

XML Digital Signature API Specification

- classes for, 643–645
- detached signatures in, 646–649
- discussed, 642–643
- document signing with, 645
- enveloped signatures in, 649–652
- in JCA, 642–654
- signing process in, 645–646
- validating signatures in, 652–653

XML document

- for configuration objects, 271–272
- in JSP20-compliant web containers, 362–363

XML schema

- annotations generating, 301–302
- JAXB classes from, 276–277

XML Schema Definition (XSD)

- for configuration data model, 275–276
- development of, 270
- in JAXB, 271
- for XML format definitions, 273–280

XmlAccessorType, 296–297

XmlAttribute, 292, 297

XmlElement, 292, 297

XmlElementWrapper, 292, 297–298

XMLEncoder/Decoder API

- classes in, 265
- customization of, 268–269
- discussed, 262
- EJB serialization using, 262–263, 265–269, 265–270
- file format of, 264
- Java Serialization API vs., 262–263, 269
- in JDK, 269
- usage of, 269–270
- XML serialization format in, 263–264

XMLFormatter, 52–54, 58

XmlID, 292, 305

XmlIDREF, 292, 305

XmlJavaTypeAdapter, 293, 298–300

XmlRootElement, 280–281, 291, 293, 295

XmlTransient, 293, 300–301

XmlType, 295–296, 293–296

XmlValue, 293

X/Open SQL Call Level Interface (CLI), 312

XP (Extreme Programming), 87, 91–92

XSD. See **XML Schema Definition**

XWork

- validating, 415
- as WebWork component, 144, 399
- WebWork wrapped to, 423

XWork Action

- for adding data, 413–414
- for browsing data, 412–413
- discussed, 399, 412–415

